Diversity-Guided Efficient Multi-Objective Bayesian Optimization with Batch Evaluations (DGEMO)

Luković et al.

Reviewed & Presented by Joon Hyeok Kim

Three-Minute Summary of DGEMO

DGEMO solves $\underbrace{\text{multi-objective problem}}_{\text{MOO}}$ with $\underbrace{\text{black-box functions}}_{\text{BO}}$ $\underbrace{\text{fast in parallel}}_{\text{MOBO}}$.

1. Multi-Objective Optimization (MOO)

- Pareto Frontier Approximation
- Hypervolume Indicator

2. Bayesian Optimization (BO)

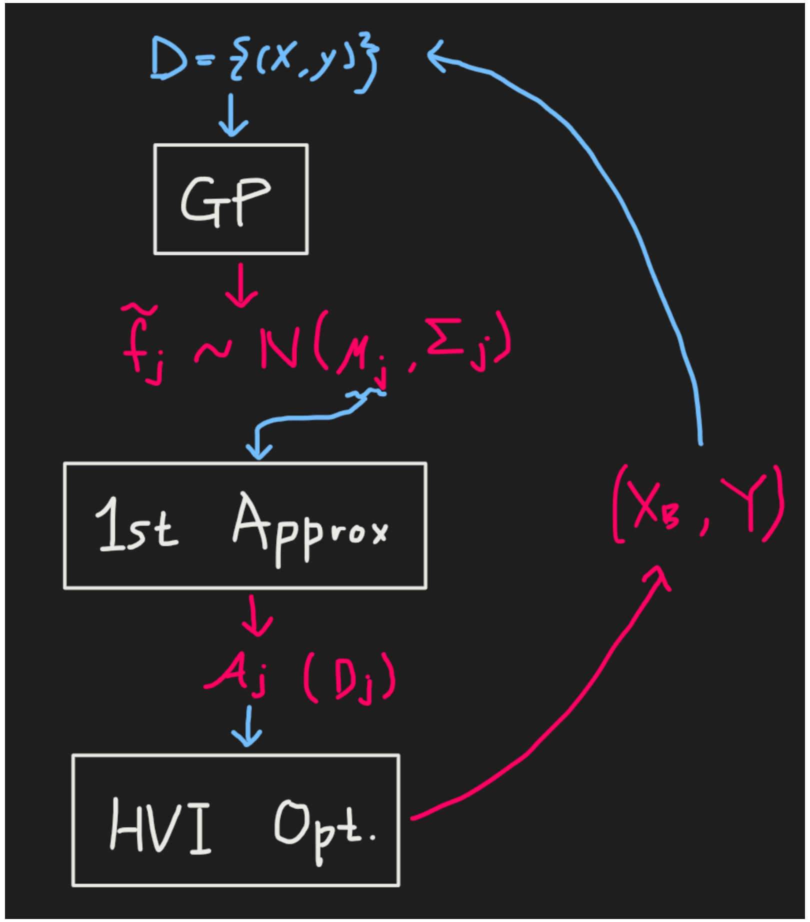
- Gaussian Process

3. Multi-Objective Bayesian Optimization (MOBO)

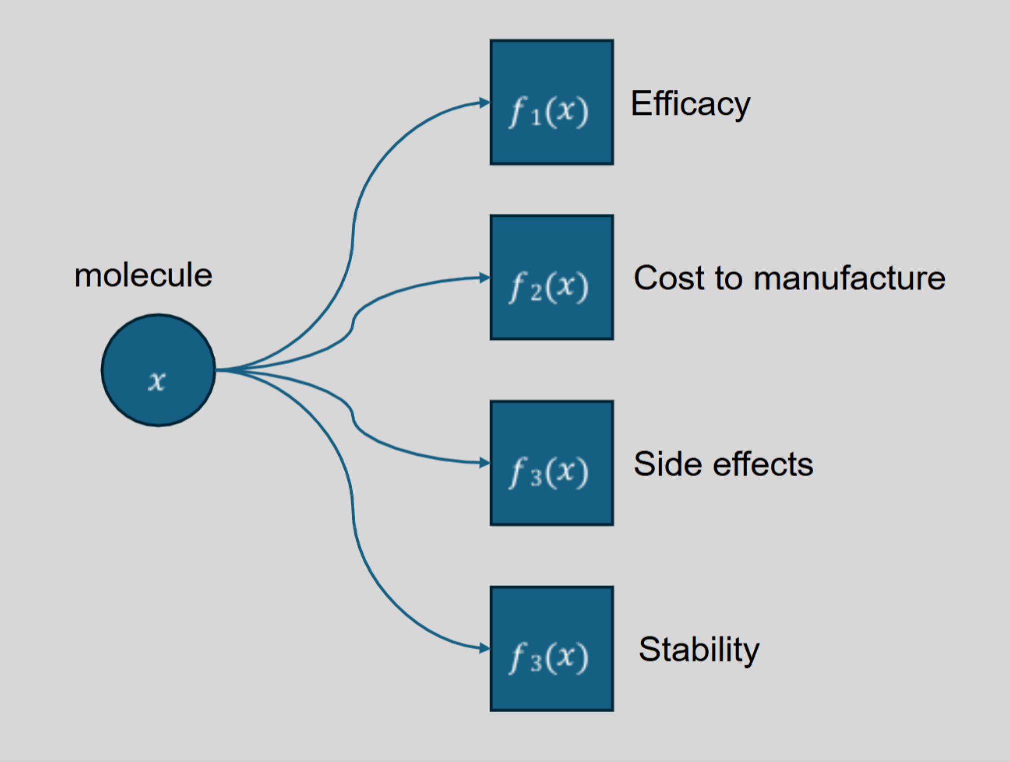
- Diversity Region Batch Selection : Running in parallel = Faster!

Outline

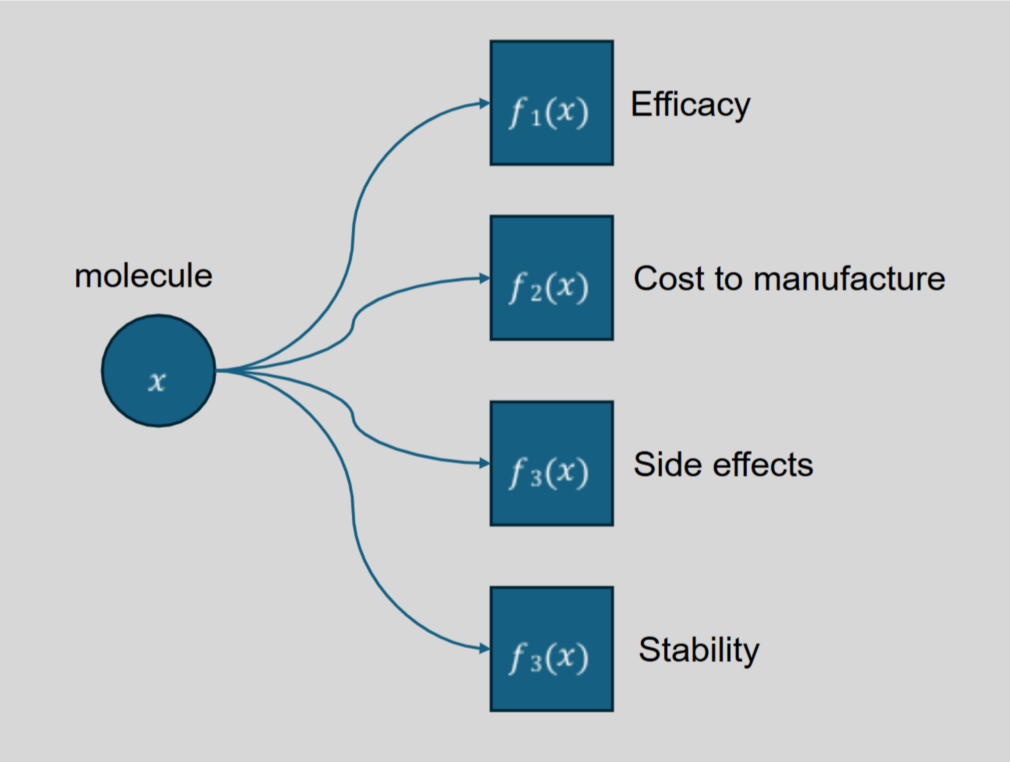
1. Multi-Objective Optimization (MOO)
    - Pareto Front
    - Hypervolume Index (HVI)
2. Batch Selection Method (MOBO)
    - Sequential Selection vs Batch Selection
3. DGEMO
    - Pareto Front Approximation
    - Batch Selection Strategy



1. Multi-Objective Optimization (MOO)



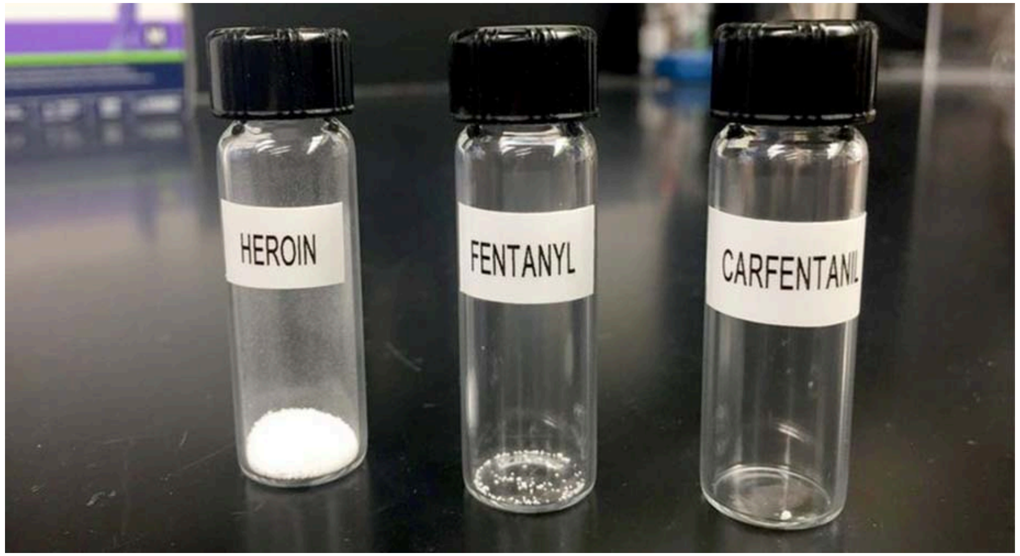
Recall the example of finding the optimal molecule for medicine.



Main Issue : Conflicting Objectives

i.e.) Trade-off between Objectives

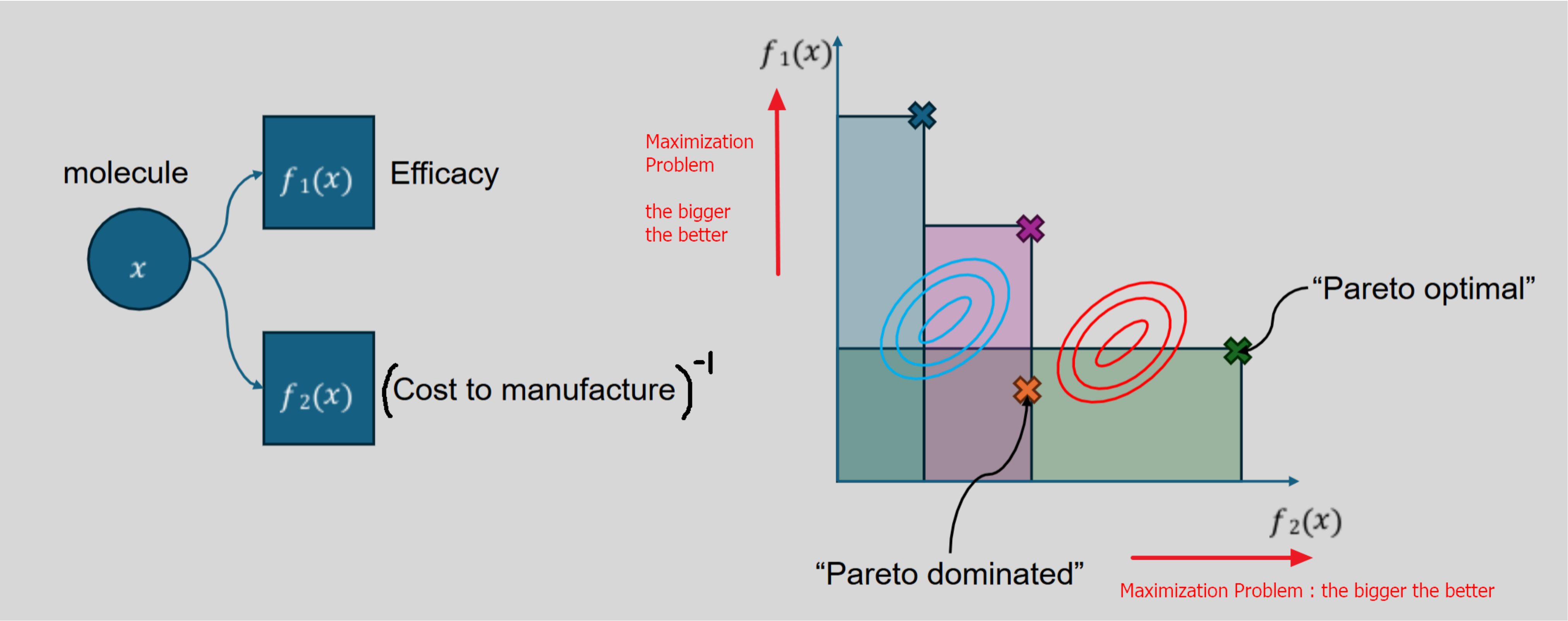
e.g.) The more effective, the more unstable, costly, severe side effect



How can we measure and compare the performance in a Multi-Objective case?

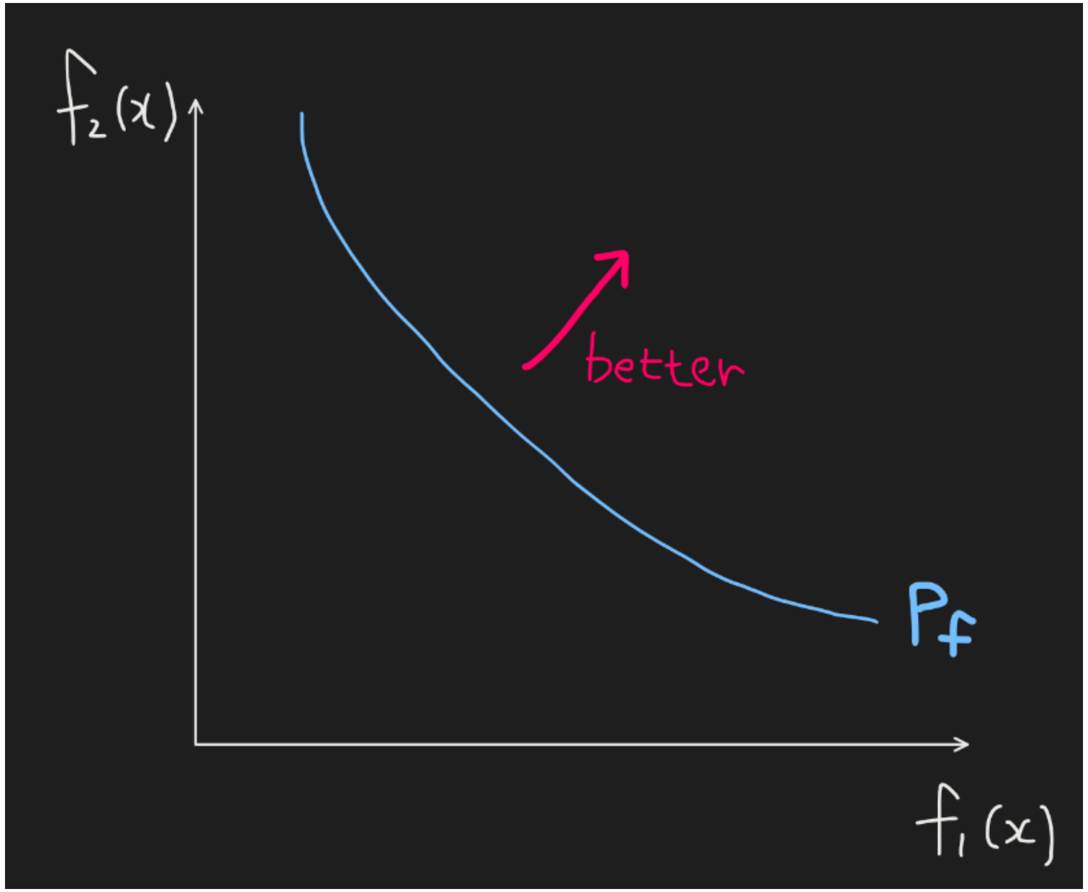
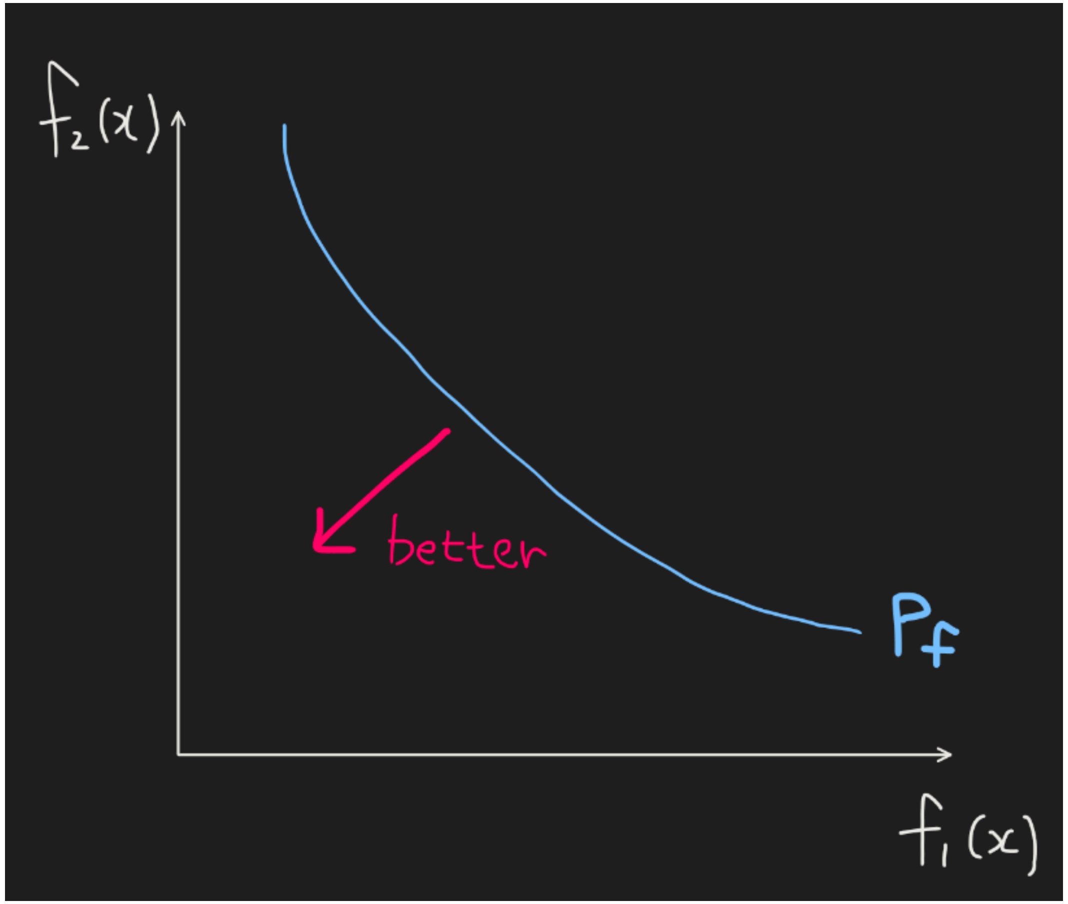
Rule) Pareto Optimality

x^* is Pareto optimal if there is **no** other solution x s.t. $\begin{cases} f_i(x) \geq f_i(x^*) & \forall \text{ objective } i \\ f_i(x) > f_i(x^*) & \exists \text{ objective } i \end{cases}$



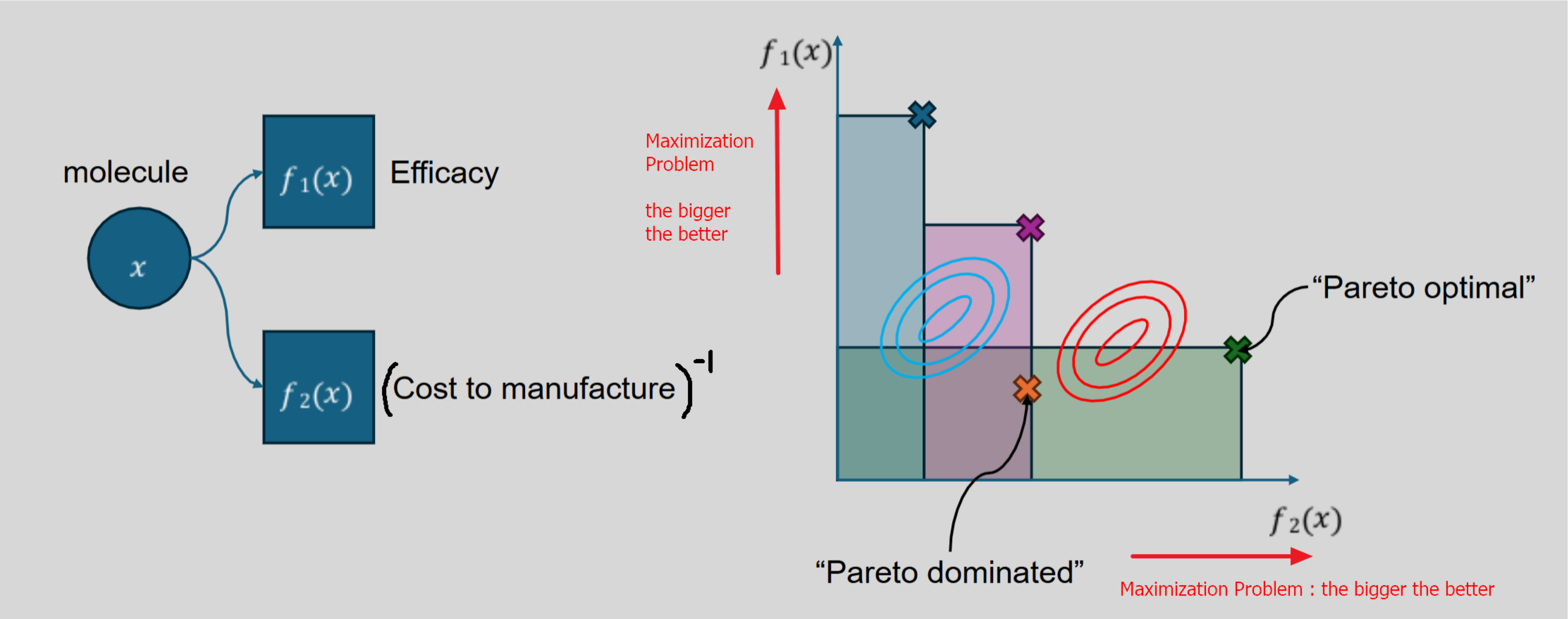
Concept) Pareto Front(ier)

- The set of Pareto optimal points, i.e. the solution set of MOO.

Maximization Problem	Minimization Problem
	

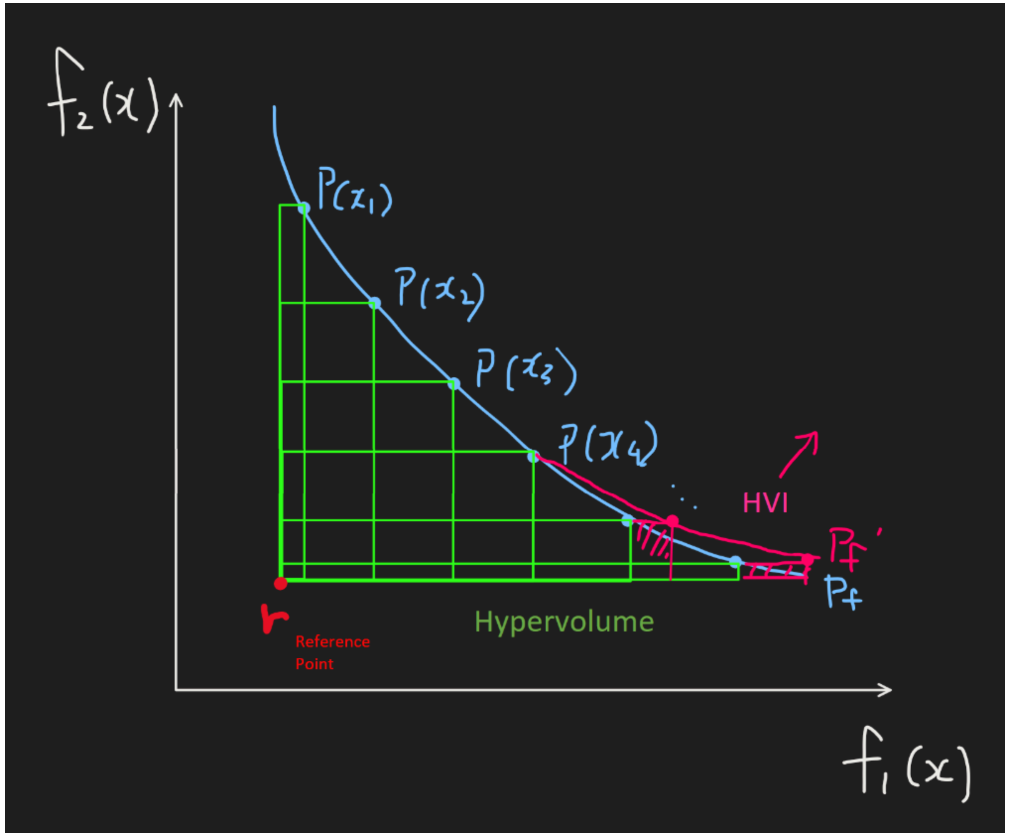
The Goal of Multiple Objective Optimization Problem

- Get the Pareto front.
  - i.e.) the set of Pareto optimal solutions
- Points on the Pareto fronts are better off than other available points.



How to measure the improvement on Pareto Fronts

1. Calculate the initial Hypervolume of the Pareto Front.
2. Update the Pareto Front and calculate the Hypervolume again.
3. If the Hypervolume increased, then there was an improvement : HVI



Hypervolume

- Def.)
  - Let
    - $\mathcal{P}_f$: a Pareto Front **approximation** in an m-dim'l performance space
    - $r \in \mathbb{R}^m$: a reference point
      - i.e.) a fixed point deliberately chosen so that its performance is "inferior" to that of all candidate solutions (or Pareto-optimal solutions)
  - Then the **hypervolume** $\mathcal{H}(\mathcal{P}_f)$ is defined as
    - $\mathcal{H}(\mathcal{P}_f) = \displaystyle\int_{\mathbb{R}^m} \mathbb{1}_{H(\mathcal{P}_f)}(z)dz$
      - where
        - $H(\mathcal{P}_f) = \{z \in Z \mid \exists 1 \le i \le |\mathcal{P}_f| : r \preceq z \preceq \mathcal{P}_f(i)\}$
        - $\mathcal{P}_f(i)$: the i-th solution in $\mathcal{P}_f$
        - $\preceq$: the relation operator of objective dominance
        - $\mathbb{1}_{H(\mathcal{P}_f)} = \begin{cases} 1 & \text{if } z \in H(\mathcal{P}_f) \\ 0 & \text{otherwise} \end{cases}$: a Dirac Delta function

Hypervolume Improvement (HVI)

- Def.)
  - $\mathrm{HVI}(P, \mathcal{P}_f) = \mathcal{H}(\mathcal{P}_f \cup P) - \mathcal{H}(\mathcal{P}_f)$
- Meaning)
  - How much the hypervolume would **increase** if a set of new points $P(\mathbf{p}_1, \cdots, \mathbf{p}_n) \subset \mathbb{R}^m$ is added to the current Pareto front approximation $\mathcal{P}_f$

MOO vs MOBO

MOO : Multi Objective Optimization

- Goal)
  - Solve problems involving several conflicting objectives and optimizes for a set of Pareto-optimal solutions
- e.g.)
  - MOEA, NSGA-II, MOEA/D

MOBO : Multi Objective Bayesian Optimization

- Goal)
  - Solve the **MOO** problem with Bayes Opt.
- Types)
  - Sequential Selection vs Batch Selection

2. Batch Selection Method (MOBO)

Sequential Selection vs Batch Selection

Sequential Selection Models

input: initial dataset $\mathcal{D}$ ▸ can be empty
repeat
  $x \leftarrow \textsc{policy}(\mathcal{D})$ ▸ select the next observation location
  $y \leftarrow \textsc{observe}(x)$ ▸ observe at the chosen location
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x, y)\}$ ▸ update dataset
until termination condition reached ▸ e.g., budget exhausted
return $\mathcal{D}$

- A.K.A. Single Point Method
- e.g.) PareEGO, EHI, SUR, PESMO, MESMO, USeMO

Batch Selection Models

input: initial dataset $\mathcal{D}$ ▸ can be empty

repeat

$\boxed{X_B = (\mathbf{x}_1, \cdots, \mathbf{x}_b)} \leftarrow \text{Policy}(\mathcal{D})$ .ect the next observation location

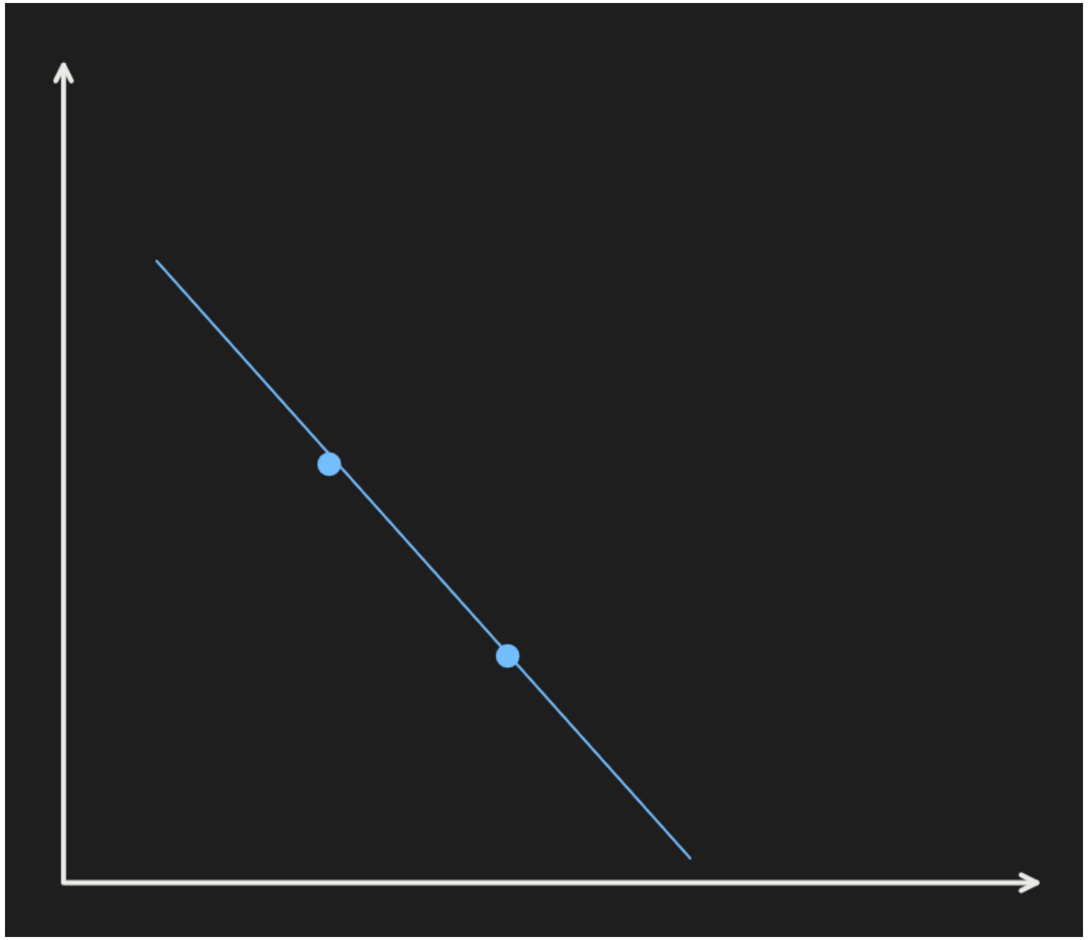
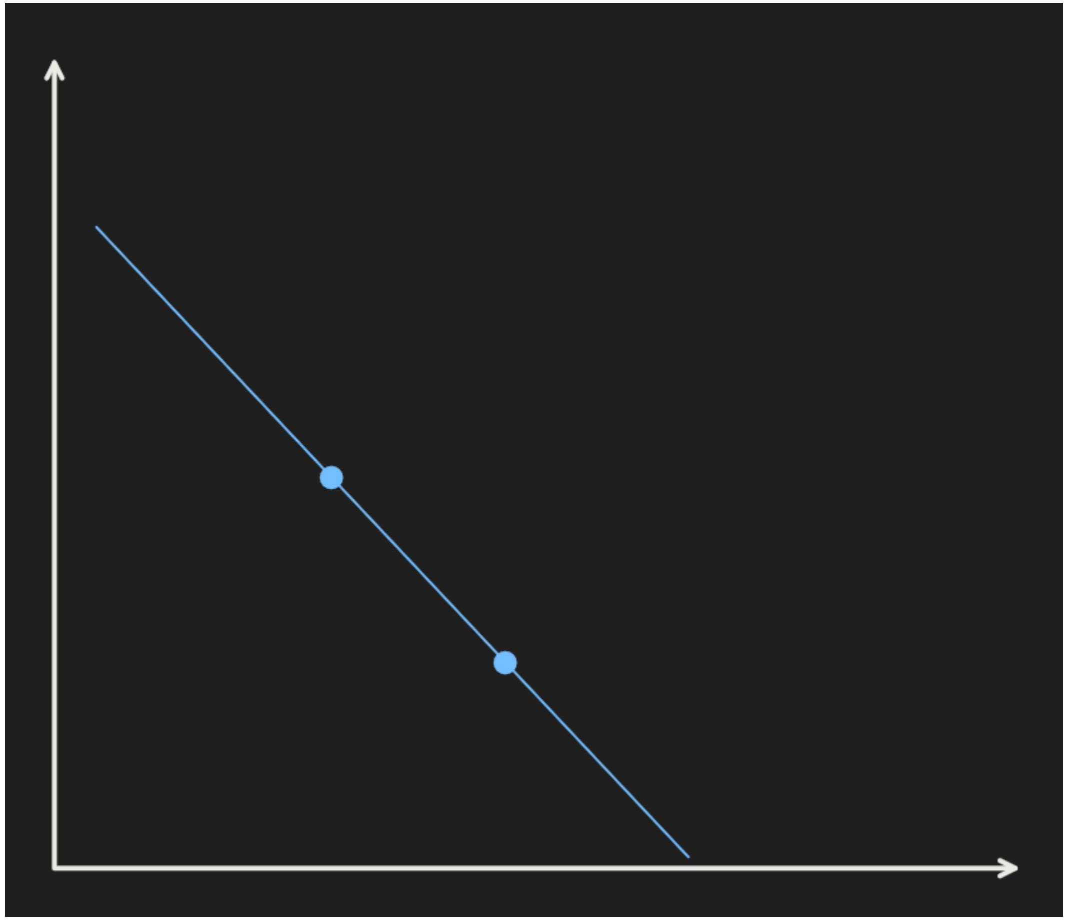
$\boxed{Y = (F(\mathbf{x}_1), \cdots, F(\mathbf{x}_b))} \leftarrow \text{Observe}(X_B)$ ▸ observe at the chosen location

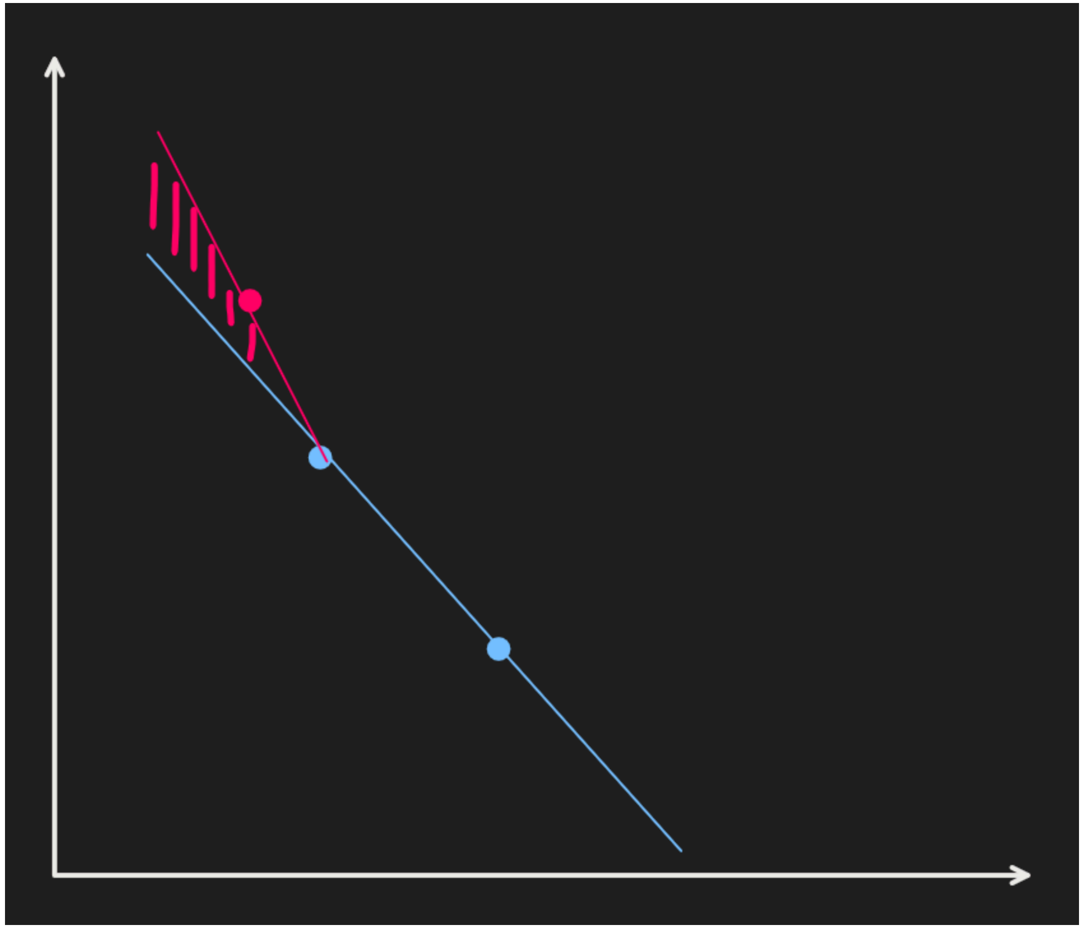
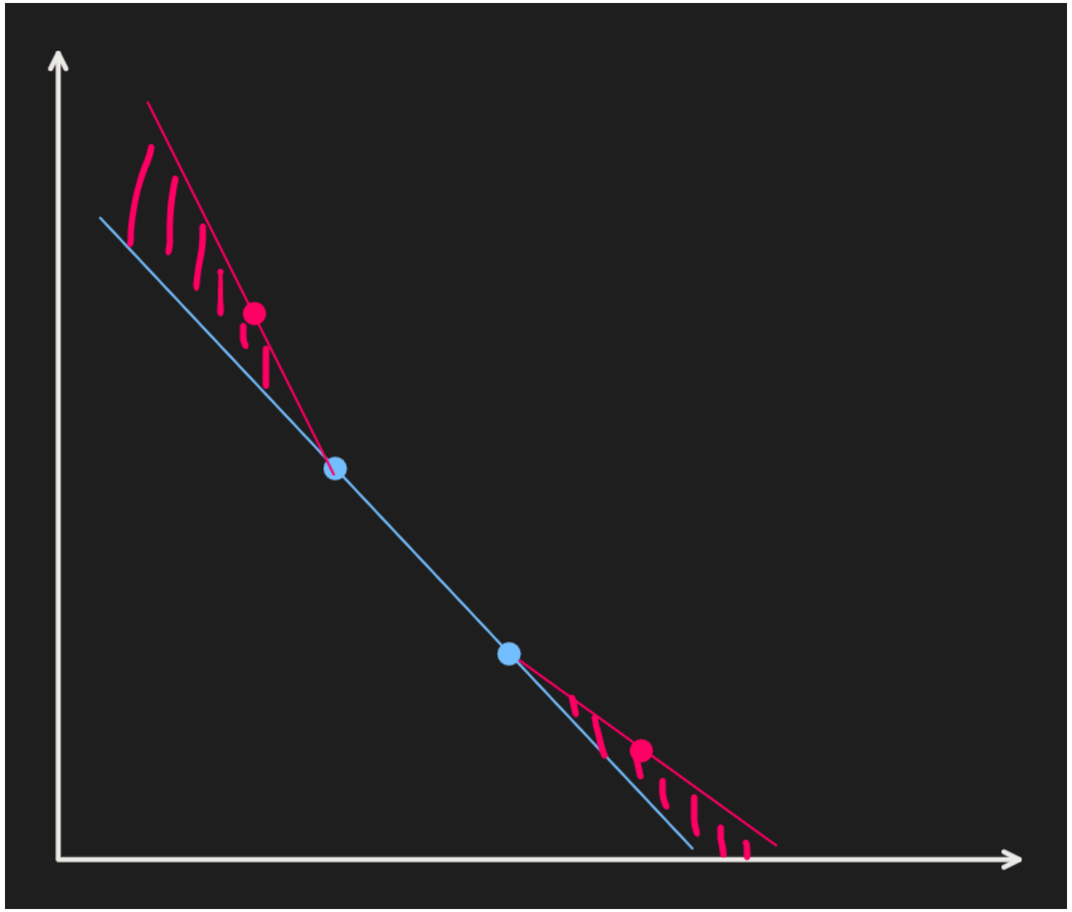
$\mathcal{D} \leftarrow \boxed{\mathcal{D} \cup \{(X_B, Y)\}}$ ▸ update dataset

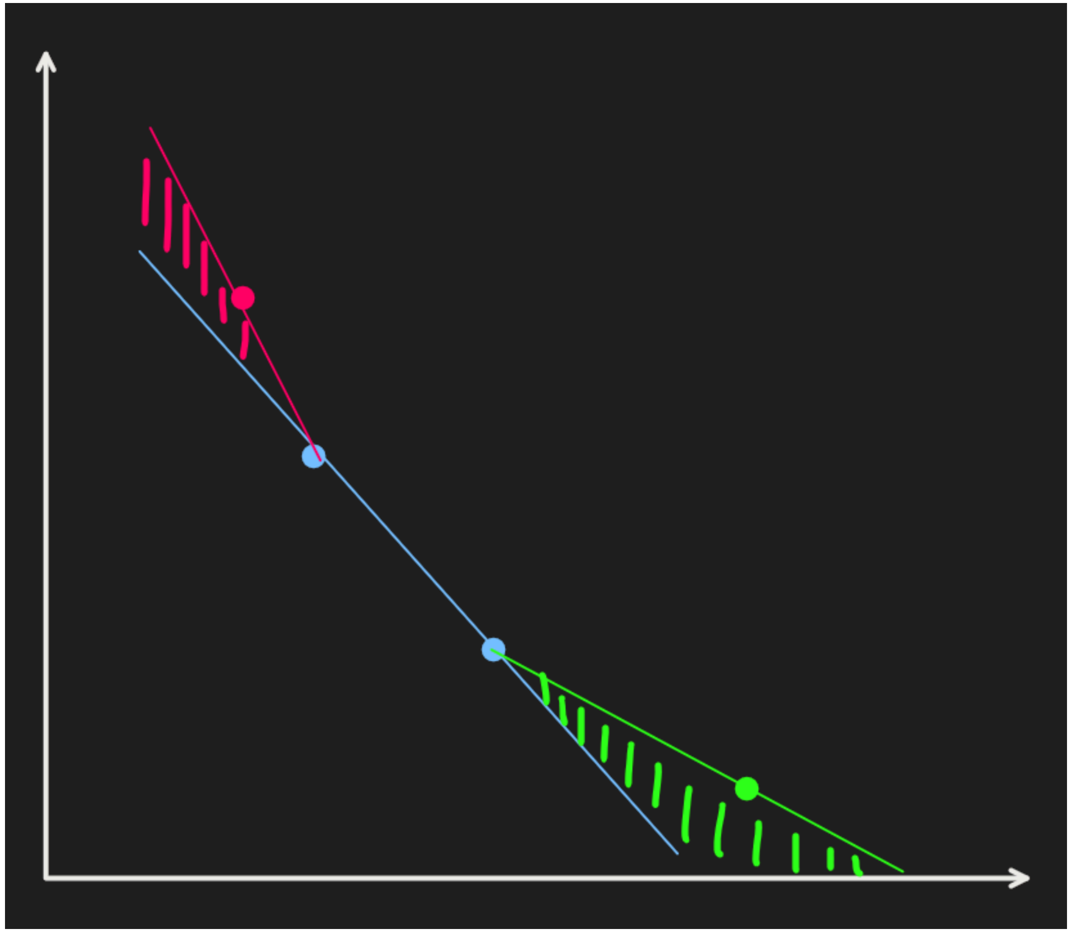
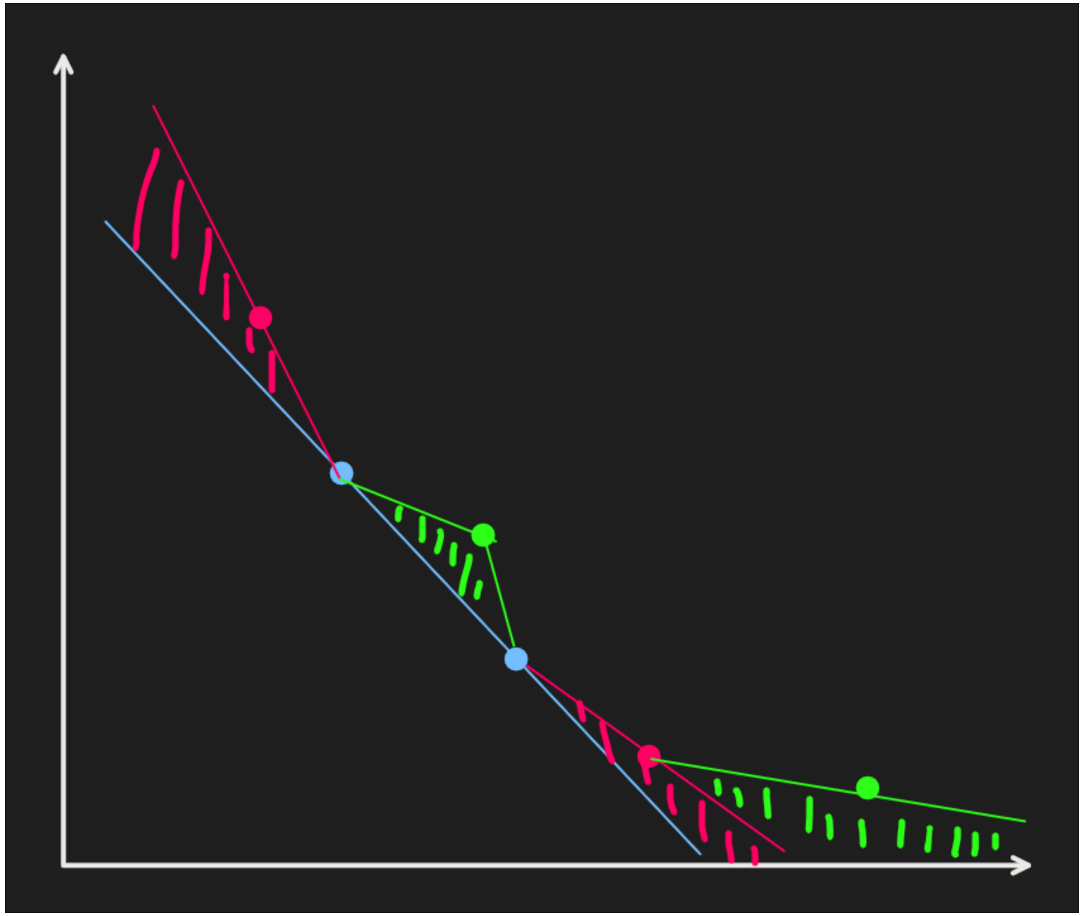
until termination condition reached ▸ e.g., budget exhausted

return $\mathcal{D}$

- Fast because it can run parallel : $X_B = (\mathbf{x}_, \cdots, \mathbf{x}_b)$
- With some sacrifice in accuracy
- e.g.) MOEA/D-EGO, MOBO/D, TSEMO (Thompson Sampling), BS-MOBO

Sequential Selection	Batch Selection
	

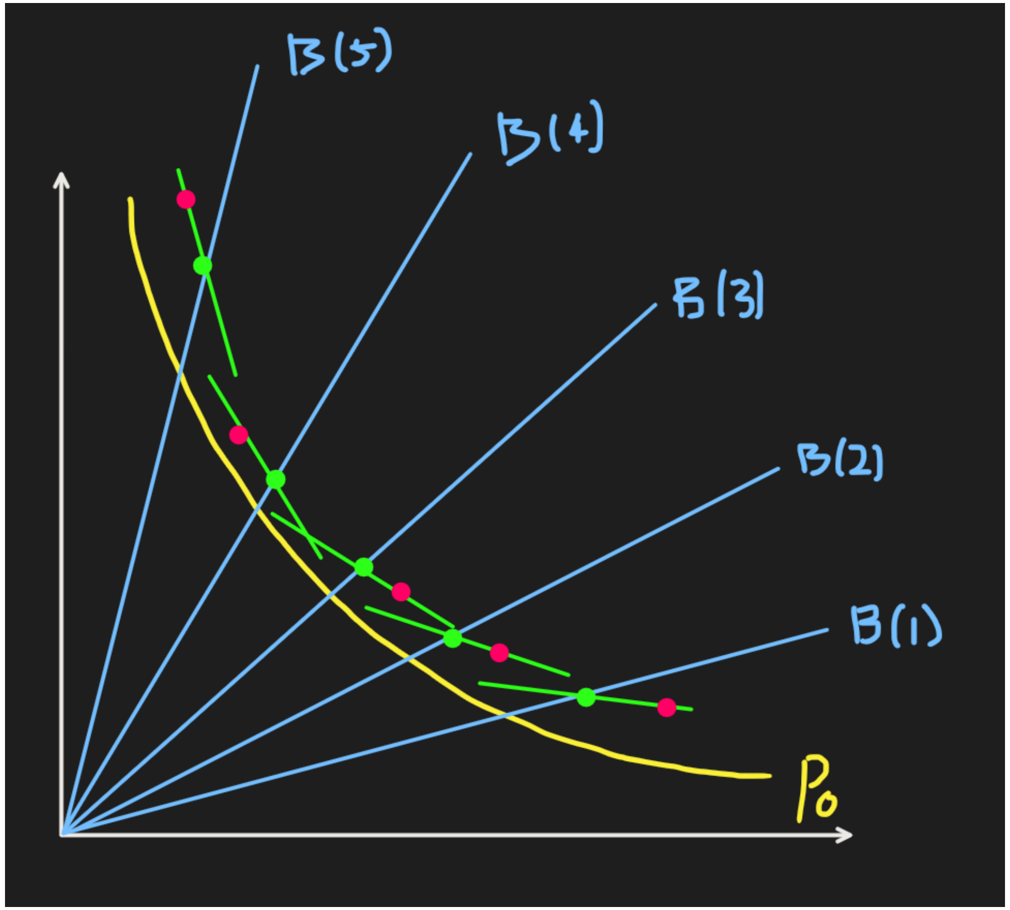
Sequential Selection	Batch Selection
	

Sequential Selection	Batch Selection
	

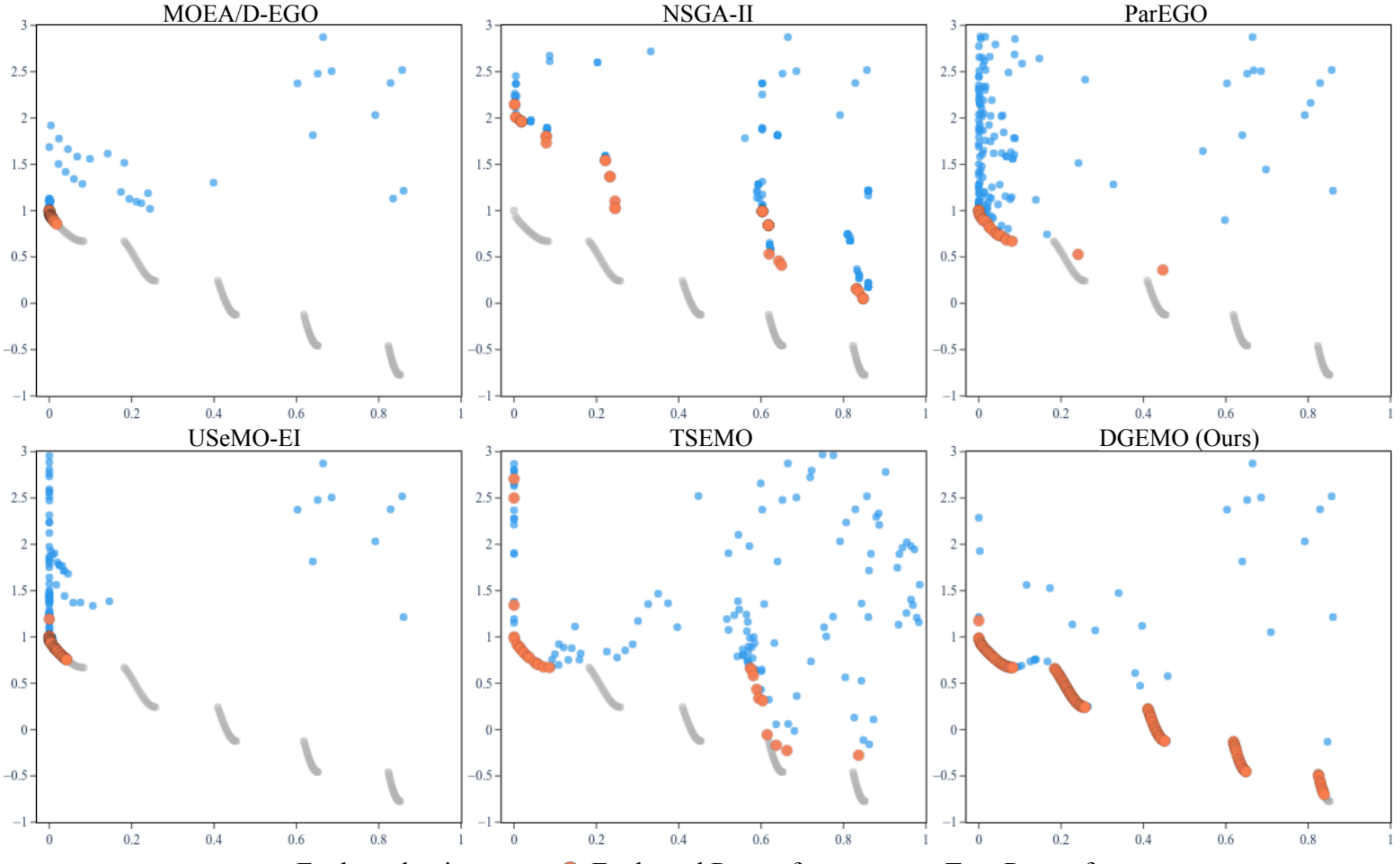
3. DGEMO

DGEMO is a Batch Selection MOBO

with Diversity Region Batch Selection Strategy



Just a small spoiler — the diversity region worked!

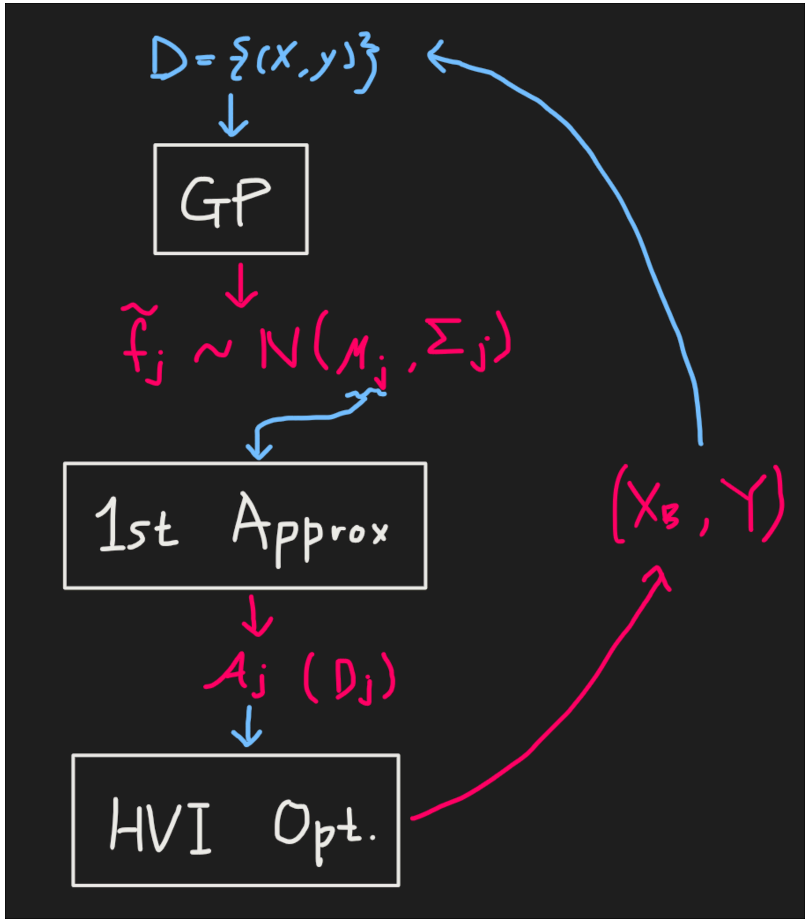


3. DGEMO

DGEMO is a Batch Selection MOBO

with Batch Selection Strategy that utilizes...

1. GP as a surrogate model
2. **First-order Approximation** $(\mathcal{A}_i)$ on Pareto Front
3. HVI maximization w.r.t. **Diversity Regions** $(\mathcal{D}_i)$
   - where $\mathcal{A}_i = \mathcal{D}_i$



Problem Setting) Multiple Objective Problem

Def.) Design Space and Constraints

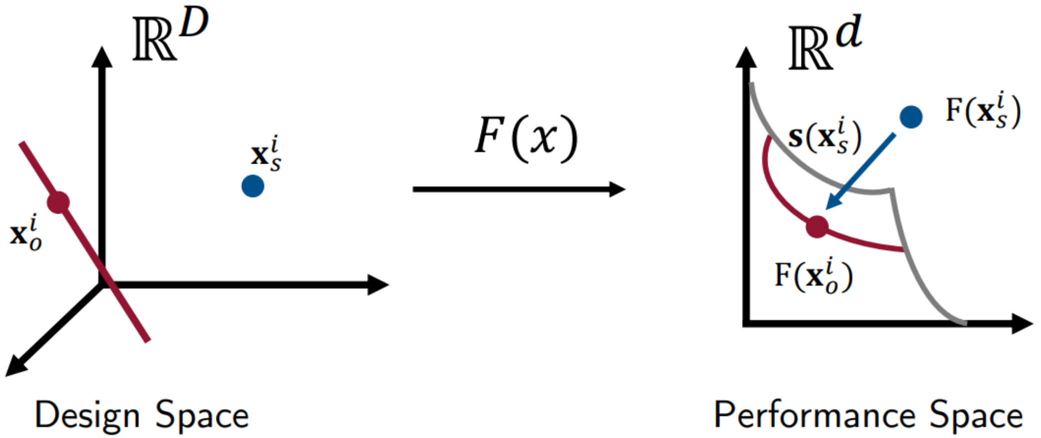
- $\mathcal{X} = \{\mathbf{x} = (x^1, \cdots, x^D) \in \mathbb{R}^D : g_j(\mathbf{x}) \leq 0, \forall j \in \{1, \cdot, K\}\}$
    - the design space where
        - g_j represents a single constraint on $\mathbf{x}$
        - $G(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^K$ s.t. $G(\mathbf{x}) = (g_1(\mathbf{x}), \cdots, g_K(\mathbf{x}))$
            - the concatenation of the K constraints

Def.) Performance metric and Space

- $f_i : \mathbb{R}^D \to \mathbb{R}$: the i-th performance metric (objective function)
- $F(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^d$ s.t. $F(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_d(\mathbf{x}))$
    - the concatenation of the d performance metrics
    - $d \geq 2$: multi-objective problem!
    - $d \ll D$
- $\mathcal{S} = F(\mathcal{X}) \subseteq \mathbb{R}^d$: the performance space

Be Careful : Design Space vs Performance Space

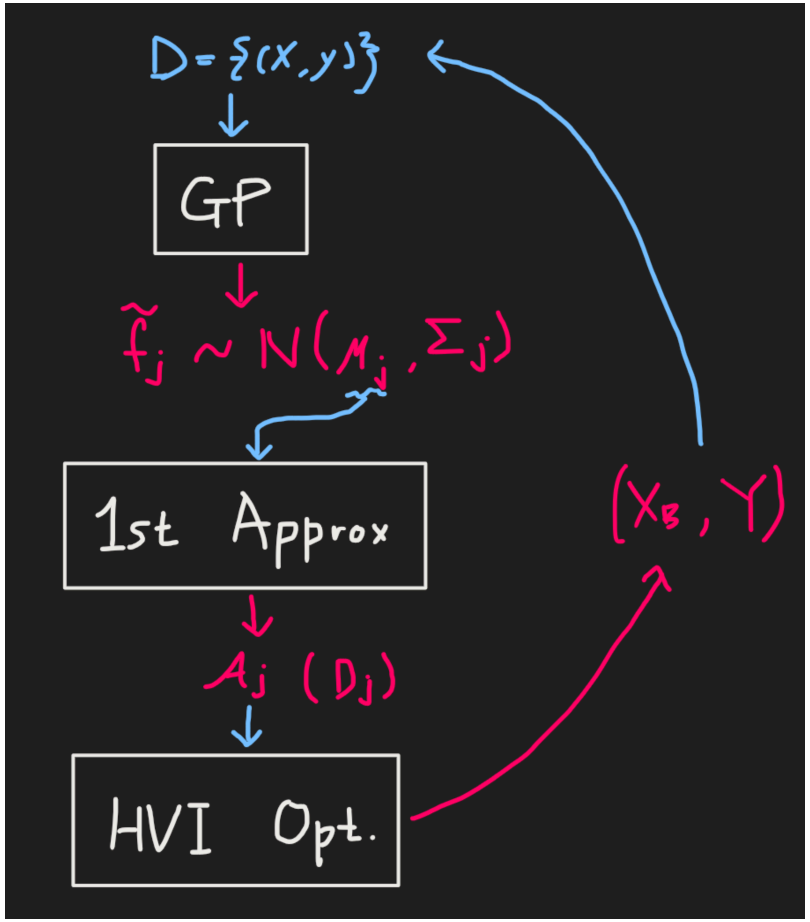
Design Space: $\mathcal{X} \subseteq \mathbb{R}^D$	$\rightarrow$	Performance Space: $F(\mathcal{X}) \subseteq \mathbb{R}^d$
$\mathbf{x} = (x_1, \cdots, x_D)$	$F : \mathbb{R}^D \rightarrow \mathbb{R}^d$ $F = (f_1, \cdots, f_d)$	$F(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_d(\mathbf{x}))$
- Sample in here: $\mathbf{x}_s$ - Solution is in here: $\mathbf{x}^*$ - $B(j) = \{\mathbf{x}_{j1}, \cdots, \mathbf{x}_{jK}\}$		- Performance : $F(\mathbf{x}_s), F(\mathbf{x}^*)$ - Pareto Front : $\mathcal{P}_f$



3-1. GP as a surrogate model.

Problem)

- All we have is the data $\mathcal{D} = \{X, Y\}$
  - Each data point looks like $(\underbrace{x_1, x_2, \cdots, x_D}_{D \text{ features}}, \underbrace{y_1, y_2, \cdots, y_d}_{d \text{ objectives}})$.
- We want to find $F(X) = (f_1(X), f_2(X), \cdots, f_d(X))$



For the j-th independent objective function $f_j,\ \forall j \in \{1, \cdots, m\}$...

1. Prior

- Mean Function : $m_j : \mathcal{X} \to \mathbb{R}$
  - $m_j(\mathbf{x}) = 0$
- Kernel Function $k_j : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
  - Matern Kernel
    - $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)$
      - where
        - σ^2 : the variance parameter
        - ℓ : the length scale
        - ν : the smoothness parameter
        - K_ν : the modified Bessel function

2. Likelihood)

- The log marginal likelihood of on the available dataset $\{X, Y\}$
- $p_j(\mathbf{y}|\mathbf{x}, \theta_j)$:
  - where θ_j is the parameters set
    - i.e.) $\sigma^2, \ell, \nu \in \theta_j$
  - Recall, $\mathbf{y} = \mathbf{f}(\mathbf{x}) + \epsilon$

3. Posterior

- $\tilde{f}_j \sim N(\mu_j(\mathbf{x}), \Sigma_j(\mathbf{x}, \mathbf{x}))$
  - where
    - $\begin{cases} \mu_j(\mathbf{x}) = m_j(\mathbf{x}) + k_j K_j^{-1} Y = k_j K_j^{-1} Y \\ \Sigma_j(\mathbf{x}) = k_j(\mathbf{x}, \mathbf{x}) - k_j K_j^{-1} k_j^\top \end{cases}$ for $\begin{cases} k_j = k_j(\mathbf{x}, X) \\ K_j = k_j(X, X) \end{cases}$

Result

For the d number of objectives f_j, we have $\tilde{f}_j \sim N(\mu_j, \Sigma_j)$.

$$\tilde{F}(\mathbf{x}) = (\tilde{f}_1(\mathbf{x}), \cdots, \tilde{f}_d(\mathbf{x})) \quad \text{where } \tilde{f}_j \sim N(\mu_j, \Sigma_j)$$

Additionally, during the first order approximation, we will use only the mean of $\tilde{f}_j$.

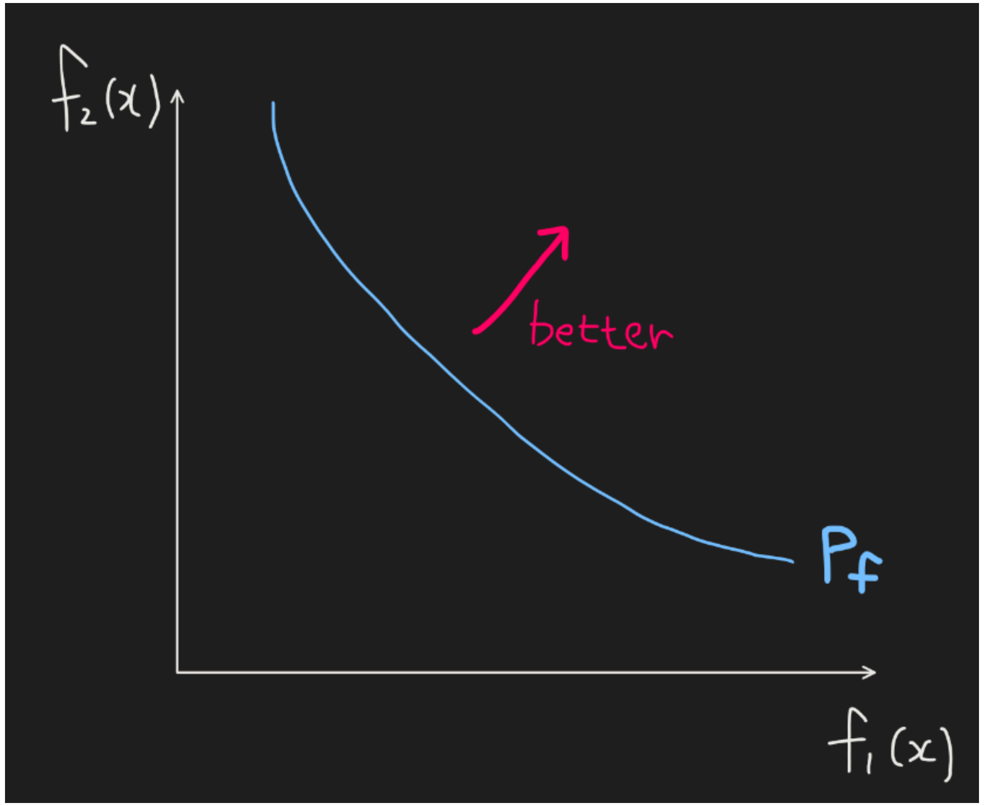
- $\tilde{f}_j = \mu_j, \ \forall j$

$$\therefore \tilde{F} = (\mu_1(\mathbf{x}), \cdots, \mu_d(\mathbf{x}))$$

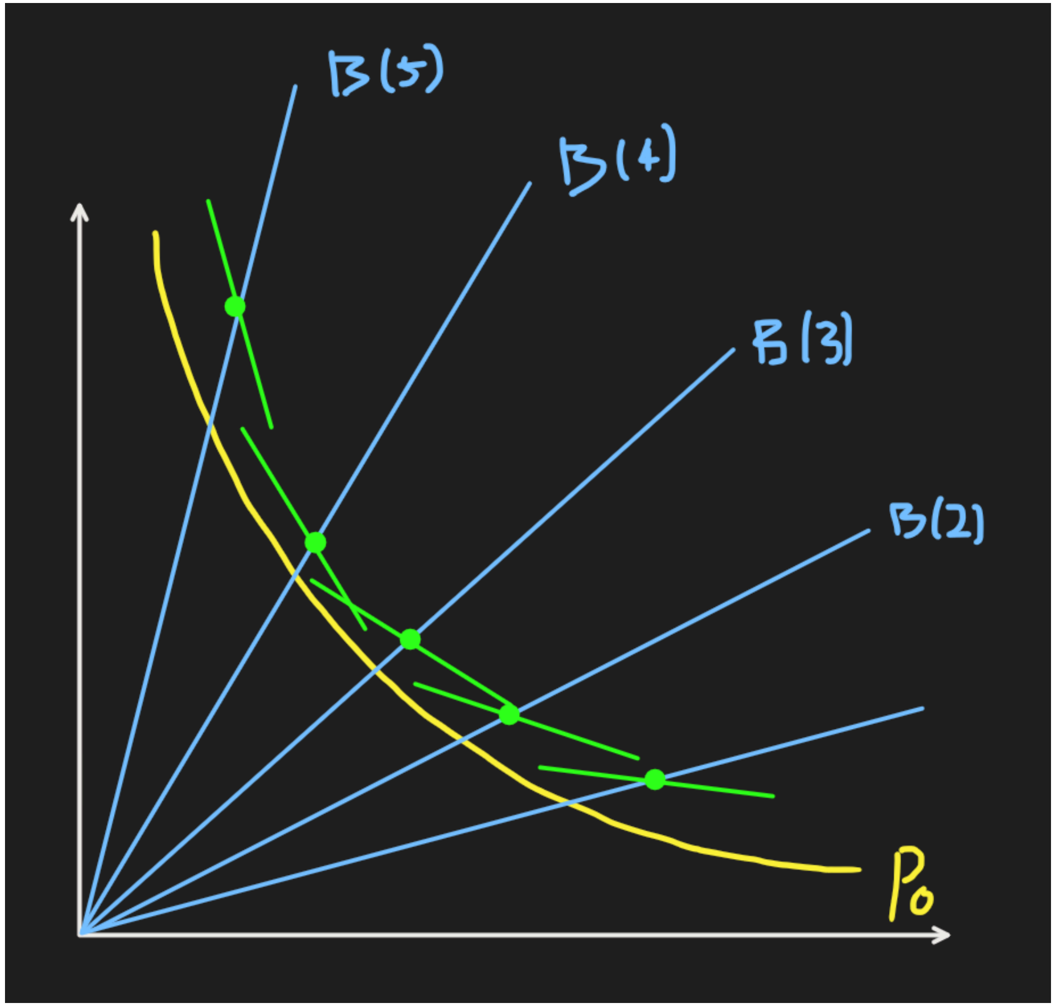
Time to simply get the Pareto Front using $\tilde{F}$?

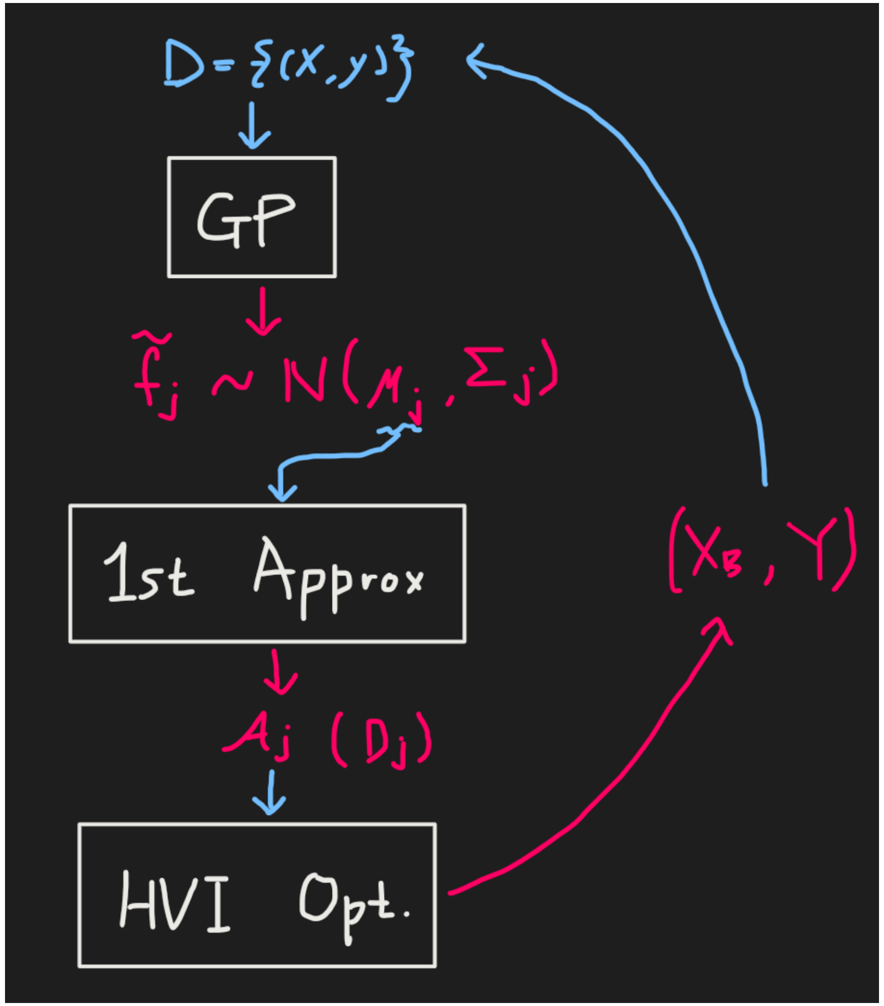
Problem) Pareto Front is not nice and friendly in higher-dimensional space

Recall that we had d number of objectives.
Thus, the performance space is the subset of $\mathbb{R}^d$.

$d = 2$	$d = 10$
	

3-2. First-Order Approximation (on the Pareto Front)



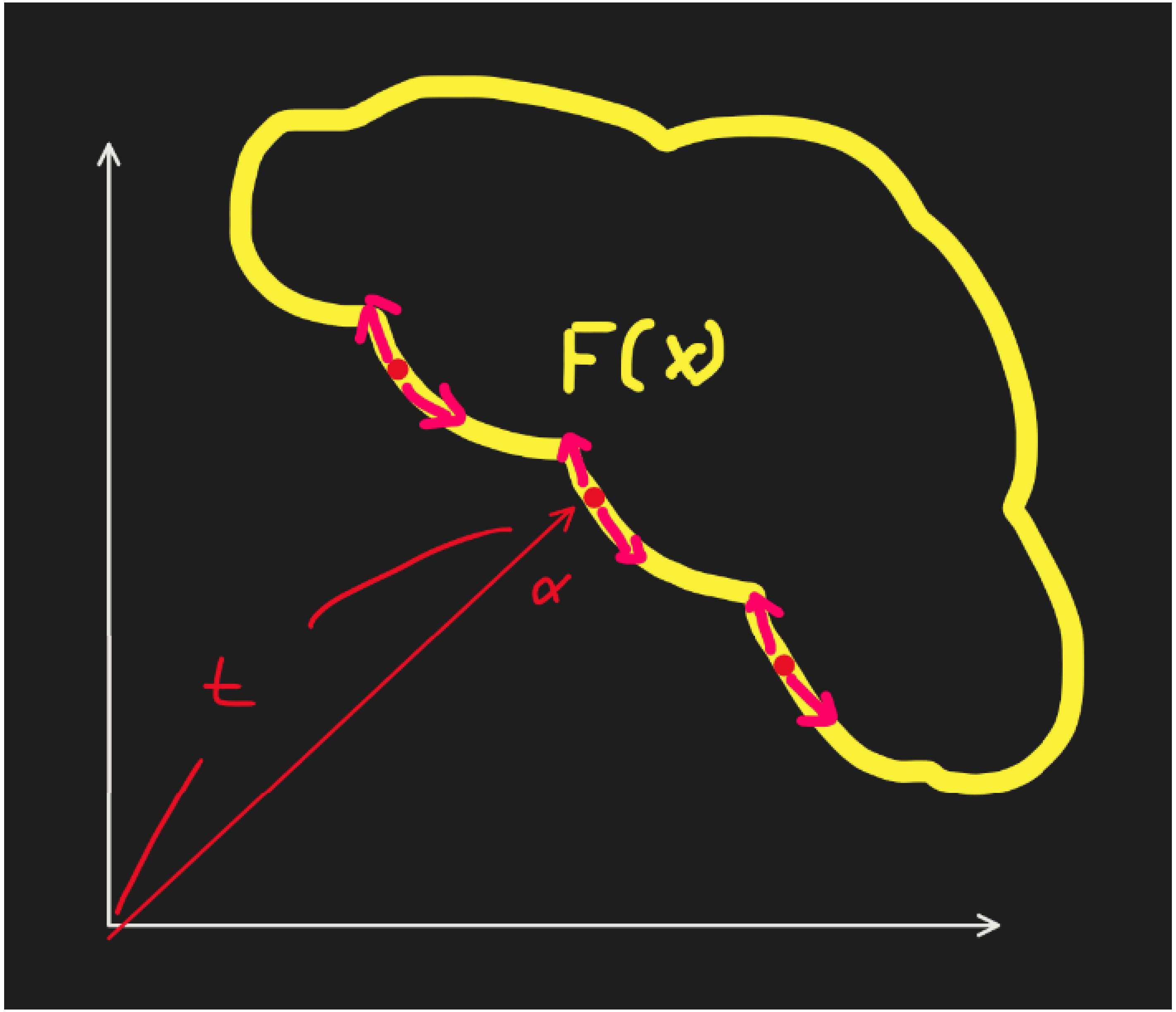


Solution) Pareto Front Approximation

Schulz et al. "Interactive Exploration of Design Trade-Offs"

Algorithm 1 Pareto set discovery given performance metrics F and design constraints that define $\mathcal{X}$.

procedure ParetoFrontDiscovery($\mathcal{X}$, F)
    B: performance buffer array
    $B(i) \leftarrow \emptyset, \forall i$
    do
        $\mathbf{x}_s^0, \ldots, \mathbf{x}_s^{N_s} \leftarrow$ stochasticSampling(B, F, $\mathcal{X}$)
        for each $\mathbf{x}_s^i$ **do**
            $D(\mathbf{x}_s^i) \leftarrow$ selectDirection(B, $\mathbf{x}_s^i$)
            $\mathbf{x}_o^i \leftarrow$ localOptimization($D(\mathbf{x}_s^i)$, F, $\mathcal{X}$)
            $M^i \leftarrow$ firstOrderApproximation($\mathbf{x}_o^i$, F, $\mathcal{X}$)
            updateBuffer(B, $F(M^i)$)
        if buffer not updated on past N_i iterations **then**
            break
    while within computation budget
    return B



F(x)
α
t

Question

But how can we find the intersection point between the ray and the Pareto front if we don't know what the Pareto frontier looks like?

Solution

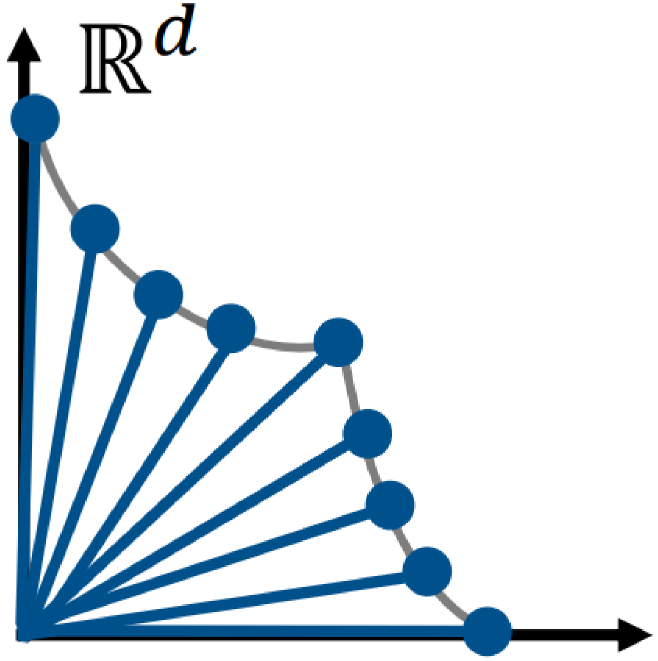
Get some candidate points that seem to be close to the Pareto Front and choose the best point from them.

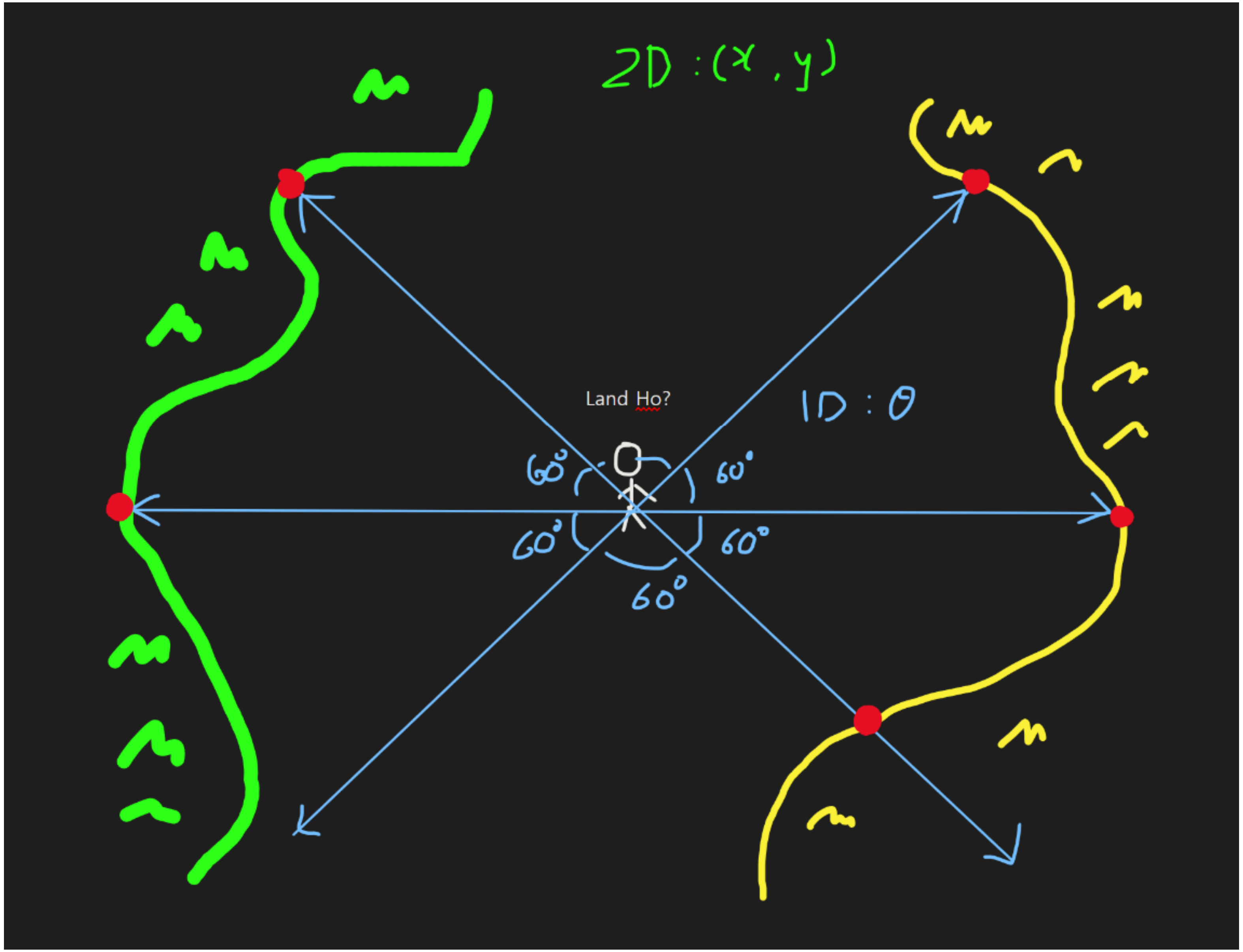
How?)

1. Setting the Performance Buffer (Buffer Cell)
2. Stochastic Sampling
3. Local Optimization

1. Setting the Performance Buffer (Buffer Cell)

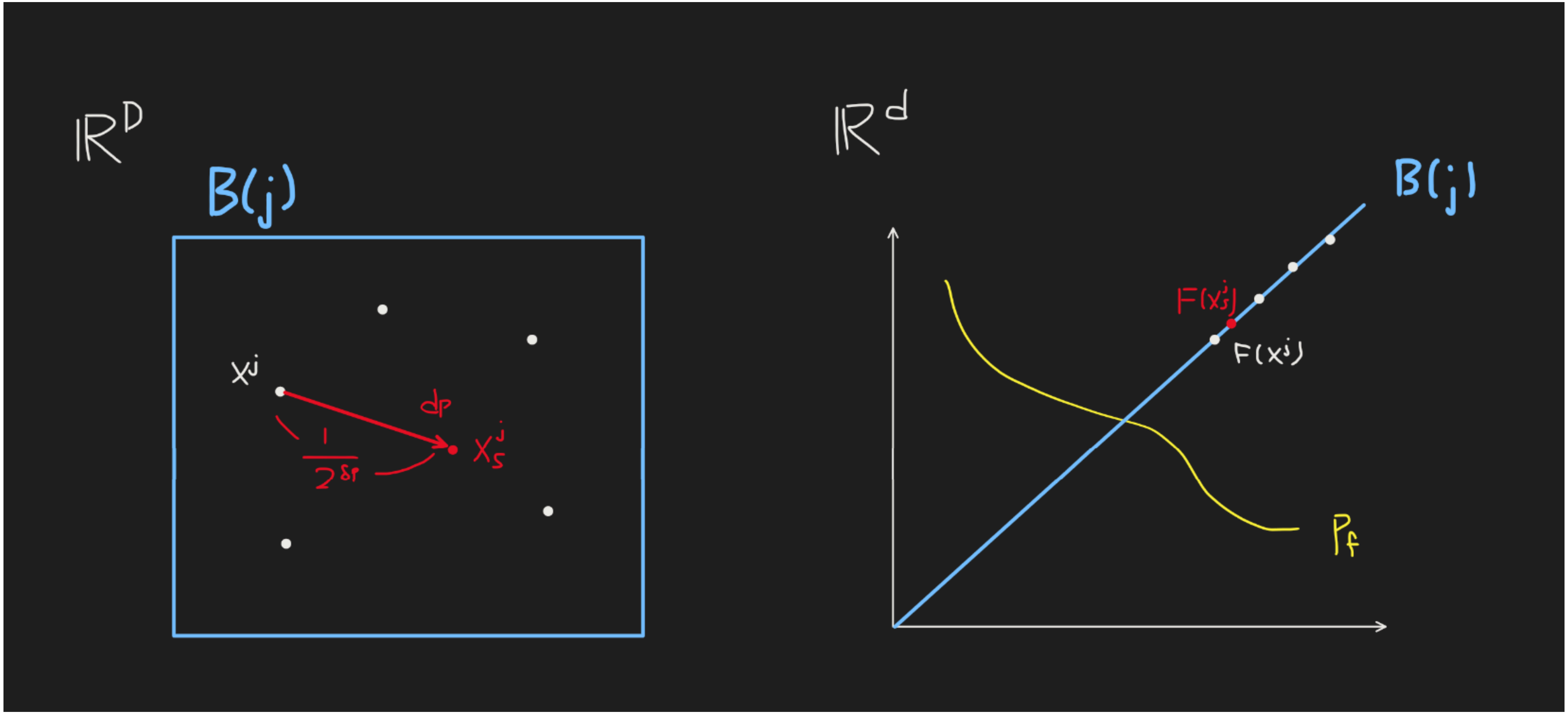
- $(d-1)$-dimensional array discretized using (hyper)spherical coordinates
- Partitions the **performance space** $\mathcal{S} = F(\mathcal{X}) \subseteq \mathbb{R}^d$
- Each buffer cell $B(j)$ contains a list of candidate solutions.
    - Why multiple candidates?)
        - Useful for extracting sparse approximation of the Pareto front
        - top $K(=50)$ candidates




2D : (x , y)
Land Ho?
1D : θ
60°
60°
60°
60°
60°
60°

2. Stochastic Sampling

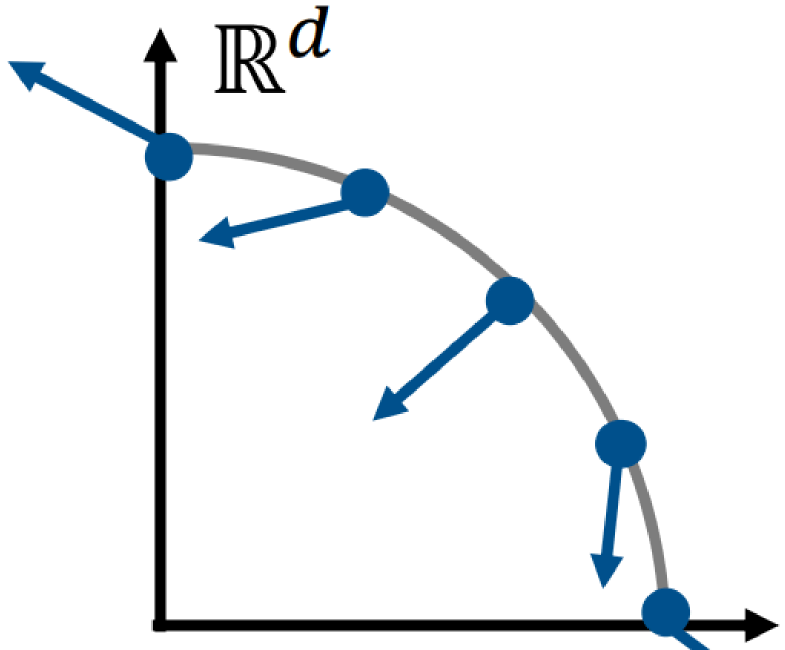
- Goal)
  - Choose N_S samples in the design space $\mathcal{X}$
    - i.e.) $\mathbf{x}_s^i \in \mathcal{X}$ for $i = 1, \cdots, N_S$
- How?)
  - First iteration
    - Uniformly sample from $\mathcal{X}$
  - Rest of the iteration
    - $B(j)$ from the previous state contains the point $\mathbf{x}^j$ with the minimal distance to the origin.
    - Sample $\mathbf{x}_s = \mathbf{x}^j + \dfrac{1}{2^{\delta_p}}\mathbf{d}_p$
      - where
        - δ_p : a uniform random number in $[0, \delta_p]$ (Scaling Factor)
        - $\mathbf{d}_p$: a uniform random unit vector that defines the stochastic direction

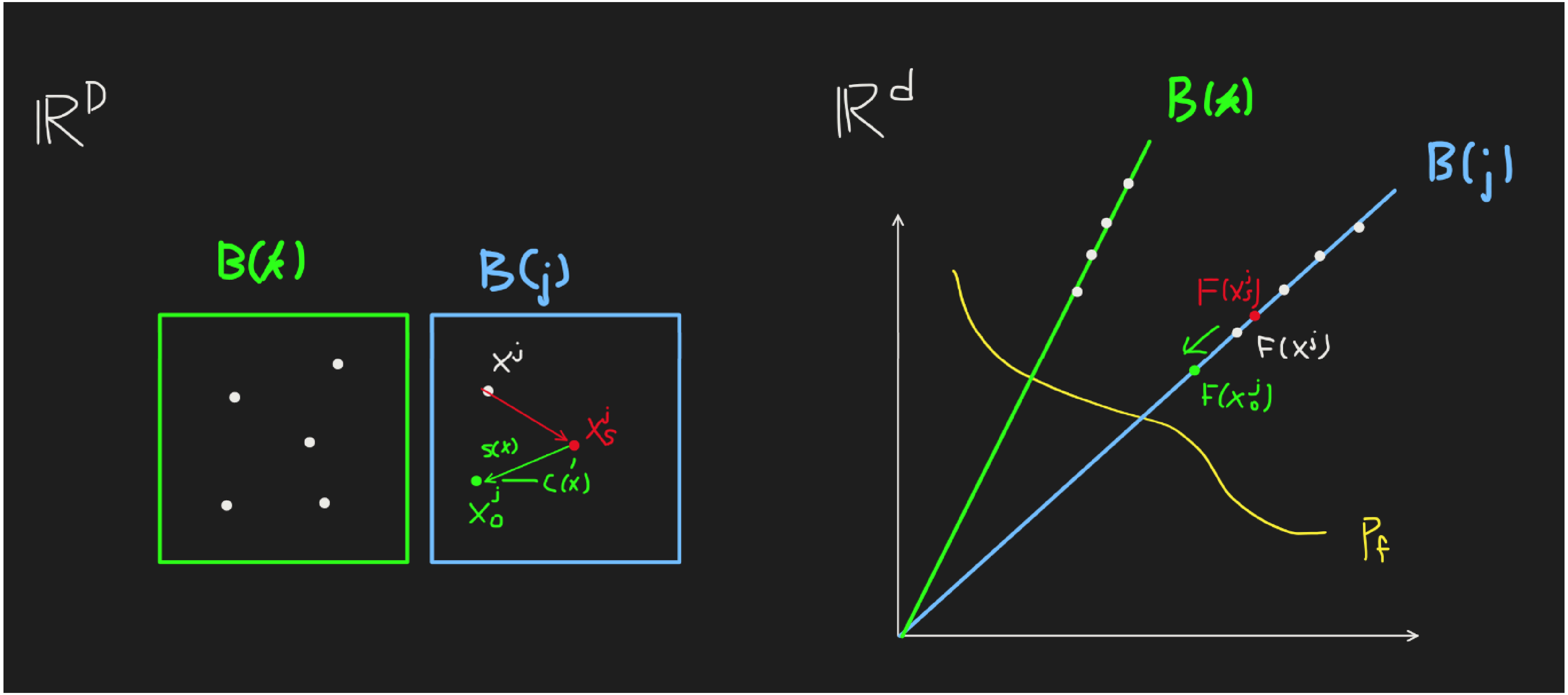


$\mathbb{R}^D$
B(j)
x^j
dp
$\frac{1}{2^{\delta p}}$
x_s^j
$\mathbb{R}^d$
B(j)
$F(x_s^j)$
$F(x^j)$
P_f

3. Local Optimization

- Idea from Compromise Programming (Zeleny)
- Goal)
  - For each $\mathbf{x}_s^i$, find $\mathbf{x}_o^i$ that optimizes for Pareto optimality
  - A scalarization scheme is used to convert the problem into **single** objective problem
    - **Diversification** is essential to avoid having solutions cluster in certain areas, failing to provide a good representation for the shape.





$\mathbb{R}^D$
$B(k)$
$B(j)$
x^j
x_s^j
$s(x)$
$c(x)$
x_0^j
$\mathbb{R}^d$
$B(k)$
$B(j)$
$F(x_s^j)$
$F(x^j)$
$F(x_0^j)$
P_f

- Problem Setting)
  - $\mathbf{x}_o = \arg\min_{\mathbf{x}\in\mathcal{X}} \|F(\mathbf{x}) - \mathbf{z}(\mathbf{x_s})\|^2$
    - where $\mathbf{z}(\mathbf{x_s}) \in \mathbb{R}^d$ is a reference point defined for each sample.
      - This paper uses $\mathbf{z}(\mathbf{x_s}) = \mathbf{x}_s + \mathbf{s}(\mathbf{x_s})C(\mathbf{x}_s)$
        - where
          - $\mathbf{s}(\mathbf{x_s})$ is a unit length search direction pushing $\mathbf{x}_s$ toward the Pareto front $\mathcal{P}$
            - Select the search direction assigned to a cell on the neighborhood of cell j selected uniformly at random, within distance δ_N
          - $C(\mathbf{x}_s) = \delta_s\|\mathbf{x}_s\|$ is a scaling factor depending on the distance to origin
            - Key to **Diversification**!

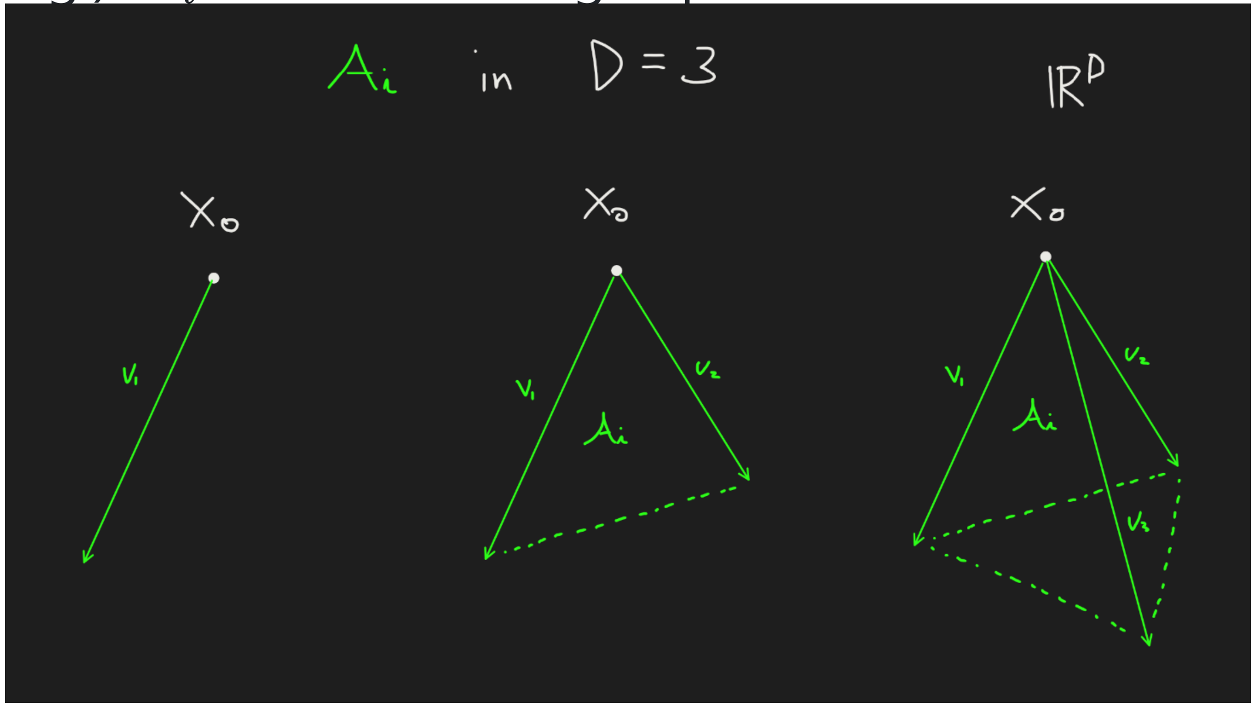
Result)

Although we don't know the actual Pareto Front, we have candidates.

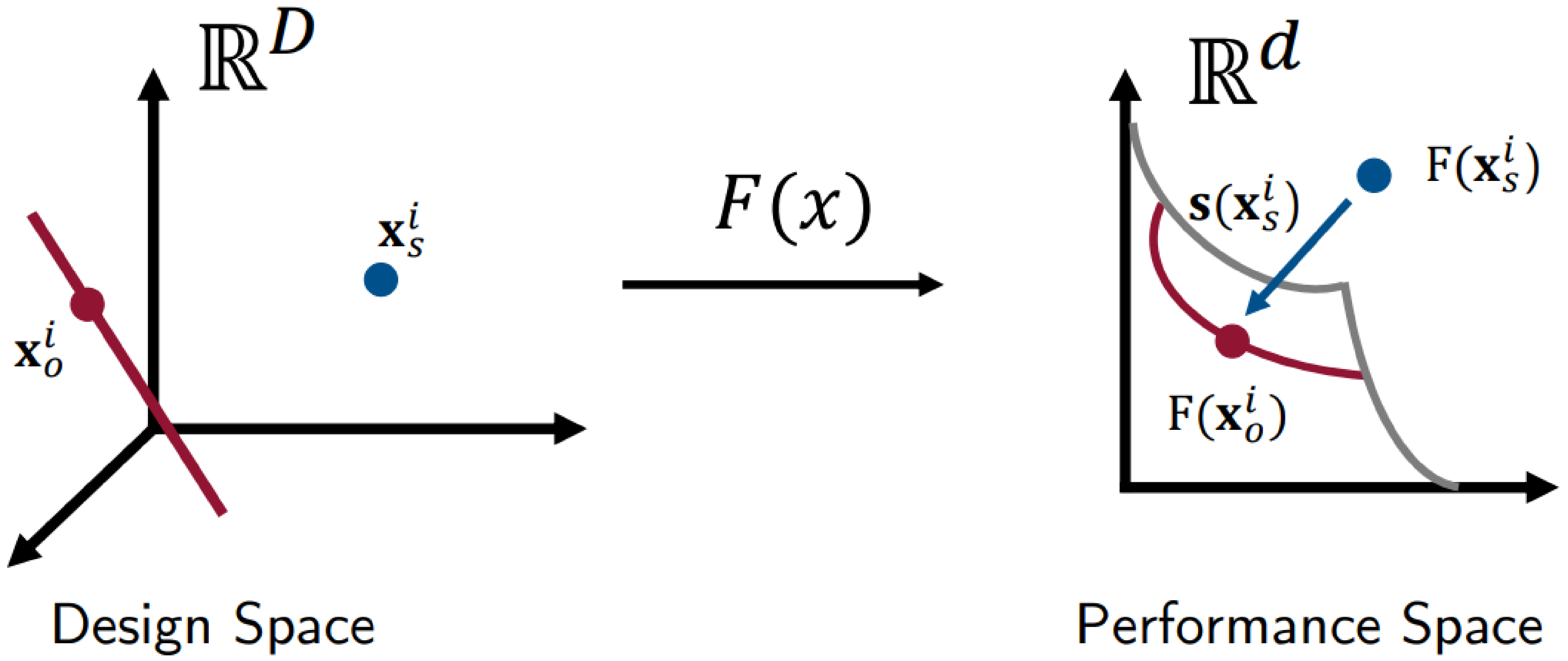
- We have Performance Buffers $B(j)$ that partitions the performance space.
- For each $B(j)$ we have K number of candidate points $\mathbf{x}_o$.

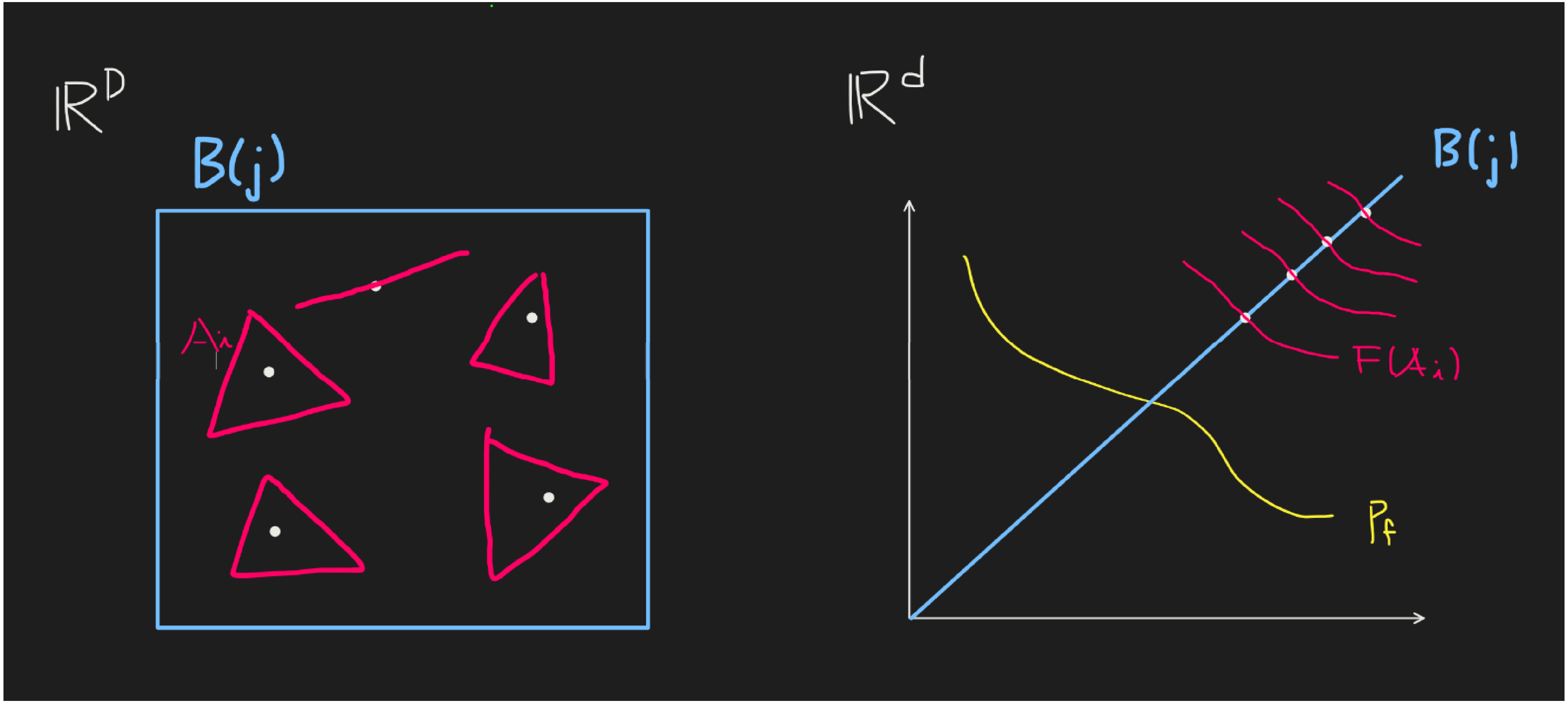
First-Order Approximation of the Pareto Front

- Goal)
  - For each point $\mathbf{x}_o^i$ perform first-order approximation to find an affine subspace $\mathcal{A}_i$ around $\mathbf{x}_o^i$
  - $\mathcal{A}_i$ is stored in a matrix M^i which is defined by the $d-1$ directions for local exploration.
    - e.g.) $\mathcal{A}_i$ in $D=3$ design space



- How?)
  - First-order approximation using the KKT condition.
  - The output is the affine subspace $\mathcal{A}_i$ represented by the point $\mathbf{x}_o$ and the bases represented by the matrix M_i





$\mathbb{R}^D$
$B(j)$
A_i
$\mathbb{R}^d$
$B(j)$
$F(A_i)$
P_f

KKT Conditions

- Assuming that
  - f_i and g_k are continuously differentiable
  - the vectors $\{\nabla g_{k'}(\mathbf{x}^*) | k' \text{ is an index of an active constraint}\}$ are linearly independent
- Then for any solution $\mathbf{x}^*$ to $\min_x f_i(x) \text{ s.t. } x \in \mathcal{X}$
  - there exists dual variables $\alpha \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^K$ s.t.

$$\begin{cases} \mathbf{x}^* \in \mathcal{X} \\ \alpha_i \geq 0,\ \forall i \in \{1, \cdots, d\} \\ \beta_k \geq 0,\ \forall k \in \{1, \cdots, K\} & \text{: Dual Feasibility} \\ \beta_k g_k(\mathbf{x}^*) = 0,\ \forall k \in \{1, \cdots, K\} & \text{: Complementary Slackness Cond} \\ \sum_{i=1}^d \alpha_i = 1 \\ \sum_{i=1}^d \alpha_i \nabla f_i(\mathbf{x}^*) + \sum_{k=1}^K \beta_k \nabla g_k(\mathbf{x}^*) = 0 & \text{: Stationary Condition} \end{cases}$$

KKT Perturbation

- Suppose $\mathbf{x}(t) \in \mathcal{P},\ \forall t \in (-\epsilon, \epsilon)$
  - $\mathbf{x}(t) : (-\epsilon, \epsilon) \to \mathbb{R}^D$ is in the Pareto set in a neighborhood of $t = 0$.
- Taking $\alpha \in \mathbb{R}^D, \beta \in \mathbb{R}^K$ the KKT dual variables corresponding to $\mathbf{x}^* = \mathbf{x}(0)$, we have
  - $H\mathbf{x}'(0) \in \mathrm{Im}\left(DF^\top(\mathbf{x}^*)\right) \oplus \mathrm{Im}\left(DG_{K'}^\top(\mathbf{x}^*)\right)$
    - where
      - $H = \sum_{i=1}^{d} \alpha_i H_{f_i}(\mathbf{x}^*) + \sum_{k=1}^{K'} \beta_k H_{g_k}(\mathbf{x}^*) \in \mathbb{R}^{D \times D}$
        - for $\begin{cases} H_{f_i}(\mathbf{x}^*) = [H_{f_1}(\mathbf{x}^*) \cdots H_{f_d}(\mathbf{x}^*)] \in \mathbb{R}^{D \times D \times d} & \text{Hessian of } F(\mathbf{x}) \\ H_{g_k}(\mathbf{x}^*) = [H_{g_1}(\mathbf{x}^*) \cdots H_{g_K}(\mathbf{x}^*)] \in \mathbb{R}^{D \times D \times K} & \text{Hessian of } G(\mathbf{x}) \end{cases}$
      - $\mathbf{x}'(0)$: the first order differentiation of $\mathbf{x}(t)$ on $t = 0$

- $DF(\mathbf{x}^*) = \begin{bmatrix} \nabla f_1(\mathbf{x}^*)^\top \\ \vdots \\ \nabla f_d(\mathbf{x}^*)^\top \end{bmatrix} \in \mathbb{R}^{D \times d}$: the Jacobian of $F(\mathbf{x})$
- $DG_{K'}(\mathbf{x}^*) = \begin{bmatrix} \nabla g_1(\mathbf{x}^*)^\top \\ \vdots \\ \nabla g_K(\mathbf{x}^*)^\top \end{bmatrix} \in \mathbb{R}^{D \times K}$: the Jacobian of $G(\mathbf{x})$
- $\mathrm{Im}(\cdot)$: the Span of the column space of $\cdot$

First-order or Second order?

- Recall the Second Order Taylor approximation
  - $f(x+x') \approx f(x) + \nabla f(x)^\top x' + \frac{1}{2} x'^\top H_f(x) x'$
- Put $x' = \mathbf{x}'(0)$.
- Then we have the second order term of $\frac{1}{2}\mathbf{x}'(0)^\top \underbrace{H_f(x)\mathbf{x}'(0)}_{\text{here!}}$

Approximation using...

- $H_f(x)$: the curvature of f
- $\mathbf{x}'(0)$: the direction of f

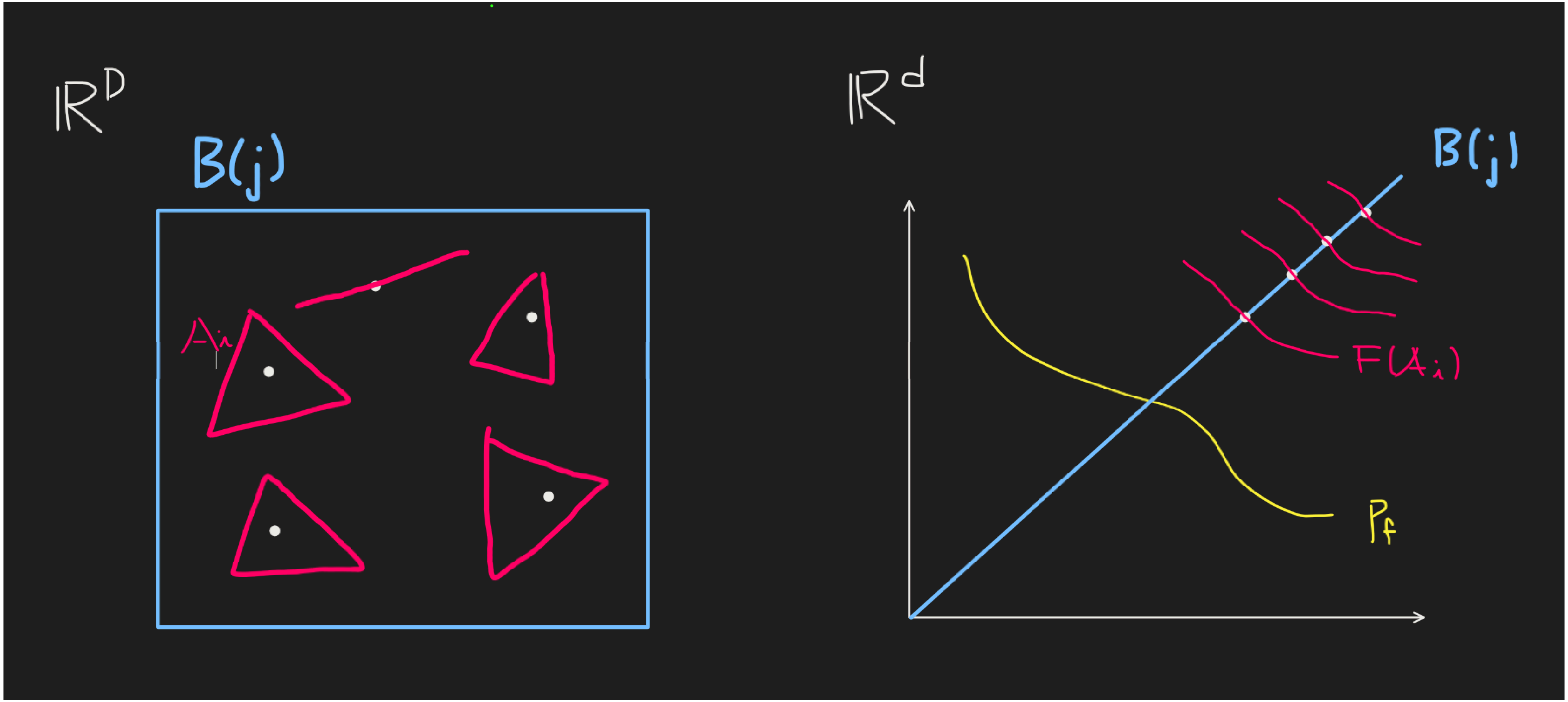
Property) First order approximation reduces the dimensionality!

$\mathbf{x}'(0) \in \mathbb{R}^{d+K'-1}$ where K' is the number of active constraints.

- Meaning)
  - Recall that the design space was $\mathcal{X} \subseteq \mathbb{R}^D$.
  - This approximation reduces the dimension of the direction that $\mathbf{x} \in \mathcal{X}$ must move to $d + K' - 1 < D$ dimension.
    - where
      - d : the number of objective functions
      - K' : the number of active constraints
  - Projecting on the subspace of Hessian and Jacobian.
    - cf.) Dimensionality reduction and PCA

Now, we have K approximations for each performance buffer $B(j)$.

We want to choose the best $\mathcal{A}_j$ for each $B(j)$.



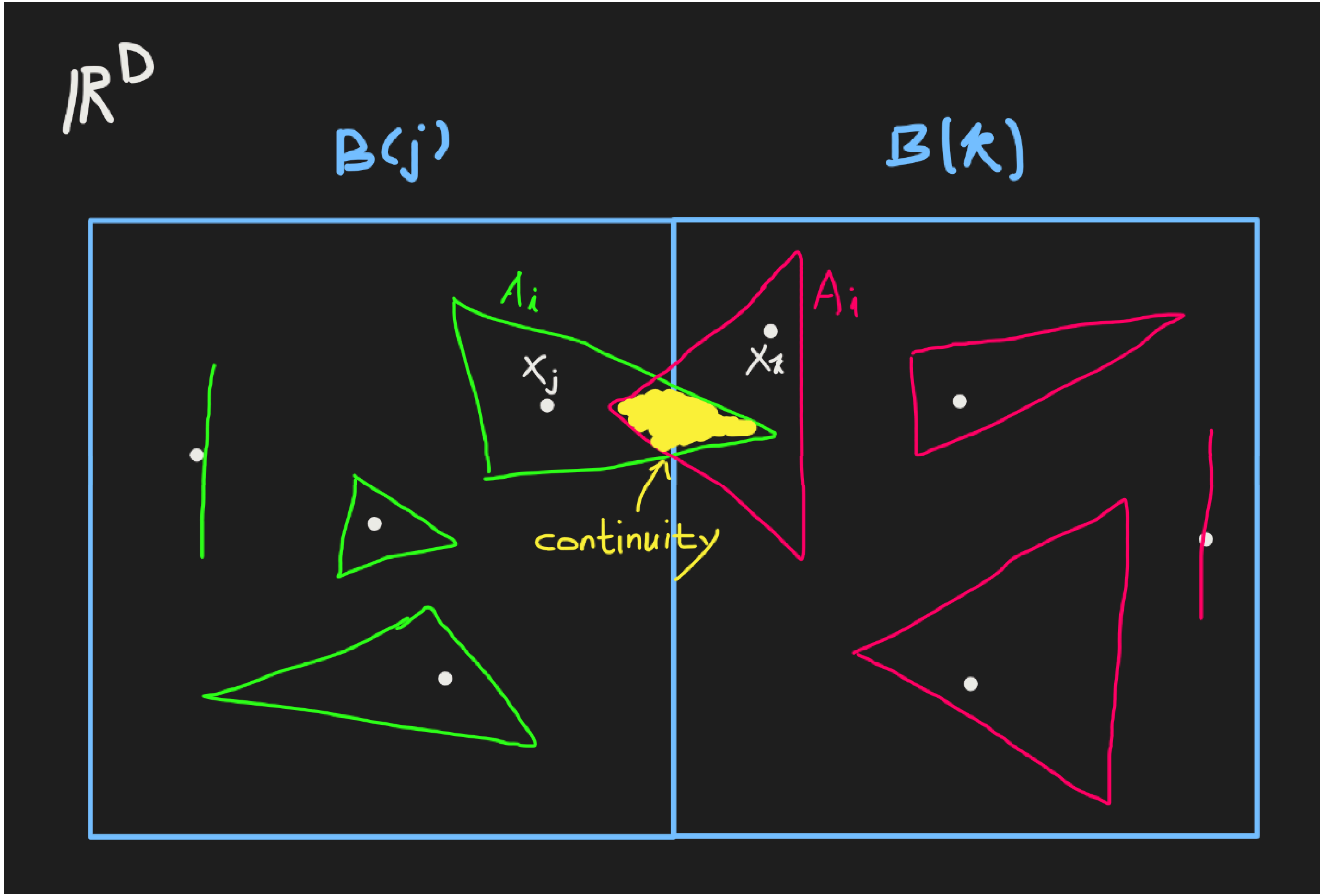
Sparse Approximation

- Goal)
  - Among K solution candidates for each buffer cell $B(j)$, find a single solution $\mathbf{x}_j^*$.
  - Output)
    - $\{(\mathbf{x}_1^*, \mathcal{A}_1^*), \cdots, (\mathbf{x}_{N_S}^*, \mathcal{A}_{N_S}^*)\}$
      - Single solution and a corresponding affine subspace for each buffer cell
        - i.e.) The Pareto Front!
    - $\bigcup_{j=1}^{N_S} \mathcal{A}_j \approx \text{Pareto Front}$

How to choose the best candidate from each Peformance Buffer $B(j)$?

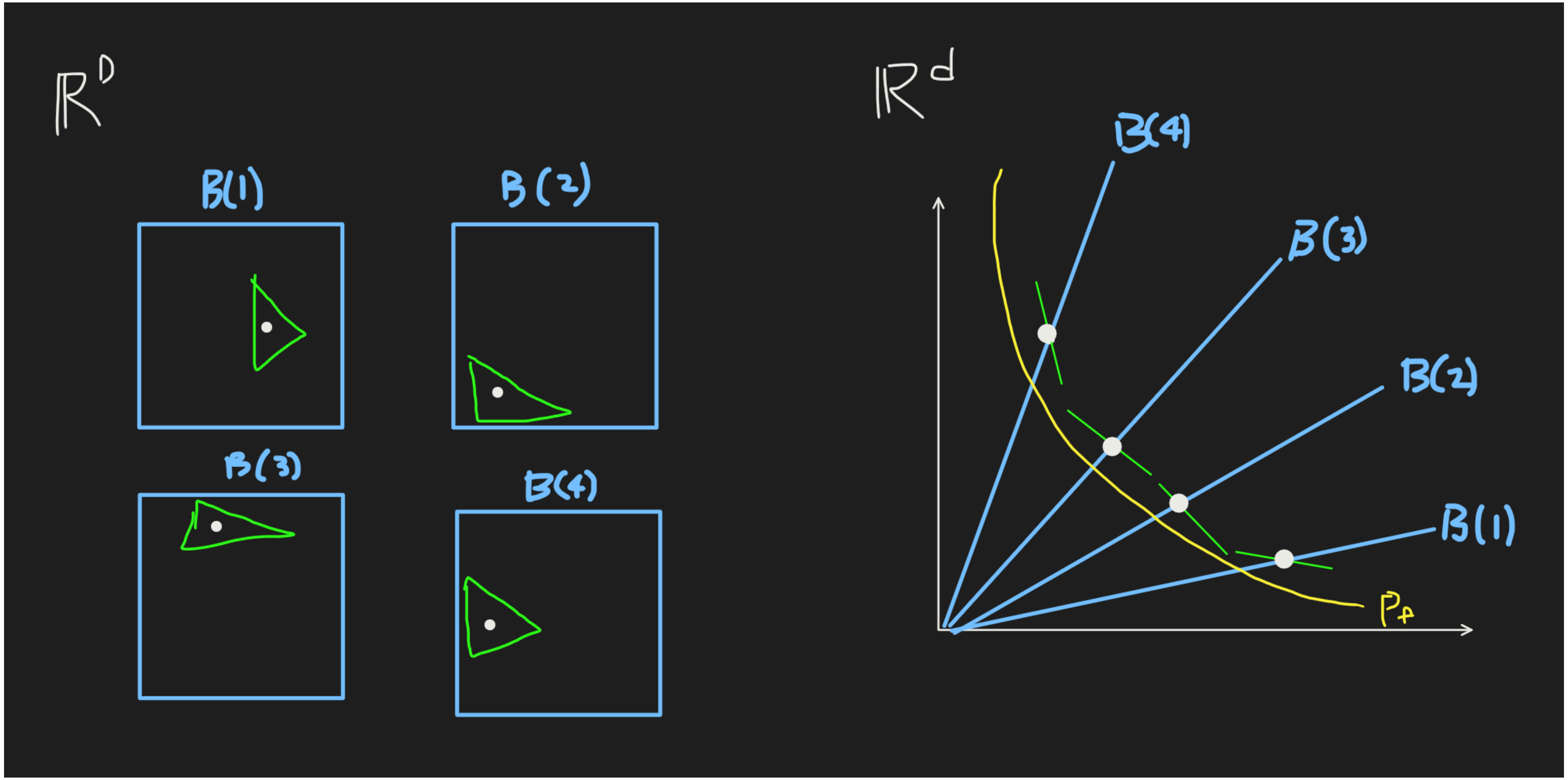
Use Graph-Cut Algorithm

- Criteria)
  - Continuity : adjacent cells share similar affine space $\mathcal{A}_i$
  - Optimality : Minimal distance to Origin
- Optimization using two constraints
  - $E_B(j,k)$: binary term for continuity
    - How?) Lable $\mathcal{A}_i$ with l_i and compare labels between solutions in adjacent buffer cells $B(j)$ and $B(k)$
  - $E_U(j,i)$: unary term for optimality

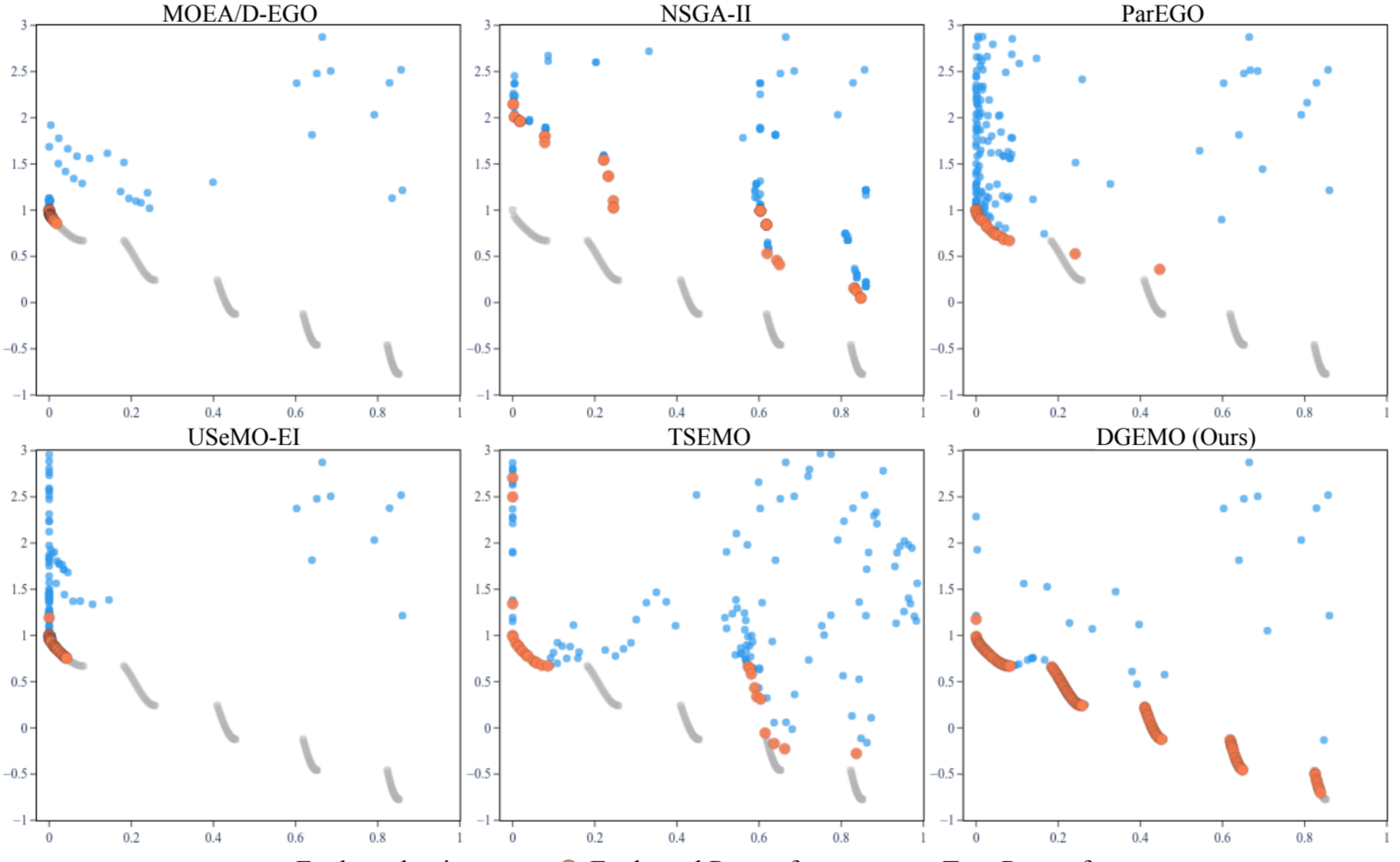


IR^D
B(j)
B(k)
A_i
x_j
A_i
x_k
continuity

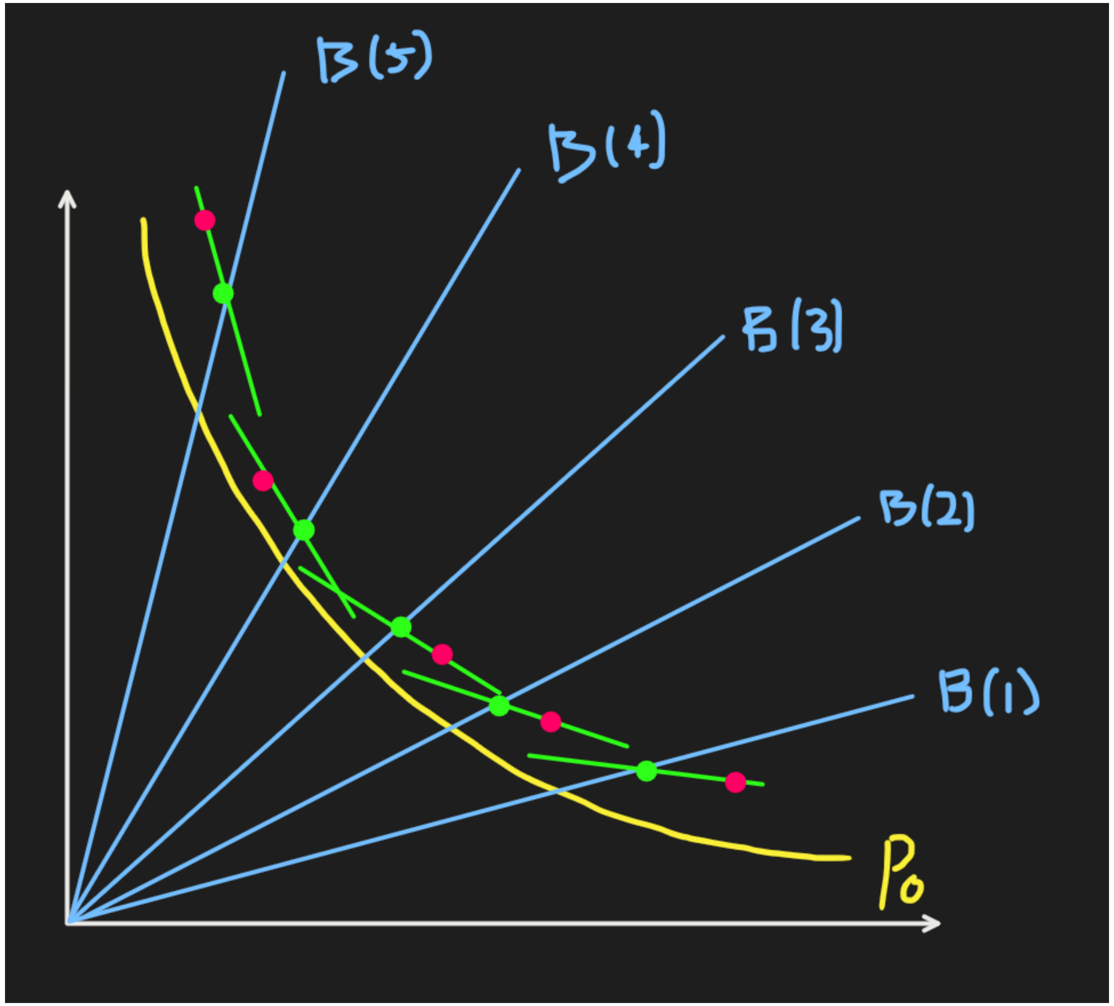
$$\bigcup_{j=1}^{N_S} \mathcal{A}_j \approx \text{Pareto Front}$$

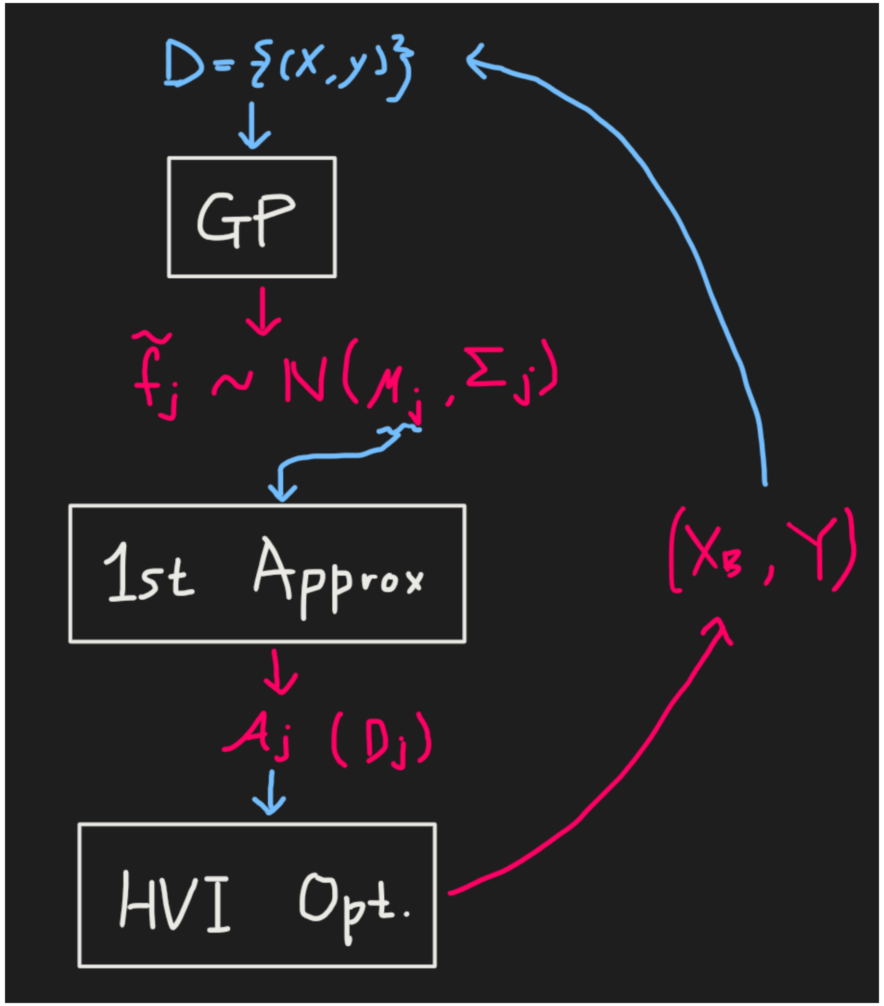


Back to our spoiler...



3-3. HVI maximization w.r.t. Diversity Regions

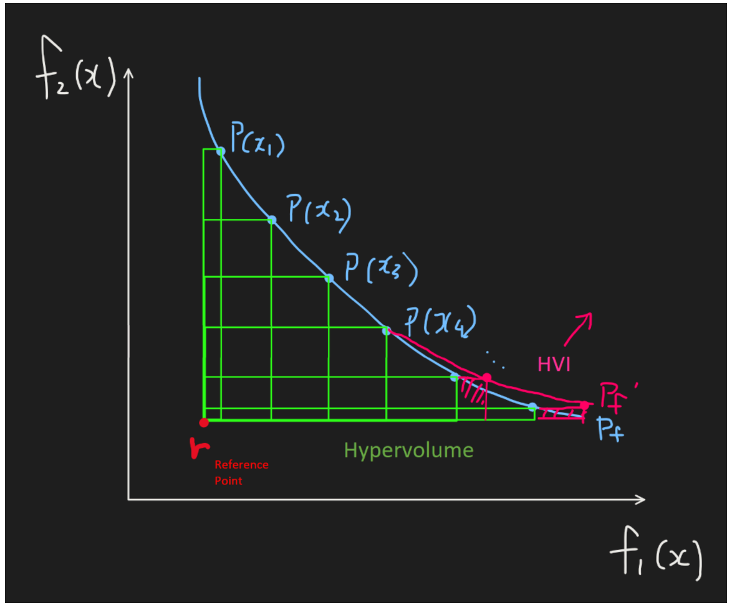


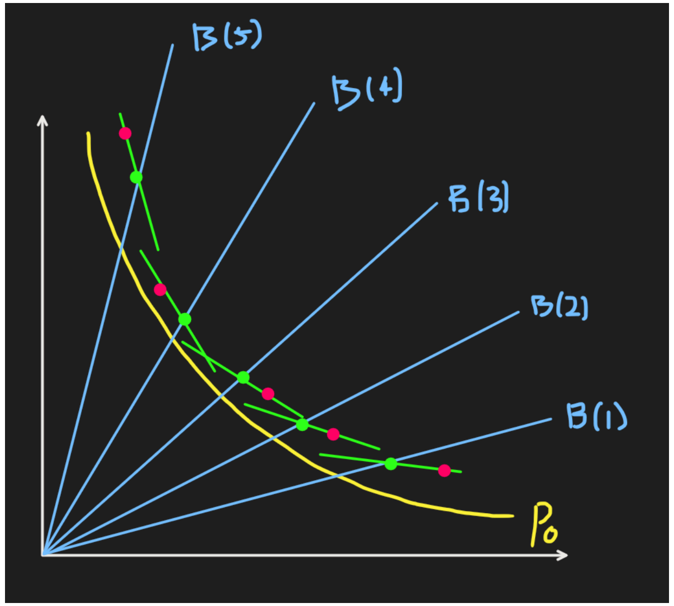


$$\arg\max_{X_B} \mathrm{HVI}(Y_B, \mathcal{P}_f) \quad \text{s.t.} \quad \max_{1\le i\le|\mathcal{D}|} \delta_i(X_B) - \max_{1\le i\le|\mathcal{D}|} \delta_i(X_B) \le 1$$

where

$$\begin{cases} X_B = \{\mathbf{x}_1, \cdots, \mathbf{x}_b\} : \text{a set of b samples in a batch} \\ Y_B = \{\tilde{f}(\mathbf{x}_1), \cdots, \tilde{f}(\mathbf{x}_b)\} \\ \mathcal{P}_f : \text{the current Pareto Front} \\ \delta_i(X) : \text{the number of elements in } X \text{ that belongs to the region } \mathcal{D}_i (= \mathcal{A}_i) \end{cases}$$



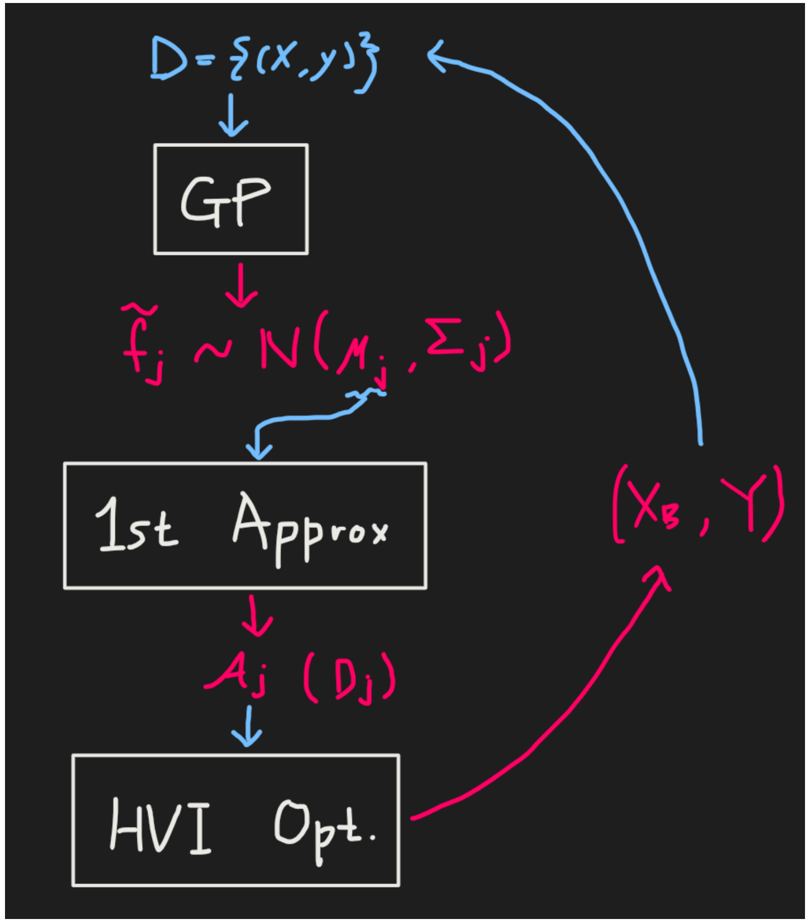


Looks great, but...

Recap : DGEMO

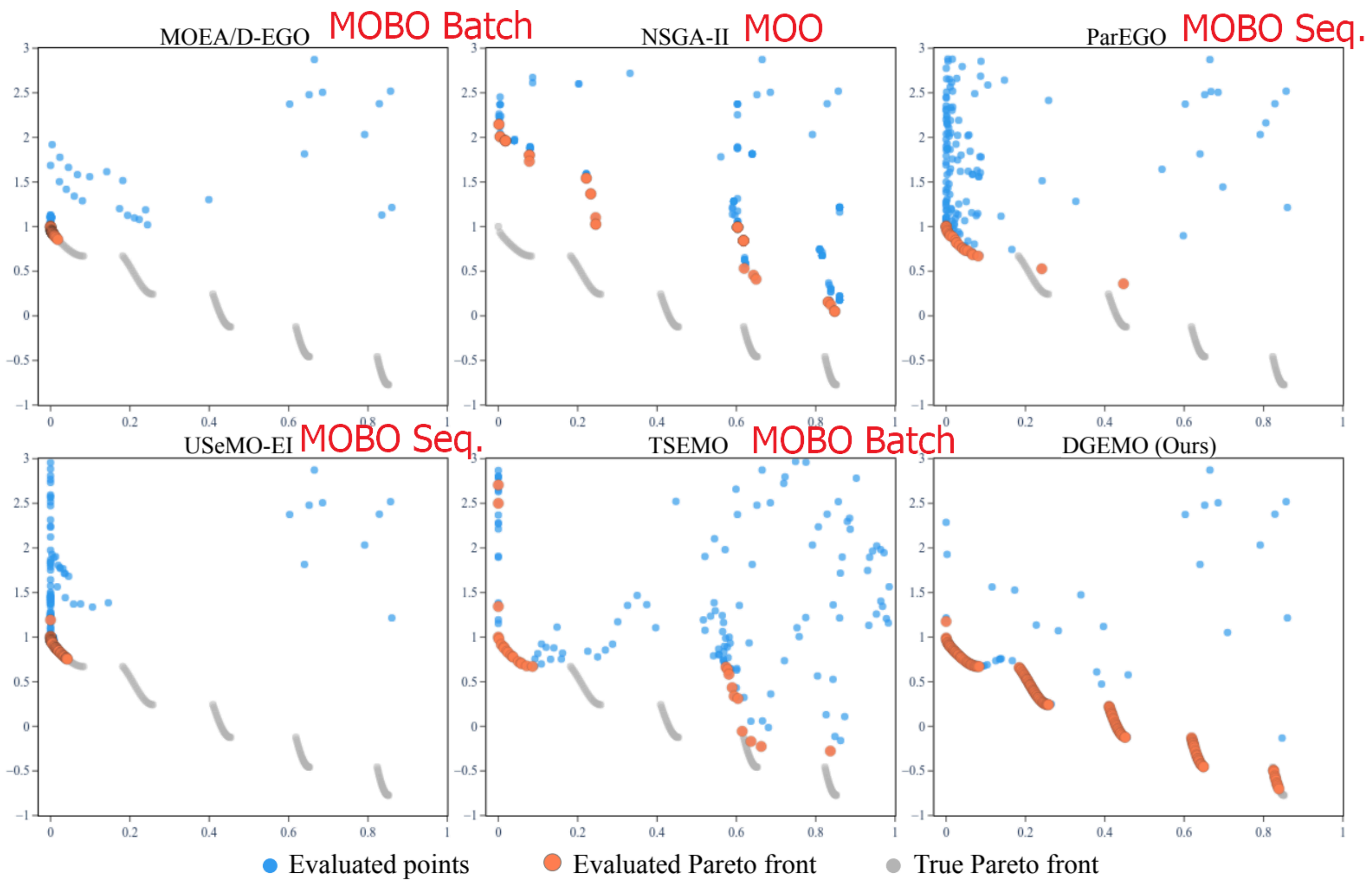
A MOBO model using the Batch Selection Strategy with

- First-order Approximation on Pareto Front
- HVI maximization w.r.t. Diversity Region

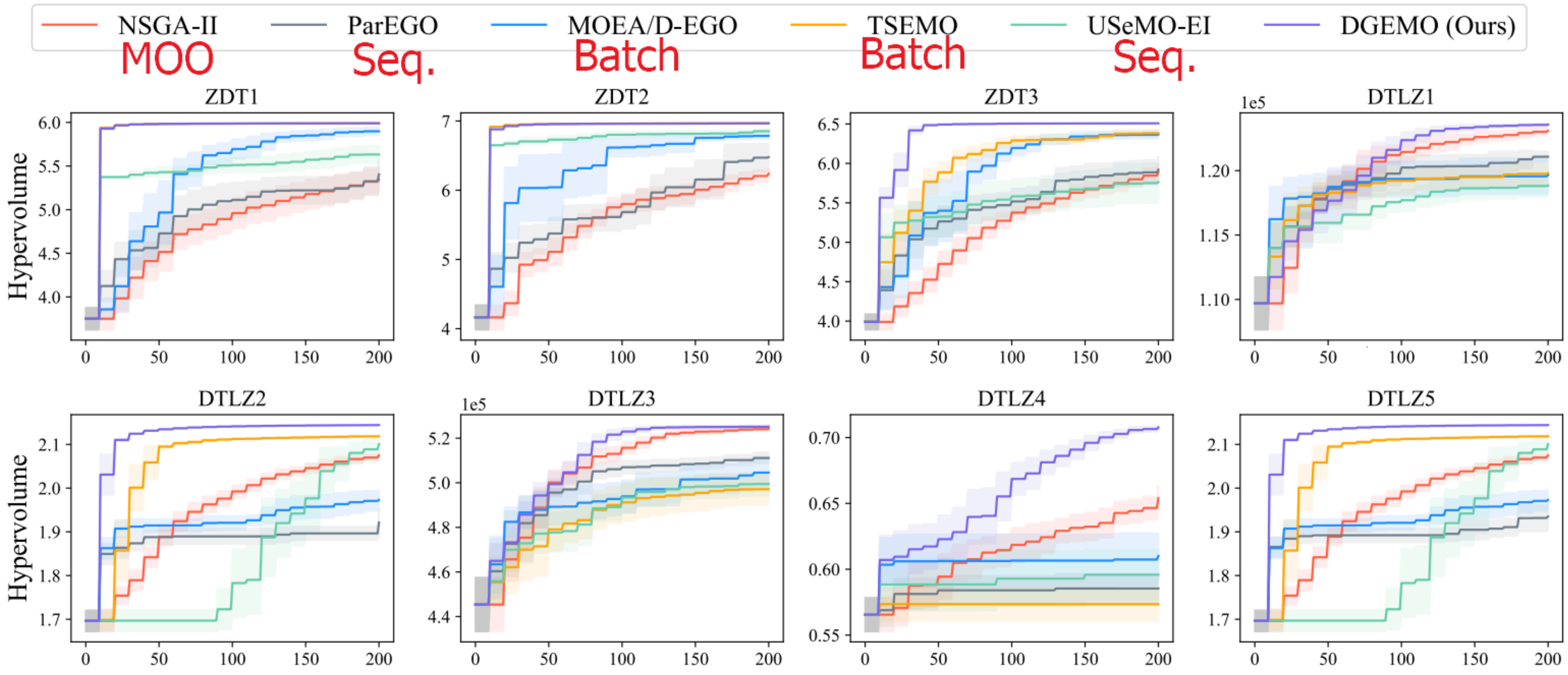


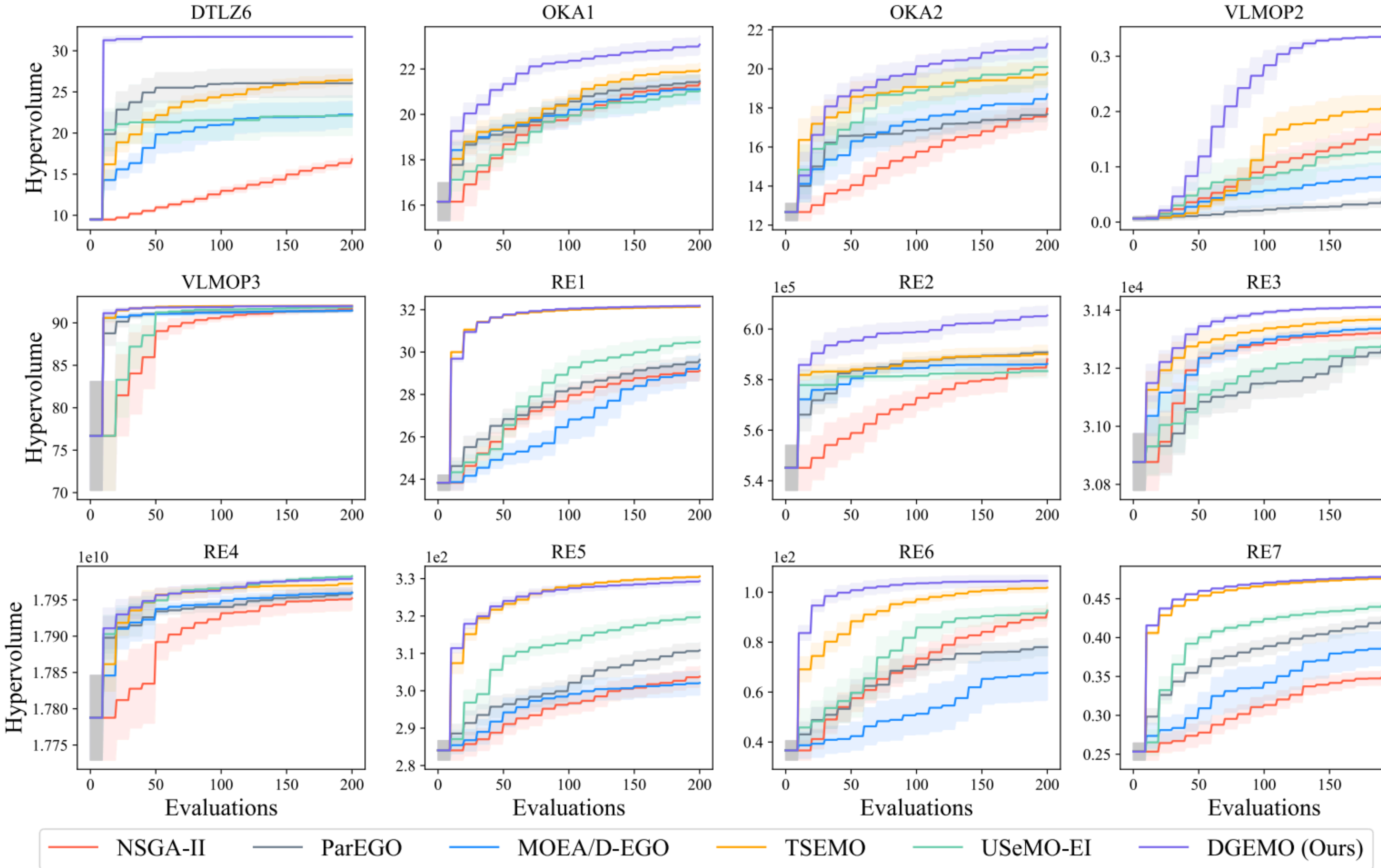
Result & Performance

DGEMO successfully discovered all regions of interest!



DGEMO converged fast!





DTLZ6
OKA1
OKA2
VLMOP2
VLMOP3
RE1
RE2
RE3
RE4
RE5
RE6
RE7
Hypervolume
Evaluations
1e5
1e4
1e10
1e2
NSGA-II
ParEGO
MOEA/D-EGO
TSEMO
USeMO-EI
DGEMO (Ours)

Table 1: Averaged surrogate prediction error of points proposed by different algorithms, including DGEMO, on all benchmark problems. The scale of the error varies across different problems due to the problem definition, as some objectives have extremely high but valid values.

Method	ZDT1	ZDT2	ZDT3	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	OKA1
ParEGO	7.87	2.29	34.9	1.76e+04	6.95	3.83e+04	**30.1**	6.93	304	679
MOEA/D-EGO	76	8.11	106	2.09e+04	5.75	4.09e+04	124	5.75	6.37e+03	3.63e+03
TSEMO	65.9	5.48	331	1.78e+05	41.6	4.12e+05	607	41.6	1.43e+03	544
USeMO-EI	352	1.39	420	2.25e+04	22.3	5.54e+04	92.2	22.3	3.08e+04	1.55e+03
DGEMO (Ours)	**5.25**	**0.933**	**11.6**	**8.24e+03**	**1.59**	**1.5e+04**	71.6	**1.59**	**275**	**233**

Method	OKA2	VLMOP2	VLMOP3	RE1	RE2	RE3	RE4	RE5	RE6	RE7
ParEGO	**58.8**	**5.4**	61.6	5.93	2.86e+16	98.6	**1.95e+08**	575	292	0.597
MOEA/D-EGO	297	9.22	69.4	10.2	1.23e+15	264	1.19e+09	284	1.64e+03	1.26
TSEMO	641	31.9	284	153	1.19e+17	1.32e+03	1.47e+10	229	1.33e+03	0.734
USeMO-EI	1.21e+03	17.5	115	16.4	7.74e+15	386	6.72e+09	**114**	976	1.14
DGEMO (Ours)	140	6.26	**21**	**3.09**	**3.25e+11**	**92.7**	2.13e+08	1.22e+03	**288**	**0.343**

Thank you for your patience!