Solving Portfolio Optimization Problems Using MOEA/D and Lévy Flight

Yifan He

(Master's Program in Computer Science)

Advised by Hitoshi Kanoh

Submitted to the Graduate School of
Systems and Information Engineering
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering
at the
University of Tsukuba

March 2020

Abstract

Portfolio optimization (PO) is a financial task that requires an investment allocation of capital on a set of financial assets to achieve a better trade-off between return and risk. PO belongs to the big category of the multi-objective optimization problem, a class of optimization tasks containing more than one conflicting objectives. Therefore, recent studies applied multi-objective evolutionary algorithms (MOEAs) to solve the PO. Among the large group of MOEAs, MOEA/D is a method decomposing a hard MOOP into several simple single-objective optimization problems, and solving them simultaneously in one single run.

The mutation method is one of the important components in a MOEA. In some complex problems, an efficient mutation method should well control the balance between local search and global search, and sometimes utilizes the structural information of the problem, in order to converge to the real optimal. For MOEA/D, researchers have tried to use mutation methods, such as differential evolution mutation, to enhance the performance.

This thesis proposes an MOEA/D-based method, injecting an efficient mutation method named Lévy Flight (LF). LF is the moving pattern of the creature's foraging activity in a large space, without any given information on food source. However, the search efficiency of LF is not only derived from its biological meaning. Mathematically, it well controls the balance between local search and global search by a probability distribution, yet I show that LF utilizes the structural information of the PO in the later discussion.

In the experiments, the proposed algorithm is first compared with several literature methods, including three MOEA/D-based algorithms (MOEA/D-DEM, MOEA/D-DE, and MOEA/D-GA) and NSGA-II, to show its superiority on the PO tasks. Five PO benchmarks sized from 31 to 225 in OR library are used with unit constraint. Six evaluation metrics, namely generation distance, spacing, maximum spread, spread, inverted generation distance, and hypervolume, are used to provide a comprehensive assessment. The second experiment compares mutation methods based on different probability distributions. By tracking the distance between parents and offspring, I show LF contributes to the improvement by promoting global search early in the optimization. In addition, an explanation of this behavior is provided in the discussion, considering the structure of the PO. What is more, the last two experiments show extra evidence of the efficiency of LF, by comparison on LF with different truncation and comparison on the mutation methods based on normal distributions with different variance.

Overall speaking, the proposed method shows a superior performance on the PO, because LF well controls the balance between local search and global search, and utilizes the structural information of the problem. The property of LF that searches efficiently in the beginning phase can speed up the whole optimization. Based on this result, one may design an adaptive or hybrid strategy to implement a practical PO solution. However, as the discussion in this thesis is based on benchmarks with only one simple constraint, the performance of this method in real-world PO remains a further investigation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multi-objective optimization problems (MOOPs) consist of several conflicting objectives and require that an optimization algorithm finds an optimal set of trade-offs rather than a single optimal solution. In the financial world, researchers and investors face a famous MOOP known as portfolio optimization (PO). The goal of this problem is to find an optimal allocation of capital among a finite set of available financial assets, by maximizing portfolio return and minimizing portfolio risk simultaneously.

The first importance of the PO comes from the financial point of view. Investors such as banks, asset management companies and financial consultants are faced with the challenges of managing their funds, assets, and stocks. PO provides a computational method towards the optimal asset allocation. Another importance of the PO comes from its mathematical property. One of the challenges of the PO comes from its complex and large search space. Although the unconstrained model of modern portfolio theory contributed by Markowitz [1] states that PO can be solved using a quadratic programming method, realistic constraints make the problem NP-hard [2]. Therefore, the experience of studying the PO may help in other difficult optimization tasks holding similar properties.

To solve this complex problem, evolutionary algorithms (EAs) are frequently applied. By assigning different weights to each objective, PO can be addressed in a single-objective form. However, considering its natural bi-objective formulation, researchers have paid increasing efforts on methods using multi-objective evolutionary algorithms (MOEAs). Some domination-based MOEAs, such as MOGA, PAES, SPEA2 and NSGA-II, have been assessed on PO benchmarks [3, 4, 5, 6], or applied on practical PO applications [7, 8]. In addition, several multi-objective variants of swarm intelligence methods, namely non-dominated sorting multi-Objective particle swarm optimization (NS-MOPSO) [9], multi-objective bacteria foraging optimization (MOBFO) [10], and multi-objective co-variance based artificial bee colony (M-CABC) [11], have been introduced to the PO literature in the recent years. However, only a few PO researchers have paid attention to MOEA based on decomposition (MOEA/D) [12, 13, 14]. This powerful decomposition-based MOEA has not been well-discussed on the PO yet.

Recently, researchers have paid attention to Lévy Flight (LF) for solving hard optimization problems. LF is a special random walk, consisting of short motions as well as

long trajectories. Prior studies have shown the strong search capability of this mutation method [15, 16]. In this thesis, I propose a method named MOEA/D-Lévy, which injects LF into MOEA/D (**Chapter** 3), and assess it on PO with unit constraint. This modification is motivated by the efficient global search performance of LF. As one of the main challenges in PO is the high-dimensionality of the search space, the expected global search capability of LF can overcome this difficulty during optimization.

The experiments include a comparison with literature methods and a comparison with mutations based on different probability distributions. The results of six evaluation metrics, as well as a statistical test on all five datasets in a frequently used PO benchmark in OR Library [17], indicate that this method outperforms the comparison methods in most cases (**Section** 4.2). Additionally, I use an experiment to show how the addition of LF contributes to the optimization process (**Section** 4.3). **Fig.** 1.1 illustrates this contribution, by showing the population of five methods on the objective space at the 10th generation when optimizing for the Nikkei dataset. It is interesting to notice that at the very beginning of optimization the proposed method, represented by circles, can achieve a relatively good solution set compared to other literature methods. This may be caused by a compound factor of LF and the repair method, as well as the characteristics of the PO problem. I also provide another two experiments to show further evidence on this finding (**Section** 4.4 and **Section** 4.5).

The main contributions of this study include the comprehensive assessment of multiple algorithms, as well as analysis and explanation showing that LF well controls the balance between local search and global search, and utilizes the structural information of the problem when solving the PO. Based on the conclusion of this study, one may further improve the proposed algorithm, or apply it into practical use after a proper effort.

To my best knowledge, no research has applied LF into MOEA/D in the PO literature. The code and data used for the experiments in this thesis are available at a public repository[1].

[1] https://github.com/Y1fanHE/po_with_moead-levy

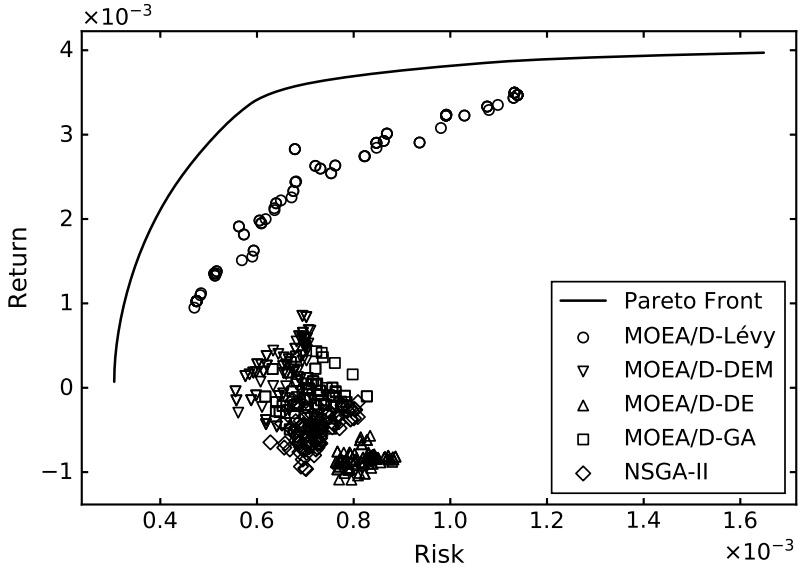


Fig. 1.1: Population of different methods for the Nikkei dataset at the 10th generation. The proposed method (MOEA/D-Lévy) quickly explores a larger area of the objective space.

Chapter 2

Background

2.1 Portfolio Optimization

2.1.1 Modern Portfolio Theory

It is not hard to realize the importance of diversification during the investment, as the old phrase "Don't put all your eggs in one basket." The portfolio is an implementation of this idea, allocating capital into multiple assets. **Fig.** 2.1 provides an example of a portfolio on four stocks. The whole pie represents all capital and the rate in the sectors represents invest rate on the corresponding assets (e.g. 30% of the capital has been allocated into Google stock).

The modern portfolio theory (MPT) proposed by Markowitz [1] provides a guideline to select a portfolio. This theory assumes that, for an investor, the expected portfolio return is a desirable thing and the variance of portfolio return (or risk) is an undesirable thing. In other words, the expected portfolio return should be maximized while the variance of portfolio return should be minimized. Furthermore, in MPT, the return of i-th asset is modeled as a random variable R_i with expected value $E(R_i)$ and variance $V(R_i)$, and a portfolio R is considered as a linear combination of multiple assets in (2.1), where w_i represents the invest rate of i-th asset. As the expected value of a weighted sum is the weighted sum of expected values, we have the expected portfolio return in (2.2). The risk of a portfolio is computed as the variance of a weighted sum. It can be expressed in (2.3) using co-variance between two assets $\sigma_{ij} = E\{[R_i - E(R_i)][R_j - E(R_j)]\}$.

$$R = w_1R_1 + w_2R_2 + \cdots + w_nR_n \tag{2.1}$$

$$E(R) = w_1E(R_1) + w_2E(R_2) + \cdots + w_nE(R_n) \tag{2.2}$$

$$V(R) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_iw_j\sigma_{ij} \tag{2.3}$$

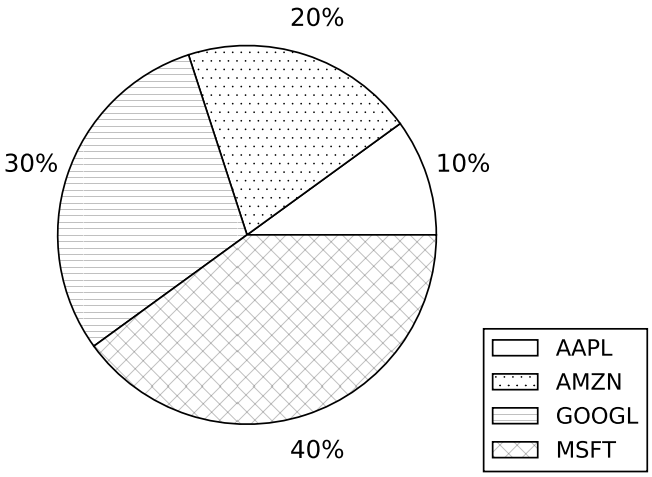


Fig. 2.1: A sample portfolio on four assets

2.1.2 Unconstrained Portfolio Optimization Problem

This thesis deals with the PO model based on MPT. In this unconstrained PO model, the first objective return in (2.4) is being maximized, while the second objective risk in (2.5) is being minimized. $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ is a vector representing the invest ratios of n assets. r_i is the expected return rate of i-th asset, and σ_{ij} is co-variance between the return rate of i-th asset and j-th asset. It is interesting to notice that this " unconstrained " problem includes two constraints, mathematically speaking. The first constraint in (2.6) requires to allocate all of the capital during the investment. This is usually represented as the sum of investments being equal to one (unit constraint). The second constraint in (2.6) indicates that no short selling is allowed during the investment.

$$\max_{\boldsymbol{w}} E = \sum_{i=1}^{n} w_i r_i \tag{2.4}$$

$$\min_{\boldsymbol{w}} V = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_{ij} \tag{2.5}$$

subject to,

$$\sum_{i=1}^{n} w_i = 1 \tag{2.6}$$

$$0 \leq w_i \leq 1 \tag{2.7}$$

2.2 Multi-Objective Optimization Problem

2.2.1 Definition

One may easily realize that there are more than one objectives in unconstrained PO (**Section** 2.1.2). This problem can be categorized into multi-objective optimization problem (MOOP). A MOOP can be defined as an optimization task which contains more than one conflicting objectives. Here, "conflicting" means that there is no such a solution holding optimal values of all objectives. Consider the following minimization examples.

$$f(x) = x^2,\ g(x) = (x-2)^2,\ h(x) = x,\ \text{subject to } x \geq 0$$

- For function $f(x)$, the minimum can be achieved at $x = 0$.
- For function $g(x)$, the minimum can be achieved at $x = 2$.
- For function $h(x)$, the minimum can be achieved at $x = 0$.

Thus, if one is going to minimize $f(x)$ and $g(x)$ at the same time, this optimization task is a MOOP, for the minimum of two objectives cannot be achieved at the same time. However, if one is to minimize $f(x)$ and $h(x)$, as both objectives hold the minimum $x = 0$, this task is not a MOOP. In other words, by minimizing one of the objectives, the other objective can be minimized simultaneously.

Mathematically, a minimization MOOP with M objectives can be defined as follows, where $\boldsymbol{x}$ is a design vector on the n-dimensional decision space, Ω represents the feasible region of the problem. As the objective $\boldsymbol{f}(\boldsymbol{x})$ is a vector, MOOP is sometimes called vector optimization problem.

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})\} \tag{2.8}$$

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Omega \tag{2.9}$$

2.2.2 Domination and Pareto Front

In a single-objective optimization problem (SOOP), it is easy to compare two candidates, by taking a subtraction on their objective values. However, in a MOOP, the existence of multiple objectives make the comparison harder. Consider an example of house renting with two objectives to be minimized, the price per month and distance from the university. For house A, it costs 45,000 Yen per month and 300 meters away from the university. B costs 23,000 Yen per month but 2,000 meters away from the university. C costs 60,000 Yen per month and 1,500 meters from the university. It is obvious that a rational person should choose A rather than C because it is cheaper and closer to the university. However, when considering A and B, there is no winner. B is 22,000 Yen cheaper than A, but A is 1,700 meters closer to the university. Thus, people who do not care about money may prefer A, but others may prefer B.

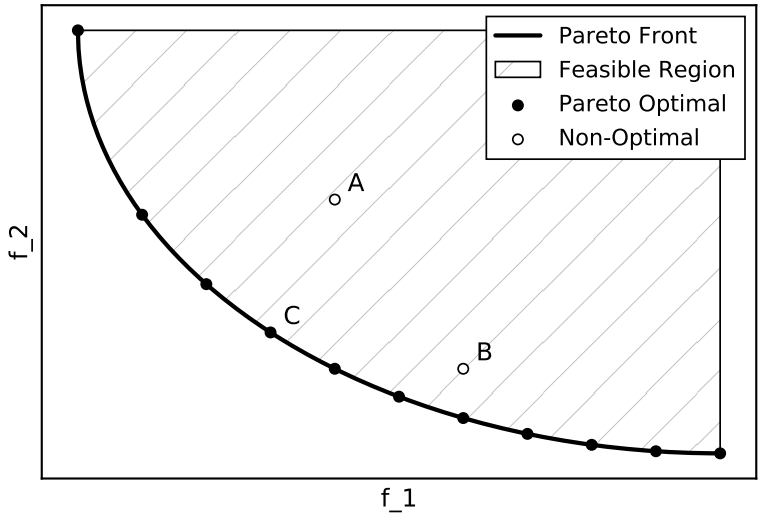


Fig. 2.2: An intuition of domination and Pareto Front

The above case can be expressed mathematically using domination. The mathematical definition of domination in a M-objective minimization MOOP is as follows. The symbol "$\prec$" is read as "dominates", i.e., "$\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{y})$" is read as $\boldsymbol{f}(\boldsymbol{x})$ dominates $\boldsymbol{f}(\boldsymbol{y})$.

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{y}) \Leftrightarrow & \forall m = 1, \ldots, M,\ f_m(\boldsymbol{x}) \leq f_m(\boldsymbol{y}) \wedge \\ & \exists m = 1, \ldots, M,\ f_m(\boldsymbol{x}) < f_m(\boldsymbol{y}) \end{aligned} \tag{2.10}$$

In a minimization SOOP, the optimal can be defined as the solution(s) which are non-larger than any other feasible solutions. Similarly, optimal in a MOOP can be defined as solutions that are non-dominated by any other feasible solutions. This definition can be mathematically described as follows, where Ω is the feasible region and $PF(\Omega)$ is optimal solutions of this feasible region. Such a set of optimal solutions is named as Pareto Front. Such an optimal solution on the Pareto Front is called Pareto optimal.

$$PF(\Omega) = \{\boldsymbol{\omega} \in \Omega \mid \{\boldsymbol{\omega}' \in \Omega \mid \boldsymbol{\omega}' \prec \boldsymbol{\omega}\} = \emptyset\} \tag{2.11}$$

Fig. 2.2 provides an example of domination and Pareto Front. In the minimization problem in this graph, we have C dominates A. A and B are non-dominated by each other. In addition, C is one of the Pareto optimal in this feasible region.

2.2.3 Difficulties in Multi-Objective Optimization Problem

A MOOP is usually considered more difficult than a SOOP for the following difference [18].

- **Two goals vs. one goal:** In a SOOP, the solution set is required to converge as close as possible to the true optimal. However, in a MOOP, despite convergence, the solution set is also required to hold a good diversity to well estimate the whole Pareto Front.

- **Two spaces vs. one space:** It is important to notice that one is dealing with two spaces, decision space, and objective space, in a MOOP, especially when considering diversity. This diversity can be a requirement in both two spaces. Additionally, there is no need for diversity in these two spaces to be matched.

2.3 Multi-Objective Evolutionary Algorithm based on Decomposition

2.3.1 Overview

Researchers have come up with a series of MOEAs. This group of optimization methods can achieve a set of estimated Pareto optimal solutions in one single run. The first MOEA may be vector evaluated genetic algorithm (VEGA) [19]. In VEGA, the whole population is randomly divided into several equal-sized partitions, and each partition is evaluated with one individual objective function (e.g. for a bi-objective problem, the population will be divided into two equal-sized sub-populations, one is evaluated with the first objective, and the other is evaluated with the second objective). During the reproduction process, offspring are generated based on parent candidates from the same partition. This algorithm allows one to achieve trade-off solutions in a few generations, but after a large generation of evolution, the solution set usually converges to the best individuals for each objective. The main reason is that this method only considers the corresponding objective for a solution during the evaluation. However, the rest of the objectives should not be ignored in the sense of MOOP.

Since then, MOEA researchers have not made too much progress until the concept of domination was introduced to design MOEAs in Goldberg's work [20]. Based on his suggestion, researchers have developed many different implementation of MOEAs, such as multi-objective GA (MOGA) [21], non-dominated sorting GA (NSGA) [22], niched-Pareto GA (NPGA) [23], strength Pareto EA (SPEA) [24], Pareto archived evolutionary strategy (PAES) [25] and NSGA-II [26].

Take NSGA-II as an example. This algorithm first assigns ranks to every solution in the population using a domination-based method called fast non-dominated sorting. As shown in **Fig.** 2.3, the first rank contains non-dominated solutions in the population. The second rank contains non-dominated solutions of the population excluding the first rank. Thus, the solutions with a lower rank dominate those with a higher rank. Then, in order to keep diversity, NSGA-II assigns crowding distance to the solutions with the same rank. This distance is computed as the sum of the distance between the two closest solutions to the current solution, in terms of each objective (i.e., show in **Fig.** 2.4). A partial order, which prefers firstly a lower rank and then a larger crowding distance only when the ranks are the same, is applied in both binary tournament selection and elite preservation process.

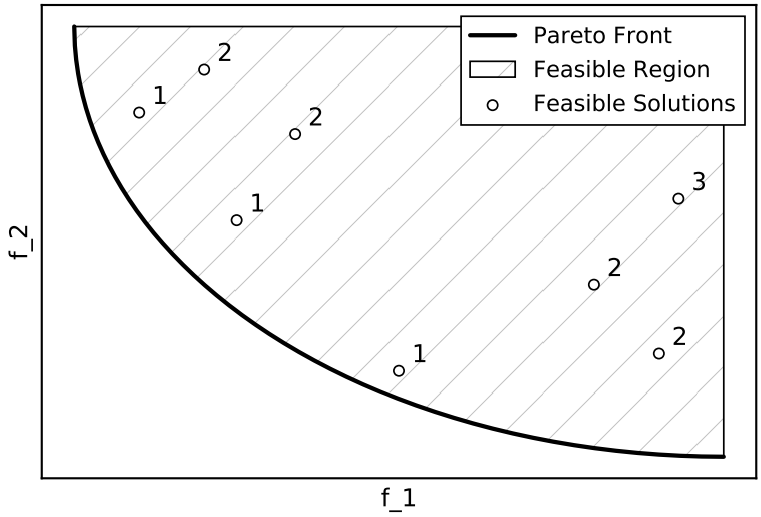


Fig. 2.3: An intuition of fast non-dominated sorting result. The number near a solution shows the non-dominated rank.

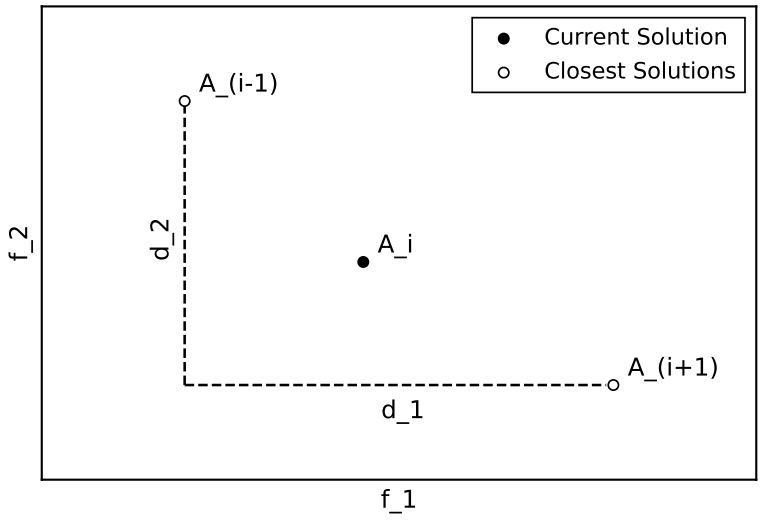


Fig. 2.4: An intuition of crowding distance assignment. The crowding distance of A_i is $d_1 + d_2$ if A_{i+1} and A_{i-1} are the closest solutions in terms of each objective to A_i.

However, this algorithm may not perform a good diversity when addressing some hard problems, as the partial order prefers convergence first rather than diversity. One example of this limitation was reported in Fukumoto's study [27].

In 2007, Zhang proposed a multi-objective evolutionary algorithm based on decomposition (MOEA/D) using the concept of decomposition rather than domination [28]. Generally, decomposition is a technique transforming an entire, large and complex problem to several partial, small and simple problems, and thus provides efficiency. In **Section** 2.3.2 and **Section** 2.3.3, several decomposition methods and MOEA/D will be introduced.

2.3.2 Decomposition

Decomposition is a classical method to solve a MOOP, transforming a vector optimization task to a scalar optimization task. Considering the following example. To implement a decomposition, one simple approach is to compute the linear combination of original objectives as a new optimization problem. This new problem is usually called sub-problem.

$$\min \boldsymbol{f}(x) = \{f_1(x), f_2(x)\}$$
$$f_1(x) = x^2,\ f_1(x) = (x-2)^2,\ x \geq 0$$

- **Sub-problem 1:** $s_1(x) = 0.1 \cdot f_1(x) + 0.9 \cdot f_2(x) = x^2 - 3.6x + 3.6$. The minimum is achieved at $x = 1.8$.
- **Sub-problem 2:** $s_2(x) = 0.5 \cdot f_1(x) + 0.5 \cdot f_2(x) = x^2 - 2x + 2$. The minimum is achieved at $x = 1$.
- **Sub-problem 3:** $s_3(x) = 0.8 \cdot f_1(x) + 0.2 \cdot f_2(x) = x^2 - 0.8x + 0.8$. The minimum is achieved at $x = 0.4$.

One may find that by using different weight combinations, the optimal of sub-problem is different. By using multiple weight combinations, one can achieve a set of Pareto optimal. Several frequently used decomposition approaches will be introduced in the following paragraphs.

Weighted Sum Approach

The example above applies the weighted sum approach. A mathematical expression (in minimization case) of this method is present as follows, where $\boldsymbol{\lambda}$ is a M-dimensional weight vector. As the weighted sum approach decomposes Pareto Front into tangent points, it cannot achieve all Pareto optimal if the front does not hold convexity.

$$\min_{\boldsymbol{x}} g^{ws}(\boldsymbol{x} \mid \boldsymbol{\lambda}) = \sum_{m=1}^{M} \lambda_m f_m(\boldsymbol{x}) \tag{2.12}$$

$$\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M),\ \lambda_1 + \cdots + \lambda_M = 1 \tag{2.13}$$

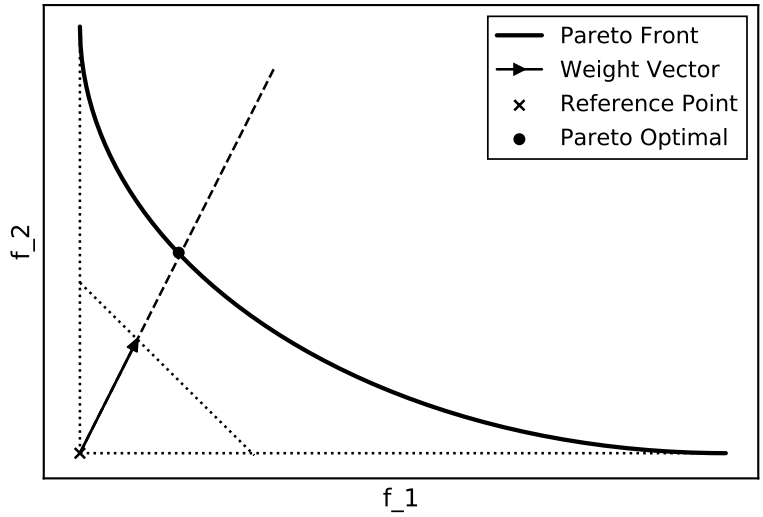


Fig. 2.5: An intuition of weighted Tchebycheff approach

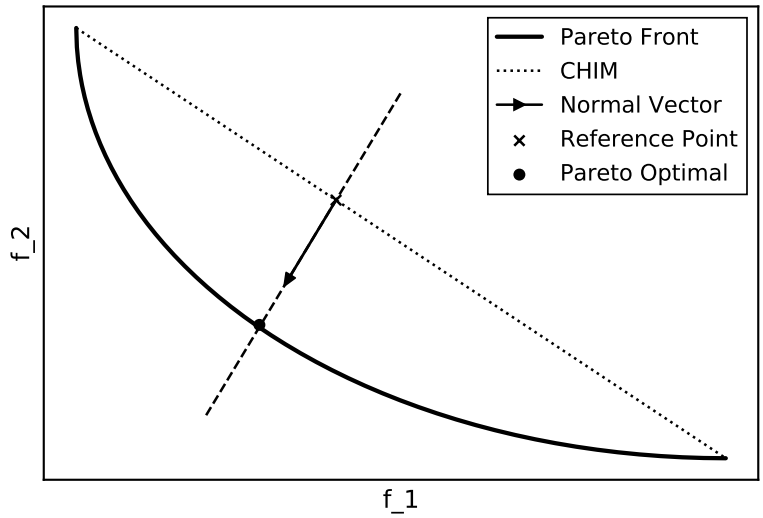


Fig. 2.6: An intuition of NBI-style Tchebycheff approach

Weighted Tchebycheff Approach

Another frequently used method is the weighted Tchebycheff approach. This approach decomposes the Pareto Front into the intersection of the front and a straight line determined by a weight vector $\boldsymbol{\lambda}$ and a reference point (utopia point) $\boldsymbol{z^*}$, as shown in **Fig.** 2.5. Thus, by using this approach, all Pareto optimal can be retrieved. The mathematical expression of the weighted Tchebycheff approach (in minimization case) is present as follows.

$$\min_{\boldsymbol{x}} g^{te}(\boldsymbol{x} \mid \boldsymbol{\lambda}, \boldsymbol{z^*}) = \max_{m=1,\dots,M} \{\lambda_m | f_m(\boldsymbol{x}) - z_m^*|\} \tag{2.14}$$

$$\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_M),\ \lambda_1 + \cdots + \lambda_M = 1 \tag{2.15}$$

$$\boldsymbol{z^*} = (z_1^*, \dots, z_M^*),\ z_m^* = \min\{f_m(\boldsymbol{x})\} \tag{2.16}$$

Normal Boundary Intersection style Tchebycheff Approach

Normal Boundary Intersection style (NBI-style) Tchebycheff approach [12] is a variant of the weighted Tchebycheff approach. In this method, the reference point $\boldsymbol{z^*}$ is a point in the Convex Hull of Individual Minima (CHIM) rather than the utopia point. Correspondingly, the weight vector $\boldsymbol{\lambda}$ in this method is the normal vector of CHIM. In a bi-objective optimization problem, CHIM is the straight line connecting two extreme points of the feasible region. The mathematical expression of the NBI-style Tchebycheff approach in the bi-objective minimization case is present as follows, where $\boldsymbol{F}^1$ and $\boldsymbol{F}^2$ are two extreme points, and a is a parameter between 0 and 1. Similar to the weighted Tchebycheff approach, the NBI-style Tchebycheff approach also decomposes the Pareto Front into the intersection of the front and the straight line (shown in **Fig.** 2.6).

$$\min_{\boldsymbol{x}} g^{tn}(\boldsymbol{x} \mid \boldsymbol{\lambda}, \boldsymbol{z^*}) = \max_{m=1,2} \{\lambda_m (f_m(\boldsymbol{x}) - z_m^*)\} \tag{2.17}$$

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2),\ \lambda_1 = |F_2^1 - F_2^2|,\ \lambda_2 = |F_1^1 - F_1^2| \tag{2.18}$$

$$\boldsymbol{z^*} = a \cdot \boldsymbol{F}^1 + (1 - a) \cdot \boldsymbol{F}^2 \tag{2.19}$$

All three approaches can only achieve one optimal at once. To retrieve multiple Pareto optimal, a set of weight vectors (i.e., for weighted sum and weighted Tchebycheff approach) or reference points (i.e., for NBI-style Tchebycheff approach) are required. Additionally, this set of vector or points should be uniformly distributed in order to achieve an optimal set with good diversity. The methods to generate such a set of vectors $\{\boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(N)}\}$ or points $\{\boldsymbol{z}^{*(1)}, \dots, \boldsymbol{z}^{*(N)}\}$ are present as follows.

$$\boldsymbol{\lambda}^{(i)} = (\lambda_1^{(i)}, \dots, \lambda_M^{(i)}),\ \lambda_m^{(i)} \in \left\{ \frac{1}{H}, \dots, \frac{H}{H} \right\} \tag{2.20}$$

$$\boldsymbol{z^{*(i)}} = a^{(i)} \cdot \boldsymbol{F}^1 + [1 - a^{(i)}] \cdot \boldsymbol{F}^2,\ a^{(i)} = \frac{N - i}{N - 1} \tag{2.21}$$

Algorithm 1 Original MOEA/D using weighted Tchebycheff approach

Generate a set of weight vectors $\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(N)}\}$;
Determine neighbor;
Initialize population $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$;
Compute reference point $\boldsymbol{z}^*$;
while stopping criteria **do**
    for $\boldsymbol{x}^{(i)}$ in population **do**
        Select parents from neighbor of $\boldsymbol{x}^{(i)}$;
        Reproduce an offspring $\boldsymbol{y}$ by GA operator;
        Update reference point $\boldsymbol{z}^*$;
        for $\boldsymbol{x}^{(p)}$ in neighbor of $\boldsymbol{x}^{(i)}$ **do**
            if $g^{te}(\boldsymbol{y} \mid \boldsymbol{\lambda}^{(p)}, \boldsymbol{z}^*) \leq g^{te}(\boldsymbol{x}^{(p)} \mid \boldsymbol{\lambda}^{(p)}, \boldsymbol{z}^*)$ **then**
                Set $\boldsymbol{x}^{(p)} = \boldsymbol{y}$;
            end if
        end for
    end for
end while

2.3.3 Procedure of the Algorithm

As shown in **Section** 2.3.2, to retrieve a set of different Pareto optimal, one can try to solve multiple sub-problems. By applying single-objective EA (SOEA), one can retrieve a set of non-dominated solutions, however, in multiple runs.

MOEA/D provides a framework which can maintain the non-dominated set in one single run by defining neighbor between solutions. In MOEA/D, one individual candidate serves as the solution of one sub-problem (e.g. $\boldsymbol{x}^{(1)}$ is the solution of sub-problem 1, $\boldsymbol{x}^{(2)}$ is the solution of sub-problem 2). These sub-problems are defined by different weight vectors. If the distance between two vectors is close, the parameter (or the property) of the sub-problems are close as well. The neighbor of a sub-problem is the sub-problems determined by closest weight vectors to its corresponding vector. Similarly, the neighbor of a solution is a group of solutions which are corresponding with neighbor sub-problems. A good solution of a sub-problem should hold a good performance in terms of its neighbor sub-problems as well. Thus, though only one individual is corresponding with one sub-problem, the algorithm can make use of its neighbor during the reproduction process. In MOEA/D, the parents, which are used during generic operations, are selected from the neighbor solutions. In addition, the generated offspring can be used to update neighbor solutions as well.

What is more, the applying weighted Tchebycheff approach, the utopia point of the feasible region is required. However, in MOEA/D, the algorithm uses the utopia point of the population instead of that of the feasible region. Once a better offspring is searched, this utopia point will also be updated during the evolutionary process.

The entire algorithmic description of MOEA/D is present in **Algorithm** 1 (i.e., weighted Tchebycheff approach is used as the decomposition method).

2.4 Lévy Flight

2.4.1 Definition

Lévy Flight is a special random walk whose individual steps are taken from a heavy-tailed distribution. Thus, LF is a random process combining short motions and long trajectories. This property helps escape from local optimal, and thus improves the performance in an optimization task.

Fig. 2.7 illustrates the probability density function (PDF) of Cauchy distribution (i.e., an example of heavy-tailed distribution). **Fig.** 2.8 shows a comparison of a 100-step random walk in two-dimensional space using standard normal distribution and Cauchy distribution. Both are started at point $(0, 0)$. The search area of the random walk using Cauchy distribution is much larger than the random walk using the standard normal distribution.

LF can be expressed in the following mathematical formulation, where $x^{(t)}$ represents the position of a point at time t, and ξ is a random variable with a probability density of P. In this thesis, the heavy-tailed distribution applied in LF is named as symmetrical stable distribution. An introduction of this group of distributions is provided in **Section** 2.4.3.

$$x^{(t)} = x^{(t-1)} + \xi \tag{2.22}$$

$$\xi \sim P \tag{2.23}$$

2.4.2 Lévy Flight Foraging Hypothesis

When looking at random searches from the view of biological encounters, researchers have argued that creatures have evolved to exploit LF, for it optimizes random searches. This statement is the so-called Lévy Flight Foraging Hypothesis. The following paragraphs provide an interpretation of this hypothesis by recalling Viswanathan's review study [15].

Biological encounters occur because living organisms need to interact with other individuals, both intra-species (e.g. mating) and inter-species (e.g. predation). Therefore, the encounter rate may serve as a reasonable evaluation of the fitness of the organisms. For instance, with a low encounter rate, an organism cannot access the food source efficiently, and thus holds a low fitness in the sense of evolutionary nature.

Furthermore, it is easy to realize that this encounter rate can be determined by multiple factors, and within them, search strategies may be the most important one. Viswanathan's study states that without the prior information of the randomly and sparsely located targets, searching by LF tends to be a good choice through the analytical results. The LF described in the same study is a random walk based on the following step length distribution.

$$P(\xi) \sim \xi^{-\mu},\ 1 < \mu \leq 3 \tag{2.24}$$

In the same study, the author has also provided a review on empirical studies showing that living organisms perform LF or LF-like searches during foraging motion, including microorganisms (amoeba), fruit flies, honey bees, sharks, sea turtles, etc. All these examples may imply LF as an efficient search strategy which is derived from biological evolution.

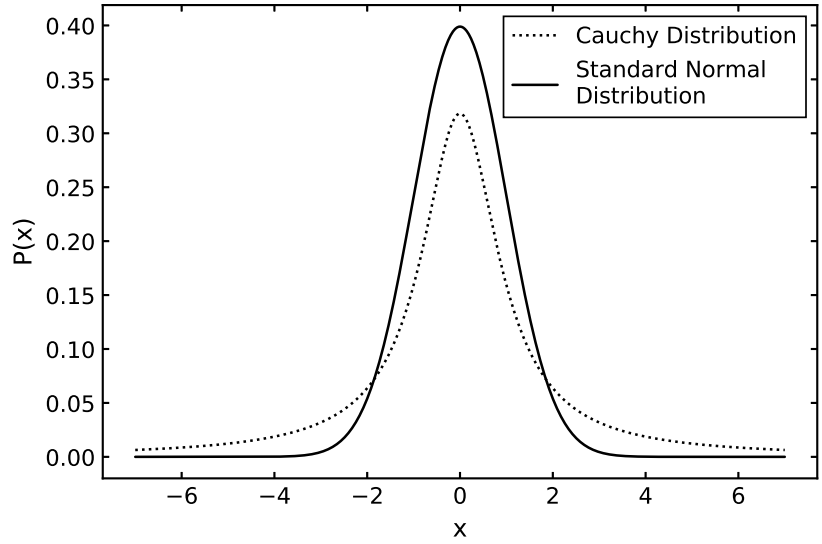


Fig. 2.7: PDF of Cauchy distribution and standard normal distribution

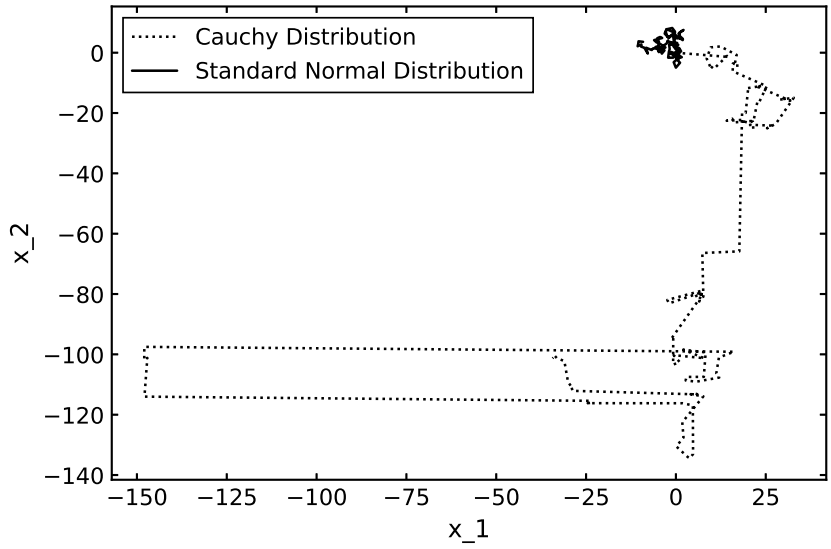


Fig. 2.8: Random walks by Cauchy distribution and standard normal distribution

2.4.3 Symmetrical Stable Distribution

The analytical form of symmetrical stable distributions can only be found for several special cases. A general form of this distribution is given in Mantegna's work [29]. In this formula, β is index parameter ($0 < \beta \leq 2$), and γ is a scaling parameter which is usually set to 1 for convenience. One may simplify the formula to (2.26) by taking the first term of its power series expansion [30].

$$L(x) = \int_0^\infty \exp(-\gamma q^\beta) \cos(qx) dq \tag{2.25}$$

$$L(x) \sim |x|^{-1-\beta}, \ |x| \to \infty \tag{2.26}$$

One of the important properties of the symmetrical stable distribution is that, it holds an infinity variance when $0 < \beta < 2$, and an undefined expectation value when $0 < \beta < 1$. Therefore, it can be applied to search even an unbounded space when $0 < \beta < 1$, as the average position does not exist [30].

2.4.4 Implementation of Lévy Flight

Several prior studies have provided the computer simulation methods to generate LF, however, approximately.

Mantegna's Algorithm

Mantegna's algorithm [29] generates random variables from a symmetrical stable distribution with $0.3 \leq \beta \leq 1.99$, using the following formula, where Γ is the Gamma function.

$$L\acute{e}vy(\beta) \sim \frac{u}{|v|^{1/\beta}}, \ u \sim N(0, \sigma_u^2), \ v \sim N(0, \sigma_v^2) \tag{2.27}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \ \sigma_v = 1 \tag{2.28}$$

Gutowski's Algorithm

Gutowski's algorithm [30] produces a random sequence distributed following (2.29). It can be simply implemented by applying the formula in (2.30) where u is drawn from a uniform distribution between 0 and 1. This algorithm can be used for all $\beta \in (0, 2)$, however, it simulates a right-skewed distribution rather than the symmetrical one (i.e., $L\acute{e}vy(\beta) \geq 0$).

$$L(x) \sim \frac{C}{(1+x)^{1+\beta}} \tag{2.29}$$

$$L\acute{e}vy(\beta) \sim u^{-\frac{1}{\beta}} - 1 \tag{2.30}$$

In this thesis, as the problem is real-coded, I apply Mantegna's algorithm, for it can easily generate negative values from the symmetrical property.

2.5 Related Works

2.5.1 Variants of MOEA/D

Recently, researchers have tried to combine MOEA/D with mutation methods in other meta-heuristics to enhance its search capability, such as particle swarm optimization (PSO), differential evolution (DE), and ant colony optimization (ACO) [31, 32, 33]. MOEA/D-DE [32] injects the DE operator and polynomial mutation operator into MOEA/D. The DE operator is present as follows, where $\boldsymbol{x}^{(i)}$, $\boldsymbol{x}^{(j)}$ and $\boldsymbol{x}^{(k)}$ are parents, $\boldsymbol{y}$ is offspring, and $rand$ is a random number between 0 and 1.

$$\boldsymbol{y} = \begin{cases} \boldsymbol{x}^{(i)} + F \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)}), & rand < CR \\ \boldsymbol{x}^{(i)}, & rand \geq CR \end{cases} \tag{2.31}$$

The original DE operator includes a DE mutation step and a crossover step with the original parent. However, the authors of MOEA/D-DE have suggested to set CR to 1.0 to deal with complicated problems, which means only the DE mutation step will be implemented. Additionally, they have designed a diversity keeping strategy, including an upper limitation for updating neighbor, and a small probability to select parents from the whole population rather than the neighbor. The parent selection scheme of MOEA/D-DE has been well discussed by Tanabe [34]. This study has reported that using curr/1 (i.e., select current individual as $\boldsymbol{x}^{(i)}$) and WR (i.e., $\boldsymbol{x}^{(i)}$, $\boldsymbol{x}^{(j)}$, $\boldsymbol{x}^{(k)}$ can be the same individual) or WPR (i.e., $\boldsymbol{x}^{(j)}$ and $\boldsymbol{x}^{(k)}$ cannot be same, but either can be the same as $\boldsymbol{x}^{(i)}$) outperforms other settings.

2.5.2 Portfolio Optimization Using MOEAs

Since the genetic algorithm was firstly introduced into PO literature in a multi-objective form in 1993 [35], prior studies have assessed a large group of MOEAs. One frequently used benchmark in these assessments is the OR library [17], which contains five PO datasets, namely Hangseng, DAX 100, FTSE 100, S&P 100 and Nikkei. VEGA, Fuzzy VEGA, MOGA, SPEA2, and NSGA-II have been compared on Hangseng with constraints [3]. Another two comparison studies using the same dataset have reported that NSGA-II outperforms PESA, PAES and APAES [4, 5]. NSGA-II, PESA, and SPEA2 have also been assessed on DAX 100 with constraints [6]. The results have shown that NSGA-II and SPEA2 hold the best average performance. In Branke ' s study [36], envelope-based MOEA has been assessed on Hangseng, S&P 100, and Nikkei with realistic constraints.

While EAs are modeled from the evolutionary process, swarm intelligence (SI) is a group of algorithms based on the self-organization of individuals. A survey has reported an increasing attention on using MOEAs with SI to solve PO [37]. MOPSO has been compared with PSFGA, SPEA2, and NSGA-II on Hangseng [38]. The results have indicated that MOPSO outperforms the other three methods significantly. NS-MOPSO, MOBFO, and M-CABC have been proposed and assessed on all five PO datasets in the OR library with constraints [9, 10, 11].

Despite performance assessment using benchmarks, researchers have also shown strong

interest in developing MOEAs for practical PO. NSGA-II, SPEA2, and IBEA have been compared on solving PO with financial data in the Venezuelan market [7]. Five domination based MOEAs have been tested on cardinality constrained PO with a dataset containing over 2000 assets [8]. Variants of MOPSO have been proposed to solve the constrained PO with a realistic dataset [39, 40]. Several studies have focus on designing specific initialization methods, problem guided mutation and constraint handling techniques on PO [41, 42, 43, 44].

While most of the prior studies are based on domination methods, few researchers have focused on solving PO with decomposition-based MOEAs. Zhang ' s study [12] has proposed a new decomposition method and used MOEA/D-DE to solve a constrained PO. The experimental results show that MOEA/D-DE performs better than NSGA-II-DE. A new weight vector generation approach to achieve an evenly distributed vector set has been proposed by Zhang [13]. In Zhou's study [14], researchers have combined a data envelopment analysis technique with MOEA/D and assessed this method on ZDT1-3 benchmarks as well as PO application on 10 stocks. The results show that the proposed method can outperform MOEA/D. Thus, performance assessment and application of MOEA/D on PO have not been well discussed so far.

In Zhang ' s study [12], researchers have reported that the original weighted Tchebycheff decomposition cannot achieve an evenly distributed optimal set, for the scale of two objectives are usually different in PO. To solve this problem, they have proposed the NBI-style Tchebycheff decomposition approach (i.e., described in **Section** 2.3.2). To generate offspring, they have used the DE mutation operator. In that work, the three parents are randomly selected from the neighbor of the current individual. What is more, although original MOEA/D-DE [32] uses diversity keeping strategies, the usage of such strategies has not been reported in Zhang's study.

2.5.3 Lévy Flight and Optimization

One biological application of LF is the LF foraging theory [15] in **Section** 2.4.2. This theory states that creatures have evolved to use LF during foraging for its optimized random search capability. Despite this natural example, LF has been applied in optimization problems in various areas such as physics, biology, statistics, finance, and economics [45].

In addition, researchers have developed and enhanced metaheuristics using LF. Cuckoo Search (CS) [46] is an efficient optimization method using LF to implement a global search. Researchers have also used LF to enhance PSO and ABC [16, 47, 48, 49]. In Zhang's study [50], a modified CS has been injected into MOEA/D to solve a spectrum allocation problem. However, the mis-setting of evaluation times (i.e., they do not set an equal evaluation time for all algorithms in the experiment) and the unclear algorithm description (i.e., they do not report some details in the numerical process when describing algorithm procedure) may confuse other researchers when understanding and implementing this method. Thus, there remains a proper assessment of MOEA/D injected with LF. Some studies have applied the single-objective form of CS to solve PO [51, 52, 53]. However, to my best knowledge, there is no research injecting LF into MOEA/D in the PO literature.

Chapter 3

Proposed Method

3.1 Lévy Flight Mutation

The LF mutation in the proposed method is similar to the DE mutation (i.e., described in **Section** 2.5), utilizing the difference between individuals. However, the scaling factor in LF mutation is a vector generated from heavy-tailed distribution rather than a constant. What is more, LF mutation only uses two parents, while DE mutation uses three parents. The formulation of this mutation method is as follows, where $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ are parents, $\boldsymbol{y}$ is offspring, $\oplus$ means entry-wise multiplication, α_0 is a constant parameter, and $\boldsymbol{Lévy(\beta)}$ is a vector where each component is generated using Mantegna ' s algorithm [29] (i.e., $0.3 \leq \beta \leq 1.99$).

$$\boldsymbol{y} = \boldsymbol{x}^{(i)} + \alpha_0 \cdot (\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}) \oplus \boldsymbol{Lévy(\beta)} \tag{3.1}$$

There are several reasons motivating this modification. First, DE mutation utilizes the difference between individuals. This difference is generally considered to be larger at the beginning but smaller at the end. Thus, in the ending phrase, DE mutation usually implements local search. By introducing LF, the small chance of implementing global search can help to escape from the local optimal. Second, the performance of DE mutation depends on the diversity of the population. In MOEA/D, when there is a high-quality offspring generated, multiple neighbors will be updated to exactly the same value (i.e., same as the generated offspring). This may cause the diversity of the population decreasing rapidly. However, by introducing LF, the mutation can hold a good performance even when the population does not hold a good diversity.

3.2 Repair Method

The offspring generated by LF mutation may not satisfy the constraints in (2.6) and (2.7). Thus, it is necessary to apply a proper constraint handling technique after reproduction. In this research, a general repair method following the description in Algorithm 2 has been applied. These repair steps will first set negative variables to 0, and then scale the entire vector (offspring) so that the summation of all variables equals to 1.

Algorithm 2 Repair Method

1: **for** y_i in offspring $\boldsymbol{y}$ **do**
2: **if** $y_i < 0$ **then**
3: Set $y_i = 0$;
4: **end if**
5: **end for**
6: Compute $s = \sum y_i$;
7: **for** y_i in offspring $\boldsymbol{y}$ **do**
8: Set $y_i = y_i/s$;
9: **end for**

3.3 Proposed MOEA/D-Lévy Algorithm

An entire algorithmic description of the proposed method is presented in **Algorithm** 3. In the following chapters, the proposed method will be named as MOEA/D-Lévy for convenience. In MOEA/D-Lévy, the population is initialized using a uniform distribution between 0 and 1. When selecting parents, the current individual is selected as $\boldsymbol{x^i}$, while $\boldsymbol{x^j}$ is randomly selected from the neighbor of the current individual. MOEA/D-Lévy injects LF mutation and polynomial mutation operators into MOEA/D. When an offspring is generated, the algorithm will implement repair steps to satisfy constraints. NBI-style Tchebycheff decomposition approach (i.e., described in **Section** 2.3.2) has been applied to deal with different scale of two objectives in PO. I have also applied the diversity keeping strategies proposed in Li ' s study [32] (i.e., described in **Section** 2.5.1), including a small proportion $1 - \sigma$ to select parents from the whole population (i.e., described in Algorithm 3, Line 6 to 10), where $rand(0, 1)$ means a random number generated from uniform distribution with range 0 to 1, as well as an upper limitation n_r for updating neighbor (i.e., described in **Algorithm** 3, Line 20 to 23).

Compared with the original MOEA/D algorithm [28], the main modification of the proposed method includes NBI-style Tchebycheff decomposition (the original MOEA/D applies Tchebycheff method), diversity keeping strategies (the original MOEA/D does not apply it), and reproduction method based on LF and polynomial mutation (the original MOEA/D applies GA operator).

Algorithm 3 Proposed MOEA/D-Lévy

Determine neighbor;
Initialize population $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$;
Compute extreme points $\boldsymbol{F^1}, \boldsymbol{F^2}$;
while stopping criteria **do**
    for $\boldsymbol{x}^{(i)}$ in population **do**
        if $rand(0,1) < \sigma$ **then**
            Set $B(i)$ as neighbor of $\boldsymbol{x}^{(i)}$;
        else
            Set $B(i)$ as population $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$;
        end if
        Select parents from $B(i)$;
        Reproduce an offspring $\boldsymbol{y}$ by LF mutation and polynomial mutation;
        Repair $\boldsymbol{y}$ to satisfy constraints;
        Update extreme points $\boldsymbol{F^1}, \boldsymbol{F^2}$;
        Set update counter $n_c = 0$;
        Re-arange $B(i)$ in random order;
        for $\boldsymbol{x}^{(p)}$ in neighbor of $B(i)$ **do**
            if $g^{tn}(\boldsymbol{y} \mid \boldsymbol{\lambda}, \boldsymbol{z}^{*(p)}) \leq g^{tn}(\boldsymbol{x}^{(p)} \mid \boldsymbol{\lambda}, \boldsymbol{z}^{*(p)})$ **then**
                Set $\boldsymbol{x}^{(p)} = \boldsymbol{y}$;
                Set $n_c = n_c + 1$;
                if $n_c \geq n_r$ **then**
                    Break;
                end if
            end if
        end for
    end for
end while

Chapter 4

Experiments

4.1 Experimental Method

4.1.1 Portfolio Optimization Benchmark

The benchmarks used are all five PO datasets in the OR library [17], containing 31, 85, 89, 98, and 225 assets from 1992 to 1997, respectively. A summary of the datasets is presented in **Table** 4.1. For each dataset, the mean and standard deviation of the return rate of all assets, and the correlation coefficient of all asset pairs are provided in this benchmark. Additionally, Pareto Front for each dataset is approximated with 2000 points.

4.1.2 Evaluation Metrics

When evaluating MOEAs, it is important to consider both convergence and diversity performance. Although there are several metrics assessing convergence, such as generation distance (GD), and diversity, such as spacing (S), maximum spread (MS), spread (Δ), a recent trend in the literature is to assess the two performance at one time using overall metrics, such as inverted generation distance (IGD) and hypervolume (HV). In this study, all six metrics (i.e., GD, S, MS, Δ, IGD and HV) are applied. The calculation methods of GD, S, MS, Δ and HV are referred to Chapter 8 in Deb ' s book [18]. For IGD, I refer to Coello ' s study [54] which first proposed the IGD metric. A smaller value in GD, S, Δ and IGD indicates a better performance, while for MS and HV, a larger value represents a better performance.

Table 4.1: A summary of the datasets used in experiments

Dataset	Region	Size	Time Period
Hangseng	Hongkong	31	
DAX 100	Germany	85	
FTSE 100	U.K.	89	1992∼1997
S&P 100	U.S.	98	
Nikkei	Japan	225	

- **GD**: $d(\boldsymbol{v}, P^*)$ is the minimum Euclidean distance between a non-dominated solution $\boldsymbol{v}$ and Pareto Front P^*.

$$\frac{\sum_{\boldsymbol{v} \in A} d(\boldsymbol{v}, P^*)}{|A|} \tag{4.1}$$

- **S**: d_i is the minimum ***Manhattan distance*** between i-th non-dominated solution and another one.

$$\sqrt{\frac{1}{N'} \sum_{i=1}^{N'} (\bar{d} - d_i)^2} \tag{4.2}$$

- **MS**: f_m^i is the m-th objective of i-th non-dominated solution.

$$\sqrt{\sum_{m=1}^{M} (\max_{i=1,\ldots,N'} f_m^i - \min_{i=1,\ldots,N'} f_m^i)^2} \tag{4.3}$$

- **Δ**: d_i is the Euclidean distance between two consecutive non-dominated solutions; d_f and d_l are the Euclidean distance between extreme solutions of Pareto Front and boundary solutions of non-dominated set.

$$\frac{d_f + d_l + \sum_{i=1}^{N'-1} |d_i - \bar{d}|}{d_f + d_l + (N' - 1)\bar{d}} \tag{4.4}$$

- **IGD**: $d(\boldsymbol{v}, A)$ is the minimum Euclidean distance between a solution $\boldsymbol{v}$ in Pareto Front P^* and non-dominated set A.

$$\frac{\sum_{\boldsymbol{v} \in P^*} d(\boldsymbol{v}, A)}{|P^*|} \tag{4.5}$$

- **HV**: v_i is the hypercube constructed with a reference point and i-th non-dominated solution as the diagonal corners. To compute the reference point, the Nadir point of solutions generated by all algorithms in the final generation will be used.

$$\text{volume} \left(\bigcup_{i=1}^{N'} v_i \right) \tag{4.6}$$

4.1.3 Experimental Settings

Totally, there are four comparison experiments. In Experiment I, MOEA/D-Lévy is compared with four other literature methods. This comparison aims to assess the performance of the proposed method. In Experiment II, LF mutation is compared with three other mutation methods based on different probability distributions. In this comparison, I show how the long trajectories in LF contribute to the search power. Experiment III provides additional evidence for the explanation in Experiment II, by comparison on LF mutation with different truncation. In Experiment IV, mutation methods based on normal distributions with different variance are compared.

For Experiment I and II, both experiments are implemented with 51 repetitions. For every single run, the values of the six metrics at the final generation are calculated. All five datasets share the same parameter setting. The parameters are set as follows. For all methods, population size and maximum generation are set to 100 and 1500. An early stop criterion (convergence) is set when the variation of IGD is not larger than 1e-05 for 100 continuous generations. Neighbor size T, proportion σ to select parents from the neighbor and upper limitation n_r for updating neighbors in MOEA/D-based methods are set to 20, 0.9 and 2. These settings (i.e., T, σ and n_r) are the same as Li ' s study [32]. The parameter settings of the mutation methods that differentiate the compared algorithms will be presented in **Section** 4.2 and **Section** 4.3. These parameters are fine-tuned by a pre-experiment, using the Nikkei dataset. I perform 30 runs, stopping at the 300-th generation, and choose the parameters that receive the best average IGD to be used in the formal experiments. **Fig.** 4.1 shows an example for β parameter tuning on the proposed algorithm.

For Experiment III and IV, both experiments are implemented with 51 runs. However, only the IGD metric is calculated for every single run. Parameter settings for MOEA/D framework are the same as the former two experiments, but in Experiment III, the maximum generation is set to 100. The detailed reason for this parameter setting is stated in **Section** 4.4. Also, only Nikkei dataset is used in these two experiments.

The reference points used to compute HV in Experiment I and II are reported in Table 4.2. The numerical results of all four experiments are presented in the tables in the corresponding sections. The medians of the best algorithm in each metric are in bold font. In addition, those best medians decorated with underlines indicate that the corresponding algorithms perform better than algorithms with second best medians, through a Wilcoxon Rank Sum Test with a significant level of 5%.

Table 4.2: Reference points (return, risk) used on five datasets

Dataset	Reference Point
Hangseng	(0.0026, 0.0048)
DAX 100	(0.0019, 0.0028)
FTSE 100	(0.0024,0.0028)
S&P 100	(0.0018,0.0031)
Nikkei	(-0.0026,0.0017)

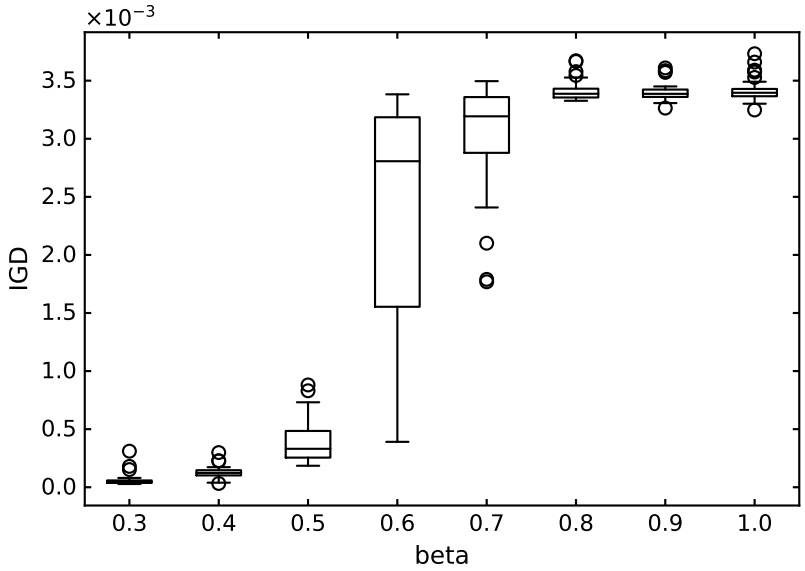


Fig. 4.1: An example of parameter tuning on MOEA/D-Lévy

4.2 Experiment I: Comparison with Literature Methods

4.2.1 Parameter Settings

In this experiment, MOEA/D-Lévy and four comparison methods, namely MOEA/D-DEM, MOEA/D-DE, MOEA/D-GA, and NSGA-II, are included. Among these five methods, four are based on MOEA/D framework described in Li ' s study [32] (i.e., a small proportion to select parents from the whole population and an upper limitation for updating neighbors are set), while the decomposition method applied is NBI-style Tchebycheff approach [12]. The selection method of all methods except the two GA-based algorithms is to select the current visited individual as one of the parents and to randomly select the other individual as the other parents (i.e., same as Li ' s study [32]). For MOEA/D-GA, all the parents are randomly selected from neighbors or the whole population, and a binary tournament selection is applied in NSGA-II. The mutation method in MOEA/D-Lévy has been described in Line 12, Algorithm 3. In MOEA/D-DEM, the mutation is the same as Li ' s study [32] (i.e., DE mutation and polynomial mutation). In MOEA/D-DE, the mutation method is only based on DE mutation. This setting is applied in Zhang ' s study [12]. For two GA-based algorithms, SBX crossover and polynomial mutation are applied. This setting is the same as the original NSGA-II [26] and MOEA/D [28] algorithms. In Table 4.8, detailed mutation methods in Experiment I are listed.

Following the pre-experimental tuning described in **Section** 4.1.3, I obtained the following parameter values. In MOEA/D-Lévy, α_0 and β of LF mutation are set to 1e-05 and 0.3. In MOEA/D-DEM and MOEA/D-DE, scaling factor F of DE mutation is set to 1.3. In MOEA/D-Lévy and MOEA/D-DEM, the mutation rate is set to $1/n$ (i.e., n is the size of

dataset). In MOEA/D-GA and NSGA-II, the crossover rate is set to 0.7. In MOEA/D-GA, the mutation rate is 0.05 and in NSGA-II, the mutation rate is 0.01.

4.2.2 Experimental Results

As an example, **Fig.** 4.2 shows IGD changing by generations on Nikkei in Experiment I (best run of IGD). **Fig.** 4.3 illustrates the final population on the same dataset and experiment, as well as a zoom-in. The final population of other datasets and experiments are shown in **Fig.** 4.4. The results for each dataset are detailed as follows.

- **Hangseng**: MOEA/D-Lévy holds the best median in terms of MS, Δ and IGD, while MOEA/D-GA performs best on GD, S, and HV. Both methods show a statistical significance compared with the methods holding the second-best median.
- **DAX 100**: MOEA/D-Lévy shows a statistically significant superiority in terms of MS, Δ, IGD and HV, while MOEA/D-GA performs better in GD and S with a statistical significance.
- **FTSE 100**: MOEA/D-Lévy shows a clear superiority on MS, Δ, IGD and HV, while MOEA/D-GA and MOEA/D-DE perform best on GD and S, respectively.
- **S&P 100**: MOEA/D-Lévy holds the best median on Δ, IGD and HV. Especially, on Δ and HV, there is a significant difference between the second-best method. MOEA/D-GA, MOEA/D-DE, and MOEA/D-DEM are the best method in terms of GD, S, and MS, respectively.
- **Nikkei**: MOEA/D-Lévy performs best on Δ, IGD and HV. Especially, on Δ and HV, there is a significant difference between the second-best method. NSGA-II, MOEA/D-DE, and MOEA/D-DEM are the best method in terms of GD, S, and MS, respectively.

In addition, it is hard to find a difference when considering the plots of the solution distributions. On FTSE 100, MOEA/D-DE performs relatively worse on retrieving low-risk portfolios (i.e., it has fewer solutions in the bottom left regions in **Fig.** 4.4c). On S&P 100 and Nikkei, the two GA-based algorithms only retrieve a part of the front (i.e., show in **Fig.** 4.4d and 4.4e). On Nikkei, the solution set retrieved by MOEA/D-DE is relatively narrow and far from the Pareto Front (i.e., show in **Fig.** 4.4e).

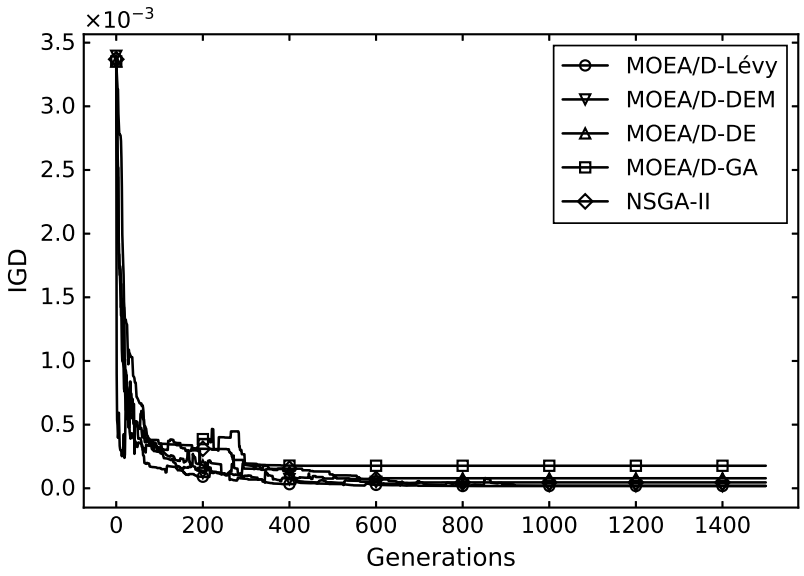


Fig. 4.2: IGD by generations on Nikkei in Experiment I

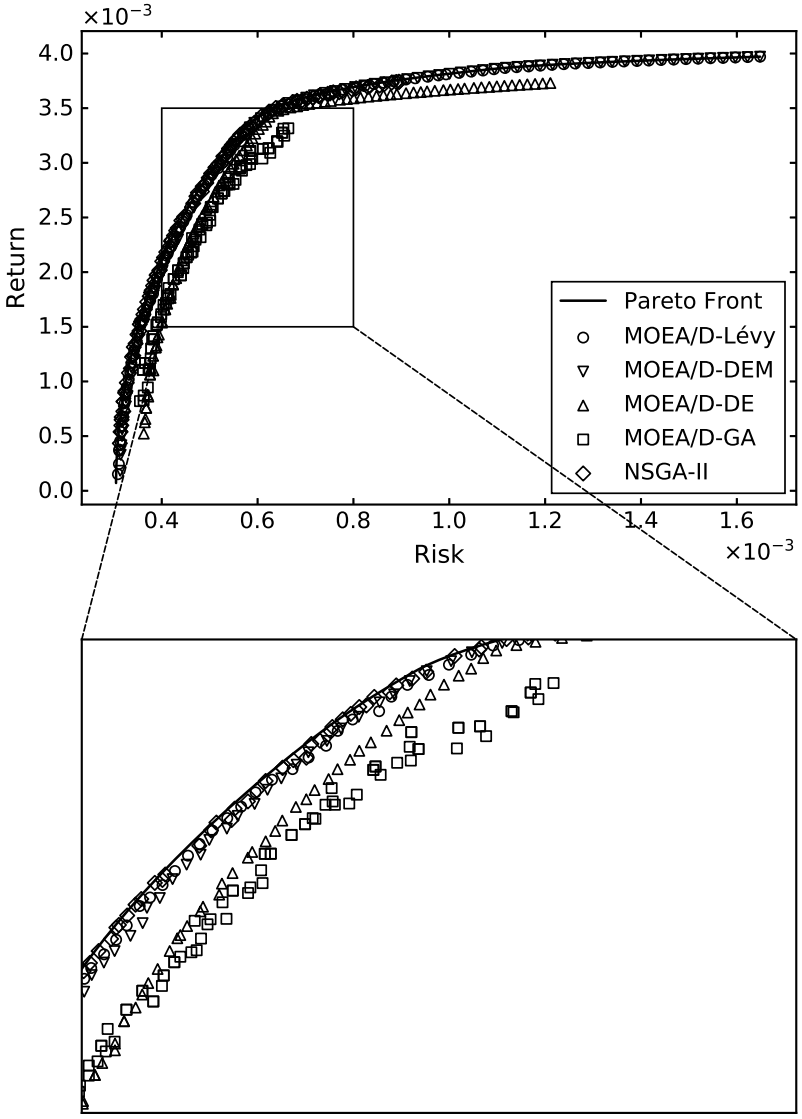


Fig. 4.3: Final population and zoom-in on Nikkei in Experiment I

Table 4.3: Numerical results on Hangseng in Experiment I

Metric		MOEA/D				NSGA-II
		Lévy	DEM	DE	GA	
GD	Best	4.51E-06	4.21E-06	3.49E-06	1.74E-06	9.06E-06
	Median	5.78E-06	7.26E-06	7.98E-06	**2.29E-06**	1.19E-05
	Std.	8.78E-07	2.22E-06	1.18E-04	2.60E-07	1.34E-06
S	Best	1.56E-05	1.50E-05	8.51E-06	9.38E-06	3.94E-05
	Median	2.06E-05	2.34E-05	1.80E-05	**1.53E-05**	4.92E-05
	Std.	7.18E-06	9.07E-06	6.76E-06	5.71E-06	4.19E-06
MS	Best	9.13E-03	9.23E-03	9.00E-03	8.92E-03	9.06E-03
	Median	**8.96E-03**	8.89E-03	8.54E-03	8.25E-03	8.63E-03
	Std.	1.01E-04	1.98E-04	9.93E-04	3.16E-04	2.18E-04
Δ	Best	2.47E-01	2.53E-01	2.33E-01	2.61E-01	4.48E-01
	Median	**2.64E-01**	2.87E-01	2.87E-01	2.80E-01	4.94E-01
	Std.	3.11E-02	3.99E-02	8.20E-02	1.40E-02	3.50E-02
IGD	Best	2.90E-05	2.99E-05	3.15E-05	2.98E-05	3.92E-05
	Median	**3.13E-05**	3.50E-05	6.03E-05	7.54E-05	5.01E-05
	Std.	2.75E-06	8.54E-06	2.44E-04	3.97E-05	1.55E-05
HV	Best	2.64E-05	2.64E-05	2.64E-05	2.64E-05	2.63E-05
	Median	2.64E-05	2.63E-05	2.63E-05	**2.64E-05**	2.63E-05
	Std.	9.76E-09	2.64E-08	2.21E-06	1.38E-08	1.31E-08

Table 4.4: Numerical results on DAX 100 in Experiment I

Metric		MOEA/D				NSGA-II
		Lévy	DEM	DE	GA	
GD	Best	5.85E-06	7.11E-06	7.32E-06	1.83E-06	5.37E-06
	Median	7.98E-06	9.53E-06	1.65E-05	**2.78E-06**	8.04E-06
	Std.	1.11E-06	1.81E-06	9.27E-05	6.36E-07	1.99E-06
S	Best	2.76E-05	2.34E-05	1.64E-05	1.59E-05	2.27E-05
	Median	3.25E-05	3.24E-05	2.98E-05	**2.48E-05**	4.34E-05
	Std.	7.29E-06	7.37E-06	6.61E-06	5.68E-06	6.94E-06
MS	Best	8.13E-03	8.12E-03	8.36E-03	7.37E-03	7.83E-03
	Median	**7.77E-03**	7.71E-03	7.40E-03	6.04E-03	7.20E-03
	Std.	1.59E-04	2.31E-04	7.26E-04	3.70E-04	5.20E-04
Δ	Best	3.88E-01	4.07E-01	3.60E-01	4.54E-01	5.70E-01
	Median	**4.07E-01**	4.49E-01	4.36E-01	5.81E-01	6.68E-01
	Std.	2.72E-02	3.79E-02	7.72E-02	3.65E-02	5.11E-02
IGD	Best	3.28E-05	3.62E-05	4.40E-05	7.20E-05	4.14E-05
	Median	**4.16E-05**	4.90E-05	9.48E-05	1.54E-04	6.55E-05
	Std.	7.87E-06	1.60E-05	9.39E-05	3.75E-05	3.34E-05
HV	Best	1.91E-05	1.90E-05	1.90E-05	1.91E-05	1.90E-05
	Median	**1.90E-05**	1.90E-05	1.89E-05	1.84E-05	1.90E-05
	Std.	1.16E-08	2.34E-08	1.08E-06	4.22E-07	4.44E-07

Table 4.5: Numerical results on FTSE 100 in Experiment I

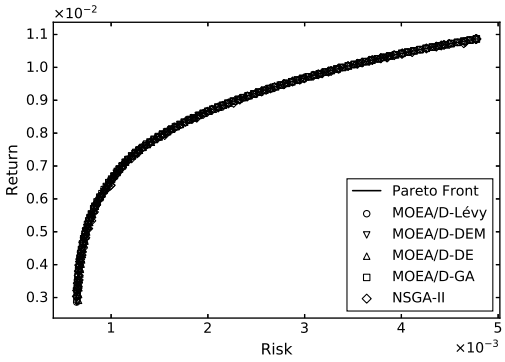
Metric		MOEA/D				NSGA-II
		Lévy	DEM	DE	GA	
GD	Best	5.39E-06	6.32E-06	7.01E-06	2.84E-06	7.38E-06
	Median	7.12E-06	9.58E-06	1.83E-05	**5.05E-06**	9.25E-06
	Std.	7.30E-07	2.41E-06	1.55E-04	1.60E-06	9.51E-07
S	Best	1.59E-05	1.38E-05	9.87E-06	1.14E-05	2.39E-05
	Median	2.09E-05	2.01E-05	**1.72E-05**	1.97E-05	2.99E-05
	Std.	3.96E-06	4.54E-06	3.34E-06	5.10E-06	2.24E-06
MS	Best	5.85E-03	5.79E-03	5.74E-03	5.47E-03	5.67E-03
	Median	**5.54E-03**	5.40E-03	5.15E-03	4.91E-03	5.45E-03
	Std.	1.46E-04	1.93E-04	5.08E-04	4.19E-04	1.69E-04
Δ	Best	4.06E-01	4.25E-01	4.21E-01	4.27E-01	5.47E-01
	Median	**4.33E-01**	4.72E-01	4.51E-01	5.05E-01	6.06E-01
	Std.	3.38E-02	3.06E-02	6.30E-02	6.88E-02	3.32E-02
IGD	Best	2.36E-05	2.90E-05	4.26E-05	4.07E-05	3.22E-05
	Median	**3.83E-05**	5.31E-05	9.09E-05	8.76E-05	4.74E-05
	Std.	1.08E-05	2.16E-05	1.47E-04	4.33E-05	1.38E-05
HV	Best	1.37E-05	1.37E-05	1.37E-05	1.37E-05	1.37E-05
	Median	**1.37E-05**	1.37E-05	1.36E-05	1.34E-05	1.37E-05
	Std.	5.72E-09	1.64E-08	1.36E-06	6.19E-07	8.35E-08

Table 4.6: Numerical results on S&P 100 in Experiment I

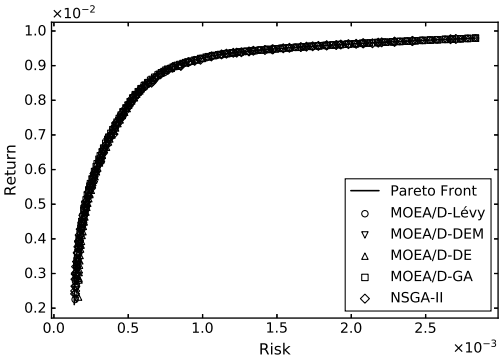
Metric		MOEA/D				NSGA-II
		Lévy	DEM	DE	GA	
GD	Best	1.05E-05	1.13E-05	1.27E-05	3.20E-06	1.01E-05
	Median	1.25E-05	1.75E-05	3.92E-05	**5.02E-06**	1.29E-05
	Std.	1.64E-06	3.45E-06	7.64E-05	1.86E-06	1.42E-06
S	Best	2.08E-05	2.01E-05	1.72E-05	1.77E-05	2.51E-05
	Median	2.51E-05	2.76E-05	**2.31E-05**	2.35E-05	3.58E-05
	Std.	5.10E-06	5.26E-06	4.78E-06	5.06E-06	3.68E-06
MS	Best	7.72E-03	7.86E-03	7.60E-03	6.87E-03	7.29E-03
	Median	7.44E-03	**7.55E-03**	7.23E-03	5.82E-03	6.57E-03
	Std.	1.47E-04	1.61E-04	2.53E-04	4.35E-04	3.52E-04
Δ	Best	3.29E-01	3.30E-01	3.29E-01	4.22E-01	5.37E-01
	Median	**3.45E-01**	3.74E-01	3.66E-01	5.57E-01	6.48E-01
	Std.	2.95E-02	3.37E-02	2.41E-02	3.88E-02	3.71E-02
IGD	Best	3.30E-05	3.30E-05	3.64E-05	4.88E-05	4.08E-05
	Median	**3.97E-05**	4.35E-05	7.12E-05	1.36E-04	7.08E-05
	Std.	9.23E-06	7.65E-06	4.16E-05	4.13E-05	2.21E-05
HV	Best	1.87E-05	1.87E-05	1.87E-05	1.87E-05	1.87E-05
	Median	**1.87E-05**	1.86E-05	1.85E-05	1.79E-05	1.83E-05
	Std.	1.51E-08	3.19E-08	4.79E-07	5.03E-07	3.13E-07

Table 4.7: Numerical results on Nikkei in Experiment I

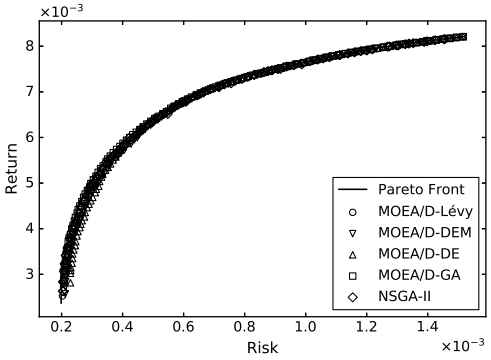
Metric		MOEA/D				NSGA-II
		Lévy	DEM	DE	GA	
GD	Best	5.45E-06	5.10E-06	5.93E-05	9.95E-06	2.54E-06
	Median	7.26E-06	8.06E-06	1.45E-04	3.31E-05	**4.24E-06**
	Std.	1.16E-06	1.66E-04	7.09E-05	2.08E-04	2.01E-06
S	Best	1.28E-05	1.28E-05	5.99E-06	0.00E+00	1.05E-05
	Median	1.76E-05	2.08E-05	**1.15E-05**	1.92E-05	1.45E-05
	Std.	2.88E-06	5.77E-06	4.53E-06	9.72E-06	1.84E-06
MS	Best	4.09E-03	4.23E-03	4.29E-03	2.63E-03	3.36E-03
	Median	3.93E-03	**3.94E-03**	2.96E-03	2.20E-03	2.88E-03
	Std.	1.38E-04	5.39E-04	5.48E-04	4.65E-04	2.54E-04
Δ	Best	3.94E-01	3.99E-01	3.17E-01	8.40E-01	6.09E-01
	Median	**4.34E-01**	4.81E-01	5.58E-01	9.34E-01	6.81E-01
	Std.	3.89E-02	9.05E-02	1.00E-01	3.54E-02	2.95E-02
IGD	Best	1.77E-05	1.90E-05	7.90E-05	1.77E-04	4.64E-05
	Median	**2.39E-05**	2.73E-05	2.23E-04	2.41E-04	9.69E-05
	Std.	1.15E-05	4.09E-04	6.95E-05	5.29E-04	3.67E-05
HV	Best	8.31E-06	8.29E-06	7.96E-06	7.87E-06	8.19E-06
	Median	**8.29E-06**	8.26E-06	7.23E-06	7.54E-06	7.94E-06
	Std.	1.59E-08	9.52E-07	3.43E-07	1.20E-06	1.08E-07



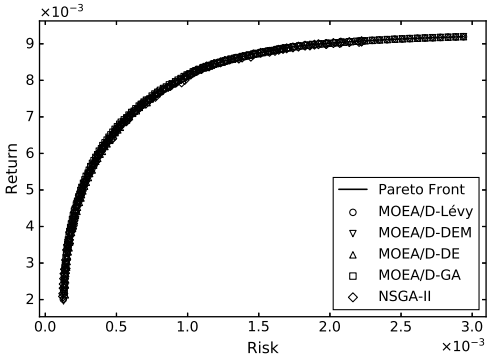
(a) Hangseng



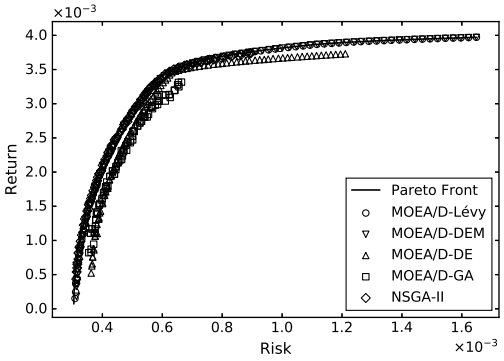
(b) DAX 100



(c) FTSE 100



(d) S&P 100



(e) Nikkei

Fig. 4.4: Final population on five datasets in objective space (Experiment I)

4.3 Experiment II: Comparison with Other Distribution-based Mutation Methods

4.3.1 Parameter Settings

In this experiment, mutation methods based on four probability distributions, namely Lévy-stable distribution (LEVY), uniform distribution (UNIF), standard normal distribution (NORM) and constant (CONST) are compared to show the effectiveness of LF. Among these four methods, MOEA/D framework and selection method in Li ' s study [32] are applied. The mutation operators of these methods are similar to DE mutation, but the scaling factors are drawn from the probability distributions mentioned above. In addition, two parents are selected in LEVY, while three are selected in the other three mutation methods. As the goal of this experiment is to show how LF contributes to optimization, no polynomial mutation is applied in all methods. Table 4.9 presents the detailed formula of the mutation methods in Experiment II.

Following the pre-experimental tuning method described in **Section** 4.1, I obtained the following parameter values. Parameter C is set to 1.0 and 0.5 and in UNIF and NORM, respectively. The parameters of LF mutation in LEVY are the same as that of MOEA/D-Lévy in Section 4.2. The parameters of CONST is the same as that of MOEA/D-DE in **Section** 4.2. It is interesting to notice that CONST and MOEA/D-DE are the same method.

4.3.2 Experimental Results

- **Hangseng**: NORM performs best on GD, MS, IGD, and HV. CONST holds the best S, and UNIF holds the best Δ.
- **DAX 100**: NORM is the best method in terms of five metrics except for S, especially on GD, MS and IGD, the statistical test shows a significant difference. For the S metric, CONST performs best.
- **FTSE 100**: LEVY performs best on MS, Δ, IGD and HV. On MS and IGD, it shows a statistical significance in the comparison with NORM which holds second place. NORM performs best on GD, while CONST performs best on S.
- **S&P 100**: UNIF holds the best GD and S, while NORM performs best on the rest metrics. LEVY holds the second place on MS, IGD, and HV, and shows a comparable performance.
- **Nikkei**: LEVY shows a significant superiority in terms of GD, MS, Δ, IGD and HV. For S, CONST performs best.

In addition, it is also hard to find a difference when considering the plots of the solution distributions. On FTSE 100, CONST performs relatively worse on retrieving low-risk portfolios (i.e., it has fewer solutions in the bottom left regions in 4.5c). On Nikkei, the solution set retrieved by CONST is relatively narrow and far from the Pareto Front (i.e., show in and 4.5e).

Table 4.8: A summary of mutation methods in Experiment I

Method	Mutation Formula
MOEA/D-Lévy	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + \alpha_0 \cdot (\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}) \oplus \boldsymbol{Lévy(\beta)}$ polynomial mutation
MOEA/D-DEM	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + F \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)})$ polynomial mutation
MOEA/D-DE	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + F \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)})$
MOEA/D-GA NSGA-II	SBX crossover and polynomial mutation

Table 4.9: A summary of mutation methods in Experiment II

Method	Mutation Formula
LEVY	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + \alpha_0 \cdot (\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}) \oplus \boldsymbol{Lévy(\beta)}$
UNIF	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + C \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)}) \oplus \boldsymbol{Unif(-1, 1)}$
NORM	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + C \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)}) \oplus \boldsymbol{N(0, 1)}$
CONST	$\boldsymbol{y} = \boldsymbol{x}^{(i)} + F \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)})$

Table 4.10: Numerical results on Hangseng in Experiment II

Metric		LEVY	UNIF	NORM	CONST
GD	Best	4.95E-06	4.57E-06	4.24E-06	3.49E-06
	Median	8.10E-06	7.07E-06	**7.02E-06**	7.98E-06
	Std.	1.44E-06	1.62E-06	1.44E-06	1.18E-04
S	Best	1.43E-05	1.43E-05	1.37E-05	8.51E-06
	Median	1.98E-05	1.83E-05	1.85E-05	**1.80E-05**
	Std.	4.95E-06	4.66E-06	4.45E-06	6.76E-06
MS	Best	9.02E-03	9.09E-03	9.10E-03	9.00E-03
	Median	8.85E-03	8.82E-03	**8.86E-03**	8.54E-03
	Std.	1.39E-04	2.04E-04	1.87E-04	9.93E-04
Δ	Best	2.48E-01	2.47E-01	2.47E-01	2.33E-01
	Median	2.64E-01	**2.63E-01**	2.65E-01	2.87E-01
	Std.	2.26E-02	1.94E-02	1.60E-02	8.20E-02
IGD	Best	2.97E-05	2.88E-05	2.89E-05	3.15E-05
	Median	3.40E-05	3.40E-05	**3.27E-05**	6.03E-05
	Std.	5.92E-06	9.76E-06	8.27E-06	2.44E-04
HV	Best	2.64E-05	2.64E-05	2.64E-05	2.64E-05
	Median	2.63E-05	2.64E-05	**2.64E-05**	2.63E-05
	Std.	1.42E-08	1.69E-08	1.43E-08	2.21E-06

Table 4.11: Numerical results on DAX 100 in Experiment II

Metric		LEVY	UNIF	NORM	CONST
GD	Best	7.23E-06	4.76E-06	4.88E-06	7.32E-06
	Median	9.37E-06	7.43E-06	**6.85E-06**	1.65E-05
	Std.	9.63E-07	1.39E-06	1.47E-06	9.27E-05
S	Best	2.58E-05	2.61E-05	2.70E-05	1.64E-05
	Median	3.28E-05	3.09E-05	3.19E-05	**2.98E-05**
	Std.	3.96E-06	3.97E-06	5.22E-06	6.61E-06
MS	Best	8.00E-03	8.02E-03	8.13E-03	8.36E-03
	Median	7.75E-03	7.76E-03	**7.82E-03**	7.40E-03
	Std.	1.59E-04	1.48E-04	1.36E-04	7.26E-04
Δ	Best	3.85E-01	3.94E-01	3.89E-01	3.60E-01
	Median	4.12E-01	4.08E-01	**4.07E-01**	4.36E-01
	Std.	2.32E-02	2.47E-02	2.55E-02	7.72E-02
IGD	Best	3.40E-05	3.34E-05	3.20E-05	4.40E-05
	Median	4.39E-05	4.07E-05	**3.83E-05**	9.48E-05
	Std.	1.14E-05	7.89E-06	5.77E-06	9.39E-05
HV	Best	1.90E-05	1.91E-05	1.91E-05	1.90E-05
	Median	1.90E-05	1.90E-05	**1.90E-05**	1.89E-05
	Std.	1.19E-08	1.28E-08	1.50E-08	1.08E-06

Table 4.12: Numerical results on FTSE 100 in Experiment II

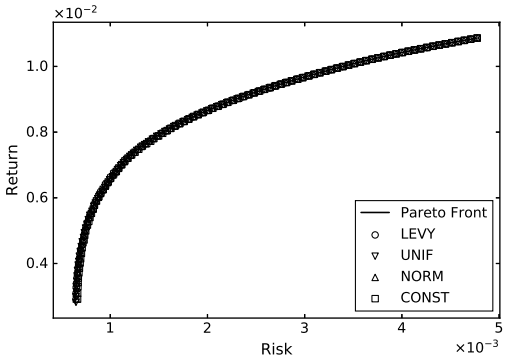
Metric		LEVY	UNIF	NORM	CONST
GD	Best	6.56E-06	5.58E-06	4.83E-06	7.01E-06
	Median	8.48E-06	8.26E-06	**7.26E-06**	1.83E-05
	Std.	1.12E-06	2.24E-06	5.79E-05	1.55E-04
S	Best	1.73E-05	1.51E-05	1.62E-05	9.87E-06
	Median	2.06E-05	1.75E-05	1.87E-05	**1.72E-05**
	Std.	3.94E-06	1.56E-06	3.02E-06	3.34E-06
MS	Best	5.82E-03	5.48E-03	5.69E-03	5.74E-03
	Median	**5.47E-03**	5.19E-03	5.35E-03	5.15E-03
	Std.	1.54E-04	1.29E-04	2.07E-04	5.08E-04
Δ	Best	4.09E-01	4.19E-01	4.13E-01	4.21E-01
	Median	**4.35E-01**	4.45E-01	4.36E-01	4.51E-01
	Std.	2.03E-02	1.13E-02	2.60E-02	6.30E-02
IGD	Best	2.62E-05	3.95E-05	3.02E-05	4.26E-05
	Median	**4.29E-05**	7.26E-05	5.51E-05	9.09E-05
	Std.	1.27E-05	1.66E-05	4.61E-05	1.47E-04
HV	Best	1.37E-05	1.37E-05	1.37E-05	1.37E-05
	Median	**1.37E-05**	1.37E-05	1.37E-05	1.36E-05
	Std.	8.46E-09	2.48E-08	4.42E-07	1.36E-06

Table 4.13: Numerical results on S&P 100 in Experiment II

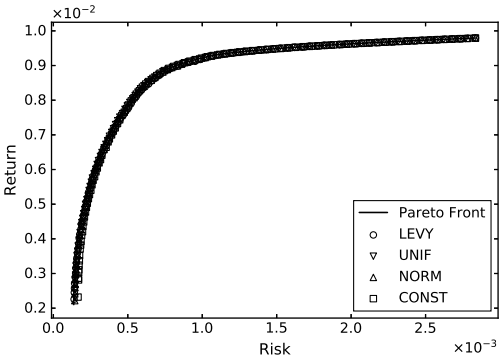
Metric		LEVY	UNIF	NORM	CONST
GD	Best	1.09E-05	7.66E-06	7.01E-06	1.27E-05
	Median	1.41E-05	**1.01E-05**	1.04E-05	3.92E-05
	Std.	1.83E-06	2.67E-05	2.99E-05	7.64E-05
S	Best	2.20E-05	2.07E-05	2.07E-05	1.72E-05
	Median	2.58E-05	**2.29E-05**	2.42E-05	2.31E-05
	Std.	4.15E-06	2.61E-06	4.55E-06	4.78E-06
MS	Best	7.75E-03	7.47E-03	7.64E-03	7.60E-03
	Median	7.32E-03	7.20E-03	**7.35E-03**	7.23E-03
	Std.	1.60E-04	1.33E-04	1.65E-04	2.53E-04
Delta	Best	3.28E-01	3.32E-01	3.30E-01	3.29E-01
	Median	3.55E-01	3.49E-01	**3.49E-01**	3.66E-01
	Std.	2.04E-02	1.86E-02	3.04E-02	2.41E-02
IGD	Best	3.35E-05	3.62E-05	3.12E-05	3.64E-05
	Median	4.72E-05	5.23E-05	**4.28E-05**	7.12E-05
	Std.	1.11E-05	1.42E-05	1.31E-05	4.16E-05
HV	Best	1.87E-05	1.87E-05	1.88E-05	1.87E-05
	Median	1.87E-05	1.87E-05	**1.87E-05**	1.85E-05
	Std.	1.73E-08	1.65E-07	1.78E-07	4.79E-07

Table 4.14: Numerical results on Nikkei in Experiment II

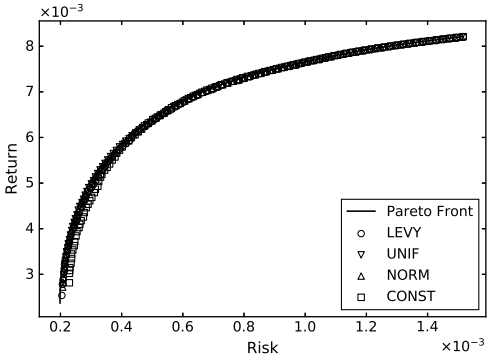
Metric		LEVY	UNIF	NORM	CONST
GD	Best	5.92E-06	5.45E-06	5.38E-06	5.93E-05
	Median	**7.71E-06**	4.25E-05	4.10E-05	1.45E-04
	Std.	1.42E-06	1.93E-04	2.65E-05	7.09E-05
S	Best	9.54E-06	1.24E-05	8.75E-06	5.99E-06
	Median	1.66E-05	1.54E-05	1.50E-05	**1.15E-05**
	Std.	5.35E-06	2.62E-06	3.05E-06	4.53E-06
MS	Best	4.14E-03	4.08E-03	4.07E-03	4.29E-03
	Median	**3.90E-03**	3.68E-03	3.70E-03	2.96E-03
	Std.	1.57E-04	5.59E-04	3.12E-04	5.48E-04
Δ	Best	3.89E-01	3.84E-01	3.76E-01	3.17E-01
	Median	**4.33E-01**	4.44E-01	4.55E-01	5.58E-01
	Std.	5.61E-02	9.82E-02	6.73E-02	1.00E-01
IGD	Best	1.84E-05	1.83E-05	1.95E-05	7.90E-05
	Median	**2.72E-05**	4.16E-05	4.85E-05	2.23E-04
	Std.	1.68E-05	4.51E-04	2.98E-05	6.95E-05
HV	Best	8.31E-06	8.29E-06	8.28E-06	7.96E-06
	Median	**8.29E-06**	8.11E-06	8.10E-06	7.23E-06
	Std.	2.80E-08	9.83E-07	1.34E-07	3.43E-07



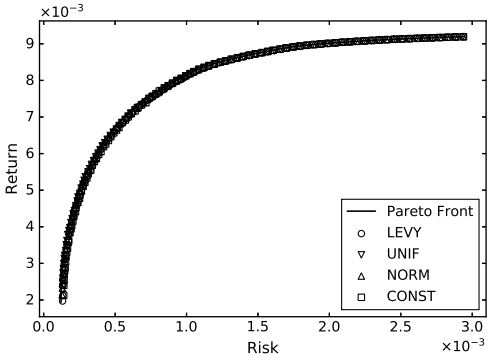
(a) Hangseng



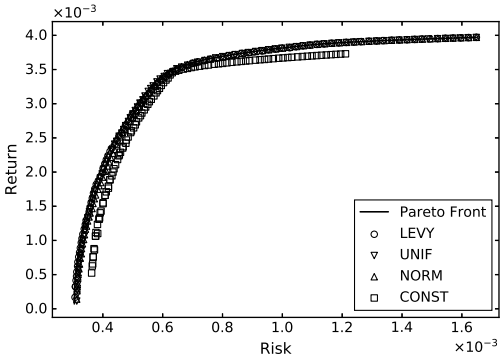
(b) DAX 100



(c) FTSE 100



(d) S&P 100



(e) Nikkei

Fig. 4.5: Final population on five datasets in objective space (Experiment II)

4.3.3 Discussion

The numerical results show a good performance of the proposed MOEA/D-Lévy and LEVY. It is easy to realize that those long trajectories caused by LF can enhance the global search capability of algorithms by comparing the results of LEVY and NORM in Experiment II, as the main difference between a heavy-tailed distribution and standard normal distribution is the occasional generation of large numbers. In this discussion, I show further insights on how long trajectories contribute to this observed improvement.

To record long trajectories, I compute the Euclidean distance between a repaired offspring $\boldsymbol{y}$ and its parent $\boldsymbol{x^i}$ in the mutation methods based on LEVY, UNIF, NORM and CONST, using Nikkei dataset. This distance is the length of the trial vector. In addition, I record the frequency that this offspring is successfully updated. Based on the parameter settings in **Section** 4.1.3, the possible frequency for one offspring is 0, 1 or 2. **Fig.** 4.7a, 4.8a and 4.9a show the frequency of trajectories in different length, at 1st, 3rd and 10th generation in one certain run, respectively. **Fig.** 4.7b, 4.8b and 4.9b illustrate the corresponding populations at each generation. **Fig.** 1.1 shows a population on objective space at 10th generation in Experiment I. **Fig.** 4.6 shows the frequency of successfully updated long trajectories (i.e., length is larger than 0.2) by generations in MOEA/D-Lévy in the same run.

It is interesting to notice that there are some long trial vectors successfully updated in the beginning phase of LEVY, while the same observation does not occur in UNIF, NORM, and CONST. The long trajectory usually represents a global search. As a result, LEVY achieves early in the optimization a solution set that is widely spread across the objective space (i.e., show in **Fig.** 4.7b and 4.8b). In **Fig.** 1.1 and 4.9b, methods based on LF form a relatively better front than other methods. **Fig.** 4.10a to 4.10f illustrates the population of five algorithms in Experiment I at different generations in a certain run. In the beginning phase of the optimization, MOEA/D-Lévy forms multiple "sub-sets" while the other methods hold only one front. This may indicate that methods based on LF can search multiple areas at the same time, while other mutations can only deal with one.

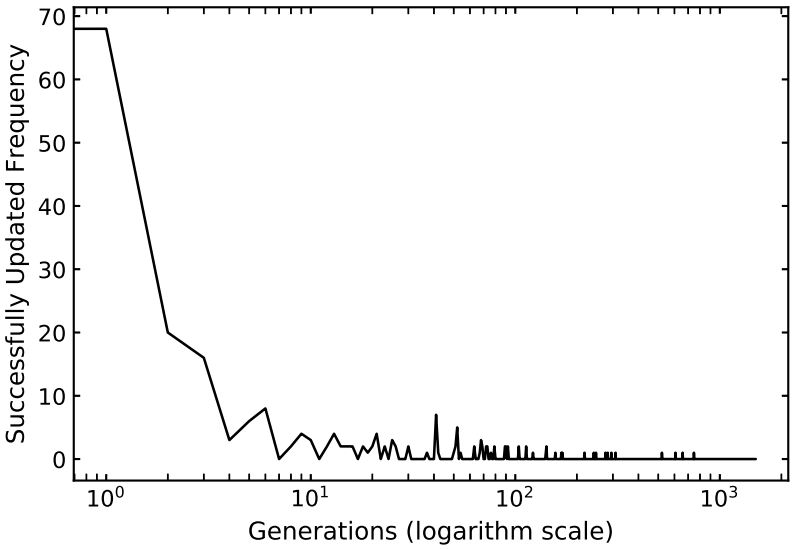
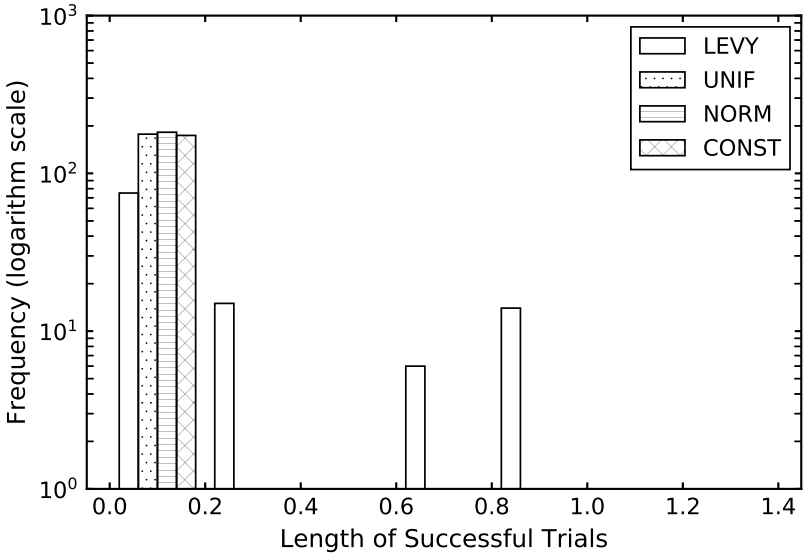
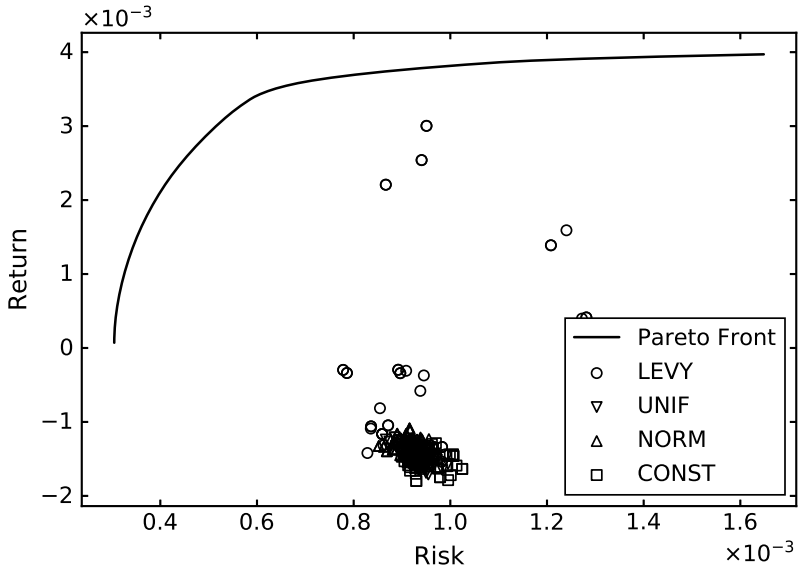


Fig. 4.6: Successfully updated long trials by generations

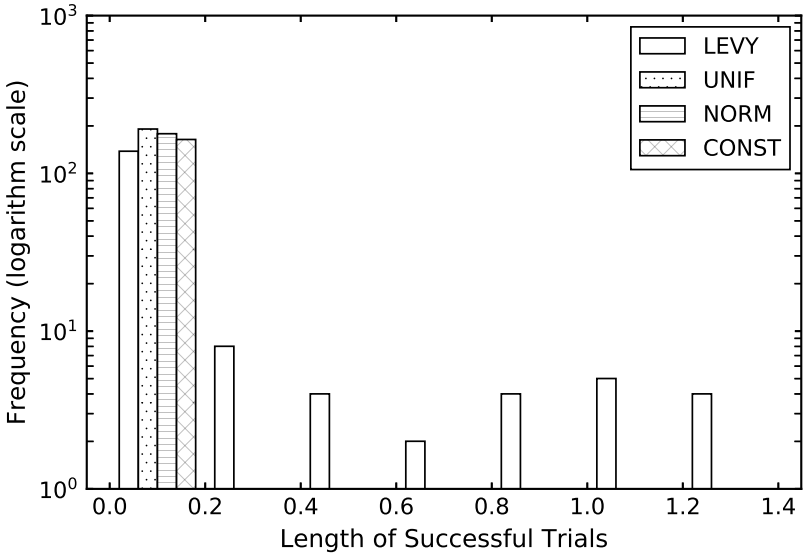


(a) Successful trials

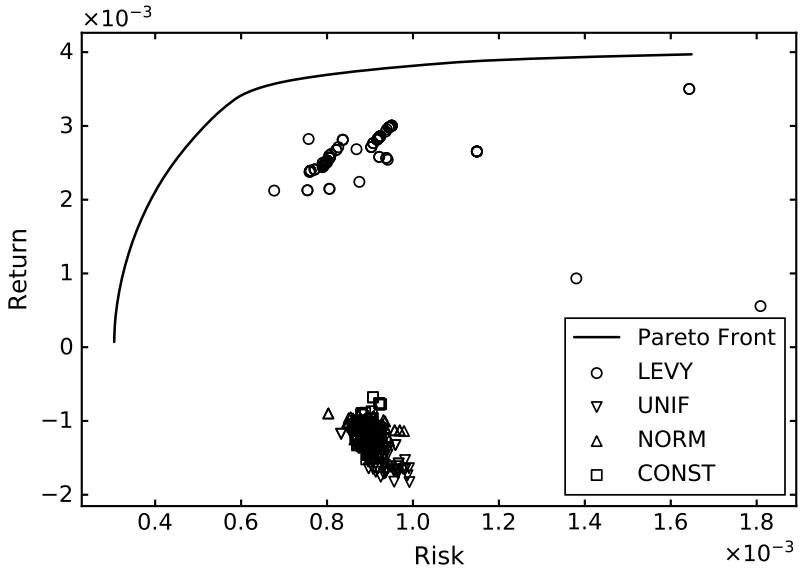


(b) Population in objective space

Fig. 4.7: Experiment II, Nikkei Dataset (1st generation). Upper: Frequency of "Succesful trials" (when the mutation operator generates an offspring that is better than its parent) against the length of the mutation step. Lower: population in the objective space.

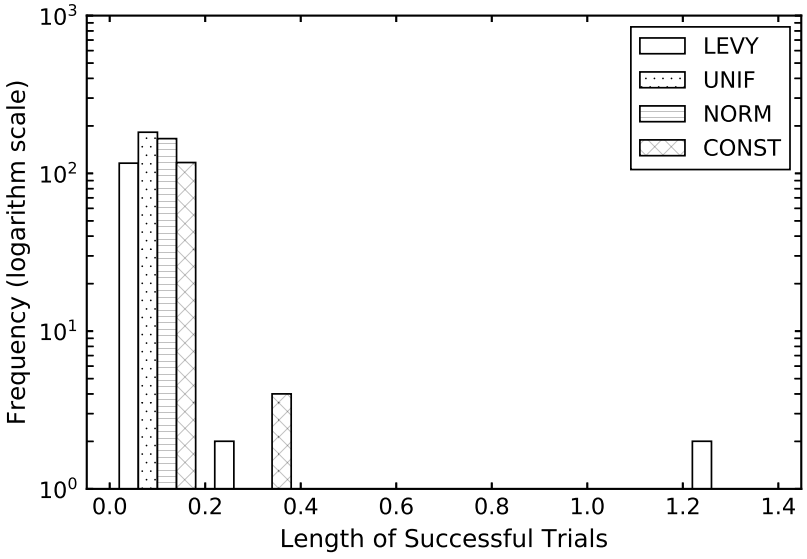


(a) Successful trials

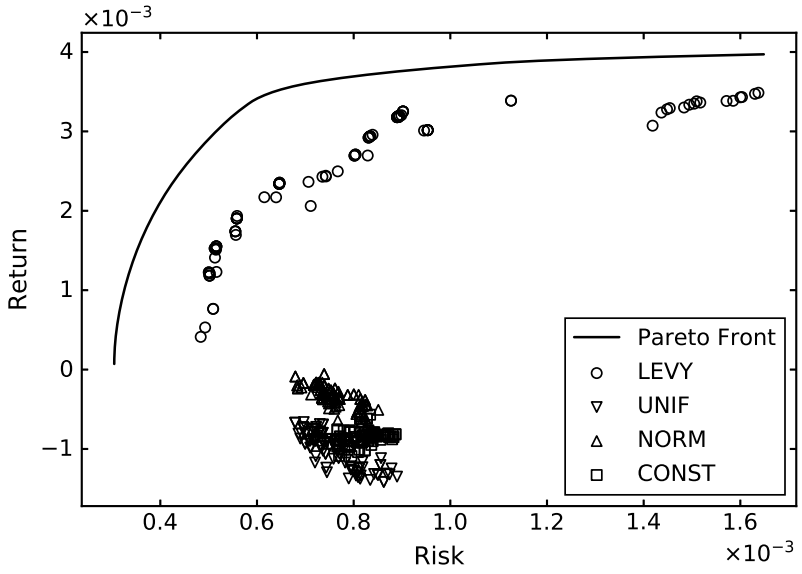


(b) Population in objective space

Fig. 4.8: Experiment II, Nikkei Dataset (3rd generation). Upper: Frequency of "Succesful trials" (when the mutation operator generates an offspring that is better than its parent) against the length of the mutation step. Lower: population in the objective space.

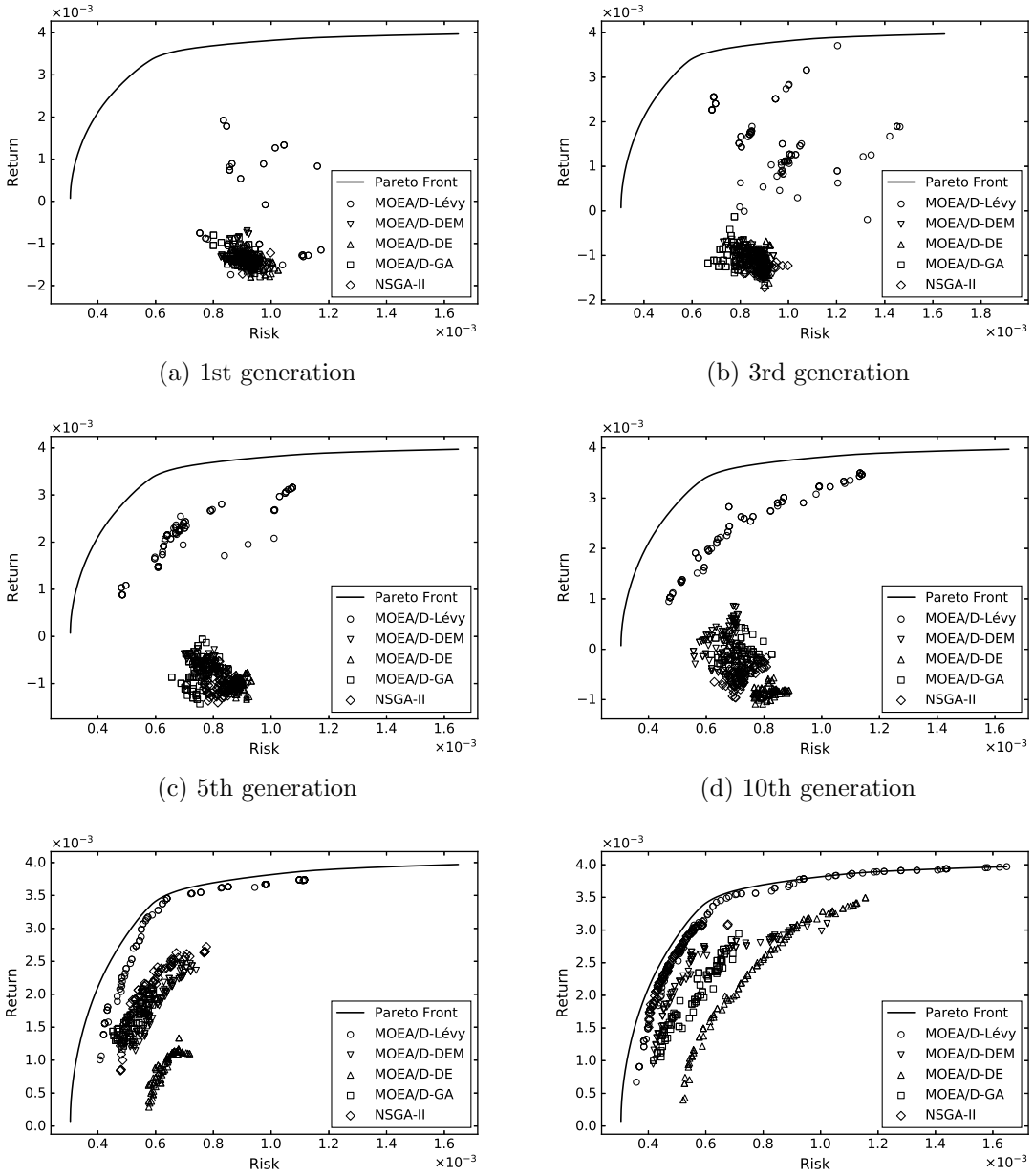


(a) Successful trials



(b) Population in objective space

Fig. 4.9: Experiment II, Nikkei Dataset (10th generation). Upper: Frequency of "Succesful trials" (when the mutation operator generates an offspring that is better than its parent) against the length of the mutation step. Lower: population in the objective space.



(a) 1st generation

(b) 3rd generation

(c) 5th generation

(d) 10th generation

(e) 50th generation

(f) 100th generation

Fig. 4.10: Experiment I: Populations of different algorithms over generations in the Nikkei dataset. MOEA/D-Lévy covers a wider area early in the optimization, which leads to a better distribution over the Pareto Front around the 100th generation.

4.3.4 An Explanation of Long Trajectories Considering the Case of Portfolio Optimization

To explain this behavior, consider that, in PO, trial vectors or trajectories represent the re-allocation of capital. The heavy-tailed distribution holds a large probability to generate small values, and a small probability to generate large values. **Fig.** 4.11 presents an example of this procedure. The variable (asset) that receives a large number during mutation will get more capital allocation. What is more, to satisfy the constraint that the summation of variables equals to 1 in (2.6), the repair steps will implement a proper scaling, and thus the other variables (assets) will be reduced. Therefore, the re-allocation of capital will be centralized on some assets rather than equally distributed across all assets. If these assets hold relatively high returns, the algorithm can find portfolio candidates with high returns early in the optimization. As the initial population is centralized at the low-risk area in objective space because of the uniform initialization, this mutated candidate has a large probability to be successfully updated. Following this procedure, the solution set will be distributed in multiple areas on the search space. Then, the algorithm will mainly update solutions by local search but in multiple search areas, as the accepted long trials decrease by generation in **Fig.** 4.6. Such different search patterns may lead to the observed improvement compared with other methods.

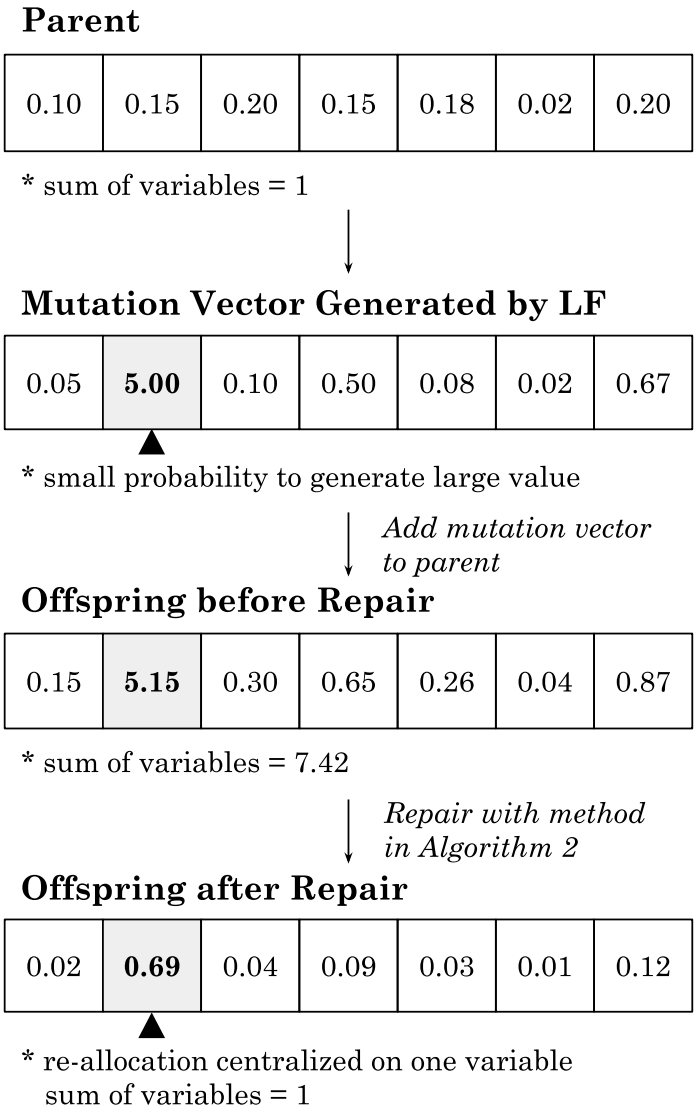


Fig. 4.11: How LF mutation helps exploration in PO. An interaction between long trajectories and unit constraint leads to candidates with high allocation in only some assets.

4.4 Experiment III: Comparison on Lévy Flight Mutation with Different Truncation

4.4.1 Parameter Settings

Another way to show the effectiveness of long trajectories is to implement a truncation directly on LEVY so that it will not generate large steps. Mathematically, this process can be expressed as follows, where $\epsilon > 0$.

$$L\acute{e}vy_{\epsilon}(\beta) = \max\{-\epsilon, \min\{\epsilon, L\acute{e}vy(\beta)\}\} \tag{4.7}$$

As long trajectories are accepted mainly in the beginning phase of optimization, this section discusses based on the results at 100th generation for LF with truncation with different ϵ settings, using the Nikkei dataset. To simplify the discussion, only one of the overall metrics, IGD, is used in the evaluation. In this 51-repeat experiments, parameter ϵ are set to {infinity, 1e+07, 1e+06, 1e+05, 1e+04}. Other parameters such as α_0 and β are the same as LEVY in **Section** 4.3.1. When ϵ is set to infinity, it implements exactly the same as LEVY.

4.4.2 Experimental Results and Discussion

Table 4.15 shows the IGD results of LEVY with different truncation. **Fig.** 4.12 illustrates IGD by generations for the best runs. When ϵ is set to 1e+07, it holds the best median (1.46e-04) and best values in terms of IGD. However, when ϵ is set to infinity and 1e+06, the algorithm still performs good results (i.e., around 2e-04). When ϵ is set to 1e+05 and 1e+04, the performance of the algorithm becomes worse (i.e., around 3e-03). In addition, the evolutionary process shows that for the first ϵ settings, IGD decreases rapidly in the first 20th generations. However, the same observation cannot be found when a smaller ϵ is applied.

The results indicate that when applying a truncation, IGD becomes worse, especially when ϵ is set to some relatively small values (i.e., 1e+05 and 1e+04). In addition, the evolutionary process shows that when ϵ is set to the small values, the algorithm cannot explore a wide area, for the IGD does not show a rapid decrease. As a small truncation can limit the length of trajectories into a small interval, these results may provide another evidence for the effectiveness of long trajectories in LF.

Table 4.15: IGD of LEVY with different ϵ at 100th generation on Nikkei

ϵ	infinity	1e+07	1e+06	1e+05	1e+04
Best	8.08E-05	5.92E-05	9.48E-05	1.51E-03	3.18E-03
Median	1.91E-04	**1.46E-04**	2.00E-04	3.00E-03	3.34E-03
Std.	9.71E-05	8.31E-05	8.13E-05	4.02E-04	1.20E-04

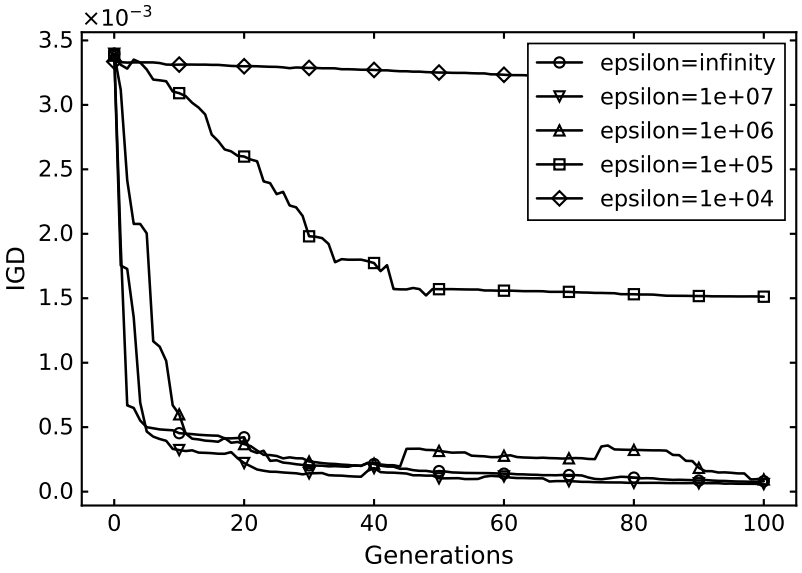


Fig. 4.12: IGD by generations of LEVY with different ϵ on Nikkei

4.5 Experiment IV: Comparison on Mutation based on Normal Distributions with Different Variance

4.5.1 Parameter Settings

In **Section** 4.3.3, the effectiveness of long trajectories in the beginning phase has been shown by tracking the Euclidean distance between parent and offspring. Also, in **Section** 4.4.2, this effectiveness has been proved by comparison with truncated LF. However, these long trajectories can also be found when applying normal distribution with a large variance. In this section, NORM (i.e., a comparison method in Experiment II) with different parameter settings is compared. The formulation of this method is recalled in (4.8).

$$\boldsymbol{y} = \boldsymbol{x}^{(i)} + C \cdot (\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(k)}) \oplus \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{1}) \tag{4.8}$$

The parameter C is set to $\{0.5,\ 1,\ 5,\ 10,\ 50,\ 100\}$. Other parameter settings are the same as **Section** 4.3.1. When C is set to 0.5, it performs exactly the same as NORM in Experiment II.

4.5.2 Experimental Results and Discussion

Table 4.16 presents the distribution of IGD of different parameter C in the final generation for 51 runs on the Nikkei dataset. The parameter C controls the variance of the normal distribution, and a larger C shows a larger variance. **Fig.** 4.13 illustrates how IGD changes by generations in the best run of each parameter setting.

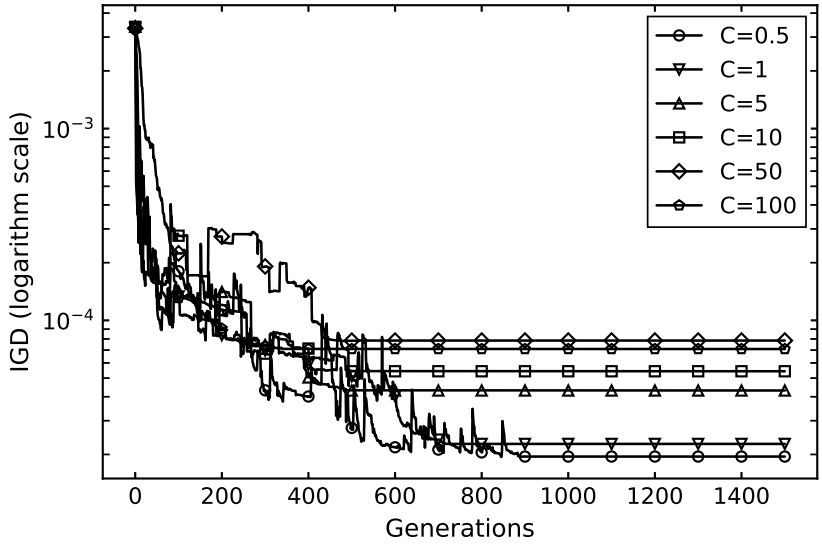


Fig. 4.13: IGD by generations of NORM with different C on Nikkei

Table 4.16: IGD of NORM with different C on Nikkei

C	0.5	1	5	10	50	100
Best	1.95E-05	2.27E-05	4.31E-05	5.43E-05	7.85E-05	7.09E-05
Median	**2.27E-05**	5.14E-05	1.31E-04	1.35E-04	1.85E-04	2.19E-04
Std.	4.31E-05	3.35E-05	9.47E-05	9.44E-05	1.00E-04	1.17E-04

As a result, the best median of IGD holds when C is set to 0.5, and a larger variance leads to a worse performance in terms of IGD. In addition, one may notice that, for normal distribution with large variance (e.g. $C = 100$), IGD decreases rapidly at the beginning, but also converges quickly. This may indicate that, by applying a large variance to the normal distribution, the algorithm does find a promising area by long trajectories. However, it cannot implement an efficient exploitation in the later steps, for the proportion of short motion becomes less when applying a larger variance. Therefore, the performance of LF mutation is not only derived from its global search capability but a balance between global search and local search.

Chapter 5

Conclusions

5.1 Summary of the Research

In this thesis, a novel method has been proposed to solve the MOOP formulation of PO by injecting LF into MOEA/D as a mutation method. The main reason for this modification is derived from the efficient global search capability of LF and the high-dimensional difficulty of the problem. One of the main contributions of this work is to analyze and explain how LF improves the results.

In the experiments, the proposed method was first compared with several literature methods, namely MOEA/D-DEM, MOEA/D-DE, MOEA/D-GA, and NSGA-II. This comprehensive assessment was implemented on five frequently used PO benchmark with unit constraint. The results on six evaluation metrics have shown the superiority of the proposed method (e.g. MOEA/D-Lévy holds the best IGD and HV on the largest problem Nikkei).

The main focus of the thesis is on analyzing the reason that LF improves the performance. Thus, LF mutation (LEVY) has been compared with three other distribution-based mutation methods (UNIF, NORM, and CONST). Except for the good performance of LEVY (e.g. it performs best in terms of IGD and HV on Nikkei), the experimental results have shown that methods using LF mutation can retrieve a relatively better solution set at the beginning of the optimization. By tracking the distance between parents and offspring during the evolutionary process, I have found that LF mutation promotes global search at the beginning and search multiple areas of the objective space at the same time. To explain this behavior, I have provided an interpretation considering the property of the PO. That is, the re-allocation of capital becomes centralized on several assets when applying LF mutation, and thus, the algorithm can search the area with high return and high risk efficiently even in the beginning phase.

Additionally, two extra experiments have shown further evidence of the efficiency of LF mutation. By comparing LF mutation with different truncation, the effectiveness of long trajectories has been again confirmed. In the comparison on mutation methods based on normal distributions with different variance, the results have indicated that the performance of the proposed method is not only derived from its strong global search capability but the good balance between global search and local search.

From an experimental point of view, this thesis has shown that the proposed method, or more specifically LF mutation, holds a good control on the balance between global search and local search, and utilization on the structural information of the problem when solving PO.

5.2 Limitations and Future Work

This thesis discusses based on a simple PO model and benchmark data for convenience. However, several limitations on the problem model may remain a further investigation.

- The experiments and discussion of this thesis are based on PO benchmarks with a simple unit constraint. Also, the financial data in these benchmarks are out-of-date (from 1992 to 1997). Thus, one cannot exactly estimate the performance of the proposed algorithm in a realistic PO based on this study.
- The Markowitz Mean-Variance model assumes that the return of an asset follows a normal distribution. However, some asset return follows an asymmetrical distribution, and thus variance may not be a good assessment for risk. To fix this problem, Markowitz has suggested to use a semi-variance as a risk assessment. Another solution is to introduce a skewness parameter into the model.

Despite the limitations, other future work concerns further discussion on the effectiveness of LF, new proposals to improve the performance, adaption to a practical system and possibility to extend the method on other problems. The following ideas may be good for a trial.

- The long trajectories or global search help in the beginning phase in this simple problem. However, one may get different observations when LF interacts with difficult constraints.
- In **Fig.** 4.6, few long trajectories get successfully updated with the generation increased. Thus, one may design an adaptive strategy to reduce global search but enhance local search in the middle and ending period of optimization.
- The risk assessment in the PO is computed using co-variance between asset returns, and thus turns the problem difficult. However, a better performance may be achieved by designing a non-separable search strategy or specifically designed data structure.
- In **Section** 4.3, the results have shown that the proposed method can achieve a relatively good solution set even at the beginning phase. Therefore, one may design a hybrid strategy and utilize this property to speed up the computation, which is an important concern in the practical application.

5.3 Publications

Several publications in the following list are derived from this thesis.

- Yifan He, Claus Aranha and Hitoshi Kanoh. Solving Portfolio Optimization Problems based on MOEA/D and Lévy Flight. In *Symposium of the Japanese Society of Evolutionary Computation 2019*, pages 75-82. JSEC, Miyagi, sep 2019.
- Yifan He, Claus Aranha. Solving Portfolio Optimization Problems Using MOEA/D and Lévy Flight. *Swarm and Evolutionary Computation*, **under review**.

Acknowledgements

First, I would like to express my deep gratitude to Prof. Hitoshi Kanoh (狩野 均　教授) and Dr. Claus Aranha, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work.

I would also like to thank my thesis committee: Prof. Jun Sakuma, for his advice and hard questions. My grateful thanks are also extended to Prof. Yohei Akimoto, for his advice and critiques on experimental design.

I would also like to extend my thanks to my fellows in Knowledge System Lab, for their help and discussion during the lab seminar. I would like to express my special thanks to Mr. Yuri Lavinas in Mathematical Modeling and Algorithm Lab for the discussion on MOEA/D.

My sincere thanks also go to Dr. Pinata Winoto and Dr. Tiffany Y. Tang, for enlightening me at first glance of research. Also, I would like to thank my friends at Wenzhou-Kean University.

Finally, I wish to thank my parents, for giving birth to me, and for their support and encouragement throughout my study.

Bibliography

[1] Harry Markowitz. PORTFOLIO SELECTION*. *The Journal of Finance*, 7(1):77–91, mar 1952.

[2] Daniel Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140, aug 1996.

[3] Prisadarng Skolpadungket, Keshav Dahal, and Napat Harnpornchai. Portfolio optimization using multi-obj ective genetic algorithms. In *2007 IEEE Congress on Evolutionary Computation*, pages 516–523. IEEE, sep 2007.

[4] Sudhansu Kumar Mishra, Sukadev Meher, Ganapati Panda, and Abhishek Panda. Comparative performance evaluation of multiobjective optimization algorithms for portfolio management. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 1338–1342. IEEE, 2009.

[5] S.K. Mishra, G. Panda, S. Meher, R. Majhi, and M. Singh. Portfolio management assessment by four multiobjective optimization algorithm. In *2011 IEEE Recent Advances in Intelligent Computational Systems*, pages 326–331. IEEE, sep 2011.

[6] Konstantinos P. Anagnostopoulos and Georgios Mamanis. Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science*, 8(3):259–279, aug 2011.

[7] Feijoo Colomine Duran, Carlos Cotta, and Antonio J. Fernández. Evolutionary Optimization for Multiobjective Portfolio Selection under Markowitz's Model with Application to the Caracas Stock Exchange. pages 489–509. Springer, Berlin, Heidelberg, 2009.

[8] K.P. Anagnostopoulos and G. Mamanis. The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, 38(11):14208–14217, oct 2011.

[9] Sudhansu Kumar Mishra, Ganapati Panda, and Ritanjali Majhi. A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. *Swarm and Evolutionary Computation*, 16:38–51, jun 2014.

[10] Sudhansu Kumar Mishra, Ganapati Panda, and Ritanjali Majhi. Constrained portfolio asset selection using multiobjective bacteria foraging optimization. *Operational Research*, 14(1):113–145, apr 2014.

[11] Divya Kumar and K.K. Mishra. Portfolio optimization using novel co-variance guided Artificial Bee Colony algorithm. *Swarm and Evolutionary Computation*, 33:119–130, apr 2017.

[12] Qingfu Zhang, Hui Li, Dietmar Maringer, and Edward Tsang. MOEA/D with NBI-style Tchebycheff approach for portfolio management. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, jul 2010.

[13] Heng Zhang, Yaoyu Zhao, Feng Wang, Anran Zhang, Pengwei Yang, and Xiaoliang Shen. A new evolutionary algorithm based on MOEA/D for portfolio optimization. In *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pages 831–836. IEEE, mar 2018.

[14] Zhongbao Zhou, Xianghui Liu, Helu Xiao, Shijian Wu, and Yueyue Liu. A DEA-based MOEA/D algorithm for portfolio optimization. *Cluster Computing*, pages 1–10, mar 2018.

[15] G.M. Viswanathan, E.P. Raposo, and M.G.E. da Luz. Lévy flights and superdiffusion in the context of biological encounters and random searches. *Physics of Life Reviews*, 5(3):133–150, sep 2008.

[16] Hüseyin Haklı and Harun Uğuz. A novel particle swarm optimization algorithm with Levy flight. *Applied Soft Computing*, 23:333–345, oct 2014.

[17] T.-J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, nov 2000.

[18] Kalyanmoy. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

[19] James David Schaffer. Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition). 1986.

[20] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[21] Carlos M Fonseca, Peter J Fleming, et al. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Icga*, volume 93, pages 416–423. Citeseer, 1993.

[22] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[23] Je rey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence*, volume 1, pages 82–87. Citeseer, 1994.

[24] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[25] Joshua Knowles and David Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Congress on Evolutionary Computation (CEC99)*, volume 1, pages 98–105, 1999.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, apr 2002.

[27] Hiroaki Fukumoto and Akira Oyama. Benchmarking multiobjective evolutionary algorithms and constraint handling techniques on a real-world car structure design optimization benchmark problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 177–178. ACM, 2018.

[28] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, dec 2007.

[29] Rosario N. Mantegna and H. Eugene Stanley. Stochastic Process with Ultraslow Convergence to a Gaussian: The Truncated Lévy Flight. *Physical Review Letters*, 73(22):2946–2949, nov 1994.

[30] Marek Gutowski. L\'evy flights as an underlying mechanism for global optimization algorithms. *arXiv preprint math-ph/0106003*, 2001.

[31] Wei Peng and Qingfu Zhang. A decomposition-based multi-objective Particle Swarm Optimization algorithm for continuous optimization problems. In *2008 IEEE International Conference on Granular Computing*, pages 534–537. IEEE, aug 2008.

[32] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, apr 2009.

[33] Liangjun Ke, Qingfu Zhang, and Roberto Battiti. MOEA/D-ACO: A Multiobjective Evolutionary Algorithm Using Decomposition and AntColony. *IEEE Transactions on Cybernetics*, 43(6):1845–1859, dec 2013.

[34] Ryoji Tanabe and Hisao Ishibuchi. Review and analysis of three components of the differential evolution mutation operator in MOEA/D-DE. *Soft Computing*, pages 1–15, feb 2019.

[35] Salvatore Arnone, Andrea Loraschi, and Andrea Tettamanzi. A Genetic Approach to Portfolio Selection. *Neural Network World*, 3(6):597–604, 1993.

[36] J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck. Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *European Journal of Operational Research*, 199(3):684–693, dec 2009.

[37] Okkes Ertenlice and Can B. Kalayci. A survey of swarm intelligence for portfolio optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 39:36–52, apr 2018.

[38] Sudhansu Kumar Mishra, Ganapati Panda, and Sukadev Meher. Multi-objective particle swarm optimization approach to portfolio optimization. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 1612–1615. IEEE, 2009.

[39] J. J. Liang and B. Y. Qu. Large-scale portfolio optimization using multiobjective dynamic mutli-swarm particle swarm optimizer. In *2013 IEEE Symposium on Swarm Intelligence (SIS)*, pages 1–6. IEEE, apr 2013.

[40] Jianli Zhou and Jun Li. An improved multi-objective particle swarm optimization for constrained portfolio selection model. In *2014 11th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–5. IEEE, jun 2014.

[41] Yukiko Orito, Yoshiko Hanada, Shunsuke Shibata, and Hisashi Yamamoto. A New Population Initialization Approach Based on Bordered Hessian for Portfolio Optimization Problems. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1341–1346. IEEE, oct 2013.

[42] K. Liagkouras and K. Metaxiotis. A new Probe Guided Mutation operator and its application for solving the cardinality constrained portfolio optimization problem. *Expert Systems with Applications*, 41(14):6274–6290, oct 2014.

[43] Suraj S. Meghwani and Manoj Thakur. Multi-objective heuristic algorithms for practical portfolio optimization and rebalancing with transaction cost. *Applied Soft Computing*, 67:865–894, jun 2018.

[44] K. Liagkouras and K. Metaxiotis. Handling the complexities of the multi-constrained portfolio optimization problem with the support of a novel MOEA. *Journal of the Operational Research Society*, 69(10):1609–1627, oct 2018.

[45] Anis Farhan Kamaruzaman, Azlan Mohd Zain, Suhaila Mohamed Yusuf, and Amirmudin Udin. Levy Flight Algorithm for Optimization Problems - A Literature Review. *Applied Mechanics and Materials*, 421:496–501, sep 2013.

[46] Xin-She Yang and Suash Deb. Cuckoo Search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 210–214. IEEE, 2009.

[47] Wei Ma, Zhengxing Sun, Junlou Li, Mofei Song, Xufeng Lang, and Cheng Le. An Artificial Bee Colony Algorithm Guided by Lévy Flights Disturbance Strategy for Global Optimization. pages 493–503. Springer, Cham, 2015.

[48] R. Jensi and G. Wiselin Jiji. An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing*, 43:248–261, jun 2016.

[49] I. Aydogdu, A. Akin, and M.P. Saka. Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution. *Advances in Engineering Software*, 92:1–14, feb 2016.

[50] Ruining Zhang, Xuemei Jiang, and Ruifang Li. Improved decomposition-based multi-objective cuckoo search algorithm for spectrum allocation in cognitive vehicular network. *Physical Communication*, 34:301–309, jun 2019.

[51] Elham Shadkam, Reza Delavari, Farzad Memariani, and Morteza Poursaleh. Portfolio selection by the means of cuckoo optimization algorithm. aug 2015.

[52] Farshad Faezy Razi and Naeimeh Shadloo. Portfolio selection using hybrid algorithm of data envelopment analysis based on Kohonen neural network and Cuckoo algorithm. *Journal of Information and Optimization Sciences*, 37(4):549–567, jul 2016.

[53] Sarah El-Bizri and Nashat Mansour. Metaheuristics for Portfolio Optimization. pages 77–84. Springer, Cham, 2017.

[54] Carlos A. Coello Coello and Margarita Reyes Sierra. A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm. pages 688–697. Springer, Berlin, Heidelberg, 2004.

Appendix A

Pre-experiments

This appendix presents the results of pre-experiments, implementing the parameter tuning steps for all methods used in the experiments in **Chapter** 4. I select the parameters based on the average IGD performance at 300th generation when optimizing the Nikkei dataset with 30 repeats. Several paremeters are directly set to the values used in prior studies. For MOEA/D-based algorithms, T is set to 20, σ is set to 0.9, and n_r is set to 2. For NSGA-II, binary tournament is used (i.e., tournament size is 2). For GA-based algorithms, the index parameters η_c and η_m for SBX crossover and polynomial mutation are set to 20. For all algorithms, the population size is set to 100.

A.1 Parameter Tuning for MOEA/D-Lévy

The tuned parameters for MOEA/D-Lévy is summarized in the following table.

- The scaling parameter α_0 is tuned among {1e-06, 1e-05, 1e-04, 1e-03, 1e-02, 1e-01}, when $\beta = 0.4$ and $p_m = 0$. Based on the results in **Fig.** A.1 and **Table** A.2, this α_0 is set to 1e-05 in the later experiments.
- The index parameter β for the stable distribution is then tuned among {0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}, when α = 1e-05 and $p_m = 0$. The results in **Fig.** A.2 and **Table** A.3 show that when $\beta = 0.3$, the best IGD is achieved.
- The mutation rate p_m for the polynomial mutation is tuned among {0.001, $1/n$=0.004, 0.010, 0.050, 0.100} ($n = 225$ is the dimension of the problem), when α_0 = 1e-05 and $\beta = 0.3$. **Fig.** A.3 and **Table** A.4 show that the best IGD is achieved when p_m is set to $1/n$.

Table A.1: A summary of paremeters used in MOEA/D-Lévy

α_0	1e-05
β	0.3
p_m	$1/n$

Table A.2: Results of parameter tuning on α_0 in MOEA/D-Lévy

α_0	Mean	Std.
1e-06	9.22E-05	4.35E-05
1e-05	**6.18E-05**	3.57E-05
1e-04	6.61E-05	2.85E-05
1e-03	7.90E-05	3.03E-05
1e-02	1.34E-04	4.95E-05
1e-01	1.77E-04	7.49E-05

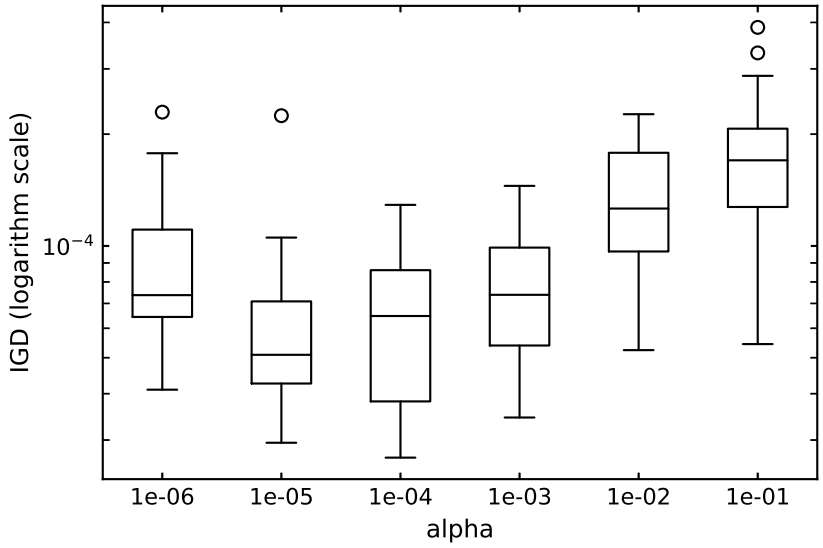


Fig. A.1: IGD by α_0 in MOEA/D-Lévy at 300th generation on Nikkei

Table A.3: Results of parameter tuning on β in MOEA/D-Lévy

β	Mean	Std.
0.3	**6.32E-05**	5.60E-05
0.4	1.28E-04	5.50E-05
0.5	3.96E-04	1.88E-04
0.6	2.39E-03	9.36E-04
0.7	3.04E-03	4.67E-04
0.8	3.42E-03	9.10E-05
0.9	3.40E-03	7.50E-05
1.0	3.42E-03	1.08E-04

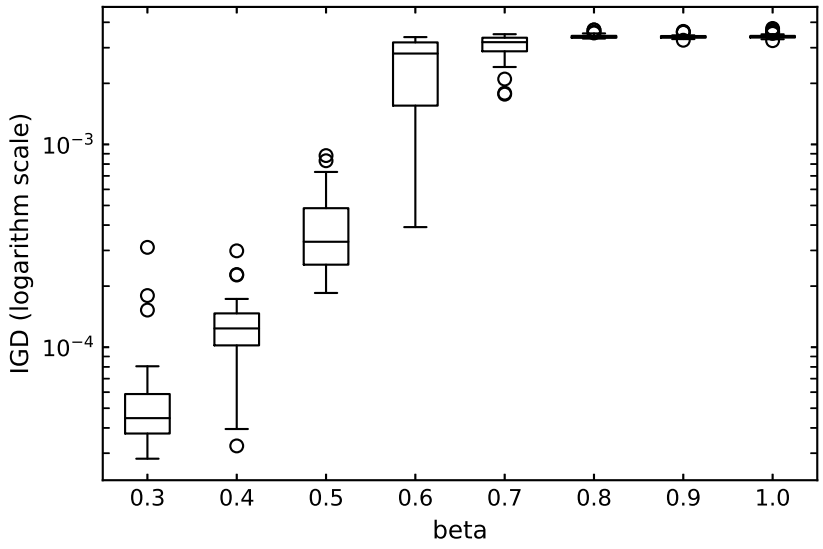


Fig. A.2: IGD by β in MOEA/D-Lévy at 300th generation on Nikkei

Table A.4: Results of parameter tuning on p_m in MOEA/D-Lévy

p_m	Mean	Std.
0.001	5.90E-05	3.68E-05
$1/n = 0.004$	**5.00E-05**	1.75E-05
0.010	7.20E-05	2.81E-05
0.050	2.05E-04	5.79E-05
0.100	3.30E-04	5.30E-05

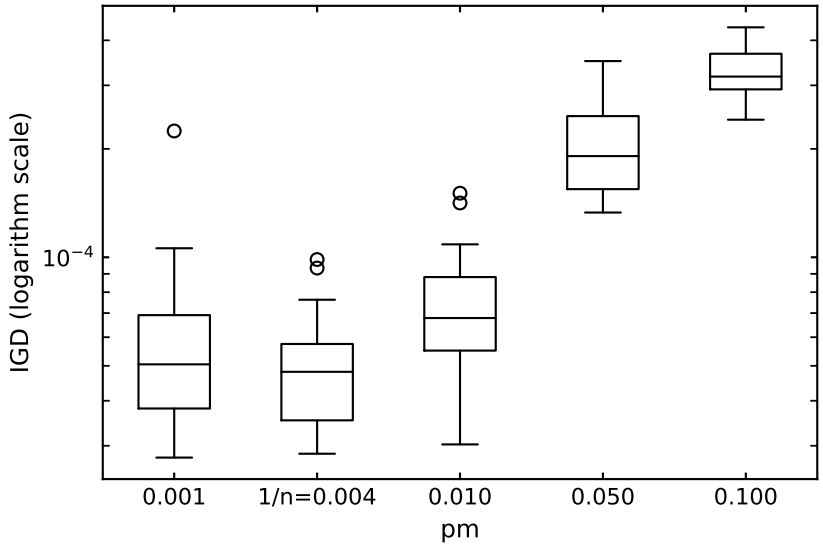


Fig. A.3: IGD by p_m in MOEA/D-Lévy at 300th generation on Nikkei

A.2 Parameter Tuning for MOEA/D-DEM

The tuned parameter for MOEA/D-DEM is present in **Table** A.5.

- The scaling parameter F for the DE mutation is tuned among {0.1, 0.5, 0.9, 1.3, 1.7, 2.0}, when p_m is set to 0. **Fig.** A.4 and **Table** A.6 shows the tuning results. Based on the average IGD in the table, I select F as 1.3.
- The mutation rate p_m is then tuned among {0.001, $1/n$=0.004, 0.010, 0.050, 0.100} ($n = 225$ is the dimension of the problem), when F is set to 1.3. **Fig.** A.5 and **Table** A.7 present the tuning result for this parameter. This mutation rate is set to $1/n$ in the later steps.

A.3 Parameter Tuning for MOEA/D-DE

The parameters used in MOEA/D-DE is based on the tuning results of MOEA/D-DEM, i.e. $F = 1.3$.

Table A.5: A summary of paremeters used in MOEA/D-DEM

F	1.3
p_m	$1/n$

Table A.6: Results of parameter tuning on F in MOEA/D-DEM

F	Mean	Std.
0.1	3.34E-04	6.30E-05
0.5	9.09E-04	4.34E-04
0.9	2.70E-04	8.60E-05
1.3	**2.54E-04**	1.12E-04
1.7	2.71E-04	1.10E-04
2.0	2.96E-04	1.20E-04

Table A.7: Results of parameter tuning on p_m in MOEA/D-DEM

p_m	Mean	Std.
0.001	5.00E-05	4.00E-05
$1/n = 0.004$	**3.00E-05**	1.80E-05
0.010	5.00E-05	5.60E-05
0.050	6.90E-04	5.40E-05
0.100	1.32E-03	5.35E-04

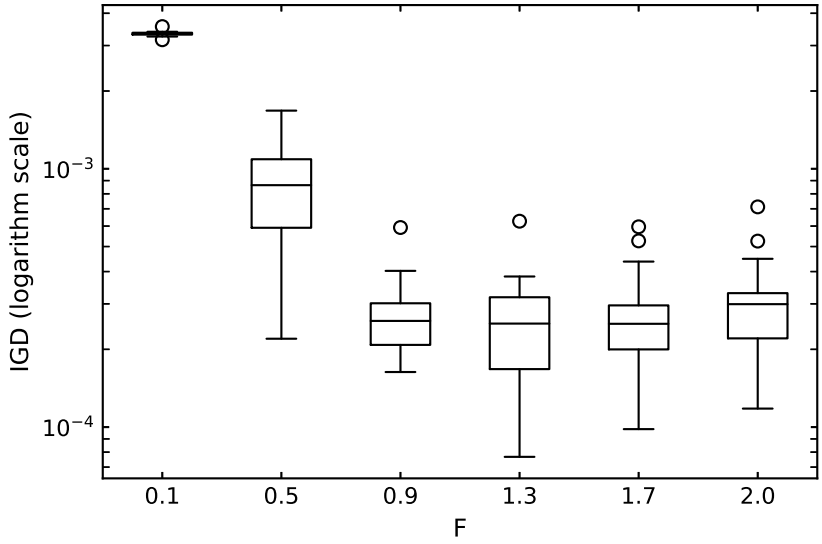


Fig. A.4: IGD by F in MOEA/D-DEM at 300th generation on Nikkei

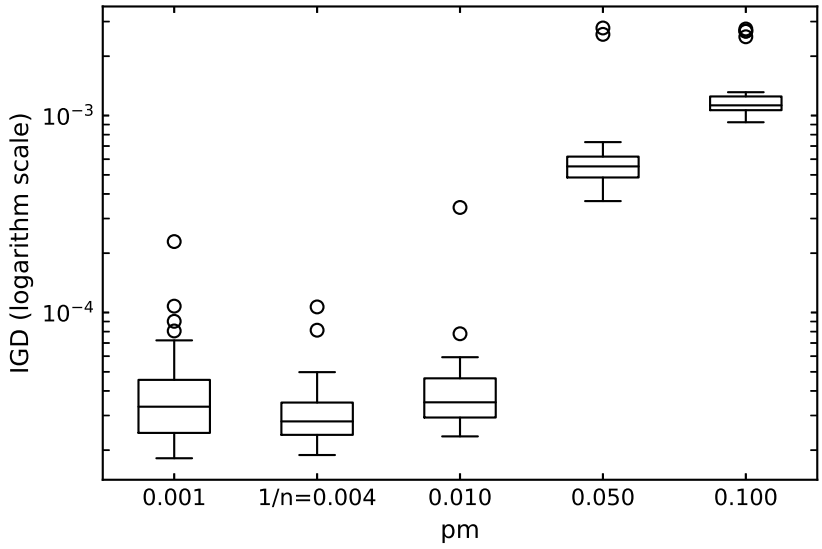


Fig. A.5: IGD by p_m in MOEA/D-DEM at 300th generation on Nikkei

A.4 Parameter Tuning for MOEA/D-GA

A summary of the tuned parameter for MOEA/D-GA is present in **Table** A.8.

- The crossover rate p_c is first tuned among {0.6, 0.7, 0.8, 0.9, 1.0}, when the mutation rate p_m is set to 0.05. Based on the results in **Fig.** A.6 and **Table** A.10, this rate is set to 0.7.
- The mutation rate p_m is then tunes among {0.001, $1/n$=0.004, 0.010, 0.050, 0.100}, when p_c is set to 0.7. The results is present in **Fig.** A.7 and **Table** A.11. In terms of mean, one may find p_m = 0.05 is a better choice. Another possible choice is p_m = 0.01, for it holds the best median. However, it is easy to notice that there are many outliers with this parameter setting, which is not desirable in the optimization. Therefore, I select 0.05 as the mutation rate.

A.5 Parameter Tuning for NSGA-II

The tuned parameter for NSGA-II is present in **Table** A.9.

- The crossover rate p_c is first tuned among {0.6, 0.7, 0.8, 0.9, 1.0}, when the mutation rate p_m is set to 0.05. Based on the results in **Fig.** A.8 and **Table** A.12, this rate is set to 0.7.
- The mutation rate p_m is then tunes among {0.001, $1/n$=0.004, 0.010, 0.050, 0.100}, when p_c is set to 0.7. The results is present in **Fig.** A.9 and **Table** A.13. Based on the results, the mutation rate is set to 0.01.

Table A.8: A summary of paremeters used in MOEA/D-GA

p_c	0.70
p_m	0.05

Table A.9: A summary of paremeters used in NSGA-II

p_c	0.70
p_m	0.01

Table A.10: Results of parameter tuning on p_c in MOEA/D-GA

p_c	Mean	Std.
0.6	3.71E-04	5.82E-04
0.7	**2.60E-04**	7.30E-04
0.8	6.16E-04	8.90E-04
0.9	4.16E-04	6.07E-04
1.0	4.57E-04	7.09E-04

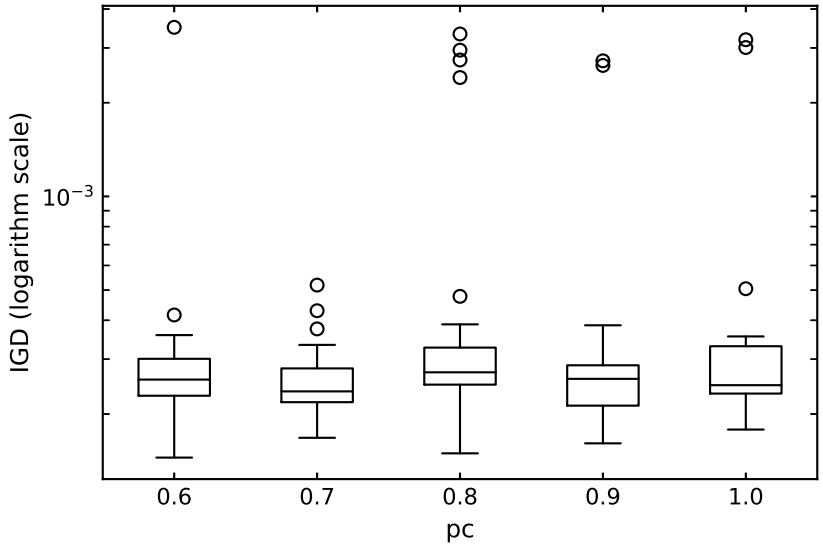


Fig. A.6: IGD by p_c in MOEA/D-GA at 300th generation on Nikkei

Table A.11: Results of parameter tuning on p_m in MOEA/D-GA

p_m	Mean	Std.
0.001	5.77E-04	1.34E-04
$1/n = 0.004$	6.19E-04	9.46E-04
0.010	4.93E-04	8.71E-04
0.050	**3.71E-04**	4.27E-04
0.100	7.41E-04	8.74E-04

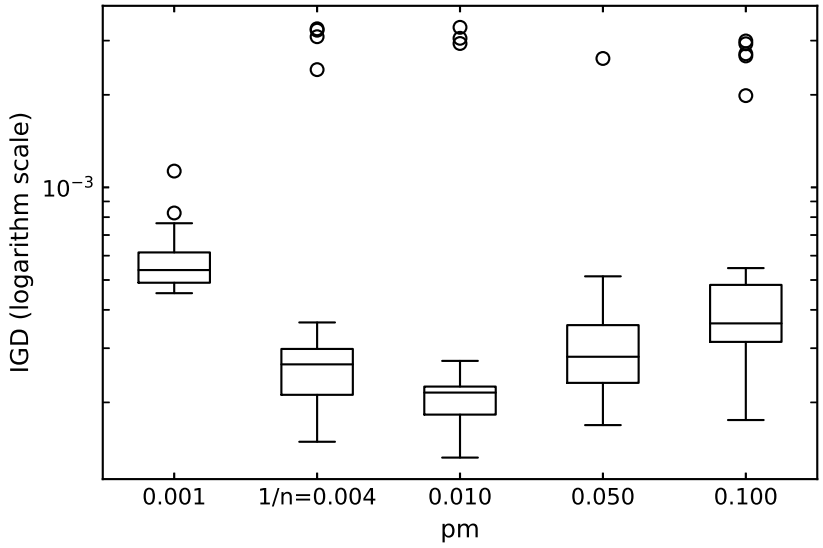


Fig. A.7: IGD by p_m in MOEA/D-GA at 300th generation on Nikkei

Table A.12: Results of parameter tuning on p_c in NSGA-II

p_c	Mean	Std.
0.6	1.74E-04	4.26E-05
0.7	**1.70E-04**	3.09E-05
0.8	1.99E-04	4.30E-04
0.9	2.13E-04	4.75E-05
1.0	2.53E-04	4.22E-05

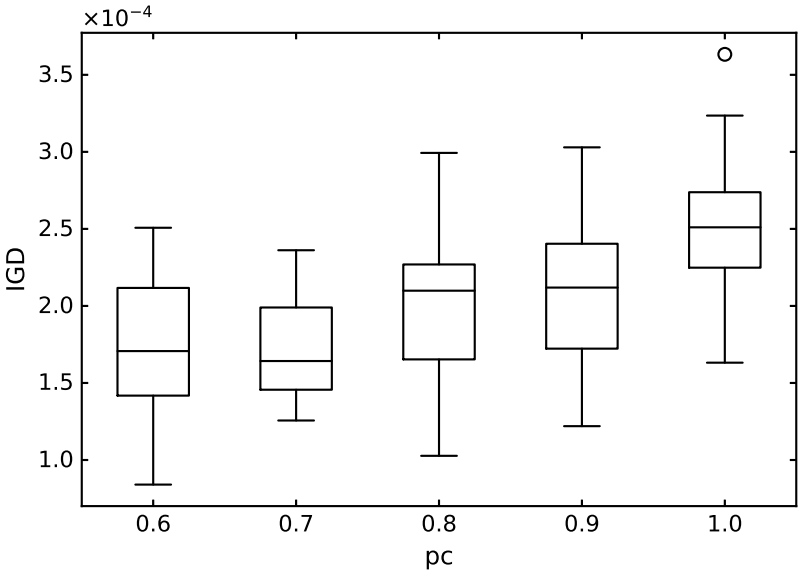


Fig. A.8: IGD by p_c in NSGA-II at 300th generation on Nikkei

Table A.13: Results of parameter tuning on p_m in NSGA-II

p_m	Mean	Std.
0.001	3.23E-04	6.66E-05
$1/n = 0.004$	1.91E-04	3.76E-05
0.010	**1.63E-04**	3.84E-05
0.050	2.13E-04	4.82E-05
0.100	3.32E-02	7.35E-05

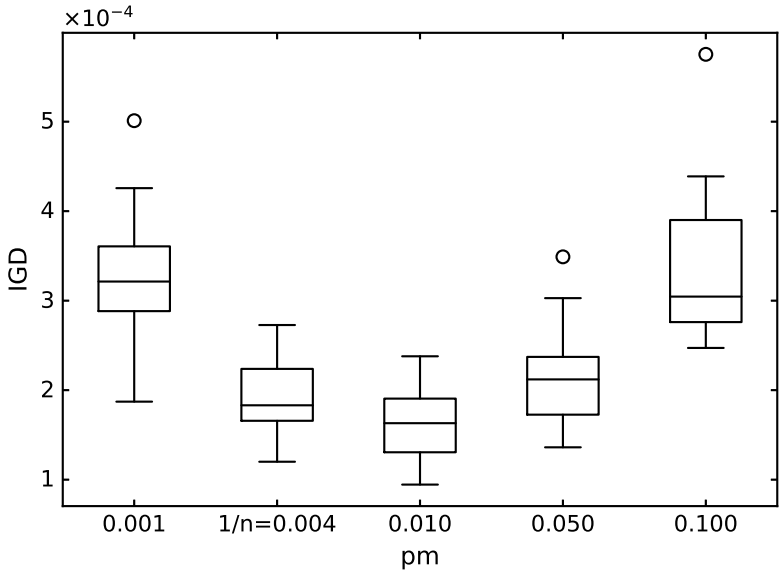


Fig. A.9: IGD by p_m in NSGA-II at 300th generation on Nikkei

A.6 Parameter Tuning for LEVY

The parameter setting of LEVY can be referred to that of MOEA/D-Lévy, i.e. α_0 is set to 1e-05 and β is set to 0.3 (**Section** A.1).

A.7 Parameter Tuning for UNIF

In UNIF, only one parameter, the scaling parameter C need to be tuned. Values among {0.05, 0.1, 0.5, 1, 5, 10, 50, 100} are tried. Based on the results in **Fig.** A.10 and **Table** A.14, the best setting of this scaling parameter is 1.

A.8 Parameter Tuning for NORM

The parameter tuning of NORM is similar as that of UNIF. As shown in **Fig.** A.11 and **Table** A.15, the best average IGD is achieved when C is set to 0.5, among the candidate settings {0.05, 0.1, 0.5, 1, 5, 10, 50, 100}.

One may notice in the table, when C is set to 5 and 10, they hold the same mean and standard deviation. This is caused by the rounding process in the statistical steps. In fact, they performs different but close values (i.e., avg. = 1.703E-04 for C at 5, and avg. = 1.674E-04 for C at 10).

A.9 Parameter Tuning for CONST

As CONST is exactly the same as MOEA/D-DE, its parameter setting can be referred to that of MOEA/D-DEM in **Section** A.2.

Table A.14: Results of parameter tuning on C in UNIF

C	Mean	Std.
0.05	3.33E-03	1.27E-04
0.1	3.32E-03	1.49E-04
0.5	1.41E-04	5.50E-05
1	**7.10E-05**	7.40E-05
5	1.01E-04	3.80E-05
10	1.25E-04	8.20E-05
50	1.71E-04	7.90E-05
100	2.31E-04	1.29E-04

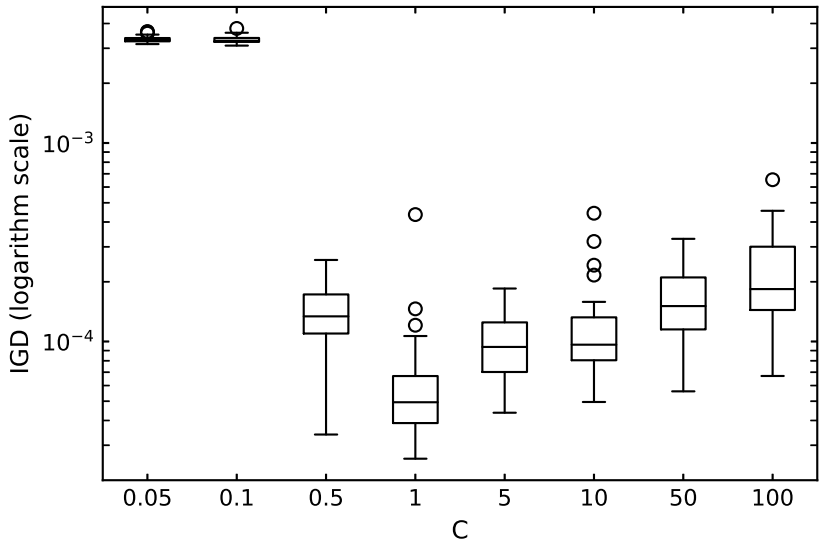


Fig. A.10: IGD by C in UNIF at 300th generation on Nikkei

Table A.15: Results of parameter tuning on C in NORM

C	Mean	Std.
0.05	3.34E-03	1.08E-04
0.1	3.19E-03	1.53E-04
0.5	**6.00E-05**	3.10E-04
1	8.00E-05	3.00E-05
5	1.70E-04	7.40E-05
10	1.70E-04	7.40E-05
50	2.00E-04	1.07E-04
100	1.90E-04	8.50E-05

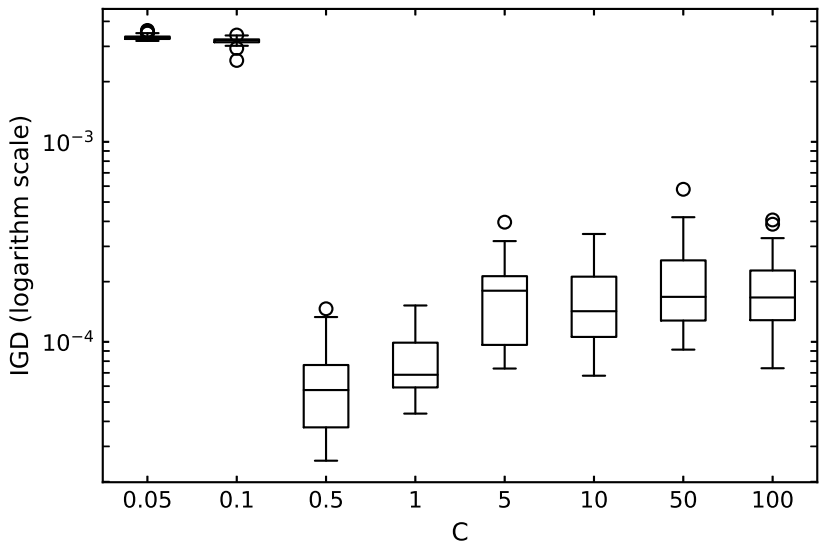


Fig. A.11: IGD by C in NORM at 300th generation on Nikkei