Multi-Objective Optimization Problems With Well-Conditioned Solutions

A thesis submitted in partial fulfillment of the requirements for the degree of

Integrated Bachelors and Masters of Science

in

Department of Mathematics

by

Harshal Bharat Dupare
18MA20015

Under the guidance of

Dr. Swanand Ravindra Khare
and
Dr. Geetanjali Panda

Department of Mathematics

Indian Institute of Technology Kharagpur
West Bengal, India
November, 2022

Certificate

*This is to certify that the work contained in this thesis entitled "**Multi-Objective Optimization Problems With Well-Conditioned Solutions**" is a bonafide work of **Harshal Bharat Dupare, Roll no. 18MA20015**, carried out in the Department of Mathematics, Indian Institute of Technology Kharagpur under my supervision and that it has not been submitted elsewhere for a degree.*

Dr. Swanand Ravindra Khare
Professor
Department of Mathematics
Indian Institute of Technology Kharagpur,
West Bengal

Dr. Geetanjali Panda
Professor
Department of Mathematics
Indian Institute of Technology Kharagpur,
West Bengal

April, 2023
Kharagpur

Acknowledgements

For their extraordinary guidance and assistance, without which this research would not have been feasible, I would like to extend my appreciation to my guides and faculty supervisors, **Dr. Swanand Ravindra Khare** and **Dr. Geetanjali Panda**. They have always encouraged me to do as much research as I can, read as many papers as I can, and test out as many concepts as I can. Whatever challenges I had while working on the project, they were always there to support me.

Finally, I want to thank the Department of Mathematics, IIT Kharagpur for giving me the chance to complete the work I have done and my family and friends for their emotional support throughout this pandemic.

Harshal Bharat Dupare

Abstract

This thesis defines the problem of Functional Multi-Objective Optimization Problem (fMOOP) in which we are looking for functions as solutions of Multi-Objective Optimization Problem, where one of the objective/constraint is to minimize/bound the condition number of such a solution function. We give motivation for studying such problems and why they are important. Then we survey the different methodologies for solving MOOP in general and summarize a few results on the condition number of matrices and function. Then we address some of the important questions regarding such problems and analyze different methods for finding well-condition solutions to fMOOP. Based on the analysis we will select a few methods and compare their experimental results.

Contents

Chapter 1

Introduction

Optimization problems are of great interest academically along with being core to the functioning of today's society by having application in a variety of fields such as social studies, economics, finance, agriculture, automotive, engineering, computer science, networking, and many others.

In general, there can be more than one objective and constraint in the formulation of optimization problems. Optimization problems are classified into 2 broad categories namely *Single Objective Optimization Problems*(SOOP) and *Multi-Objective Optimization Problems* (MOOP), the latter can be considered a superset of the former. Our focus will be on a special subset of MOOP where the *Unknown/Decision Variable* is a *function* let's call them fMOOP. fMOOP are of great interest because of their vast applications in modeling the relation between 2 sets, e.g. all of the Neural Networks so heavily researched today fall into this category with their myriads of applications.

Generally when we have a function, we want it to have some good properties like being correct, easy to compute, easy to store, behave nicely over the input set, or any other desirable properties based on the context. One such very important property for most of such functions of interest is being *Well-Conditioned*, meaning small changes in the input should not result in too big changes in the output. We focus on this problem namely the problem to find function solutions of fMOOP which are well-conditioned.

Goal of this study is to analyze the implications of restricting the condition number for solutions of fMOOP, and study how the existing methodologies perform on such problems and device new methodologies for finding well-conditioned solutions to fMOOP.

1.1 Preliminaries

1.1.1 Definitions

Aggregation Scheme: Given a value function $value(\cdot)$ over x, for $x \in X$, a method to aggregate those values to one value. Denoted as follows:

$$\underset{\forall x \in X}{Agg}\ value(x) \tag{1.1}$$

e.g. Expectation over given probability distribution $p(\cdot)$ over X.

$$\underset{\forall x \in X}{Agg}\ value(x) = \int_{\forall x \in X} value(x) p(x)\, dx \tag{1.2}$$

e.g. Exponential decaying average, given a index-function $t : X \to [|X|]$, where $[n] = \{0, 1, 2, ..., n-1\}$.

$$\underset{\forall x \in X}{Agg}\ value(x) = \sum_{\forall x \in X} \lambda^{t(x)} value(x) \tag{1.3}$$

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:

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

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:

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

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

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

Note: Whenever the dimensions of the matrix in the context are already defined or are obvious from the context we will omit the superscript denoting its dimension.

1.1.2 Problem

Given:

Domain $X_1 \cup X_2 \subseteq X$ and their associated Ranges $Y_1 \cup Y_2 \subseteq Y$, where $X_i \equiv Y_i$ such that for each $x \in X_i$ we have a corresponding $y \in Y_i$ i.e. $y \equiv x$.
Aggregation schemes $\underset{\forall x \in X_1}{Agg_1}$ defined over $\mathcal{L} : Y \times Y \to \mathbb{R}$ call $\mathcal{L}(\cdot,\cdot)$ the *Loss* function and $\underset{\forall x \in X_2}{Agg_2}$ defined over $\mathcal{R} : Y \times Y \to \mathbb{R}$ call $\mathcal{R}(\cdot,\cdot)$ the *Reward* function.
A scalar $\alpha \in \mathbb{R}^+$ and subsets $X_i \subseteq X$, $i \in \{3, 4, 5, 6\}$.
For a given $\mathcal{F}$ a family of functions $f : X \to Y$ we need to find a function $f \in \mathcal{F}$ which solves the following

Functional Multi-Objective Optimization Problem:

$$\underset{f \in \mathcal{F}}{min} \underset{\forall x \in X_1, y \in Y_1, y \equiv x}{Agg_1} \mathcal{L}(y, f(x)) \tag{1.7}$$

$$\underset{f \in \mathcal{F}}{max} \underset{\forall x \in X_2, y \in Y_2, y \equiv x}{Agg_2} \mathcal{R}(y, f(x)) \tag{1.8}$$

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

$$\underset{f \in \mathcal{F}}{min} \underset{x \in X_3}{max} \, cond_{abs}(f, x) \tag{1.9}$$

$$\underset{f \in \mathcal{F}}{min} \underset{x \in X_4}{max} \, cond_{rel}(f, x) \tag{1.10}$$

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

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

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

It is very important to mention that the above problem formulation highly depends on the nature of the mapping f and the family $\mathcal{F}$ that it belongs to. For example if f models a function for which we expect smooth change in its value w.r.t. its domain then its a reasonable formulation, and if there are no reasons to believe that it should be smooth over its domain then its not a reasonable formulation. Similarly for a chosen family $\mathcal{F}$ to model f, if $\mathcal{F}$ doesn't have smooth functions belonging to it then its not a reasonable formulations irrespective of the nature of the function that f is modeling.

Note: What we mean by minimizing over a functions $f \in \mathcal{F}$ of a family of functions $\mathcal{F}$ is that we minimize over the parameters of that family of function $\mathcal{F}$.

1.2 Motivation and Applications

1.2.1 Linear Factor Model

The problems defined in the former section was motivated from the problem on linear factor models with added constraint for the returns-to-factor matrix be orthonormal columns. We describe the problem below.

A data of returns for n time series $R_t^{n\times 1} \in \mathbb{R}^{n\times 1}$, for $t \in [T]$ and denote $R_{t\times d}^{n\times d} = [R_{t-1}^{n\times 1}, R_{t-2}^{n\times 1}, ..., R_{t-d}^{n\times 1}] \in \mathbb{R}^{n\times d}$ is given.

We need to design $F_t^{n\times k} \in \mathbb{R}^{n\times k}$ as a function of $R_{t\times d}^{n\times d}$, i.e. $F_t^{n\times k} = f(R_{t\times d}^{n\times d})$ and from that we can derive $P_t^{n\times 1} = g(F_t^{n\times k})$ such that $P_t^{n\times 1} \approx R_t^{n\times 1}$ which we can measure by *Average* aggregation scheme over the loss function $\mathcal{L}(y, \hat{y}) = ||y - \hat{y}||_2$ which we need to minimize and another *Average* aggregation scheme over the return function $\mathcal{R}(y, \hat{y}) = \frac{\langle y, \hat{y}\rangle}{||y||_2||\hat{y}||_2}$ defined over the testing period of $t \in [T, T+S] = \{T, T+1, ..., T+S-1\}$ which we need to maximize.

$$\begin{tikzcd}
R_{t\times d}^{n\times d} \arrow[rr, "f"] & & F_t^{n\times k} \arrow[d, "g"] \\
R_t^{n\times 1} \arrow[rr, equal, "\mathcal{L}"] \arrow[dr, dashed] & & P_t^{n\times 1} \arrow[dl, dashed] \\
& \mathcal{R} &
\end{tikzcd} \tag{1.13}$$

Lets assume that the functions $f(\cdot)$ and $g(\cdot)$ are linear w.r.t. their argument. In this case it can be written as

$$F_t^{n\times k} = f(R_{t\times d}^{n\times d}) = R_{t\times d}^{n\times d} \times A^{d\times k} \tag{1.14}$$

$$P_t^{n\times 1} = g(F_t^{n\times k}) = F_t^{n\times k} \times \beta^{k\times 1} \tag{1.15}$$

And the orthonormal condition requires

$$A^{d\times k}(A^{d\times k})^\top = I^{d\times d} \tag{1.16}$$

From the 1.16 we can infer that $(A^{d\times k})^T$ belongs to the set of right inverses of $A^{d\times k}$.

Further if we relax the 1.16 to bounds on condition number by $\alpha \geq 1$

$$\max_{R_{t\times d}^{n\times d} \in \mathbb{R}^{n\times d}} cond_{rel}(f, R_{t\times d}^{n\times d}) = \lim_{\epsilon \to 0} \sup_{||\delta R_{t\times d}^{n\times d}|| \leq \epsilon} \frac{||\delta R_{t\times d}^{n\times d} \times A^{d\times k}||}{||\delta R_{t\times d}^{n\times d}||} \frac{||F_t^{n\times k} \times (A^{d\times k})^\top||}{||F_t^{n\times k}||} \leq \alpha \tag{1.17}$$

which implies

$$\max_{R_{t\times d}^{n\times d} \in \mathbb{R}^{n\times d}} cond_{rel}(f, R_{t\times d}^{n\times d}) \leq ||A^{d\times k}||||(A^{d\times k})^\top|| \leq \alpha \tag{1.18}$$

Note that $\alpha = 1$ contains the set which satisfies 1.16 equation since

$$1 = ||I^{d\times d}|| = ||A^{d\times k}(A^{d\times k})^\top|| \leq ||A^{d\times k}||||(A^{d\times k})^\top|| \leq \alpha \tag{1.19}$$

So any $\alpha \geq 1$ will contain the set specififed by the condition 1.16
Combining all of that together gives us the **fMOOP** formulation of this problem as

$$\min_{\beta^{k\times 1} \in \mathbb{R}^{k\times 1}, A^{d\times k} \in \mathbb{R}^{d\times k}} \frac{1}{T-d+1} \sum_{\forall t \in [d,T]} ||R_t^{n\times 1} - R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times 1}||_2^2 \tag{1.20}$$

$$\max_{\beta^{k\times 1} \in \mathbb{R}^{k\times 1}, A^{d\times k} \in \mathbb{R}^{d\times k}} \frac{1}{S} \sum_{\forall t \in [T,T+S]} \frac{\langle R_t^{n\times 1}, R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times 1} \rangle}{||R_t^{n\times 1}||_2 ||R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times 1}||_2} \tag{1.21}$$

subject to

$$\max_{R_{t\times d}^{n\times d} \in \mathbb{R}^{n\times d}} cond_{rel}(f, R_{t\times d}^{n\times d}) \leq \alpha \tag{1.22}$$

1.2.2 Principle Time Series

Another task related to time series is to represent a set of n time series by less number of time series $k < n$. Which is some way is an application of the Linear Factor Models.
Given:
The definition of $R_t^{n\times 1}$ and $R_{t\times d}^{n\times d}$ are same as before, but here we need to design latent time series denote them by $F_t^{k\times 1} \in \mathbb{R}^{k\times 1}$ as a function of $R_{t+1\times d}^{n\times d}$, i.e. $F_t^{k\times 1} = f(R_{t+1\times d}^{n\times d})$ which reduces the n time series observed from time $t-d$ to time t to set of k time series at time t such that its possible to sufficiently recover the n time series at time t by another

function $g(F_t^{k\times 1}) = P_t^{n\times 1}$ such that $P_t^{n\times 1} \approx R_t^{n\times 1}$ which we can measure by *Average* aggregation scheme over the loss function $\mathcal{L}(y, \hat{y}) = ||y - \hat{y}||_2$, which we need to minimize.

$$\begin{tikzcd}
R_{t+1\times d}^{n\times d} \arrow[rr, "f"] & & F_t^{k\times 1} \arrow[d, dashed, "g"] \\
R_t^{n\times 1} \arrow[rr, equal, "\mathcal{L}"] \arrow[dr, dashed] & & P_t^{n\times 1} \arrow[dl, dashed] \\
& \mathcal{R} &
\end{tikzcd} \tag{1.23}$$

Further we can restrict the functions $f(\cdot)$ and $g(\cdot)$ to be linear w.r.t. their argument. In that case it can be written as

$$F_t^{k\times 1} = f(R_{t+1\times d}^{n\times d}) = A^{k\times n} \times R_{t+1\times d}^{n\times d} \times B^{d\times 1} \tag{1.24}$$

$$P_t^{n\times 1} = g(F_t^{k\times 1}) = C^{n\times k} \times F_t^{k\times 1} \tag{1.25}$$

For small changes in our time series data $\delta R_{t+1\times d}^{n\times d}$ we expect that there is small changes in the latent time series $\delta F_t^{k\times 1}$, which we can model by bounding the absolute condition number by a scalar α.

$$\max_{R_{t+1\times d}^{n\times d} \in \mathbb{R}^{n\times d}} cond_{abs}(f, R_{t+1\times d}^{n\times d}) = \lim_{\epsilon\to 0} \sup_{||\delta R_{t+1\times d}^{n\times d}|| \le \epsilon} \frac{||A^{k\times n} \times \delta R_{t+1\times d}^{n\times d} \times B^{d\times 1}||}{||\delta R_{t+1\times d}^{n\times d}||} \tag{1.26}$$

Combining all of that together gives us the **fMOOP** formulation of this problem as

$$\min_{A\in\mathbb{R}^{k\times n}, B\in\mathbb{R}^{d\times 1}, C\in\mathbb{R}^{n\times k}} \frac{1}{T-d+1} \sum_{\forall t\in[d,T]} ||R_t^{n\times 1} - C^{n\times k} \times A^{k\times n} \times R_{t+1\times d}^{n\times d} \times B^{d\times 1}||_2^2 \tag{1.27}$$

subject to

$$\max_{R_{t+1\times d}^{n\times d} \in \mathbb{R}^{n\times d}} cond_{abs}(f, R_{t+1\times d}^{n\times d}) \le \alpha \tag{1.28}$$

1.2.3 Defense Against Adversarial Attacks on Neural Networks

Say we are given a trained neural network $\mathcal{N} : X \to Y$ which has learnt a mapping from X, the input set to Y, the output set. If the underlining mapping that $\mathcal{N}$ was modeled to learn was smooth w.r.t. its input then we expect that for a well trained network $\mathcal{N}$ for any input $x \in X$ and for small enough $\delta x \in X$ δx the output of the network wont change much, lets say that it won't change more than a constant $\alpha > 0$ times the norm of x i.e. $||\mathcal{N}(x + \delta x) - \mathcal{N}(x)|| = ||\delta\mathcal{N}(x)|| \le \alpha||\delta x||$. Which we can reformulate as follows

$$\underset{x \in X}{max}\ cond_{abs}(\mathcal{N}, x) = \underset{\epsilon \to 0}{lim}\ \underset{||\delta x|| \le \epsilon}{sup} \frac{||\delta\mathcal{N}(x)||}{||\delta x||} \le \alpha \tag{1.29}$$

Most of the networks which are being trained today don't account for such constraints giving the way to adversarial attacks on them, which exploit this drawbacks of the network to force them to output unreasonably wrong value.

$$\begin{tikzcd}
(x, y, y') \arrow[rr, "\mathcal{A}_\epsilon"] \arrow[d, dashed, no head] \arrow[drr, dashed, no head] & & \delta x \arrow[d, dashed, no head] \\
x \arrow[d, "\mathcal{N}"] & & x + \delta x \arrow[d, "\mathcal{N}"] \\
y & & y'
\end{tikzcd} \tag{1.30}$$

Here $\mathcal{A}_\epsilon$ the adversary which takes the input (x, y, y') being the input data x, the associate label y, and the expected forced wrong output y' and outputs the required perturbation δx, such that $||\delta x|| \le \epsilon$. When δx is added to original input data the new malicious input $x_m = x + \delta x$ forces the network $\mathcal{N}$ to output y' instead of y whereas it would have given output of y on the input x. Note that the adversary prefers smaller values of ϵ, since that implies that it can generate malicious input with as little changes as possible.
This problem can be avoided if we can train the network to also minimize the absolute condition number, $cond_{abs}(\mathcal{N}, x)$ over all the input space. We can also provide the constrains on their maximum values and model accordingly.
So if the network has been trained such that $||\delta\mathcal{N}(x)|| \le \alpha||\delta x||$ holds, then for the adversary $\mathcal{A}_\epsilon$ to make the network's $\mathcal{N}$ output perturb by $||\delta\mathcal{N}(x)||$ it will have to change the input x value perturbation of norm more than $\frac{||\delta\mathcal{N}(x)||}{\alpha} \le ||\delta x||$, which will not be possible for an adversary with $\epsilon < \frac{||\delta\mathcal{N}(x)||}{\alpha}$. Hence training with such constraints will force the adversary to make large changes to the input for designing an malicious input, which is unfavourable for the adversary.

Chapter 2

Related Works

Before we look for the related works to this problems it is important to understand the different dimensions of this problem and structure the related works accordingly. The problems stated in the section 1.1.2 has 1 major characteristic apart from being fMOOP is that it requires the solution to have bounded condition number (absolute or relative), and on top of that since such formulation has many applications it is desirable to have efficient algorithms to compute such solutions. So the following dimensions of the literate that interests us and can have impact of the problem are:

1. Methods to solve MOOP
2. Literature related to condition number
3. Efficiency in the computational aspects of both 1, and 2

As we will see that lot of research has been done in solving MOOP and we have fairly efficient methods to get the solutions. But when it comes to the implications of condition number most of the research is focused on the condition number of matrices compared to that of condition number of general functions.

2.1 Multi-Objective Optimization

2.1.1 Definitions

In MOOP since there are more than one objectives its unlikely that all of them achieve their optima at the same point, hence typically there is no single global solution. Which makes its necessary to rethink about the definition for an optimum and accordingly determine a set of points that can be deemed as the solutions. The most widely used concept concept in defining an optimal point is that of the *Pareto optimality* [6], which is defined as follows:
Below we will use $F : X \rightarrow \mathbb{R}^k$ as the objective function that we need to minimize over the input $x \in X$ and we will use minimization over all the objectives since maximization can be converted to minimization by negating the sign.

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.

All Pareto optimal points lie on the boundary of the feasible criterion space [17]. Sometimes the pareto optamility is too strong of a requirement hence we define *weakly Pareto optimal* as follows:

Weakly Pareto Optimal: A point, $x^* \in X$, is weakly Pareto optimal iff there does not exist another point, $x \in X$, such that $F(x) < F(x)$.

Pareto optimal points are weakly Pareto optimal, but weakly Pareto optimal points are not Pareto optimal.
Alternatively we define the compromise solution, which minimizes the difference between the utopia point/ideal point, defined as follows [10]:

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

In general, F^* is infeasible due to other constrains. Then the next best solution that we can attain is a solution that is as close as possible to the utopia point. We call such solution a **compromise solution** and it is Pareto optimal. The meaning of close in this context needs to be clarified and generally we need to define norms to measure closeness of the solutions for example use of Euclidean norm [11]. Another problem with this approach is of different objective have different scales so generally we need to transform the objectives to a single scale for any meaningful use of the defined norm.

The methods to solve MOOP are classified into 4 classes [39] based on the availability and involvement of a external decision maker(DM) used to convey the preference over different pareto optimal solution.

1. **No preference methods**: No DM is available and a neutral compromise solution is identified without any specification of the preference information.

2. **A priori methods**: based on the preference information given by the DM the optimal solution is found.

3. **A posteriori methods**: a good representative set of Pareto optimal solutions is found, and from among them the DM must choose the best solution.

4. **Interactive methods**: Pareto optimal solution(s) are shown to the DM, then the DM describes how the solution(s) could be improved. Then the next set of Pareto optimal solution(s) is generated based on the DM's feedback and iteratively the solutions are improved.

2.1.2 No Preference Methods

Since no preference information is provided the general one the approaches followed in [8] uses the definition of **utopia point** in 2.1.1 and then by properly scaling the objective function F to $\hat{F}$ and using a $||\cdot||$ norm defined over the objective space the following optimizations problem is formulated to minimize the following objective.

$$\underset{x \in X}{min} \, ||\hat{F}(x) - \hat{F}^{*}|| \tag{2.1}$$

Other similar methods are described in [41].

2.1.3 A Priori Methods

Methods which come under this class can further divided into other smaller classes but all of them have a common feature that is,enough information is provided a priori to compare any candidate pareto optimal solutions.
Following are some of the Important methods under this class:

Utility Function Methods: Here we have a utility function $U : \mathbb{R}^k \rightarrow \mathbb{R}$, and the goal is to solve the following SOOP

$$\underset{x \in X}{min}\, U(F(x)) \tag{2.2}$$

Notable methods which come under this utility model are

$$U(F(x)) = \sum_{\forall i \in [k]} w_i F_i(x) \tag{2.3}$$

Known as the Linear scalarization method, if all $\forall i \in [k], w_i > 0$ is a sufficient condition for the solution of 2.1.4.1 to be a pareto optima [3], but it is not a necessary condition [14].

$$U(F(x)) = \big(\sum_{\forall i \in [k]} w_i (F_i(x) - F_i^*)^p\big)^{1/p} \tag{2.4}$$

2.4 for $p > 0$,generally $p = 1, 2$ is another common extension of the formulation [47], for pareto optamility the conditions on w_i are same as they are for 2.1.4.1, along with that if any of the w_i is set to 0 then it can result in weak pareto optimality.

$$U(F(x)) = \underset{i \in [k]}{max}\, \frac{F_i(x)}{w_i} \tag{2.5}$$

2.5 is known as Hypervolume/Chebyshev Scalarization method [48], and in this case if $\forall i \in [k]\ w_i > 0$, it is shown that the solution of 2.5 converges to the Pareto front even for non-convex pareto fronts.

ϵ-Constraint Method: In this method[7] we have a single most important objective function $F_s(x)$. and the remaining objective functions are used to form additional constraints $F_i(x) \leq \epsilon_i, \forall i \in [k]/\{s\}$.

$$\underset{x \in X}{min}\, F_s(x) \tag{2.6}$$

subject to the constraints

$$F_i(x) \leq \epsilon_i,\ \ \forall i \in [k]/\{s\} \tag{2.7}$$

It is proven that the by a systematic variation of ϵ_i one can generate a set of Pareto optimal solutions[38]. If the solution of 2.6,2.7 exists then it is a weakly Pareto optimal solution [42], and if the solution is unique, then it is Pareto optimal [42].

Lexicographic Method: As the name suggests, the Objective functions are ordered as per decreasing order of importance namely $F_i(x)$ is more important than $F_j(x)$ iff $i < j$. Then, the following optimization problems are solved starting from $i = 1, 2, ..., k$.

$$\min_{x \in X} F_i(x) \tag{2.8}$$

subject to

$$F_j(x) \leq F_j(x_j^*), \forall j \in \{1, 2, ...i - 1\} \tag{2.9}$$

In 2.9 we can also have $=$ instead of $\leq$, [44]. Here x_j^* is the solution obtained at the j'th iteration, initially for $i = j = 1$ there are no constrains and $F_i(x)$ is minimized over $x \in X$.

Goal Programming Methods: Goal Programming method was developed by [2][40][5]. In this method we have been given goals g_j which are expected by the DM for the objective $F_j(x)$ respectively. To measure the deviations from the goal the sum of the absolute deviation is minimized.

$$\min_{x \in X} \sum_{\forall i \in [k]} |g_i - F_i(x)| \tag{2.10}$$

2.1.4 A Posteriori Methods

As the name suggest that the DM is involved a posteriori of finding solution, these apprpaches are also known as generate-first-choose-later approaches [20]. The goal is to produce a good enough representative subset of solutions which are Pareto optimal. In general they are classified into 2 classes.

1. **Mathematical programming methods**, which generally work by producing 1 pareto optimal solution per iteration/run of the algorithm.

2. **Evolutionary algorithms**, which produce a set of Pareto optimal solutions per iteration/run of the algorithm.

2.1.4.1 Mathematical Programming Methods

Few of the well known methods in this class are:

1. Normal Boundary Intersection Method

2. Modified Normal Boundary Intersection Method

3. Normal Constraint Method

Normal Boundary Intersection Method (NBI): As discussed the section the weighted sum method does provide a pareto optimal solution but its is very difficult to find evenly spread solution by varying the weights, to address these and other computational drawbacks Das and Dennis in their paper [18] presented the NBI method, whichis formulated as follows:

$$\underset{x \in X, t}{min}\ t \tag{2.11}$$

$$\Phi w + t\mu = F(x) - F^* \tag{2.12}$$

Where $\Phi \in \mathbb{R}^{k \times k}$ is the pay-off matrix whose Φ_{ij} entry measures the difference in the optimal value for j'th objective considering only the i'th objective to be minimized and that of the utopia point 2.1.1, i.e. $\Phi_{ij} = F_j(\underset{x \in X}{argmin}\, F_i(x)) - F_j^*$. Also $\mu = -\Phi e$, μ is known as the quasi-normal vector and $e^\top w = \sum_{i \in [k]} w_i = 1$ which is provided by the user. NBI method doesn't provide sufficient condition for finding pareto optimal solutions hence it is possible that the solutions obtained via this method are not pareto optimal and neither does this provide a necessary condition for pareto optimal solutions since for $k > 2$ for some problems it overlooks some of the pareto optimal solutions.

Modified Normal Boundary Intersection Method (NBIm): As stated in 2.1.4.1 the NBI method suffers from not even being a necessary condition for pareto optimal solutions, Das [36] and R de S Motta [30] proposed modified methods to cover these drawbacks

Normal Constraint Method (NC): Messac et al. [22][23] proposed the NC method as an alternative to the NBI method, which provided some improvements. It uses the utopia point 2.1.1 to normalize the objective vector $F(x)$ to $\hat{F}(x)$ along with a pareto filter which removes the non-dominant solutions to keep only the dominant solutions.Also it always produces pareto optimal solutions along with it's performance being independent of design objective scales.

2.1.4.2 Evolutionary Algorithms (EA)

Evolutionary algorithms are one of the very actively researched methods [45] for solving MOOP and finding pareto optimal solutions. EA are subset of the paradigm which is inspired by nature and evolution in designing algorithms for various purposes including solving optimization problems. The general procedure that and EA follows is described below:

Generic EA

function EA($\mathcal{I}$) ▷ gets input parameters $\mathcal{I}$
    $\mathcal{P} \leftarrow initialize(\mathcal{I})$ ▷ initialize solution population $\mathcal{P}$
    while $converges(\mathcal{P}) \vee terminate(\mathcal{P})$ **do** ▷ till convergence or termination
        $\mathcal{P} \leftarrow evolve(\mathcal{P}, \mathcal{I})$ ▷ evolve the population to next generation
    return $\mathcal{P}$

Some of the notable methods in EA which are commonly used are:

1. Non-dominated Sorting Genetic Algorithm-II (NSGA-II)
2. Ant Colony Optimization (ACO)
3. Particle swarm optimization (PSO)

Non-dominated Sorting Genetic Algorithm-II (NSGA-II): Proposed by K. Deb et al. in the paper [19], NSGA-II is based on elitist principle meaning only the elites of the populations survive to the next generation based on a partial-order sorting of the population, to decide the elites.

Ant Colony Optimization (ACO): it is based on idea of ant pheromone which ants use to communicate to form paths and explore more of the promising regions [32].

Particle Swarm Optimization (PSO): PSO is based on the flocking behaviour of the birds where each particle has a position (the solution) and a velocity (change in the solution) where the velocity is influenced by some neighbourhood of the particle [24].

The advantage the EA provides is that it can quickly provide a sets of solutions which even though are not guaranteed to be pareto optimal, but are non-dominant set and serve as good approximation to the entirety of the Pareto front. The disadvantages being that these algorithms are relatively slow and pareto optimality cant be guaranteed.

2.1.5 Interactive methods

Interactive methods require the *DM* to actively take part in pruning the candidates solutions suggested by the method, and based on the input of the *DM* the method adopts and suggest new solutions which are then again evaluated by the *DM* and the process is repeated to improve and adopt the solutions as per the needs of *DM*.

The generic structure of the Interactive methods is as follows [43]: The above method

---

Generic Interactive Method

$\mathcal{P} \leftarrow initialize(\mathcal{M})$ ▷ pareto optimal solution set: $\mathcal{P}$; no-preference method: $\mathcal{M}$

do

  $\mathcal{R} \leftarrow getPreference(\mathcal{P}, DM)$ ▷ preference information: $\mathcal{R}$, decision maker: DM

  $\mathcal{P} \leftarrow newSolutionSet(\mathcal{P}, \mathcal{R}, \mathcal{M})$ ▷ update solution set based on $\mathcal{R}$

while $converges(\mathcal{P}, DM) \vee terminate(\mathcal{P}, DM)$ ▷ till convergence or termination

---

can be classified based on the which method is used as $\mathcal{M}$ and what type of preference information $\mathcal{R}$ is available/provided by the *DM*.

$\mathcal{M}$ can be chosen from the various options available in a posteriori methods/no preference method 2.1.4 based on the problem type.

Generally the choices of $\mathcal{R}$ are classified into 3 classes [43].

1. **Trade-off Between Objectives**: Here the *DM* is shown various trade-off scenarios and asked their preference in regards to those trade-off, and based on that the next solution set is generated [9].

2. **Reference Point**: Here the *DM* for a get set of $\mathcal{P}$ pareto optimal solutions needs to provide a reference point w.r.t. which the next iteration will generate the updated solution set $\mathcal{P}$ [12][46].

3. **Classification of Objectives**: Here for a given $\mathcal{P}$ the *DM* classifies different objectives of the solutions such as to get more preferred solutions, and based on the classification the updates solution set $\mathcal{P}$ is generated.

2.2 Condition Number

The term ***Condition Number*** was first coined by Alan Turing in 1947 in his paper on Rounding-Off Errors in Matrix Processes [1]. He defined the condition number for matrices, further in 1966, John Rice in his paper *A Theory of Condition* [4] showed how to formulate the definition of condition number other classes of problems.

In our problem formulation may we not only require to compute the condition number of the function f to check if its bounded or not 1.11,1.12; but also to minimize the maximum over a set X_3 1.9,1.10.

Hence we need to methods which can either be used to

1. Bound the relative/absolute condition number
2. Compute relative/absolute condition number efficiently

Extensive amount of literature is available on condition number of matrices compared to that of general functions. Below we will see some results on condition number of matrix and also for general function.

2.2.1 Condition Number of Matrices

There are several papers which have proved various important properties of matrix condition number. Pierre Maréchal and Jane J. Ye in their paper *Optimizing Condition Numbers*[26] showed that for a symmetric positive semi-definite $n \times n$ matrix A minimizing the condition number $\kappa(A)$.

$$\kappa_2(A) = ||A||_2||A^{-1}||_2 \tag{2.13}$$

$$\underset{A}{min}\, \kappa_2(A) \tag{2.14}$$

2.14 is euqivalent to minimizing the following objective

$$\underset{A}{min}\, \lambda_1(A) - \kappa_2(\bar{A})\lambda_n(A) \tag{2.15}$$

where $\bar{A}$ is the optimal solution with minimum condition number, if such a solution's value of $\kappa_2(\bar{A})$ is not known, then one can use a desirable value of $\kappa_2(\bar{A})$ in its place. Here $\lambda_i(A)$ are the eigenvalues of the matrix A in decreasing order of their magnitude form $i = 1, 2, ..., n$. They also proved that that the problem of minimizing the condition number is non-smooth and non-convex optimization problem [26], which further increases the difficulty of the problem. In their paper [27], C Beltrán et al. have studied the convexity properties of the condition number over norm over frobenius inner product. Further Xiaojun Chen et al. in their paper [28] analysed the same for A being a gram matrix of functions of a scalar x, namely $A(x) = V(x)^\top V(x)$ and derived formulas for generalized gradient of $\kappa_2(A(x))$, and using exponential smoothing function they also develop a globally convergent smoothing method to solve the 2.14 problem.

G.Piazza and T.Politi in their paper [21] proved an upper bound for $\kappa_2(A)$ i.e. the condition number w.r.t. $||\cdot||_2$ (the spectral norm) for $1 \leq k \leq n$.

$$\kappa_2(A) \leq \frac{2^k}{\prod_{i=2}^{k}\sigma_i} \frac{1}{|det(A)|}\left(\frac{||A||_F}{\sqrt{n+k-1}}\right)^{n+k-1} \tag{2.16}$$

for $k = 1$ it reduces to as proved in [16].

$$\kappa_2(A) \leq \frac{2}{|det(A)|}\left(\frac{||A||_F}{\sqrt{n}}\right)^{n} \tag{2.17}$$

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

$$||A||_F = \sqrt{\sum_{i\in[n]}\sum_{j\in[m]} A_{ij}^2} \tag{2.18}$$

another interesting relation between $\kappa_2(A)$ and that of $\kappa_F(A)$ i.e. the condition number induced by Frobenius norm [15][25].

$$\frac{\kappa_F(A)}{n} \leq \kappa_2(A) \leq \kappa_F(A) \tag{2.19}$$

Hongyi Li et al. in their paper [29] have proved the following lower bounds on $\kappa_F(A)$ for A being positive definite matrix and $\beta \in \mathbb{R}$ and $tr(A)$ is the trace of matrix A.

$$\kappa_F(A) \geq 2n\frac{tr(A^{1+\beta})}{tr(A^{2\beta})det(A)^{\frac{1-\beta}{n}}} - n \tag{2.20}$$

$$\kappa_F(A) \geq 2\frac{tr(A^{\beta+1})tr(A^{\beta-1})}{tr(A^{2\beta})} - n \tag{2.21}$$

Note that 2.21 follows from 2.20 using AM-GM inequality. Further Nicholas J. Higham in their paper [13] have presented survey for computing condition number of triangular matrices

2.2.2 Condition Number of Functions

Edvin Deadman and Samuel D. Relton in their paper [31] extended the taylor's theorem for a complex function of matrices where $f : \mathbb{C} \to \mathbb{C}$ extended over matrices over $\mathbb{C}^{n\times n}$ i.e. $f : \mathbb{C}^{n\times n} \to \mathbb{C}$. For f satisfying some conditions they proved the bound on the absolute condition number of f at A for $\epsilon > 0$.

$$cond_{abs}(f, A) \leq \frac{L_\epsilon}{2\pi\epsilon^2} \max_{z\in\Gamma_\epsilon} |f(z)| \tag{2.22}$$

where Γ_ϵ is a closed counter of length L_ϵ which satisfies some condition, refer [31] for the specific conditions. A more extensive treatment and analysis of conditioning of such function is also done in the book by Nicholas J. Higham [37], based on the book a Matlab toolbox has also been made available by the name of *Matrix Function Toolbox* [35].

David H. Gutman and Javier F. Peña in their paper [33] extended the idea of condition number (relative condition number specifically) of a scalar function $f : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ extended over $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ restricted over a subset of the domain of f and sowed that the restriction has simialr properties as that of the condition number.

Chapter 3

Methodology and Results

3.1 Scope

Following are some questions that we consider important and need to be analysed

1. What types of fMOOPs confirm to restrictions on condition number of f?
2. How to determine minimum feasible value of α for a given problem and subsets $X_1, X_2, X_5, X_6 \subseteq X$?
3. For the fMOOPs which confirm such condition number restrictions how does the condition number behave over the domain X_5, X_6?
4. Based on the behaviour of the condition number what types of methods do we expect to work well?
5. What all computation methods/algorithms are available for condition number w.r.t. specific fMOOP?
6. How the transformed problems with bounds on condition number behave and how close are their solutions to the original problems solutions?

In this analysis and results we will consider following 2 main problems derived from the problems 1.2.1 and 1.2.3, both of which are defined in the Applications and Motivation section.

3.2 Theoretical Results

Condition number over composition of function: Say we have a function $f = g(h(x)) = g \circ h(x)$ where $f : X \to Y$, $h : X \to Z$, and $g : Z \to Y$ then we can write

$$cond_{abs}(f,x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta f(x)||}{||\delta x||} = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta g(h(x))||}{||\delta x||} \tag{3.1}$$

$\implies$

$$cond_{abs}(f,x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta g(h(x))||}{||\delta h(x)||} \frac{||\delta h(x)||}{||\delta x||} \tag{3.2}$$

under the assumption that $\forall \epsilon \geq 0$ and $||\delta x|| \leq \epsilon \; \exists \, \delta \geq 0$ such that $||\delta h(x)|| \leq \delta \; \forall x$ and $\lim_{\epsilon \to 0} \delta = 0$. Then we can write

$$cond_{abs}(f,x) = \lim_{\delta \to 0} \sup_{||\delta h(x)|| \leq \delta} \frac{||\delta g(h(x))||}{||\delta h(x)||} \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta h(x)||}{||\delta x||} \tag{3.3}$$

let $h(x) = z \implies \delta h(x) = \delta z$, and since z is a function of x, δz is not independent of δx hence we can write

$$cond_{abs}(f,x) \leq \lim_{\delta \to 0} \sup_{||\delta z|| \leq \delta} \frac{||\delta g(z)||}{||\delta z||} \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta h(x)||}{||\delta x||} \tag{3.4}$$

$\implies$

$$cond_{abs}(f,x) \leq cond_{abs}(g, h(x)) \times cond_{abs}(h,x) \tag{3.5}$$

the above 3.5 bound can be used to simplify the fMOOP with constraints on condition number for complex functions which are composition of simpler function whose condition number can be bound analytically, for eg. Neural Networks come under such functions.

Relation between absolute and relative condition number: given a function $f : X \to Y$ its absolute and relative condition number are given by

$$cond_{abs}(f,x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta f(x)||}{||\delta x||} \tag{3.6}$$

$$cond_{rel}(f,x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta f(x)||/||f(x)||}{||\delta x||/||x||} \tag{3.7}$$

we can write

$$cond_{rel}(f,x) = \lim_{\epsilon \to 0} \sup_{||\delta x|| \leq \epsilon} \frac{||\delta f(x)||}{||\delta x||} \frac{||x||}{||f(x)||} \tag{3.8}$$

$\implies$

$$cond_{rel}(f,x) = cond_{abs}(f,x) \frac{||x||}{||f(x)||} \tag{3.9}$$

L_2 **regularization and absolute condition number**: Consider the problem of fitting data with n data points $(X_i, Y_i), i \in [n]$, each column being 1 data point $(X, Y) \in (\mathbb{R}^{d\times n}, \mathbb{R}^{1\times n})$ using a function $f : \mathbb{R}^{d\times 1} \rightarrow \mathbb{R}^{1\times 1}$, for case of simplicity assume f is linear transform i.e. $f(X) = AX \approx Y$ where $A \in \mathbb{R}^{1\times d}$.
Consider the L_2 regularization formulations of this problem as follows

$$\underset{A\in\mathbb{R}^{1\times d}}{min} \frac{1}{n}\sum_{i\in[n]}||AX_i - Y_i||_2 + \lambda||A||_2^2 \tag{3.10}$$

Now, consider the same problem under fMOOP for minimizing the $||\cdot||_F$ frobenius norm of the transform A over the aggregation scheme of mean and $X_3 = \mathbb{R}^{d\times 1}$

$$\underset{A\in\mathbb{R}^{1\times d}}{min} \frac{1}{n}\sum_{i\in[n]}||AX_i - Y_i||_2 \tag{3.11}$$

$$\underset{A\in\mathbb{R}^{1\times d}}{min} \underset{x\in X_3}{max}\, cond_{abs}(f, x) \tag{3.12}$$

by definition

$$cond_{abs}(f, x) = \underset{\epsilon\rightarrow 0}{lim} \underset{||\delta x||_F\leq\epsilon}{sup} \frac{||A\delta x||_F}{||\delta x||_F} \tag{3.13}$$

note that $||A\delta x||_F = \sqrt{(\sum_{i\in[d]} A_i\delta x_i)^2}$ and by Cauchy–Schwarz inequality we can write $(\sum_{i\in[d]} A_i\delta x_i)^2 \leq (\sum_{i\in[d]} A_i^2)(\sum_{i\in[d]} x_i^2) = ||A||_F^2||\delta x||_F^2$, which implies

$$cond_{abs}(f, x) = \underset{\epsilon\rightarrow 0}{lim} \underset{||\delta x||_F\leq\epsilon}{sup} \frac{||A\delta x||_F}{||\delta x||_F} \leq \frac{||A||_F||\delta x||_F}{||\delta x||_F} = ||A||_F \tag{3.14}$$

note that for this case $||\cdot||_F$ is equivalent to $||\cdot||_2$, and the above upper bound implies that the problem 3.12 when minimized for the worst case using the upper bound 3.14 we get the fMOOP as follows

$$\underset{A\in\mathbb{R}^{1\times d}}{min} (||A||_2, \frac{1}{n}\sum_{i\in[n]}||AX_i - Y_i||_2) \tag{3.15}$$

which when scalarized with squaring the norm restriction gives us the standard L_2 regularization.

3.3 Experimental Results

3.3.1 Problem: Linear Factor Model

This is the same problem as defined in the section 1.2.1.
A data of returns for n time series $R_t^{n\times 1} \in \mathbb{R}^{n\times 1}$, for $t \in [T]$ and denote $R_{t\times d}^{n\times d} =$

$[R_{t-1}^{n\times1}, R_{t-2}^{n\times1}, ..., R_{t-d}^{n\times1}] \in \mathbb{R}^{n\times d}$ is given.

We need to design $A^{d\times k} \in \mathbb{R}^{d\times k}$ and $\beta^{k\times1} \in \mathbb{R}^{k\times1}$, such that $F_t^{n\times k} = R_{t\times d}^{n\times d} \times A^{d\times k}$ and $P_t^{n\times1} = F_t^{n\times k} \times \beta^{k\times1}$ and $P_t^{n\times1} \approx R_t^{n\times1}$. And the performance which is measured by *Average* aggregation scheme over the loss function $\mathcal{L}(y,\hat{y}) = ||y-\hat{y}||_2$ which we need to minimize and another *Average* aggregation scheme over the return function $\mathcal{R}(y,\hat{y}) = \frac{\langle y,\hat{y}\rangle}{||y||_2||\hat{y}||_2}$ defined over the testing period of $t \in [T, T+S] = \{T, T+1, ..., T+S-1\}$ which we need to maximize.

$$\begin{tikzcd} R_{t\times d}^{n\times d} \arrow[rr, "\times A^{d\times k}"] & & F_t^{n\times k} \arrow[d, dashed, "\times\beta^{k\times1}"] \\ R_t^{n\times1} \arrow[rr, equal, "\mathcal{L}"] \arrow[dr, dashed] & & P_t^{n\times1} \arrow[dl, dashed] \\ & \mathcal{R} & \end{tikzcd} \tag{3.16}$$

And the orthonormal condition

$$A^{d\times k}(A^{d\times k})^\top = I^{d\times d} \tag{3.17}$$

From the 3.17 we can infer that $(A^{d\times k})^\top$ belongs to the set of right inverses of $A^{d\times k}$. With the two objectives as:

$$\min_{\beta^{k\times1}\in\mathbb{R}^{k\times1}, A^{d\times k}\in\mathbb{R}^{d\times k}} \mathcal{L}_{\beta^{k\times1},A^{d\times k}} \tag{3.18}$$

Where $\mathcal{L}_{\beta^{k\times1},A^{d\times k}}$ is defined as follows

$$\mathcal{L}_{\beta^{k\times1},A^{d\times k}} = \frac{1}{T-d+1}\sum_{\forall t\in[d,T]} ||R_t^{n\times1} - R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times1}||_2^2 \tag{3.19}$$

and

$$\max_{\beta^{k\times1}\in\mathbb{R}^{k\times1}, A^{d\times k}\in\mathbb{R}^{d\times k}} \mathcal{R}_{\beta^{k\times1},A^{d\times k}} \tag{3.20}$$

Where $\mathcal{R}_{\beta^{k\times1},A^{d\times k}}$ is defined as follows

$$\mathcal{R}_{\beta^{k\times1},A^{d\times k}} = \frac{1}{S}\sum_{\forall t\in[T,T+S]} \frac{\langle R_t^{n\times1}, R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times1}\rangle}{||R_t^{n\times1}||_2||R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times1}||_2} \tag{3.21}$$

3.3.1.1 Data

We will be using the data provided by *Qube Research and Technologies (QRT)* on the *Challenge Data* site, under the challenge: *Learning factors for stock market returns prediction* [34].
The data contains (cleaned) daily returns of 50 stocks over a time period of 754 days (three years). And in the original problem they have asked for solutions with $d = 250$ and $k = 10$.

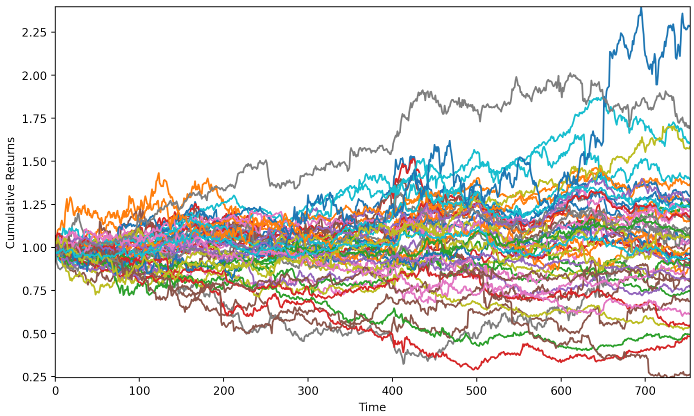

Figure 3.1: *Cumulative Returns for Daily Returns Time Series*

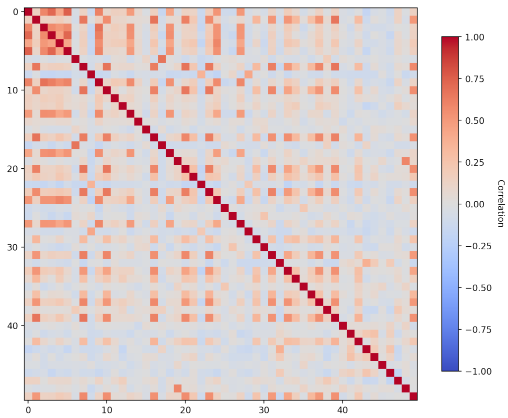

Figure 3.2: *Correlation Matrix of Daily Returns*

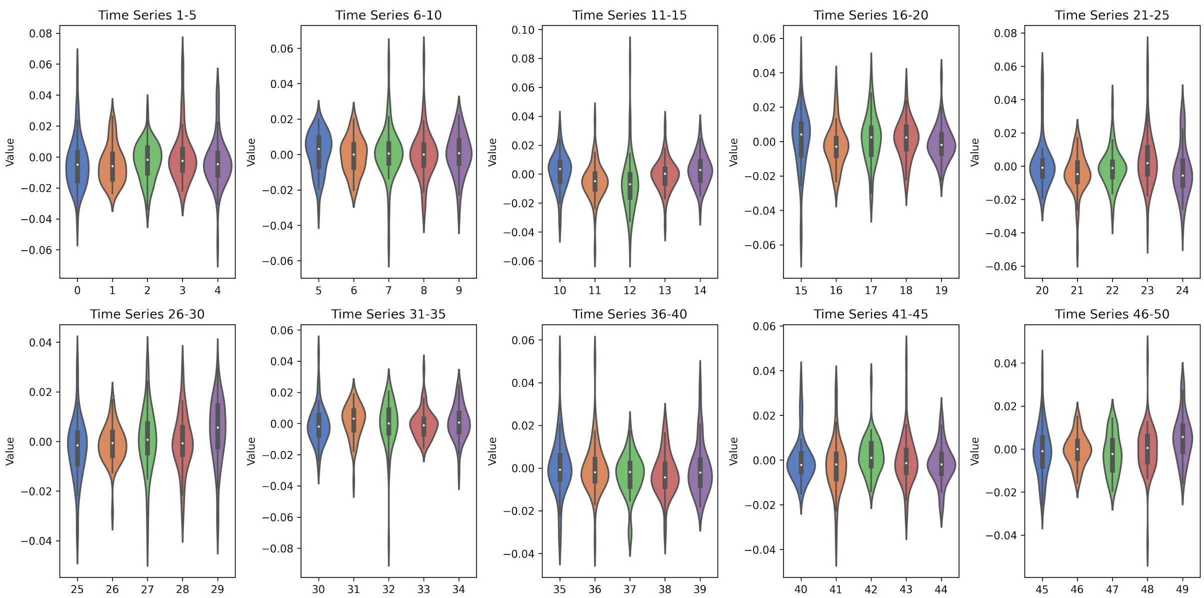

Figure 3.3: *Violin Plots for 50 Time Series*

3.3.1.2 Approaches

3.3.1.2.1 Baseline Method

The following method is based on randomized sampling method, which involves generating the matrix M randomly and applying random data and fitting them to the training data set using linear regression. This process is repeated multiple times, and the best result is selected from the attempts. Following is the pseudocode for the method:

```
Baseline Method
Set N_iter
β_optimal, A_optimal, R_optimal ← ∅, ∅, −∞
for i ← 1...N_iter do
    M_i ← sample_random_matrix()
    A_i ← gram_schmidt_algorithm(M_i)                  ▷ orthonormalize M_i to A_i
    Compute {F_{t,i}^{n×k}} for A_i over t ∈ [d, T]
    β_i ← regression({R_t^{n×1}}, {F_{t,i}^{n×k}}, t ∈ [d, T])
    if R_{β_i,A_i} > R_optimal then            ▷ update the optimal if better solution found
        β_optimal, A_optimal, R_optimal ← β_i, A_i, R_{β_i,A_i}
```

In an alternate implementation of this algorithms we can even do grid search over the entries of A, but because of the high number of possibilities we restrict this method over randomized search.

3.3.1.2.2 Gradient Based Methods

To compute the partial derivatives of the loss function $\mathcal{L}_{\beta^{k\times1},A^{d\times k}}$ with respect to $A^{d\times k}$ and $\beta^{k\times1}$, we will use the chain rule and the derivative of the matrix multiplication.

First, let's compute the derivative of $\mathcal{L}_{\beta^{k\times1},A^{d\times k}}$ with respect to $\beta^{k\times1}$. Lets Define $\Delta_t = R_t^{n\times1} - R_{t\times d}^{n\times d} \times A^{d\times k} \times \beta^{k\times1}$

$$\frac{\partial \mathcal{L}_{\beta^{k\times1},A^{d\times k}}}{\partial \beta^{k\times1}} = \frac{1}{T-d+1} \sum_{\forall t\in[d,T]} \frac{\partial}{\partial \beta^{k\times1}} ||\Delta_t||_2^2 \tag{3.22}$$

Now, let's compute the derivative of $||\Delta_t||_2^2$ with respect to $\beta^{k\times1}$:

$$\frac{\partial}{\partial \beta^{k\times1}} ||\Delta_t||_2^2 = \frac{\partial}{\partial \beta^{k\times1}} \Delta_t^\top \Delta_t = 2F_t^\top (F_t\beta - R_t) \tag{3.23}$$

Which gives

$$\frac{\partial \mathcal{L}_{\beta^{k\times 1},A^{d\times k}}}{\partial \beta^{k\times 1}} = \frac{2}{T-d+1}\sum_{\forall t\in[d,T]} F_t^\top (F_t\beta - R_t) \tag{3.24}$$

Similarly consider

$$\frac{\partial \mathcal{L}_{\beta^{k\times 1},A^{d\times k}}}{\partial A^{d\times k}} = \frac{1}{T-d+1}\sum_{\forall t\in[d,T]} \frac{\partial}{\partial A^{d\times k}} ||\Delta_t||_2^2 \tag{3.25}$$

We will use the following identities when a, b and C are not functions of X.

$$\frac{\partial(\mathbf{a}^\top \mathbf{X}\mathbf{b})}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^\top \tag{3.26}$$

$$\frac{\partial(\mathbf{a}^\top \mathbf{X}^\top \mathbf{b})}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^\top \tag{3.27}$$

$$\frac{\partial(\mathbf{X}\mathbf{a})^\top \mathbf{C}(\mathbf{X}\mathbf{b})}{\partial \mathbf{X}} = \mathbf{C}\mathbf{X}\mathbf{b}\mathbf{a}^\top + \mathbf{C}^\top \mathbf{X}\mathbf{a}\mathbf{b}^\top \tag{3.28}$$

for the equations below we are dropping the dimensions

$$||\Delta_t||_2^2 = \Delta_t^\top \Delta_t = (R_t - R_{t\times d}A\beta)^\top (R_t - R_{t\times d}A\beta) = R_t^\top R_t \tag{3.29}$$

$$||\Delta_t||_2^2 = R_t^\top R_t - R_t^\top R_{t\times d}A\beta - \beta^\top A^\top R_{t\times d}^\top R_t + \beta^\top A^\top R_{t\times d}^\top R_{t\times d}A\beta \tag{3.30}$$

Now using the identities 3.26, 3.27, 3.28 we get

$$\frac{\partial}{\partial A}||\Delta_t||_2^2 = 2(R_{t\times d}^\top R_{t\times d}A\beta\beta^\top - R_{t\times d}^\top R_t\beta^\top) \tag{3.31}$$

Which gives

$$\frac{\partial \mathcal{L}_{\beta,A}}{\partial A} = \frac{2}{T-d+1}\sum_{\forall t\in[d,T]} R_{t\times d}^\top R_{t\times d}A\beta\beta^\top - R_{t\times d}^\top R_t\beta^\top \tag{3.32}$$

For vector-valued scalar functions $v(\mathbf{x})$ and $u(\mathbf{x})$ with respect to $\mathbf{x}$, and where A is not a function of x.

$$\frac{\partial}{\partial \mathbf{x}}\left(v(\mathbf{x})u(\mathbf{x})\right) = v\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}u \tag{3.33}$$

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top \mathbf{A}\mathbf{x}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x} \tag{3.34}$$

$$\frac{\partial(\mathbf{Ax})}{\partial\mathbf{x}} = \mathbf{A}^\top \tag{3.35}$$

define $\rho_t = \frac{\langle R_t, R_{t\times d}A\beta\rangle}{||R_t||_2||R_{t\times d}A\beta||_2} = \frac{R_t^\top R_{t\times d}A\beta}{||R_t||_2||R_{t\times d}A\beta||_2} = \frac{R_t^\top F_t\beta}{||R_t||_2||F_t\beta||_2}$ and $\hat{R}_t = \frac{R_t}{||R_t||_2}$, now using 3.34, 3.33, 3.35, 3.26 and 3.28 we get

$$\frac{\partial\rho_t}{\partial\beta} = \frac{1}{||F_t\beta||_2}F_t^\top\hat{R}_t - \frac{2F_t^\top F_t\beta}{2||F_t\beta||_2^3}\hat{R}_t^\top F_t\beta \tag{3.36}$$

$$\frac{\partial\rho_t}{\partial A} = \frac{1}{||F_t\beta||_2}R_{t\times d}^\top\hat{R}_t\beta^\top - \frac{2R_{t\times d}^\top R_{t\times d}A\beta\beta^\top}{2||F_t\beta||_2^3}\hat{R}_t^\top F_t\beta \tag{3.37}$$

From that we can get

$$\frac{\partial\mathcal{R}_{\beta,A}}{\partial A} = \frac{1}{S}\sum_{\forall t\in[T,T+S]}\frac{\partial\rho_t}{\partial A} \tag{3.38}$$

$$\frac{\partial\mathcal{R}_{\beta,A}}{\partial\beta} = \frac{1}{S}\sum_{\forall t\in[T,T+S]}\frac{\partial\rho_t}{\partial\beta} \tag{3.39}$$

Also we can get derivative w.r.t. A for the frobenius norm of $AA^\top - I$

$$\frac{\partial||AA^\top - I||_F}{\partial A} = -4\cdot(I - A\cdot A^\top)\cdot A \tag{3.40}$$

Observation

Consider the set of matrices $\mathcal{O}_d^k = \{A|AA^\top = I_d, A\in\mathbb{R}^{d\times k}\}$, observe that $\mathcal{O}_k^k$ is set of all orthonormal square matrix of size $k\times k$ and for a $Q\in\mathcal{O}_k^k$ we have $QQ^\top = I$, which implies $Q^{-1} = Q^\top$ and hence $QQ^\top = Q^\top Q = I$. Using the aforementioned fact we observe that $AQ\in\mathcal{O}_d^k$ as

$$(AQ)(AQ)^\top = AQQ^\top A^\top = AA^\top \tag{3.41}$$

Now consider and iterative scheme for finding A,β which generates $\{A_i,\beta_i\}_{i=0}^{N_{iter}}$ with updates at i'th iteration denoted as $\{\Delta A_i, \Delta\beta_i\}$

$$\begin{aligned} A_{i+1} &= A_i + \Delta A_i \\ \beta_{i+1} &= \beta_i + \Delta\beta_i \end{aligned} \tag{3.42}$$

But since in our problem we have added restriction on $A\in\mathcal{O}_d^k$, which we can solve by one of the following 2 approaches

General approaches to satisfy orthonormality condition

1. Design the iteration scheme such that $A_i \in \mathcal{O}_d^k, \forall i \in \{0, 1, 2, ..., N_{iter}\}$

2. Relax the Condition on $A \in \mathcal{O}_d^k$ to $A \in \mathbb{R}^{d \times k}$ and solve using an iterative scheme; then project $A_{N_{iter}}$ to $\mathcal{O}_d^k$ i.e. solve the problem $\underset{A \in \mathcal{O}_d^k}{min} \; ||A_{N_{iter}} - A||$

First we will use the observation 3.41 to design an iterative scheme such that $A_i \in \mathcal{O}_d^k, \forall i \in \{0, 1, 2, ..., N_{iter}\}$

Lets say via some iterative method we get ΔA_i update for A_i but since $A_{i+1} = A_i + \Delta A_i$ need not belong to $\mathcal{O}_d^k$ we need some way to project ΔA_i in a space such that $A_{i+1} \in \mathcal{O}_d^k$
We define $\mathcal{O}_d^k$

$$\delta\mathcal{O}_d^k = \{\delta | \delta = A - B, A, B \in \mathcal{O}_d^k\} \tag{3.43}$$

which implies, we require to project ΔA_i in $\delta\mathcal{O}_d^k$ or to say $\Delta A_i \rightarrow proj_{\delta\mathcal{O}_d^k}(\Delta A_i)$

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 $AQ \in \mathcal{O}_d^k$ as in 3.41

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 $A - B, A + B \in \delta\mathcal{O}_d^k$ as by property 1 of $\mathcal{O}_d^k$

3. $\delta \in \delta\mathcal{O}_d^k$ and $Q \in \mathcal{O}_k^k$ we have $\delta Q \in \delta\mathcal{O}_d^k$ by 3.41

So if we could get an associated $Q_i \in \mathcal{O}_k^k$ with ΔA_i such that we can replace $A_{i+1} = A_i + proj_{\delta\mathcal{O}_d^k}(\Delta A_i)$ with $A_{i+1} = A_i Q_i$
As per the property of $\delta\mathcal{O}_d^k$ for $A, B \in \mathcal{O}_d^k$ we have $A - B, A + B \in \delta\mathcal{O}_d^k$, consider

$$\begin{aligned} T &= (A + B)(A - B)^\top \\ &= AA^\top - AB^\top + BA^\top - BB^\top \\ &= I - AB^\top + BA^\top - I \\ &= BA^\top - AB^\top \end{aligned} \tag{3.44}$$

Observe that T is skew symmetric

$$\begin{aligned} T^\top &= (BA^\top - AB^\top)^\top \\ &= AB^\top - BA^\top \\ &= -T \end{aligned} \tag{3.45}$$

Now recall Cayley Transformation for a Q which doesn't have -1 as one of its eigenvalues then there is a skew-symmetric matrix C satisfying the following properties

$$\begin{aligned} Q &= (I+C)^{-1}(I-C) = (I-C)(I+C)^{-1} \\ C &= (I-Q)(I+Q)^{-1} \end{aligned} \tag{3.46}$$

Let $B = \Delta A_i$ and $A = A_i$ and consider the $T_i = \Delta A_i A_i^\top - A_i \Delta A_i^\top$ and we generate associated $Q_i = (I - T_i)(I + T_i)^{-1}$ by using Cayley Transformation over T.
Based on the above observation we can write the following iterative scheme

Orthogonal Property Iterative Scheme (OPI Scheme)

Set N_{iter}
$A_0 \leftarrow random(\mathbb{R}^{d\times k})$
$A_0 \leftarrow gram\_schmidt\_algorithm(A_0)$ ▷ orthonormalize A_0, as $A_0 \in \mathcal{O}_d^k$
$\beta_0 \leftarrow optimization\_scheme(\mathcal{R}_{\beta,A_0}, \mathcal{L}_{\beta,A_0})$
for $i \leftarrow 0...N_{iter}$ **do**
  $\Delta A_i \leftarrow iterative\_scheme\_A(M_i)$ ▷ get ΔA_i from other iterative scheme
  $T_i \leftarrow \Delta A_i A_i^\top - A_i \Delta A_i^\top$
  $Q_i = (I - T_i)(I + T_i)^{-1}$
  $A_{i+1} \leftarrow A_i Q_i$
  $\beta_{i+1} \leftarrow optimization\_scheme(\mathcal{R}_{\beta,A_{i+1}}, \mathcal{L}_{\beta,A_{i+1}})$
Output $\beta_{N_{iter}}, A_{N_{iter}}$

The above defined iterative scheme 4 requires 2 internal methods $iterative\_scheme\_A(M_i)$ and $optimization\_scheme(\mathcal{R}_{\beta,A_{i+1}}, \mathcal{L}_{\beta,A_{i+1}})$ to get ΔA_i and β_{i+1} respectively.

3.3.1.2.3 MOOP Methods

3.3.1.2.3.1 No Preference Method

First we compute the **utopia point** as defined in 2.1.1 for this problem, here we have 2 dimensions for the objective to minimize namely $-\mathcal{R}_{\beta,A}$ and $\mathcal{L}_{\beta,A}$ and the utopia point values denoted by $-\mathcal{R}^*_{\beta,A} = min -\mathcal{R}_{\beta,A} = -max\ \mathcal{R}_{\beta,A} = -1$ as its inner-product between vectors divided by the norms induced by that inner-product in and $\mathcal{L}^*_{\beta,A} = min\ \ \mathcal{L}_{\beta,A} = 0$. And finally we have the condition that $AA^\top = I$, which either we can view as constraint or consider I as the utopia point value of $AA^\top$. Finally we can consider one of the 2 optimization problem framed under *No preference method*

No Preference Method Optimization Problem 1

$$\min_{\beta\in\mathbb{R}^{k\times 1}, A\in\mathcal{O}_d^k} ||[\mathcal{L}_{\beta,A} - \mathcal{L}^*_{\beta,A}, -\mathcal{R}_{\beta,A} + \mathcal{R}^*_{\beta,A}]|| \tag{3.47}$$

No Preference Method Optimization Problem 2

$$\min_{\beta \in \mathbb{R}^{k\times 1}, A \in \mathbb{R}^{d\times k}} ||[\mathcal{L}_{\beta,A} - \mathcal{L}^*_{\beta,A}, -\mathcal{R}_{\beta,A} + \mathcal{R}^*_{\beta,A}, AA^\top - I]|| \tag{3.48}$$

3.3.1.2.3.2 ϵ-Constraint Method

3.3.1.2.4 Condition Number Relaxation

For any $\alpha \geq 1$, 3.49 will contain the set specified by the condition 3.17 as proved in 1.16

$$\max_{R^{n\times d}_{t\times d} \in \mathbb{R}^{n\times d}} cond_{rel}(f, R^{n\times d}_{t\times d}) \leq \alpha \tag{3.49}$$

Here $f(X) = X \times A$.

Chapter 4

Conclusion and Future Work

So far we have defined and motivated the problem of solving fMOOP with restrictions on condition number and showed how it generalizes different problems of interests. We also summarized different methods to solve MOOPs and some important results on condition number of matrices and that of function. We also saw how we can simplify and bound the condition number of a function which is a composition of other simpler functions and its potential application applied to neural networks.

In Future works I'm planning to answer few of the the questions mentioned in the analysis chapter 3.1 after prioritizing them based on available time and difficulty of the question. Later I'm planning to implement few of the methods and compare their results for this problem.

References

[1] Turing, Alan M(1948): *Rounding-off errors in matrix processes*, 1: 287–308.

[2] Charnes, Abraham / Cooper, William W / Ferguson, Robert O(1955): *Optimal estimation of executive compensation by linear programming*, 2: 138–151.

[3] Zadeh, Lofti(1963): *Optimality and non-scalar-valued performance criteria*, 1: 59–60.

[4] Rice, John R(1966): *A theory of condition*, 2: 287–310.

[5] Charnes, Abraham / Clower, RW / Kortanek, KO(1967): *Effective control through coherent decentralization with preemptive goals*294–320.

[6] Pareto, Vilfredo / others u.a.(1971): *Manual of political economy.*

[7] Haimes, Yacov(1971): *On a bicriterion formulation of the problems of integrated system identification and system optimization*, 3: 296–297.

[8] Zeleny, Milan(1973): *Compromise programming.*

[9] Zionts, Stanley / Wallenius, Jyrki(1976): *An interactive programming method for solving the multiple criteria problem*, 6: 652–663.

[10] Vincent, Thomas L / Grantham, Walter Jervis(1981): *Optimality in parametric systems(Book).*

[11] Vincent, Thomas L(1983): *Game theory as a design tool.*

[12] Wierzbicki, Andrzej P(1986): *On the completeness and constructiveness of parametric characterizations to vector optimization problems*, 2: 73–87.

[13] Higham, Nicholas J(1987): *A survey of condition number estimation for triangular matrices*, 4: 575–596.

[14] Zionts, Stanley(1989): *Multiple criteria mathematical programming: an updated overview and several approaches*7–60.

[15] Datta, Biswa Nath(1995): *Numerical linear algebra and applications. Brooks.*

[16] Guggenheimer, Heinrich W / Edelman, Alan S / Johnson, Charles R(1995): *A simple estimate of the condition number of a linear system*, 1: 2–5.

[17] Athan, Timothy Ward / Papalambros, Panos Y(1996): *A note on weighted criteria methods for compromise solutions in multi-objective optimization*, 2: 155–176.

[18] Das, Indraneel / Dennis, John E(1998): *Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems*, 3: 631–657.

[19] Deb, Kalyanmoy / Pratap, Amrit / Agarwal, Sameer / Meyarivan, TAMT(2002): *A fast and elitist multiobjective genetic algorithm: NSGA-II*, 2: 182–197.

[20] Messac, Achille / Mattson, Christopher A(2002): *Generating well-distributed sets of Pareto points for engineering design using physical programming*, 4: 431–450.

[21] Piazza, G / Politi, T(2002): *An upper bound for the condition number of a matrix in spectral norm*, 1: 141–144.

[22] Messac, Achille / Ismail Yahaya, Amir / Mattson, Christopher A(2003): *The normalized normal constraint method for generating the Pareto frontier*, 2: 86–98.

[23] Messac, Achille / Mattson, Christopher A(2004): *Normal constraint method with guarantee of even representation of complete Pareto frontier*, 10: 2101–2111.

[24] Poli, Riccardo / Kennedy, James / Blackwell, Tim(2007): *Particle swarm optimization*, 1: 33–57.

[25] Chehab, Jean Paul / Raydan, Marcos(2008): *Geometrical properties of the Frobenius condition number for positive definite matrices*, 8-9: 2089–2097.

[26] Maréchal, Pierre / Ye, Jane J(2009): *Optimizing condition numbers*, 2: 935–947.

[27] Beltrán, Carlos / Dedieu, Jean Pierre / Malajovich, Gregorio / Shub, Mike(2010): *Convexity properties of the condition number*, 3: 1491–1506.

[28] Chen, Xiaojun / Womersley, Robert S / Ye, Jane J(2011): *Minimizing the condition number of a Gram matrix*, 1: 127–148.

[29] Li, Hongyi / Gao, Zongsheng / Zhao, Di(2011): *A note on the Frobenius conditional number with positive definite matrices*, 1: 1–9.

[30] S Motta, Renato de / Afonso, Silvana / Lyra, Paulo RM(2012): *A modified NBI and NC method for the solution of N-multiobjective optimization problems*, 2: 239–259.

[31] Deadman, Edvin / Relton, Samuel D(2016): *Taylor's theorem for matrix functions with applications to condition number estimation*354–371.

[32] Slowik, Adam / Kwasnicka, Halina(2017): *Nature inspired methods and their industry applications—Swarm intelligence algorithms*, 3: 1004–1015.

[33] Gutman, David H / Pena, Javier F(2021): *The condition number of a function relative to a set*, 1: 255–294.

[34] (QRT), Qube Research Technologies(2022): *Learning factors for stock market returns prediction, (https://challengedata.ens.fr/participants/challenges/72/).*

[35] Higham, Nicholas J(2022): *The Matrix Function Toolbox (https://www.mathworks.com/matlabcentral/fileexchange/20820-the-matrix-function-toolbox).*

[36] Das, Indraneel / Dennis, JE (1999): *An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection.* Short paper proceedings of the third world congress of structural and multidisciplinary optimization.: 411–413.

[37] Higham, Nicholas J (2008): *Functions of matrices: theory and computation.* , SIAM.

[38] Hwang, Ching Lai / Masud, Abu Syed Md (1979): *Methods for multiple objective decision making.* Multiple objective decision making—methods and applications.Springer: 21–283.

[39] Hwang, C L / Masud, Abu Syed Md (2012): *Multiple objective decision making—methods and applications: a state-of-the-art survey.* , Springer Science & Business Media.

[40] Ijiri, Yuji (1965): *Management goals and accounting for control.* , North Holland Publishing Company.

[41] Miettinen, Kaisa (1998): *No-Preference Methods.* Boston, MA, Springer US: 67–76.

[42] Miettinen, Kaisa (2012): *Nonlinear multiobjective optimization.* , Springer Science & Business Media.

[43] Miettinen, Kaisa / Ruiz, Francisco / Wierzbicki, Andrzej P (2008): *Introduction to multiobjective optimization: interactive approaches.* Multiobjective optimization.Springer: 27–57.

[44] Stadler, Wolfram (1988): *Multicriteria Optimization in Engineering and in the Sciences.* , Springer Science & Business Media.

[45] Vikhar, Pradnya A (2016): *Evolutionary algorithms: A critical review and its future prospects.* 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC).: 261–265.

[46] Wierzbicki, Andrzej P / Wessels, Jaap (2000): *The modern decision maker.* Model-Based Decision Support Methodology with Environmental Applications.Springer: 29–46.

[47] Yu, Po Lung / Leitmann, GJJOOT (1976): *Compromise solutions, domination structures, and Salukvadze's solution.* Multicriteria decision making and differential games.Springer: 85–101.

[48] Zhang, Richard / Golovin, Daniel (2020): *Random hypervolume scalarizations for provable multi-objective black box optimization.* : 11096–11105.