Multi-Objective Optimization Problems with Well-Conditioned Solutions

Harshal Bharat Dupare
18MA20015

Supervised by
Dr. Geetanjali Panda
Department of Mathematics
Indian Institute of Technology, Kharagpur

Table of Contents

Motivating Problem

Linear Factor Model

$$F_t^{n\times k} = f(R_{t\times d}^{n\times d}) = R_{t\times d}^{n\times d} \times A^{d\times k}$$

$$P_t^{n\times 1} = g(F_t^{n\times k}) = F_t^{n\times k} \times \beta^{k\times 1}$$

$$\begin{tikzcd}
R_{t\times d}^{n\times d} \arrow[rr, "f"] & & F_t^{n\times k} \arrow[d, "g"] \\
R_t^{n\times 1} \arrow[rr, equal, "\mathcal{L}"] \arrow[dr, dashed] & & P_t^{n\times 1} \arrow[dl, dashed] \\
& \mathcal{R} &
\end{tikzcd}$$

$$\boxed{A^{d\times k}(A^{d\times k})^\top = I^{d\times d}}$$

$$\mathcal{R}_{\beta^{k\times 1}, A^{d\times k}} = \frac{1}{S}\sum_{\forall t\in[T,T+S]} \frac{\langle R_t^{n\times 1}, R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times 1}\rangle}{||R_t^{n\times 1}||_2 ||R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times 1}||_2}$$

$$\mathcal{L}_{\beta^{k\times 1}, A^{d\times k}} = \frac{1}{2(T-d+1)}\sum_{\forall t\in[d,T]} ||R_t^{n\times 1} - R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times 1}||_2^2$$

$$\boxed{1 = ||I^{d\times d}|| = ||A^{d\times k}(A^{d\times k})^\top|| \leq ||A^{d\times k}||||(A^{d\times k})^\top|| \leq \alpha}$$

Definitions

Absolute Condition Number: *Absolute Condition Number* is defined for a function $f : X \to Y$, where X and Y have a norm $||.||$ defined over them. Denoted as follows:

$$cond_{abs}(f, x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta f(x)||}{||\delta x||} \tag{1.4}$$

Relative Condition Number: *Relative Condition Number* is defined for a function $f : X \to Y$, where X and Y have a norm $||.||$ defined over them. Denoted as follows:

$$cond_{rel}(f, x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta f(x)||/||f(x)||}{||\delta x||/||x||} \tag{1.5}$$

Induced Norm: Given a Matrix Transform $T^{m \times k} : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times k}$ and a norm $|| \cdot ||$ defined over $\mathbb{R}^{n \times m}$ and $\mathbb{R}^{n \times k}$ we define norm of the matrix transform as

$$||T^{m \times k}|| = \sup_{X \in \mathbb{R}^{n \times m}/\{\bar{0}\}} \frac{||X \times T^{m \times k}||}{||X||} \tag{1.6}$$

Generalization

Conditioned Functional Multi-Objective Optimization Problem:

$$\min_{f \in \mathcal{F}} \; Agg_1_{\forall x \in X_1, y \in Y_1, y \equiv x} \; \mathcal{L}(y, f(x)) \tag{1.7}$$

$$\max_{f \in \mathcal{F}} \; Agg_2_{\forall x \in X_2, y \in Y_2, y \equiv x} \; \mathcal{R}(y, f(x)) \tag{1.8}$$

including other problems specific objectives, but it must have atleast one of the 1.9, 1.10 as objective

$$\min_{f \in \mathcal{F}} \max_{x \in X_3} cond_{abs}(f, x) \tag{1.9}$$

$$\min_{f \in \mathcal{F}} \max_{x \in X_4} cond_{rel}(f, x) \tag{1.10}$$

or is subject to at least one of the 1.11, 1.12 as constraints

$$\max_{x \in X_5} cond_{abs}(f, x) \leq \alpha \tag{1.11}$$

$$\max_{x \in X_6} cond_{rel}(f, x) \leq \alpha \tag{1.12}$$

$f : X \to Y \approx f^* : X \to Y$

$X_i \subseteq X, i \in [6]$

$Y_i \subseteq Y, X_i \equiv Y_i, i \in [2]$

$\mathcal{F} \equiv \mathcal{F}_\theta$

$\mathcal{R} : Y \times Y \to \mathbb{R}$, Reward function

$\mathcal{L} : Y \times Y \to \mathbb{R}$, Loss function

$\alpha \in \mathbb{R}^+$

Other Applications

$$\begin{tikzcd}
(x, y, y') \arrow[r, "\mathcal{A}_\epsilon"] \arrow[d, dash, dashed] \arrow[dr, dash, dashed] & \delta x \arrow[d, dash, dashed] \\
x \arrow[d, "\mathcal{N}"] & x + \delta x \arrow[d, "\mathcal{N}"] \\
y & y'
\end{tikzcd}$$

Trained neural network, $\mathcal{N} : X \to Y$

$\alpha > 0$, $||\mathcal{N}(x + \delta x) - \mathcal{N}(x)|| = ||\delta\mathcal{N}(x)|| \leq \alpha||\delta x||$

$$\max_{x \in X} cond_{abs}(\mathcal{N}, x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta\mathcal{N}(x)||}{||\delta x||} \leq \alpha$$

$\mathcal{A}_\epsilon$, The adversary with input (x, y, y')

outputs the perturbation δx, such that $||\delta x|| \leq \epsilon$

$$\frac{||\delta\mathcal{N}(x)||}{\alpha} \leq ||\delta x|| \longrightarrow \epsilon < \frac{||\delta\mathcal{N}(x)||}{\alpha}$$

Multi-Objective Optimization Methods

Pareto Optimal: A point, $x^* \in X$, is Pareto optimal iff there does not exist another point, $x \in X$, such that $F(x) \leq F(x)$, and $F_i(x) < F_i(x)$ for at least one function.

Utopia Point : A point $F^* \in \mathbb{R}^k$ is objective space, is a utopia point iff for each $i = 1, 2..., k$, $F_i^* = \underset{x \in X}{min}\{F_i(x) | x \in X\}$.

No Preference Methods

$$\underset{x \in X}{min} ||\hat{F}(x) - \hat{F}^*||$$

A Priori Methods

ϵ-Constraint Method

A Posteriori Methods

Evolutionary Algorithms (EA)

Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

Particle Swarm Optimization (PSO)

Findings on Condition Number

$$\kappa_2(A) = ||A||_2||A^{-1}||_2 \tag{2.4}$$

$$\min_{A} \kappa_2(A) \tag{2.5}$$

2.5 is euqivalent to minimizing the following objective

$$\min_{A} \lambda_1(A) - \kappa_2(\bar{A})\lambda_n(A) \tag{2.6}$$

$|| \cdot ||_F$ is Frobenius norm and $\sigma_i, i \in [n]$ are the singular value of A in decreasing order of their magnitude.

$$||A||_F = \sqrt{\sum_{i\in[n]} \sum_{j\in[m]} A_{ij}^2} \tag{2.7}$$

another interesting relation between $\kappa_2(A)$ and that of $\kappa_F(A)$ i.e. the condition number induced by Frobenius norm [8][14].

$$\frac{\kappa_F(A)}{n} \leq \kappa_2(A) \leq \kappa_F(A) \tag{2.8}$$

Pierre Maréchal and Jane J. Ye in their paper Optimizing Condition Numbers. They also proved that that the problem of minimizing the condition number is non-smooth and non-convex optimization problem.

Theoretical Results

Condition Number over Composition of Function

$$cond_{abs}(f,x) \leq \lim_{\delta \to 0} \sup_{||\delta z|| \leq \delta} \frac{||\delta g(z)||}{||\delta z||} \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta h(x)||}{||\delta x||}$$

$$cond_{abs}(f,x) \leq cond_{abs}(g, h(x)) \times cond_{abs}(h,x)$$

L_2 Regularization & Absolute Condition Number

$$\min_{A \in \mathbb{R}^{1 \times d}} \frac{1}{n} \sum_{i \in [n]} ||AX_i - Y_i||_2 + \lambda ||A||_2^2$$

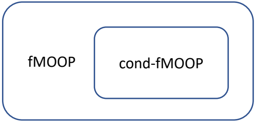
$$\min_{A \in \mathbb{R}^{1 \times d}} \frac{1}{n} \sum_{i \in [n]} ||AX_i - Y_i||_2$$

$$\min_{A \in \mathbb{R}^{1 \times d}} \max_{x \in X_3} cond_{abs}(f,x)$$

$$cond_{abs}(f,x) = \lim_{\epsilon \to 0} \sup_{||\delta x||_F \leq \epsilon} \frac{||A\delta x||_F}{||\delta x||_F}$$

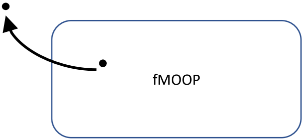
Questions on cond-fMOOP

1. For a solvable $\mathcal{P} \in$ fMOOP and $\mathcal{P}' \in$ cond-fMOOP, derived from $\mathcal{P}$. Is $\mathcal{P}'$ solvable?
2. For $\mathcal{P} \in$ cond-fMOOP over X and subsets $X_5, X_6 \subseteq X$. How to determine minimum value of α for which $\mathcal{P}$ is solvable?
3. For a $\mathcal{P} \in$ cond-fMOOP, if we transform the condition number restriction using bounds, call the new problem $\mathcal{P}'$. How close are the solutions of $\mathcal{P}'$ and $\mathcal{P}$?



Knowing the difference

It may not be possible to make α arbitrary small



Observations

$$\mathcal{O}_d^k = \{A | AA^\top = I_d, A \in \mathbb{R}^{d \times k}\}$$
$$\delta\mathcal{O}_d^k = \{\delta | \delta = A - B, A, B \in \mathcal{O}_d^k\}$$

Properties of $\mathcal{O}_d^k$

1. $A \in \mathcal{O}_d^k \implies -A \in \mathcal{O}_d^k$
2. $A \in \mathcal{O}_d^k$ and $Q \in \mathcal{O}_k^k$ we have $AQ \in \mathcal{O}_d^k$
3. $A \in \mathcal{O}_d^k$ and $Q \in \mathcal{O}_d^d$ we have $QA \in \mathcal{O}_d^k$

Properties of $\delta\mathcal{O}_d^k$

1. $\bar{\mathbf{0}} \in \delta\mathcal{O}_d^k$
2. $A, B \in \mathcal{O}_d^k$ we have $A - B, A + B \in \delta\mathcal{O}_d^k$
3. $\delta \in \delta\mathcal{O}_d^k$ and $Q \in \mathcal{O}_k^k$ we have $\delta Q \in \delta\mathcal{O}_d^k$
4. $\delta \in \delta\mathcal{O}_d^k$ and $Q \in \mathcal{O}_d^d$ we have $Q\delta \in \delta\mathcal{O}_d^k$

$$A_{i+1} = A_i + \Delta A_i$$
$$A_i Q_i \approx A_i + \Delta A_i$$
$$Q_i = (I - T_i)(I + T_i)^{-1}$$
$$2T_i = \Delta A_i^\top A_i - A_i^\top \Delta A_i$$

$$\mathbf{D} = \sum_{\forall t \in [d,T]} (R_t^{n \times d})^\top R_t^{n \times d}$$
$$\mathbf{N} = \sum_{\forall t \in [d,T]} (R_t^{n \times d})^\top R_t^{n \times 1}$$
$$\beta = (A^\top \mathbf{D} A)^{-1} A^\top \mathbf{N}$$

$$\bar{\beta} = (\bar{A}^\top \mathbf{D} \bar{A})^{-1} \bar{A}^\top \mathbf{N}$$
$$\bar{\beta} = (Q^\top A^\top \mathbf{D} A Q)^{-1} Q^\top A^\top \mathbf{N}$$
$$\bar{\beta} = Q^\top (A^\top \mathbf{D} A)^{-1} Q Q^\top A^\top \mathbf{N}$$
$$\bar{\beta} = Q^\top \beta$$

Methodologies

$\mathcal{A}_0[N]$: Baseline Method

$\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \emptyset, \emptyset, -\infty$
for $i \leftarrow 1...N$ **do**
  $M_i \leftarrow random(\mathbb{R}^{d\times k})$
  $A_i \leftarrow$ gram_schmidt_algorithm(M_i) ▷ orthonormalize M_i to A_i
  Compute $\{F_{t,i}^{n\times k}\}$ for A_i over $t \in [d, T]$
  $\beta_i \leftarrow regression(\{R_t^{n\times 1}\}, \{F_{t,i}^{n\times k}\}, t \in [d, T])$
  if $\mathcal{R}_{\beta_i, A_i} > \mathcal{R}_{optimal}$ **then** ▷ update the optimal if better solution found
    $\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta_i, A_i, \mathcal{R}_{\beta_i, A_i}$
return $\beta_{optimal}, A_{optimal}$
complexity: $O(N(dk + d^2k + ndS + k^3)) = O(N(k^3 + ndS))$

$\mathcal{A}_1[N, \mathcal{U}_\mathcal{A}, \mathcal{U}_\beta]$: Orthogonal Property Iterative Scheme (OPI Scheme)

$A_0 \leftarrow random(\mathbb{R}^{d\times k})$
$A_0 \leftarrow$ gram_schmidt_algorithm(A_0) ▷ orthonormalize A_0, as $A_0 \in \mathcal{O}_d^k$
$\beta_0 \leftarrow regression(\{R_t^{n\times 1}\}, \{F_t^{n\times k}[A_0]\}, t \in [d, T])$
for $i \leftarrow 0$ to $N-1$ **do**
  $\Delta A_i \leftarrow \mathcal{U}_\mathcal{A}[\vec{\alpha}](\mathcal{R}_{\beta,A}, \mathcal{L}_{\beta,A}, \beta_i, A_i) - A_i$
  $2T_i \leftarrow \Delta A_i^\top A_i - A_i^\top \Delta A_i$
  $Q_i \leftarrow (I - T_i)(I + T_i)^{-1}$
  $A_{i+1} \leftarrow A_i Q_i$
  $\beta_{i+1} \leftarrow \mathcal{U}_\beta[\vec{\phi}](\mathcal{R}_{\beta,A}, \mathcal{L}_{\beta,A}, \beta_i, A_i, A_{i+1})$
return β_N, A_N
complexity: $O(N(|\mathcal{U}_\mathcal{A}[\vec{\alpha}]| + k^2d + k^3 + k^2d + |\mathcal{U}_\beta[\vec{\phi}]|)) = O(N(|\mathcal{U}_\mathcal{A}[\vec{\alpha}]| + k^3 + |\mathcal{U}_\beta[\vec{\phi}]|))$

$\mathcal{A}_2[N, \mathcal{U}_\mathcal{A}, \mathcal{U}_\beta, \mathcal{U}_\mathcal{O}]$: Delayed Orthogonalization Iterative Scheme (DOI Scheme)

$A_0 \leftarrow random(\mathbb{R}^{d\times k})$
$\beta_0 \leftarrow regression(\{R_t^{n\times 1}\}, \{F_t^{n\times k}[A_0]\}, t \in [d, T])$
for $i \leftarrow 0$ to $N-1$ **do**
  $A_{i+1} \leftarrow \mathcal{U}_\mathcal{A}[\vec{\alpha}](\mathcal{R}_{\beta,A}, \mathcal{L}_{\beta,A}, \beta_i, A_i)$
  $\beta_{i+1} \leftarrow \mathcal{U}_\beta[\vec{\phi}](\mathcal{R}_{\beta,A}, \mathcal{L}_{\beta,A}, \beta_i, A_i, A_{i+1})$
$\beta^*, A^* \leftarrow \mathcal{U}_\mathcal{O}[\vec{\lambda}](\mathcal{R}_{\beta,A}, \mathcal{L}_{\beta,A}, \beta_N, A_N)$
return β^*, A^*
complexity: $O(N(|\mathcal{U}_\mathcal{A}[\vec{\alpha}]| + |\mathcal{U}_\beta[\vec{\phi}]|) + |\mathcal{U}_\mathcal{O}[\vec{\lambda}]|)$

$\mathcal{U}_\mathcal{O}^1[\vec{\lambda}](\mathcal{R}_{\beta,A}, \mathcal{L}_{\beta,A}, \beta^*, A^*)$: Iterative Closing Method

$A \leftarrow random(\mathbb{R}^{d\times k})$
$A \leftarrow$ gram_schmidt_algorithm(A) ▷ orthonormalize A
$\beta \leftarrow regression(\{R_t^{n\times 1}\}, \{F_t^{n\times k}[A]\}, t \in [d, T])$
$\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta, A, \mathcal{R}_{\beta,A}$
for $i \leftarrow 1$ to N **do**
  $\Delta A \leftarrow A^* - A$
  $Q \leftarrow$ ortho_cayley_transformation($\frac{\lambda_1}{2}\Delta A$)
  $A \leftarrow AQ$
  $\beta_{reg} \leftarrow regression(\{R_t^{n\times 1}\}, \{F_t^{n\times k}[A]\}, t \in [d, T])$
  $\Delta\beta \leftarrow (1 - \lambda_3)\frac{\partial \mathcal{R}_{\beta,A}}{\partial \beta} + \lambda_3(\beta_{reg} - beta)$
  $\beta \leftarrow \beta + \lambda_2 \Delta\beta$
  if $\mathcal{R}_{\beta,A} > \mathcal{R}_{optimal}$ **then** ▷ update the optimal if better solution found
    $\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta, A, \mathcal{R}_{\beta,A}$
return $\beta_{optimal}, A_{optimal}$
complexity: $O(k^3 + N(k^3 + ndS)) = O(N(k^3 + ndS))$

Methodologies

$NPM_1[\vec{\mu}]$: No Preference Method 1

$A \leftarrow random(\mathbb{R}^{d \times k})$
$A \leftarrow$ gram_schmidt_algorithm(A)
$\beta \leftarrow regression(\{R_t^{n\times 1}\}, \{F_t^{n\times k}[A]\}, t \in [d, T])$
$\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta, A, \mathcal{R}_{\beta,A}$
for $i \leftarrow 1$ to N **do**
  $\Delta A \leftarrow (1 - \mathcal{R}_{\beta,A})\frac{\partial \mathcal{R}_{\beta,A}}{\partial A} - \mathcal{L}_{\beta,A}\frac{\partial \mathcal{L}_{\beta,A}}{\partial A} - \frac{1}{2}\frac{\partial \|AA^\top - I\|_F^2}{\partial A}$
  $\Delta \beta \leftarrow (1 - \mathcal{R}_{\beta,A})\frac{\partial \mathcal{R}_{\beta,A}}{\partial \beta} - \mathcal{L}_{\beta,A}\frac{\partial \mathcal{L}_{\beta,A}}{\partial \beta}$
  $Q \leftarrow$ ortho_cayley_transformation$(\frac{\mu_1}{2}\Delta A)$
  $A \leftarrow AQ$
  $\beta \leftarrow \beta + \mu_2 \Delta \beta$
  if $\mathcal{R}_{\beta,A} > \mathcal{R}_{optimal}$ **then**
    $\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta, A, \mathcal{R}_{\beta,A}$
return $\beta_{optimal}, A_{optimal}$
complexity: $O(N(ndS + k^3))$

$NPM_2[\vec{\mu}]$: No Preference Method 2

$A \leftarrow random(\mathbb{R}^{d \times k})$
$A \leftarrow$ gram_schmidt_algorithm(A)
$\beta \leftarrow regression(\{R_t^{n\times 1}\}, \{F_t^{n\times k}[A]\}, t \in [d, T])$
$\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta, A, \mathcal{R}_{\beta,A}$
for $i \leftarrow 1$ to N **do**
  $\Delta A \leftarrow (1 - \mathcal{R}_{\beta,A})\frac{\partial \mathcal{R}_{\beta,A}}{\partial A} - \mathcal{L}_{\beta,A}\frac{\partial \mathcal{L}_{\beta,A}}{\partial A} - \frac{1}{2}\frac{\partial \|AA^\top - I\|_F^2}{\partial A}$
  $\Delta \beta \leftarrow (1 - \mathcal{R}_{\beta,A})\frac{\partial \mathcal{R}_{\beta,A}}{\partial \beta} - \mathcal{L}_{\beta,A}\frac{\partial \mathcal{L}_{\beta,A}}{\partial \beta}$
  $A \leftarrow A + \mu_1 \Delta A$
  $\beta \leftarrow \beta + \mu_2 \Delta \beta$
  if $\mathcal{R}_{\beta,A} > \mathcal{R}_{optimal}$ **then**
    $\beta_{optimal}, A_{optimal}, \mathcal{R}_{optimal} \leftarrow \beta, A, \mathcal{R}_{\beta,A}$
return $\beta_{optimal}, A_{optimal}$
complexity: $O(N(ndS + k^2))$

Methodologies

NSGA-II($N, G, p_c, p_m, \mathcal{C}, \mathcal{M}$)

$P \leftarrow$ random_candidate_solutions(N)
Evaluate the fitness of each candidate solution in P ▷ fitness : objective fuction
for $t \leftarrow 1$ to G **do**
  $C \leftarrow$ get_offspring_population($P, N, \mathcal{C}, \mathcal{M}$)
  $F_1, F_2, ..., F_t \leftarrow$ non_dominated_sorting($P \cup C$) ▷ F_i is the i'th front
  $P, i \leftarrow \emptyset, 1$
  while $|P| + |F_i| \leq N$ **do**
    Calculate crowding distance for each solution in F_i
    $P \leftarrow P \cup F_i$
    $i \leftarrow i + 1$
  if $|P| < N$ **then**
    Sort F_i in descending order of crowding distance
    $P \leftarrow P \cup F_i[1 : N - |P|]$
return P

Methodologies

MOPSO$(N, T_{max}, \theta, \omega, c_1, c_2, p_m, \epsilon)$ Multi-Objective Particle Swarm Optimization

Initialize the population of particles randomly:
$\{\vec{x}_i\}, \{\vec{v}_i\} \leftarrow$ initialize_position_and_velocity(N)
$\{\vec{f}(\vec{x}_i)\} \leftarrow$ objective_function$(\{\vec{x}_i\})$
$\{\vec{p}_i\}, \{\vec{f}(p_i)\} \leftarrow \{\vec{x}_i\}, \{\vec{f}_i(\vec{x}_i)\}$ ▷ $\vec{p}_i, \vec{f}(\vec{p}_i)$: best positions and objective for each $\vec{x}_i$
$A \leftarrow$ nondominated$(\{\vec{x}_i\})$ ▷ A is the archive of non-dominated solutions
$\{\vec{d}(\vec{a}_i)\} \leftarrow$ crowding_distance(A) ▷ $d(\vec{a}_i)$: average distance to $\vec{a}_i$'s neighbors in A
for $t \leftarrow 1$ to $T_{\max}$ **do**
  for $i \leftarrow 1$ to N **do**
    $\vec{p}_g \leftarrow$ get_global_best$[\theta](A)$ ▷ global best w.p. *theta* else random point from A
    $v_i \leftarrow$ update_velocity$[\omega, c_1, c_2](\vec{x}_i, \vec{v}_i, \vec{p}_i, \vec{p}_g)$
    $\vec{x}_i \leftarrow$ update_position$[\epsilon](\vec{x}_i, \vec{v}_i)$
    if True with probability p_m **then**
      $\vec{x}_i, \vec{v}_i \leftarrow$ mutate$(\vec{x}_i, \vec{v}_i)$ ▷ Mutate the solutions particle $\vec{x}_i, \vec{v}_i$
    $\vec{f}(\vec{x}_i) \leftarrow$ objective_function$(\vec{x}_i)$ ▷ Update the objective of particle $\vec{x}_i$
  $A \leftarrow$ nondominated_merge$(A, \{\vec{x}_i\})$ ▷ Update the nondominated solutions
  $\{\vec{d}(\vec{a}_i)\} \leftarrow$ crowding_distance[**True**](A)
  $\{\vec{d}(\vec{x}_i)\} \leftarrow$ crowding_distance[**False**]$(\{\vec{x}_i\})$
  $\{\vec{p}_i\}, \{\vec{f}(p_i)\} \leftarrow$ update_personal_best$(\{\vec{p}_i\}, \{\vec{f}(p_i)\}, \{\vec{x}_i\}, \{\vec{f}(x_i)\}, \{\vec{d}(x_i)\})$
return A

update_position$[\epsilon](\vec{x}_i, \vec{v}_i)$

$(\beta_i, A_i) \leftarrow x_i$
$(\Delta\beta_i, \Delta A_i) \leftarrow v_i$
$Q_i \leftarrow$ ortho_cayley_transformation$(\frac{\epsilon}{2}\Delta A_i)$
return $(\beta + \epsilon\Delta\beta_i, A_i Q_i)$

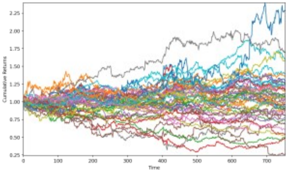
update_velocity$[\omega, c_1, c_2](\vec{x}_i, \vec{v}_i, \vec{p}_i, \vec{p}_g)$

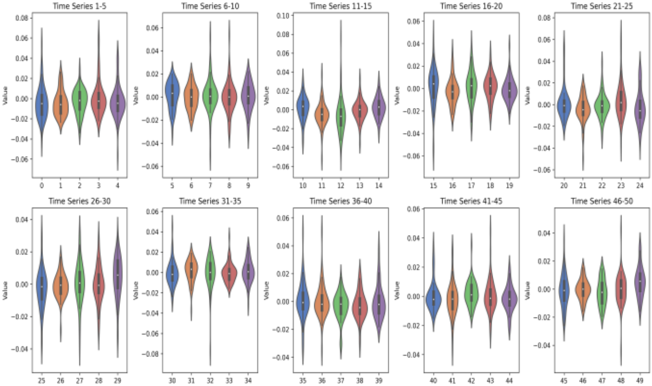
$r_1, r_2 \sim U(0,1)$
$v_i \leftarrow \omega v_i + c_1 r_1(\vec{p}_i - \vec{x}_i) + c_2 r_2(\vec{p}_g - \vec{x}_i)$
return v_i

Data

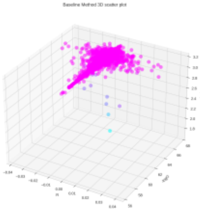
Time series returns data
For 50 time series
For 755 days

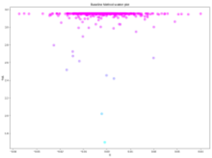
Data Source: Qube Research & Technologies (QRT) on the Challenge Data site.

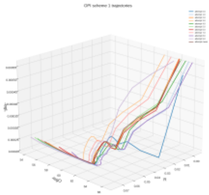


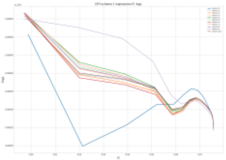


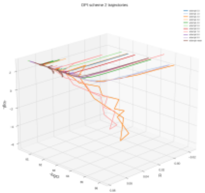
Results Visualization and Interpretation

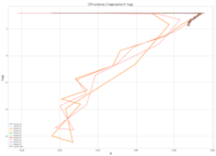


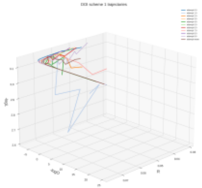




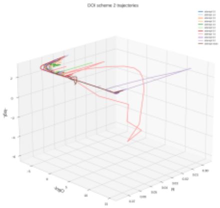


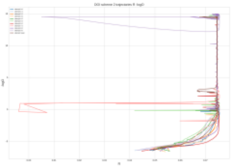


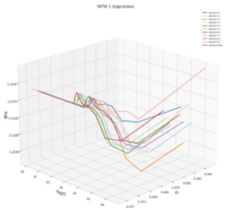


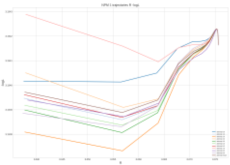


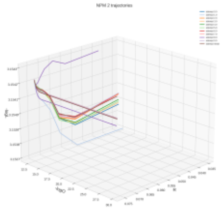
Results Visualization and Interpretation

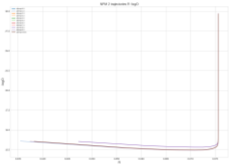




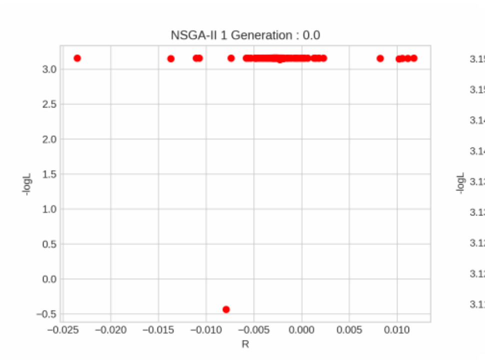


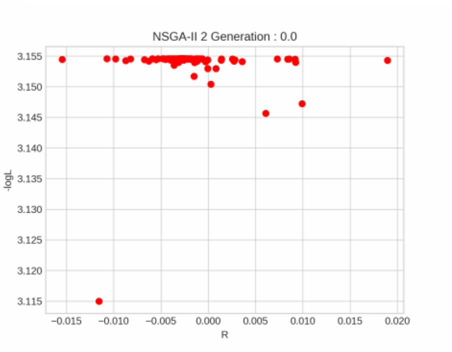




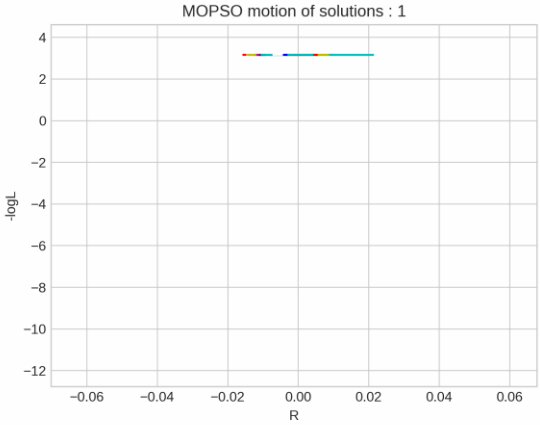


Results Visualization and Interpretation

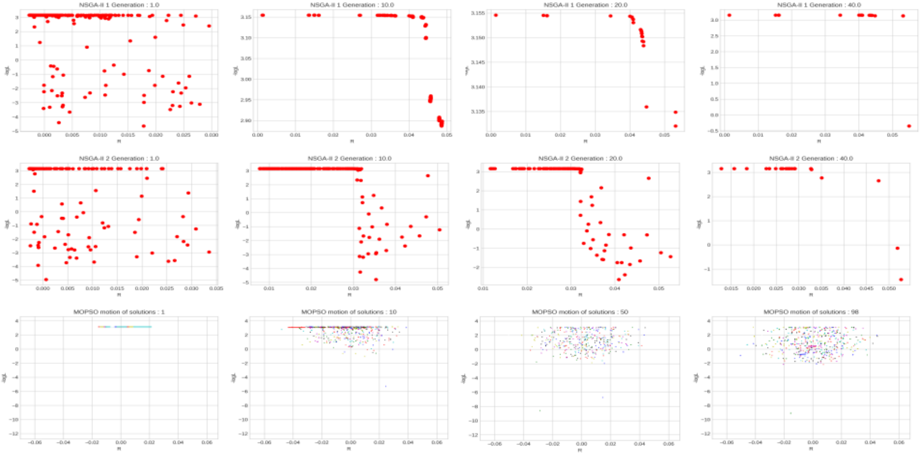




Results Visualization and Interpretation



Results Visualization and Interpretation (Backup)



Results Table and Discussion

Objective	Statistic	Baseline	OPI		DOI		NPM		NSGA-II		MOPSO
			Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	
R	mean (1e-2)	-0.272	7.53	7.43	7.53	7.43	**7.55**	**7.55**	1.90	2.63	0.00266
	std (1e-3)	2.36	3.43	4.48	1.85	3.23	1.42	1.41	9.38	7.60	**16.2**
	min (1e-2)	-3.74	-0.31	-2.22	3.11	-0.40	**3.73**	3.55	-2.35	-1.55	-5.86
	50% (1e-2)	-0.275	**7.56**	7.47	**7.56**	7.47	**7.56**	**7.56**	1.62	2.91	-8.96
	max (1e-2)	4.00	**7.56**	**7.56**	**7.56**	**7.56**	**7.56**	**7.56**	5.51	5.93	5.63
$-log(L)$	mean	**3.15**	**3.15**	3.14	**3.15**	**3.15**	**3.15**	**3.15**	3.12	3.03	1.42
	std (1e-2)	2.51	0.00147	27.6	9.56	19	0.00155	0.00216	35.9	74.6	**146**
	min	1.70	**3.15**	-6.45	2.60	-6.37	**3.15**	**3.15**	-4.66	-4.96	-11.3
	50%	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**	3.1	**3.15**	**3.15**	**3.15**	1.55
	max	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**	**3.15**
$-log(O)$	mean	**64.46**	58.34	61.80	13.82	10.06	61.58	27.22	63.60	64.15	63.38
	std	1.36	2.51	0.829	**9.54**	5.98	0.75	4.19	0.52	0.37	0.95
	min	56.41	54.09	60.5	-6.56	-6.65	**60.06**	12.43	56.32	58.47	59.65
	50%	**64.68**	57.84	61.6	16.21	14.4	61.46	29.01	63.54	64.09	63.42
	max	**67.50**	66.32	66.61	24.46	14.96	66.42	29.74	66.98	66.88	67.29

Table 3.1: Statistics on solutions generated by various method

- max achieved in R
- not 50% for some
- low std is better
- EA methods

- latter decimals

- smaller: cause R?
- DOI

Conclusion

- Defined cond-fMOOP
- Reviewed various methods to solve MOOP
- Applications of this formulation for Neural Networks
- Formulated critical question about cond-fMOOP
- Relations between cond-fMOOP and L_2 regularization
- Observed properties of original problem
- Used them to extend various methods
- Implemented those methods
- Derived Insights from those results
- Noted the advantages of using the Derived Properties.

Future Works

- Studies to determine problems in the cond-fMOOP category
- Develop a class of methods to address the cond-fMOOP problem
- Address any of the three critical questions
- Explore other properties formulations of the loss and reward functions to optimize computation of solutions
- Explore other potential applications of the presented methods beyond the Linear Factor Model problem

Thank You