

Research Article

Large-Scale Portfolio Optimization Using Multiobjective Evolutionary Algorithms and Preselection Methods

B. Y. Qu,^{1,2} Q. Zhou,¹ J. M. Xiao,¹ J. J. Liang,^{1,3} and P. N. Suganthan⁴

¹*School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China*

²*School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China*

³*School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China*

⁴*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798*

Correspondence should be addressed to J. J. Liang; liangjing@zzu.edu.cn

Received 15 October 2016; Accepted 22 January 2017; Published 20 February 2017

Academic Editor: Thomas Hanne

Copyright © 2017 B. Y. Qu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Portfolio optimization problems involve selection of different assets to invest in order to maximize the overall return and minimize the overall risk simultaneously. The complexity of the optimal asset allocation problem increases with an increase in the number of assets available to select from for investing. The optimization problem becomes computationally challenging when there are more than a few hundreds of assets to select from. To reduce the complexity of large-scale portfolio optimization, two asset preselection procedures that consider return and risk of individual asset and pairwise correlation to remove assets that may not potentially be selected into any portfolio are proposed in this paper. With these asset preselection methods, the number of assets considered to be included in a portfolio can be increased to thousands. To test the effectiveness of the proposed methods, a Normalized Multiobjective Evolutionary Algorithm based on Decomposition (NMOEA/D) algorithm and several other commonly used multiobjective evolutionary algorithms are applied and compared. Six experiments with different settings are carried out. The experimental results show that with the proposed methods the simulation time is reduced while return-risk trade-off performances are significantly improved. Meanwhile, the NMOEA/D is able to outperform other compared algorithms on all experiments according to the comparative analysis.

1. Introduction

In modern financial markets, portfolio allocation is one of the major problems faced by investors and fund managers. Investors need to choose several assets from thousands of available assets to form a single portfolio from an enormous set of possibilities in order to simultaneously maximize the return and minimize the risk. It is crucial for investors to find a perfect portfolio allocation. However, higher returns are generally associated with higher risk. Investors with varying degrees of risk aversion will demand different levels of return for taking on different degrees of risk [1, 2].

The concept of portfolio optimization has been an important tool in the development and understanding of financial markets. Portfolio optimization techniques can assist the search for the portfolio that best suits each investor's particular objective [3, 4]. As stated by the BusinessWeek [5],

the single best weapon against risk is to form portfolios with uncorrelated or negatively correlated assets because when several such assets are combined together, the overall risk of the portfolio may be less than that of the individual asset. Thus, finding a suitable combination of investments attracted attentions of investors and scholars. The major breakthrough of portfolio optimization came in 1952 with the publication of Markowitz's theory of portfolio selection [6]. Markowitz quantified return and risk of a security using statistical measures of its expected return and standard deviation. Markowitz suggested that investors should consider return and risk together and determine the allocation of funds among investment alternatives on the basis of their return-risk trade-off [7]. This theory is popularly referred to as the modern portfolio theory and it is also the theoretical basis for this work. The details of the theory are presented in Section 2.

As there are two-conflicting objectives, there is no single optimal solution to this portfolio optimization problem. Instead, there is an efficient frontier of optimal trade-off solutions. It is often desirable to have the entire efficient frontier of optimal portfolio sets that give the average return against the possible risk so that individual investors can choose the most appropriate return-risk trade-off to suit their investment objectives. In recent years, evolutionary algorithms (EAs) have become an effective tool in handling optimization problems [8]. EAs search for the solution by initializing a population of random candidates. These candidates undergo an evolutionary process based on the survival-of-the-fittest mechanism and the individuals that have superior fitness will be passed to the next generations [9].

Using EAs to solve assets allocation problem has become a trend in recent years, as EAs are able to find multiple Pareto-optimal solutions in one single run [10]. Most of the early works transformed the optimization problem into a single-objective problem using a trade-off function [10–13]. With the development of multiobjective evolutionary algorithms (MOEAs) [8, 14–16], researchers have focused their attention on using MOEAs for solving portfolio optimization problems. The first use of MOEAs for solving portfolio optimization problem can be traced back to 1993 [17]. This work adopted lower partial moments and a genetic approach to handle the problem. In 1996, Shoaf and Foster [18] used a multiobjective GA to solve the conventional Markowitz formulation. They adopted a specific encoding scheme which was able to indicate whether the holding of the particular asset would be long or short. In [19], six different MOEAs including original VEGA [20], two versions of modified VEGA, NSGAI [15], MOGA [21], and SPEA2 [22] were examined and compared using the classical mean-variance model. A mixed binary-real encoding scheme was used for all the compared algorithms. Yan et al. [23] used the semivariance as the risk and growth rate as the return. In this work, instead of considering a single investment period, T periods were used for the optimization. The weight of each investment asset must be determined by the decision-maker at the beginning of each period. Lin and Liu [24] applied genetic algorithm on Taiwanese mutual fund for portfolio selection problem with minimum transaction lots. However, due to the computational complexity of solving a large-scale quadratic programming problem, Markowitz's portfolio optimization model has not been used extensively in its original form to construct a large-scale portfolio [24]. As the number of assets increases, the complexity of the search space increases exponentially. The state-of-the-art evolutionary algorithms may not effectively solve the problem when there are more than 200 assets available to form portfolios. Lin and Liu [24] used up to 204 assets to form the optimal portfolios. In [25], Ghahtarani and Najafi used 20 stocks from the Tehran stock exchange in the portfolio optimization. Fernández and Gómez [26] adopted Markowitz's model with cardinality and bounding constraints. In this work, four different markets with different number of stocks were considered and the largest number of stocks used was 225. Motivated by these observations, two asset preselection processes are proposed in this work

for large-scale portfolio optimization so that the number of assets to be considered can be increased to a few thousands or more. Five different MOEAs known as Normalized Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D), Multiobjective Differential Evolution based on Summation Sorting (MODE-SS), Multiobjective Differential Evolution based on Nondomination Sorting (MODE-NDS), Multiobjective Comprehensive Learning Particle Swarm Optimizer (MOCLPSO), and NSGAI are used as the optimization tools to test the effectiveness of the preselection process.

The rest of the paper is outlined as follows. Section 2 presents the models of portfolio optimization. Section 3 briefly reviews the concept of multiobjective optimization and the four multiobjective optimization algorithms used in this work. Sections 4 and 5 introduce the preselection process and the constraint handling method, respectively. The experimental settings and results are presented and analyzed in Section 6. Finally, the relevant conclusions and directions for future work are discussed in Section 7.

2. Portfolio Optimization

The mathematical representation of portfolio optimization was introduced by Markowitz in 1952 and he was rewarded with a Nobel Prize in Economics in 1990 [27]. The Markowitz model assumes that investors would like to maximize return under a certain risk level or minimize the risk with a certain return level [6] and this model makes use of the mean and variance of normalized historical asset prices to compute the expected portfolio return and risk [24], respectively. The model can be expressed as a biobjective problem as follows:

$$\begin{aligned} \text{Min} \quad & \sigma^2 = \sum_{i=1}^D \sum_{j=1}^D w_i w_j \sigma_{ij} \\ \text{Max} \quad & r_p = \sum_{i=1}^D w_i r_i \\ \text{Subject to} \quad & \sum_{i=1}^D w_i = 1 \quad w_i \geq 0, \end{aligned} \quad (1)$$

where D is the number of assets in the portfolio which is also the dimensionality of the optimization problem. w_i is the weight of the i th asset to be optimized. σ^2 stands for the portfolio risk while σ_{ij} is the covariance between asset i and asset j . If $i = j$, σ_{ij} is just the variance of that particular asset. r_p is the average portfolio return and r_i is the average individual return of asset i . Due to the determined efforts of various researchers including Sharpe [28], Pang [29], Best and Hlouskova [30], and others, Markowitz's work has been widely extended. Recent works also include various constraints for the portfolio optimization problem. Speranza [31] proposed a model that takes into account the characteristics of the portfolio optimization. Chang et al. [32] extended the standard model to include cardinality constraints that limit a portfolio to have a specified number of assets and to impose limits on the proportion of the portfolio

held in a given asset. In this paper, the original form of Markowitz model is adopted.

3. Multiobjective Optimization

3.1. Mathematical Model of Multiobjective Optimization Problem. Optimization refers to finding one or more feasible solutions which correspond to extreme values of one or more objectives [14, 33]. Typically, a multiobjective optimization problem can be defined as follows:

$$\begin{aligned} \text{Min/Max } f_m(x) &= f_m(x_1, x_2, \dots, x_n), \\ m &= 1, 2, \dots, M; \\ \text{Subject to } g_j(x) &= g_j(x_1, x_2, \dots, x_n) \geq 0, \\ j &= 1, 2, \dots, J; \quad (2) \\ h_k(x) &= h_k(x_1, x_2, \dots, x_n) = 0, \\ k &= 1, 2, \dots, K; \\ x_i^L &\leq x_i \leq x_i^U, \quad i = 1, 2, \dots, n, \end{aligned}$$

where $f_m(x)$ is the m th objective function, M is the number of objectives, and $x = (x_1, x_2, \dots, x_n)$ are decision variables. $g_j(x)$ and $h_k(x)$ are inequality and equality constraint functions, respectively. x_i^L and x_i^U are the lower and upper bounds of the decision variables. If an optimization problem involves more than one objective function, it is considered as a multiobjective optimization problem and the task of this problem is to find a set of solutions which can represent the trade-off among different objectives.

3.2. Normalized Multiobjective Evolutionary Algorithm Based on Decomposition

3.2.1. Original MOEA/D. The Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) was developed by Zhang et al. in 2007 [34] and it is considered as one of the most effective multiobjective optimization algorithms at present. MOEA/D provides a new framework by combining decomposition method and evolutionary algorithm for MOPs [35]. It explicitly decomposes the MOP into a number of single optimization subproblems and solves these subproblems simultaneously by evolving a population of solutions.

In MOEA/D, various decomposition methods can be used to decompose an MOP into a set of subproblems [34]. However, the weighted Tchebycheff approach is the most prominent method as it is less sensitive to the shape of Pareto Front. Therefore, this approach is also adopted in this work. The objective of each subproblem can be represented as follows:

$$\text{Min } g^{\text{te}}(x | \lambda, z^*) = \max_{1 \leq i \leq M} \{\lambda_i |f_i(x) - z_i^*|\}, \quad (3)$$

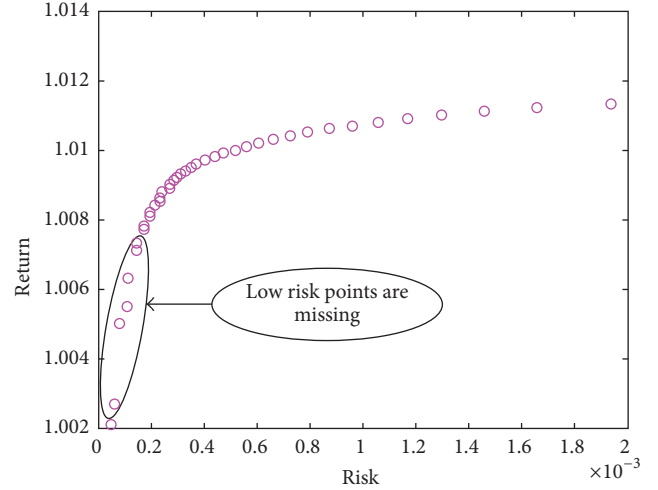


FIGURE 1: The PF obtained by MOEA/D without normalization.

where M is the number of objectives, z_i^* is the reference point for the i th objective, and λ is the weight vector. Note that the reference point can be represented as $z_i^* = \min\{f_i(x) | x \in \Omega\}$. The details of the original MOEA/D can be found in [34].

3.2.2. Normalized MOEA/D with Normalized Objectives. The MOEA/D algorithm decomposes the MOPs into a number of subproblems by a set of evenly spread weight vectors. The main idea is to narrow the gap between the objective value and the reference value in m (the number of objectives) dimensions with the assigned weight vector. In this way, MOEA/D forces the objective function to evolve in the direction of minimizing the maximum difference. However, the scales of different objectives are generally not the same for real-world applications. In such cases, most of the searching resources will be spent on the most significant objective (with largest value) while the other objectives are barely evolved. Considering portfolio optimization as an example, the Pareto Front of optimizing 100 stocks by the original MOEA/D is plotted in Figure 1. As can be seen from this plot, the solutions in the low risk area are relatively sparse. To overcome this problem, a normalization method is proposed as follows:

$$f_i^l = \frac{f_i}{f_i^{\max} - f_i^{\min}}, \quad (4)$$

where f_i , f_i^l are the original and the normalized value of the i th objective, respectively. f_i^{\max} , f_i^{\min} are the maximum and minimum value of corresponding objective function. In each generation, the objectives of all the individuals are normalized using the above formula. To demonstrate the effect of normalization, the PFs obtained by MOEA/D with and without normalization are plotted in Figure 2. It is clear that a more integrated front is obtained by using the normalization method.

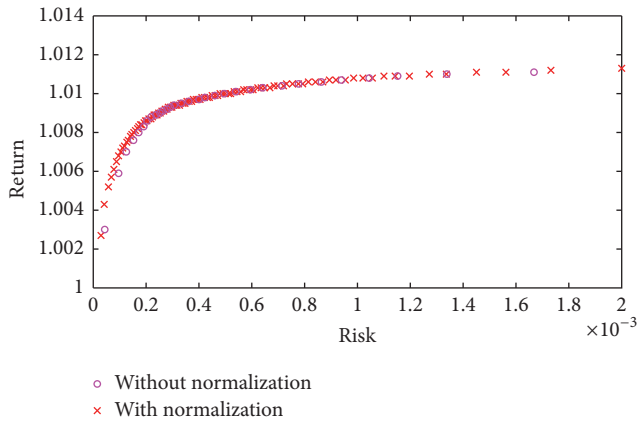


FIGURE 2: The PFs obtained by MOEA/D w/o normalization.

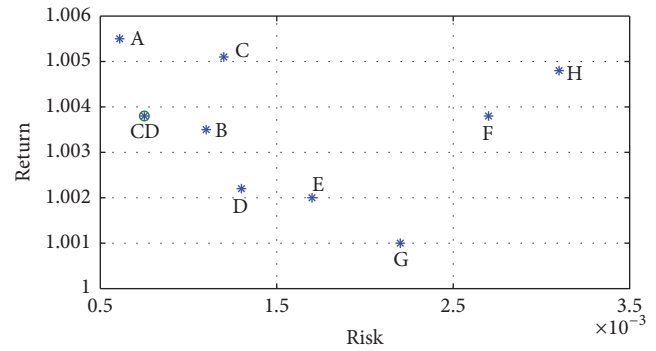


FIGURE 4: Illustration of asset preselection process 2.

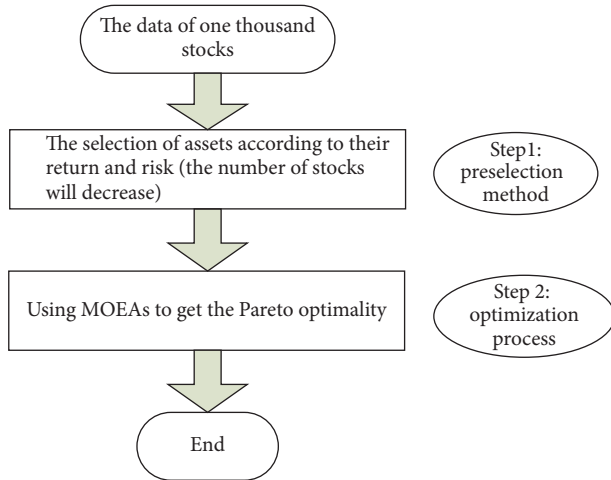


FIGURE 3: The optimization process involving 1000 stocks.

4. Large-Scale Portfolio Optimization with Preselection Methods

It is obvious that no state-of-the-art evolutionary algorithm is able to effectively solve complex multiobjective optimization problems with several thousands of variables without requiring much computational time on a standard PC. Therefore, it is important to preselect potential assets which are likely to be included in the optimized portfolios. We employ the concepts of nondominance and Pareto optimality to eliminate assets, which may not be used in the final optimal solutions. Subsequently, an MOEA is applied to the preselected assets to obtain the best approximation for the Pareto optimality. Figure 3 presents the optimization process starting with 1000 stocks. As can be seen from the figure, the number of assets is reduced within an acceptable range through Step 1.

The asset selection process is motivated by the observation that every asset has its own risk and return values. These risk and return values can be plotted and can also be subjected to nondomination sorting process. It is obvious that portfolios constructed using the assets on the efficient frontier and other frontiers just below the efficient frontier are expected to yield portfolios near the efficient frontier when the risk and return values of portfolios are subjected to nondomination sorting. Based on these observations, two assets preselection processes are proposed in this paper. The details are listed as follows:

Assets Preselection Process 1 (P-1). It is easy to understand that good single asset (high return and low risk) is likely to contribute to the final optimized portfolio. Therefore, P-1 selects the assets according to their returns and risks using nondomination sorting. In this process, each asset is considered as a portfolio and these assets are ranked using the nondomination sorting method. The selection is based on the ranks assigned to each asset until the number of selected assets reaches the predefined (user-defined) number. Although the idea of P-1 is simple, it is effective for reducing the bad assets.

Assets Preselection Process 2 (P-2). Although P-1 is simple and effective, considering only the risk and return of individual asset is insufficient when a high performance portfolio is required, because a pair of negatively correlated assets has the potential to reduce the risk depending on the degree of negative correlation. Therefore, in the asset preselection process 2 (P-2), both single assets and negatively correlated pairs of assets are considered. Figure 4 shows how the assets are selected using P-2. Imagine that we wish to select 3 assets out of the 8 assets A–H. All assets are marked according to their return and risk values. Out of the 8 assets, assume that only C and D are negatively correlated. From C and D, infinite weighted combinations can be obtained. However, there is only 1 point that has the lowest risk called CD and this point will be included in the nondomination sorting process. It is clear that point “CD” dominates asset B. Therefore, assets A, C, and D will be selected as the best 3 assets. For a pair of

assets, the combined return and risk values are computed as follows:

$$\text{Combined return} = w_1 \cdot r_1 + w_2 \cdot r_2, \quad (5)$$

$$\text{Combined risk} = \sum_{i=1}^2 \sum_{j=1}^2 \sigma_{ij} \cdot w_i \cdot w_j, \quad (6)$$

where w_1, w_2 are the fraction of the two assets held in the portfolio with $w_1 + w_2 = 1$, r_i is the return on asset i , and σ_{ij} is the covariance of returns of assets i and j . In the selection process, (6) is differentiated with respect to w_1 and set to zero. By solving the obtained equation, w_1 that gives the minimum combined risk is obtained. Substituting w_1 back to (6) will give us the minimum risk that can be obtained by that pair.

5. Constraint Handling Methods

The Markowitz (mean-variance) model initially considered only one strong constraint; that is, the summation of all asset weights must be equal to 1. This constraint is known as the real-world constraint, as the available capital must strictly meet the sum of the invested amounts in all assets which means the sum of the percentages of all the stocks should be equal to 100%. In this work, other constraints such as floor-ceiling constraints or minimum lots constraint are not considered. These constraints will be considered in our future work. To handle practical constraints and to ensure the weights to be positive (no short-sell is allowed), two different encoding schemes are adopted. The details are presented as follows.

Scheme 1 (S-1). In this encoding scheme, the weights/percentages for all the assets are firstly randomly initialized in the range of $[0, 1]$. Then all the weights are summed together to obtain a total weight. Lastly, all the initial weights are divided by the total weight to get the new weight. In this case, the summation of all the weights is always equal to 1. This scheme can be described by the following pseudocode.

The Pseudocode of S-1

Step 1. Initialize the weights $w_{1,\text{ini}}, w_{2,\text{ini}}, \dots, w_{n,\text{ini}}$ in the range of 0 to 1, where n is the number of assets in the portfolio.

Step 2. Sum all the initial weights to get the total weight. $w_{\text{total}} = w_{1,\text{ini}} + w_{2,\text{ini}} + \dots + w_{n,\text{ini}}$.

Step 3. Use the following equation to obtain the new weight and these weights meet the real-world constraint. $w_i = w_{i,\text{ini}}/w_{\text{total}}$, $i = 1, 2, \dots, n$.

Scheme 2 (S-2). This scheme is based on a variation of random keys [36]. In this scheme, all weights are randomly initialized from 0 to 0.1. All assets are sorted in the decreasing order of their weight until the summation of the included weights is greater than 1. Lastly, the weight of the last included asset will be reassigned to satisfy the unity constraint. For this scheme, the weights of many stocks are equal to 0.

6. Experiments and Results

6.1. Experimental Setup. To test the performance of the proposed methods, MATLAB R2013a was used as the programming language and simulation platform and all tests were conducted on a DELL server with Intel (R) Xeon (R) E5-2603 processor (@1.8 GHz) and 8.00 GB of memory. For all the algorithms, 100 days' closing prices (starting from 2 Jan. 2014) of 1000 stocks of Chinese stock market were downloaded and used as the historical stock data in the simulation. The population size N and the maximum number of function evaluations (*MaxFEs*) are set as 100 and 20000, respectively, for all algorithms. The other parameters are set as follows:

- (1) Normalized Multiobjective Evolutionary Algorithm based on Decomposition (NMOEA/D): The number of the weight vectors in the neighbourhood of each weigh vector is 20. The crossover rate CR is selected as 1 while the scaling factor F is 0.6. η and P_m are chosen as 20 and 1/6, respectively.
- (2) Multiobjective Differential Evolution based on Summation Sorting (MODE-SS) and Multiobjective Differential Evolution based on Nondomination Sorting (MODE-NDS) [8]: F and CR are set as 0.5 and 0.1. "DE/rand/2" is used as the mutation strategy of DE.
- (3) Multiobjective Comprehensive Learning Particle Swarm Optimizer (MO-CLPSO) [37]: P_c and P_m are set as 0.1 and 0.4, respectively.
- (4) Nondominated Sorting Genetic Algorithm II (NSGA-II) [15]: The crossover rate and mutation probabilities are selected as 0.9 and 0.1.

To evaluate the performance of the proposed algorithm and preselection methods presented in Section 3, two different encoding schemes presented in Section 4 are used, and six experiments are conducted. The details of these experiments are presented as follows. Note that all MOEAs are tested and compared on each of the six experiments and 25 independent runs are executed for each algorithm.

S-1. Experiments 1–3 are conducted using encoding Scheme 1.

Experiment 1 (S1-E1). The original 1000 stocks are used to form the portfolio and the dimension of the optimization problem is 1000 (without assets preselection process).

Experiment 2 (S1-E2P1). Optimization with 100 stocks and these 100 stocks are selected from the 1000 stocks using asset selection method 1 P-1.

Experiment 3 (S1-E3P2). Optimization with 100 stocks and these 100 stocks are selected from the 1000 stocks using asset selection method 2 P-2.

S-2. Experiments 4–6 are conducted using encoding Scheme 2.

Experiment 4 (S2-E4). Optimization with the original 1000 stocks (without assets preselection process).

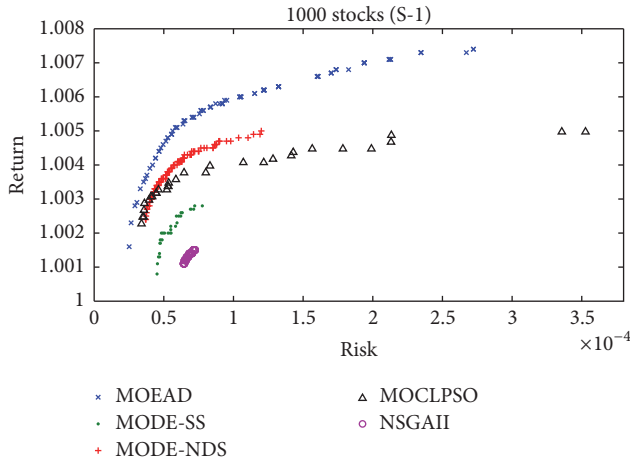


FIGURE 5: PFs of S1-E1 obtained by various algorithms (FEs = 20000).

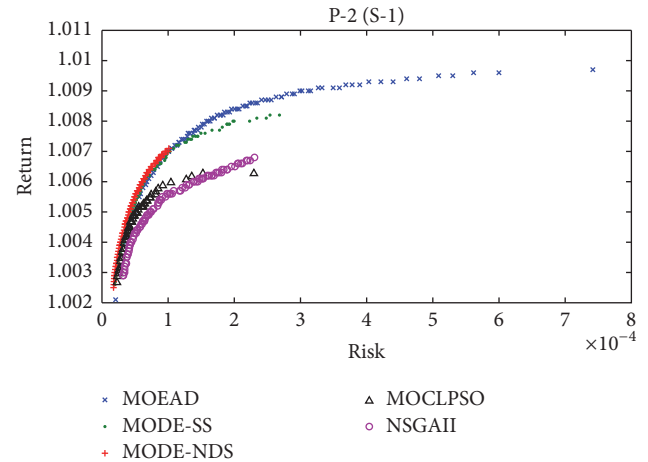


FIGURE 7: PFs of S1-E3P2 obtained by various algorithms (FEs = 20000).

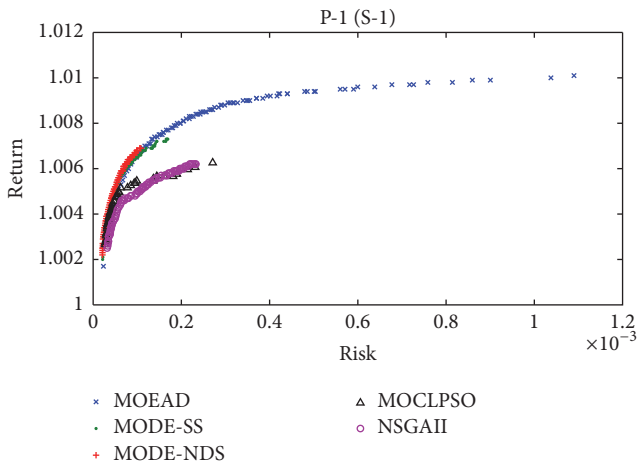


FIGURE 6: PFs of S1-E2P1 obtained by various algorithms (FEs = 20000).

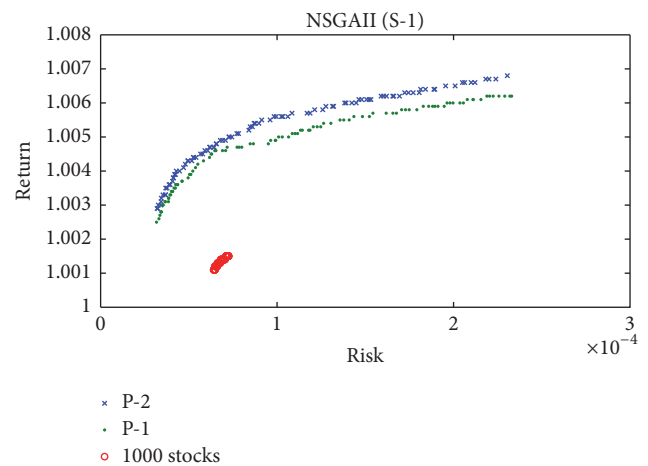


FIGURE 8: PFs obtained by NSGAI w/o preselections (FEs = 20000).

Experiment 5 (S2-E5P1). Optimization with 100 stocks and these 100 stocks are selected from the 1000 stocks using asset selection method 1 P-1.

Experiment 6 (S2-E6P2). Optimization with 100 stocks and these 100 stocks are selected from the 1000 stocks using asset selection method 2 P-2.

6.2. Experimental Results

S-1. To evaluate the performance of different algorithms with encoding Scheme 1, Experiments 1–3 were carried out. The Pareto Front of median run obtained by each algorithm is plotted in Figures 5–7. As can be seen from these plots, NMOEA/D performs the best among the algorithms for all three cases (both with and without preselection method). This is because of the high searching efficiency of NMOEA/D algorithm. The advantage of NMOEA/D is more obvious when the optimization is without the preselection method. This showed that NMOEA/D is able to solve high dimensional problem well. MODE-NDS performs a little better

than MOCLPSO and MODE-SS for S1-E1 while MODE-SS performs slightly better in S1-E2P1 and S1-E3P2. These results indicate that MODE-SS is weak in solving large-scale (1000 dimensions) problem but more efficient in solve low dimensional (100 dimensions) problems when compared with MOCLPSO and MODE-NDS. NSGAI performs the worst among the compared algorithms for all three experiments which showed that NSGAI is not as effective as others when solving multiobjective portfolio optimization problems.

To demonstrate the usefulness of the preselection processes, the return/risk trade-off curves for each algorithm without (1000 stocks) and with (100 stocks) two different preselection processes (P-1 and P-2) are plotted in Figures 8–12, respectively. As can be seen from these plots, after applying the preselection methods, the performances of all five algorithms are greatly improved. The preselection processes remove the bad stocks and keep those stocks that could potentially contribute to the portfolio. These methods not only improve the computational efficiency of the algorithms but also improve the quality of the solution. From

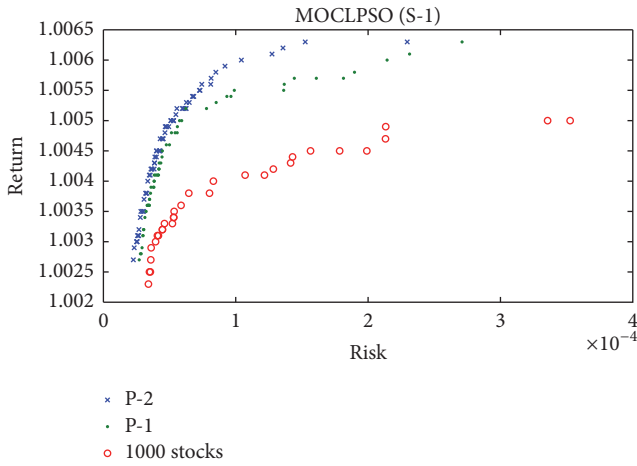


FIGURE 9: PFs obtained by MOCLPSO w/o preselections (FEs = 20000).

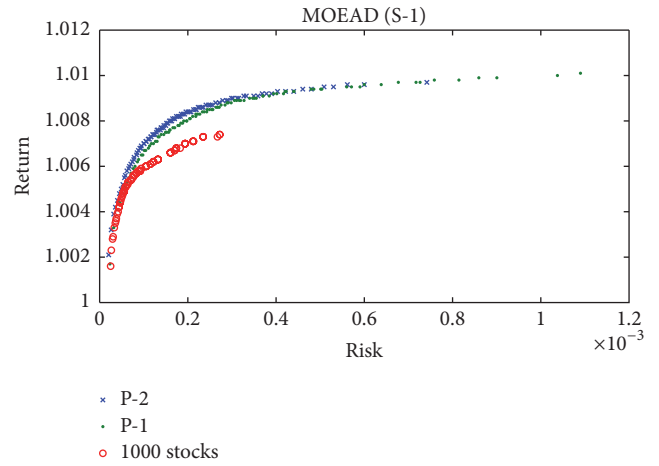


FIGURE 12: PFs obtained by MOEAD w/o preselections (FEs = 20000).

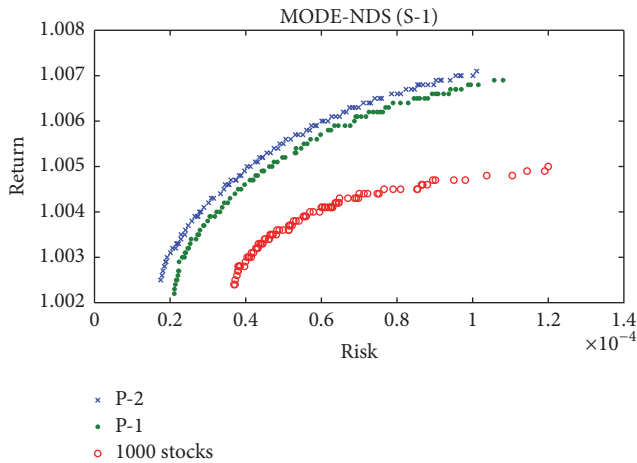


FIGURE 10: PFs obtained by MODE-NDS w/o preselections (FEs = 20000).

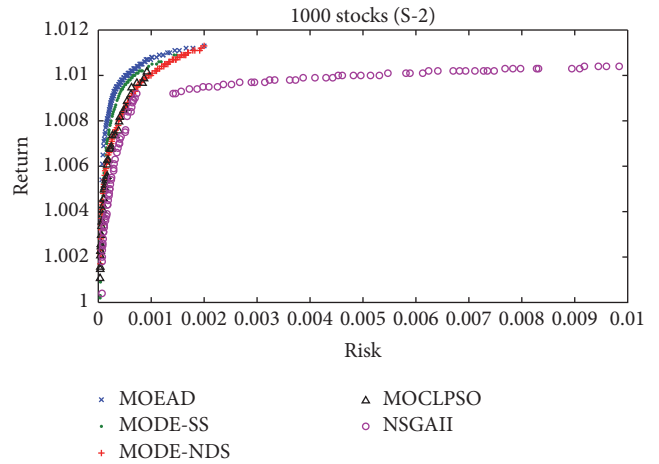


FIGURE 13: PFs of S2-E4 obtained by various algorithms (FEs = 20000).

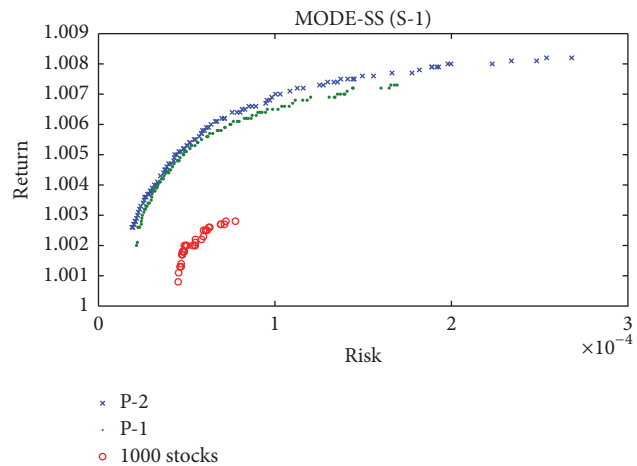


FIGURE 11: PFs obtained by MODE-SS w/o preselections (FEs = 20000).

the plots, we can also find out that preselection method 2 (P-2) generally performs better than preselection method 1 (P-1). The reason is obvious. P-2 considers both the single stocks (high return, low risk) and negatively correlated pairs while P-1 only considers single stocks. The negatively correlated pairs are able to further improve the solution quality. However, P-1 can be still useful when the requirement of accuracy is not high, as this method is faster than P-2.

S-2. In this part, the results of Experiments 4–6 are presented. Experiments 4–6 used encoding Scheme 2 and the nondominated fronts for each algorithm (median run) are presented in Figures 13–15. As can be seen from these graphs, the best results are still generated by NMOEA/D which is consistent with the results obtained in S-1. The differences among various algorithms are small compared with the results obtained in S-1, especially for S2-E5P1 and S2-E6P2. Although NSGAI performs the worst, it can still generate distributed and satisfactory fronts. Note that NMOEA/D showed good performance when optimizing the

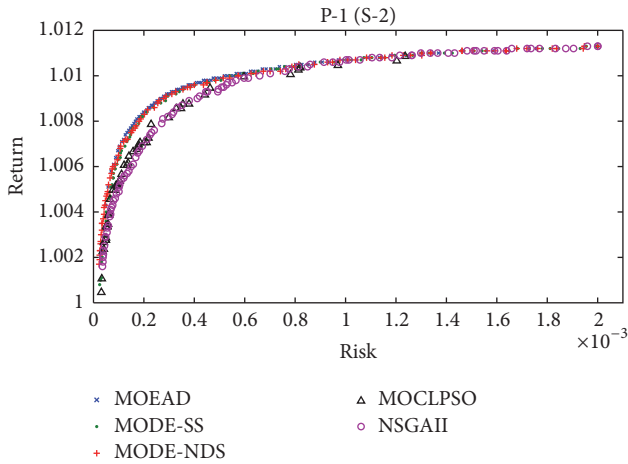


FIGURE 14: PFs of S2-E5P1 obtained by various algorithms (FEs = 20000).

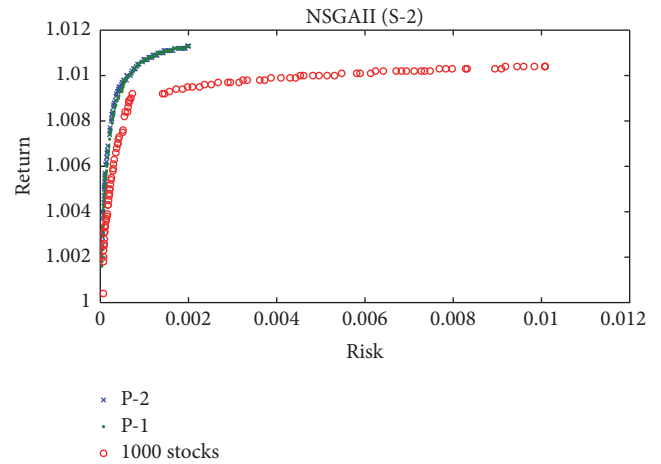


FIGURE 16: PFs obtained by NSGAI w/o preselections (FEs = 20000).

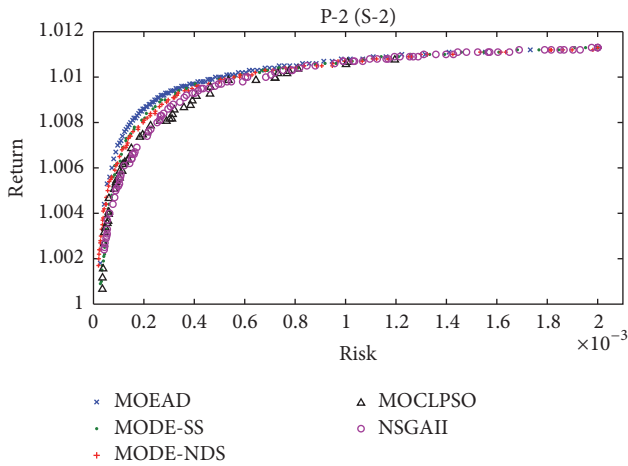


FIGURE 15: PFs of S2-E6P2 obtained by various algorithms (FEs = 20000).

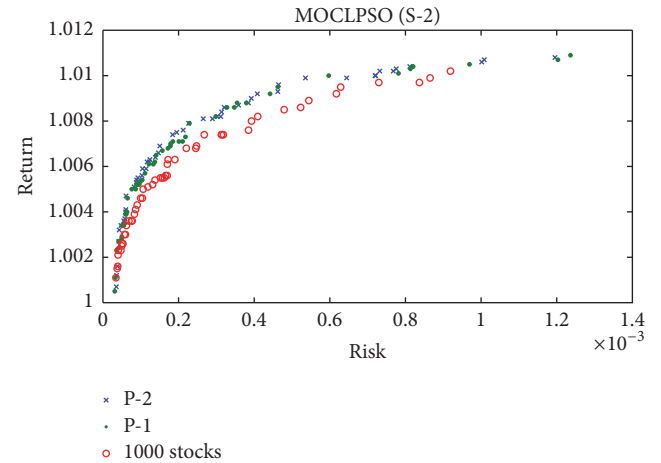


FIGURE 17: PFs obtained by MOCLPSO w/o preselections (FEs = 20000).

original 1000-stock portfolio indicating that NMOEA/D has a stronger ability in solving large-scale optimization problem. However, preselections offer more solutions at the lowest risk extreme.

Similar to the plots in S-1, the Pareto Fronts for each algorithm without (1000 stocks) and with (100 stocks) two different preselection processes (P-1 and P-2) are presented in Figures 16–20 to demonstrate the advantage of using preselection methods. The advantages of using preselection methods are quite clear for all algorithms except NMOEA/D. As shown in Figure 20, the three fronts obtained by three different methods almost coincide. Moreover, the differences between P-1 and P-2 are marginal for all algorithms. To view the difference more clearly and to further illustrate the usefulness of the preselection process, the maximum function evaluations (FEs) were reduced to 5000 for MODE-SS and NMOEA/D. The results are presented in Figures 21 and 22. As can be seen from the plot, the advantages of using preselection methods become clearer as the FEs reduce.

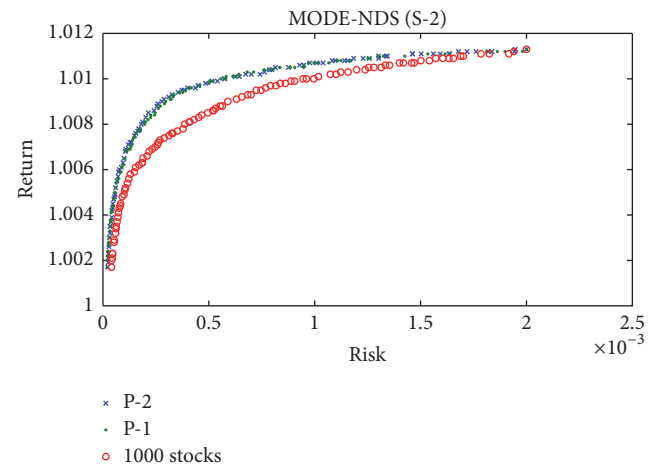


FIGURE 18: PFs obtained by MODE-NDS w/o preselections (FEs = 20000).

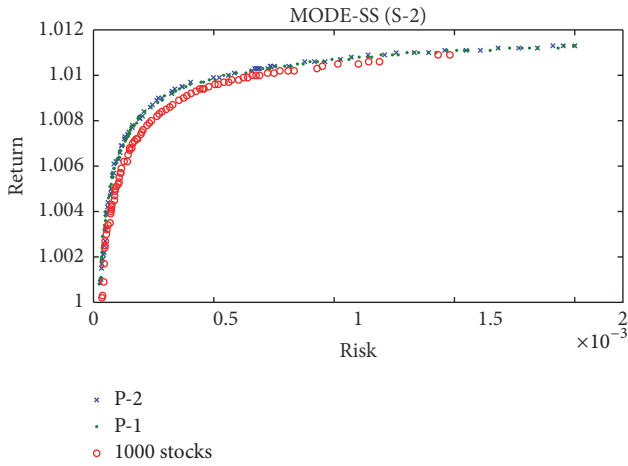


FIGURE 19: PFs obtained by MODE-SS w/o preselections (FEs = 20000).

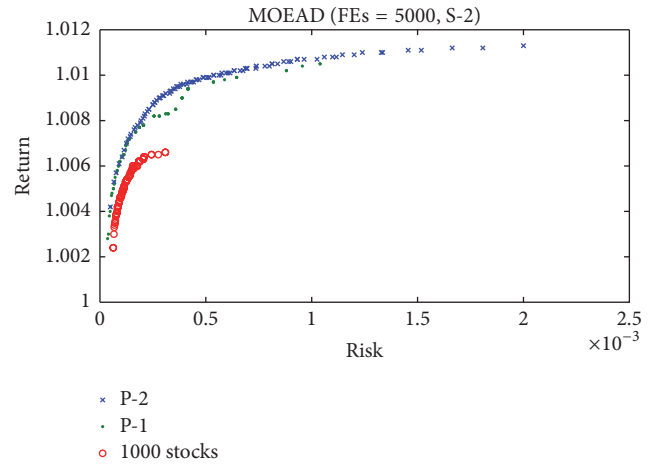


FIGURE 22: PFs obtained by NMOEAD when FEs = 5000 (FEs = 5000).

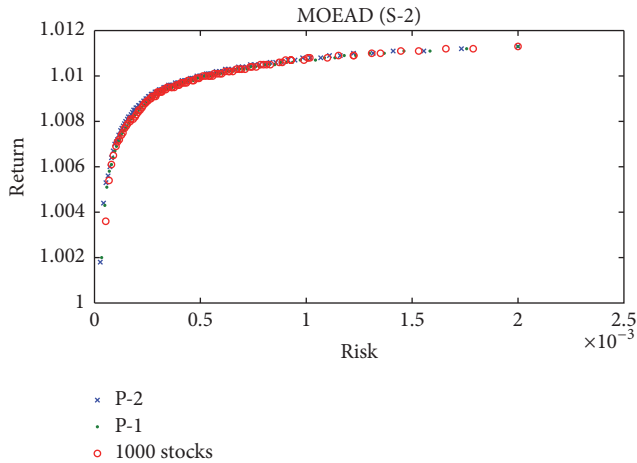


FIGURE 20: PFs obtained by MOEAD w/o preselections (FEs = 20000).

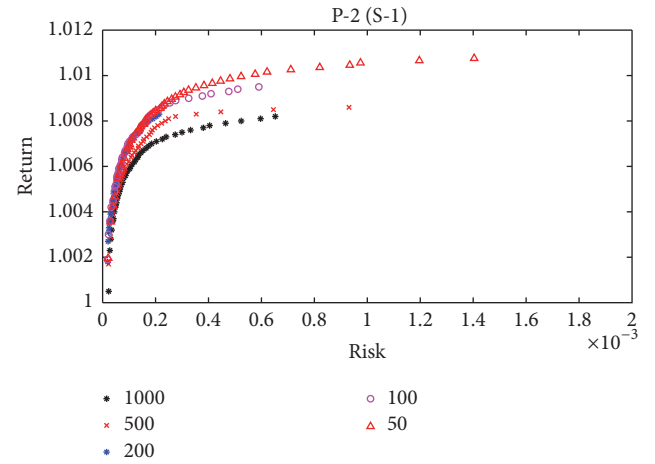


FIGURE 23: PFs obtained by using different number of stocks P-2 (S-1) (FEs = 10000).

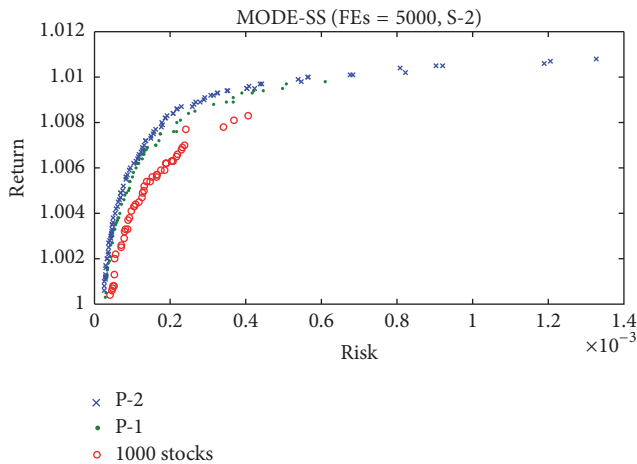


FIGURE 21: PFs obtained by MODE-SS when FEs = 5000 (FEs = 5000).

6.2.1. The Effects of Number of Stocks Selected. To compare the results of selecting different number of stocks, the following experiments are performed. Using NMOEA/D as the optimizing tool, 500, 200, 100, and 50 stocks are selected from the original 1000 stocks using preselection process P-2 and the optimized results are plotted in Figure 23 (encoding Scheme 1, S-1) and Figure 24 (encoding Scheme 2, S-2). The maximum function evaluations used are 10000. From these two graphs, it can be seen that the performance of the algorithm improves as the number of selected stocks decreases. However, if the number of stocks is further decreased to 10, the final Pareto Front becomes bad again especially for the low risk part (Figures 25 and 26 for S-1 and S-2, resp.). This observation is easy to understand. If the selected number of stocks is too small, it is difficult to cover a large number of negatively correlated assets and these assets are the main contributors of reducing the portfolio risk. Based on these observations, 50–200 is generally the desirable range for the number stocks to be preselected.

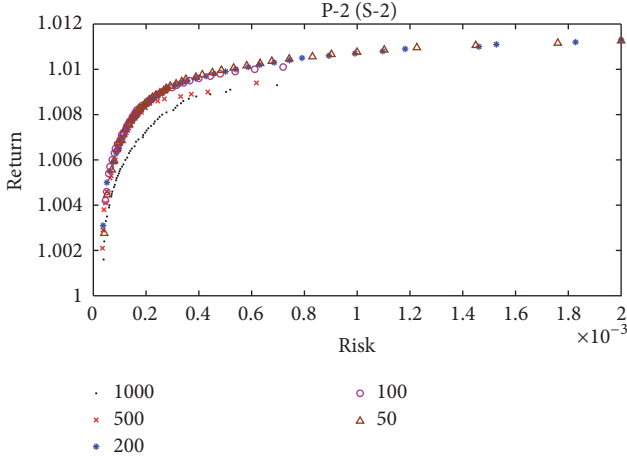


FIGURE 24: PFs obtained by using different number of stocks P-2 (S-2) (FEs = 10000).

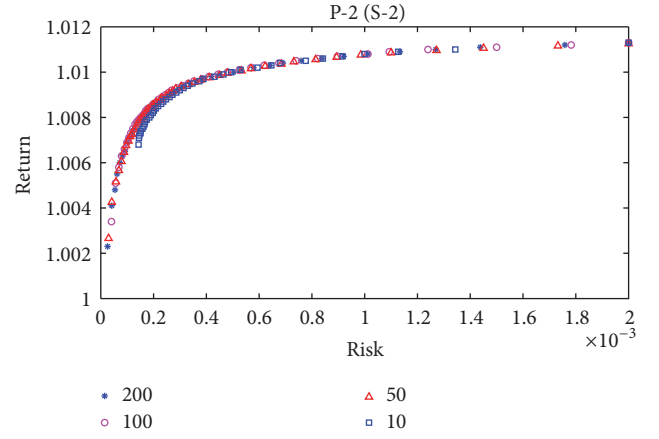


FIGURE 26: Further comparisons of different number of stocks P-2 (S-2) (FEs = 10000).

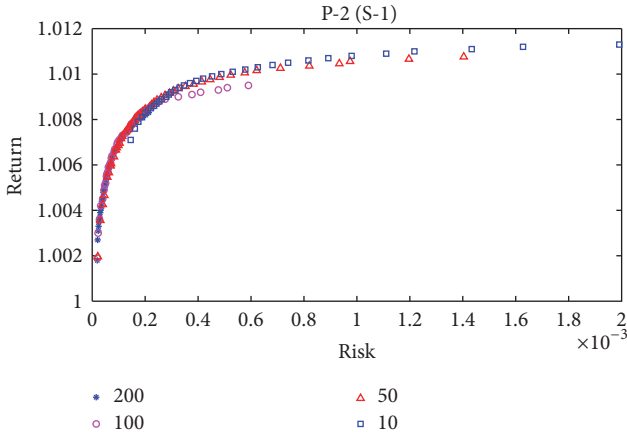


FIGURE 25: Further comparisons of different number of stocks P-2 (S-1) (FEs = 10000).

6.2.2. Numerical Analysis of the Results. To analyze the experimental results, Table 1 is presented. In this table, the best risk, best return, and the best compromise solutions of different algorithms and different methods are listed. The best compromise solution is obtained by using a fuzzy-based method which can be represented as follows [38]:

$$\mu_i = \begin{cases} 1 & \text{if } F_i \leq F_i^{\min} \\ \frac{F_i^{\max} - F_i}{F_i^{\max} - F_i^{\min}} & \text{if } F_i^{\max} < F_i \leq F_i^{\min} \\ 0 & \text{if } F_i \geq F_i^{\max}, \end{cases} \quad (7)$$

where μ_i is the membership value of the i th objective function (F_i). The normalized membership value $\mu[k]$ for each nondominated solution k is calculated using

$$\mu[k] = \frac{\sum_{i=1}^M \mu_i[k]}{\sum_{j=1}^{N_{\text{pareto}}} \sum_{i=1}^M \mu_i[j]}, \quad (8)$$

where M is the number of objectives while N_{pareto} is the number of solutions in the nondominated front. The best compromise solution is the one with the largest normalized membership value.

As can be seen from Table 1, with the preselection processes, the results (best risk, best return, and best compromise solution) obtained are either better or similar to those obtained without preselection. Moreover, the risk obtained by P-2 is always smaller than the risk obtained by P-1 when the returns of the two methods are the same. This advantage of P-2 is because of the negatively correlated pairs which can reduce the risk of the portfolio. To further demonstrate the advantage of preselection process, the simulation times (in seconds) for each algorithm with/without the preselection process are recorded in Table 2. It is clear that the simulation time is reduced greatly by using the preselection process for all the tested algorithms.

Table 3 presents the values of R indicator for each algorithm. The method of calculating R indicator can be obtained from [39]. The best results of mean values are identified by using boldface. As can be seen from this table, the best results are generated by either P-1 or P-2 (most cases generated by P-2). Moreover, t -test is applied on the R indicator. The numerical values 1 represent that the results obtained by using preselection methods are statistically superior to those obtained without using preselection method. This also demonstrated the effectiveness of the preselection processes. One observation is that the mean values obtained by other algorithms are better than that obtained by NMOEA/D for S-2. This seems to be contradictory with the results obtained from the figures. This is because NMOEA/D is able to generate better results in the 45-degree part (low risk high return), but the low risk low return part is missing and this will lead to bad indicator values. However, the investors are more interested in the low risk high return part and we can still conclude that NMOEA/D performs the best from this point of view.

TABLE 1: The numerical results of various algorithms (FEs = 20000).

Algorithms	Methods	S-1			S-2		
		Best risk	Best return	Best compromise solution	Best risk	Best return	Best compromise solution
NSGAI	1000 stocks						
	Risk	$5.4190e - 05$	$8.8293e - 05$	$5.6357e - 05$	$3.3399e - 05$	$1.9601e - 03$	$6.0823e - 04$
	Return	1.0007	1.0021	1.0014	1.0002	1.0113	1.0098
	P-1 (200 stocks)						
	Risk	$2.4064e - 05$	$7.4367e - 04$	$1.4351e - 04$	$2.1958e - 05$	$1.9345e - 03$	$3.2171e - 04$
	Return	1.0017	1.0082	1.0061	1.0010	1.0113	1.0091
	P-2 (200 stocks)						
	Risk	$2.1818e - 05$	$8.2492e - 04$	$1.2878e - 04$	$1.9771e - 05$	$1.9064e - 03$	$4.0098e - 04$
Return	1.0022	1.0087	1.0063	1.0011	1.0113	1.0095	
MODE-SS	1000 stocks						
	Risk	$4.0824e - 05$	$1.3633e - 04$	$7.7637e - 05$	$2.3137e - 05$	$1.9303e - 03$	$3.0775e - 04$
	Return	1.0012	1.0037	1.0031	1.0004	1.0113	1.0093
	P-1 (200 stocks)						
	Risk	$1.9843e - 05$	$5.2692e - 04$	$1.1955e - 04$	$2.2217e - 05$	$1.9399e - 03$	$3.3046e - 04$
	Return	1.0018	1.0085	1.0071	1.0011	1.0113	1.0094
	P-2 (200 stocks)						
	Risk	$1.7240e - 05$	$6.4681e - 04$	$1.4896e - 04$	$2.0021e - 05$	$1.9092e - 03$	$2.7307e - 04$
Return	1.0024	1.0091	1.0077	1.0006	1.0113	1.0092	
MODE-NDS	1000 stocks						
	Risk	$2.7804e - 05$	$7.9330e - 04$	$1.6792e - 04$	$2.3248e - 05$	$1.9198e - 03$	$3.7301e - 04$
	Return	1.0016	1.0067	1.0060	1.0004	1.0113	1.0092
	P-1 (200 stocks)						
	Risk	$1.8281e - 05$	$2.0738e - 04$	$6.6082e - 05$	$1.9139e - 05$	$1.9423e - 03$	$3.1554e - 04$
	Return	1.0017	1.0082	1.0060	1.0013	1.0113	1.0093
	P-2 (200 stocks)						
	Risk	$1.4330e - 05$	$1.8111e - 04$	$6.4452e - 05$	$1.7303e - 05$	$1.9307e - 03$	$3.2112e - 04$
Return	1.0018	1.0081	1.0062	1.0011	1.0113	1.0094	
MOCLPSO	1000 stocks						
	Risk	$2.2991e - 05$	$2.8116e - 04$	$6.3625e - 05$	$2.6386e - 05$	$2.0001e - 03$	$5.0291e - 04$
	Return	1.0016	1.0059	1.0042	1.0010	1.0113	1.0096
	P-1 (200 stocks)						
	Risk	$2.3381e - 05$	$3.5102e - 04$	$8.6957e - 05$	$2.8300e - 05$	$2.0001e - 03$	$4.5441e - 04$
	Return	1.0025	1.0071	1.0059	1.0003	1.0113	1.0096
	P-2 (200 stocks)						
	Risk	$2.2449e - 05$	$3.1597e - 04$	$8.2379e - 05$	$2.8384e - 05$	$2.0001e - 03$	$2.8508e - 04$
Return	1.0025	1.0070	1.0060	1.0009	1.0113	1.0089	

TABLE 1: Continued.

Algorithms	Methods	S-1			S-2		
		Best risk	Best return	Best compromise solution	Best risk	Best return	Best compromise solution
NMOEA/D	1000 stocks						
	Risk	$1.9760e-05$	$1.4485e-03$	$1.9534e-04$	$2.5375e-05$	$2.0001e-03$	$2.9807e-04$
	Return	1.0012	1.0098	1.0075	1.0017	1.0113	1.0093
	P-1 (200 stocks)						
	Risk	$2.0411e-05$	$1.0229e-03$	$2.4117e-04$	$2.4623e-05$	$1.9998e-03$	$3.4813e-04$
	Return	1.0018	1.0105	1.0087	1.0022	1.0113	1.0095
	P-2 (200 stocks)						
	Risk	$1.5985e-05$	$1.2743e-03$	$2.4025e-04$	$2.2294e-05$	$1.9998e-03$	$3.0229e-04$
Return	1.0015	1.0106	1.0089	1.0010	1.0113	1.0094	

TABLE 2: The simulation time of various algorithms (FEs = 20000).

Algorithms	Methods	S-1	S-2
NSGAI	1000	176.53	175.22
	P-1 (200 stocks)	18.51	18.66
	P-2 (200 stocks)	18.35	18.82
MODE-NDS	1000	178.08	175.86
	P-1 (200 stocks)	18.92	17.54
	P-2 (200 stocks)	18.78	17.72
MOCLPSO	1000	183.96	180.25
	P-1 (200 stocks)	22.05	21.79
	P-2 (200 stocks)	22.19	21.93
MODE-SS	1000	173.58	174.82
	P-1 (200 stocks)	9.33	10.29
	P-2 (200 stocks)	9.32	10.26
NMOEA/D	1000	175.71	175.39
	P-1 (200 stocks)	12.60	13.72
	P-2 (200 stocks)	12.56	13.90

7. Conclusion

In this paper, an efficient procedure for solving large-scale portfolio optimization problem was introduced. The key features of this method are the asset preselection processes. Two different asset preselection processes were proposed. Large-scale optimization requires long computational time and large memory to compute the optimal portfolios. However, with the asset preselection process, the assets that may not be potentially included in the final portfolio are removed thereby reducing the computational time substantially while improving the quality of the generated efficient portfolios within the same number of function evaluations. Our extensive results showed that the quality of the generated efficient portfolios improved significantly due to the asset preselection procedures, especially for the selection process 2 which

considers both single asset and negatively correlated pairs. Moreover, the Normalized MOEA/D algorithm performed the best due to its high searching efficiency. In future, we are planning to include various real-world constraints such as the floor-ceiling constraint and cardinality constraint in the optimization process. These constraints will introduce additional difficulties for the optimization algorithms and constraint handling methods thereby further highlighting the importance of asset preselection process. With these constraints, the problem formulation will be readily applicable to real-world situations and the results obtained can be attractive to the investors.

Competing Interests

The authors declare that they have no competing interests.

TABLE 3: R indicators of various algorithms (FEs = 20000).

Algorithms	S-1			S-2		
	1000 stocks	P-1 (200 stocks)	P-2 (200 stocks)	1000 stocks	P-1 (200 stocks)	P-2 (200 stocks)
NSGAI						
Best	0.0045	6.23E-04	5.00E-04	1.75E-04	9.23E-05	9.45E-05
Worst	0.0054	0.0021	0.0017	9.13E-04	2.06E-04	1.83E-04
Mean	0.0051	0.0014	0.0012	5.13E-04	1.47E-04	1.24E-04
Std	2.71E-04	3.81E-04	2.95E-04	2.35E-04	2.19E-05	3.71E-05
t -test	—	1	1	—	1	1
MODE-SS						
Best	0.0031	4.51E-04	2.85E-04	7.46E-05	6.49E-05	5.34E-05
Worst	0.0044	0.0011	9.29E-04	5.21E-04	1.38E-04	8.45E-05
Mean	0.0039	8.20E-04	6.05E-04	2.23E-04	7.73E-05	6.60E-05
Std	3.26E-04	1.62E-04	2.11E-04	1.48E-04	1.39E-05	8.27E-06
t -test	—	1	1	—	1	1
MODE-NDS						
Best	0.0012	5.65E-04	5.94E-04	8.52E-05	6.36E-05	5.39E-05
Worst	0.003	0.0014	0.0014	8.48E-04	2.27E-04	2.55E-04
Mean	0.0021	0.001	9.47E-04	3.24E-04	8.59E-05	8.43E-05
Std	5.34E-04	2.11E-04	1.94E-04	2.17E-04	3.09E-05	5.77E-05
t -test	—	1	1	—	1	1
MOCLPSO						
Best	0.0016	0.001	0.0011	1.53E-04	1.25E-04	1.06E-04
Worst	0.0028	0.0018	0.0018	0.0011	2.22E-04	1.71E-04
Mean	0.0022	0.0014	0.0014	3.99E-04	1.47E-04	1.36E-04
Std	3.04E-04	2.29E-04	1.97E-04	2.37E-04	2.08E-05	1.63E-05
t -test	—	1	1	—	1	1
NMOEA/D						
Best	3.31E-04	1.30E-04	8.84E-05	1.69E-04	1.72E-04	1.60E-04
Worst	0.0012	2.96E-04	3.39E-04	0.001	2.73E-04	3.56E-04
Mean	8.69E-04	2.17E-04	1.89E-04	3.91E-04	2.07E-04	2.16E-04
Std	2.10E-04	4.78E-05	1.02E-04	2.37E-04	2.60E-05	8.16E-05
t -test	—	1	1	—	1	1

Acknowledgments

This research is partially supported by National Natural Science Foundation of China (61673404, 61305080, 61473266, and 61379113) and Postdoctoral Science Foundation of China (2014M552013) and the Scientific and Technological Project of Henan Province (132102210521, 152102210153). Dr. P. N. Suganthan acknowledges support offered by C4T Cambridge Create Program.

References

- [1] D. W. Blackwell, M. D. Griffiths, and D. B. Winters, *Modern Financial Markets*, John Wiley & Sons, Hoboken, NJ, USA, 2007.
- [2] L. H. Liu and F. A. Deng, "The stock investment strategy based on Vague Sets," *Science Technology and Engineering*, vol. 12, pp. 4584–4587, 2012.
- [3] M. Guidolin and A. Timmermann, "Asset allocation under multivariate regime switching," *Journal of Economic Dynamics and Control*, vol. 31, no. 11, pp. 3503–3544, 2007.
- [4] G. Chow, "Portfolio selection based on return, risk, and relative performance," *Financial Analysts Journal*, vol. 51, no. 2, pp. 54–60, 1995.
- [5] E. F. Brigham and J. F. Houston, *Fundamentals of Financial Management*, Thomson/South-Western, Mason, Ohio, USA, 11th edition, 2007.
- [6] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [7] P. N. Kolm, R. Tütüncü, and F. J. Fabozzi, "60 Years of portfolio optimization: practical challenges and current trends," *European Journal of Operational Research*, vol. 234, no. 2, pp. 356–371, 2014.
- [8] B. Y. Qu and P. N. Suganthan, "Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection," *Information Sciences*, vol. 180, no. 17, pp. 3170–3181, 2010.

- [9] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences*, vol. 197, no. 15, pp. 131–143, 2012.
- [10] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.
- [11] A. Loraschi, A. Tettamanzi, M. Tomassini, and P. Verda, "Distributed genetic algorithms with an application to portfolio selection problems," in *Proceedings of the Congress on Artificial Neural Networks and Genetic Algorithms, Alès, France*, pp. 384–387, Springer, Heidelberg, Germany, 1995.
- [12] A. Loraschi and A. Tettamanzi, "An evolutionary algorithm for portfolio selection in a downside risk framework," *The European Journal of Finance*, vol. 6, pp. 8–12, 1995.
- [13] S. Arnone, A. Loraschi, and A. Tettamanzi, "A genetic approach to portfolio selection," *Neural Network World*, vol. 6, pp. 597–604, 1993.
- [14] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, West Sussex, UK, 2001.
- [15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [16] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [17] S. Arnone, A. Loraschi, and A. Tettamanzi, "A genetic approach to portfolio selection," *Neural Network World*, vol. 3, no. 6, pp. 597–604, 1993.
- [18] J. Shoaf and J. Foster, "A genetic algorithm solution to the efficient set problem: a technique for portfolio selection based on the Markowitz model," in *Proceedings of the 27th Annual Meeting of the Decision Sciences Institute*, pp. 571–573, November 1996.
- [19] P. Skolpadungket, K. Dahal, and N. Harnpornchai, "Portfolio optimization using multi-objective genetic algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 516–523, Singapore, September 2007.
- [20] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.
- [21] C. Fonseca and P. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 416–423, June 1993.
- [22] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," in *Proceedings of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN '02)*, pp. 95–100, 2002.
- [23] W. Yan, R. Miao, and S. Li, "Multi-period semi-variance portfolio selection: model and numerical solution," *Applied Mathematics and Computation*, vol. 194, no. 1, pp. 128–134, 2007.
- [24] C.-C. Lin and Y.-T. Liu, "Genetic algorithms for portfolio selection problems with minimum transaction lots," *European Journal of Operational Research*, vol. 185, no. 1, pp. 393–404, 2008.
- [25] A. Ghahtarani and A. A. Najafi, "Robust goal programming for multi-objective portfolio selection problem," *Economic Modelling*, vol. 33, pp. 588–592, 2013.
- [26] A. Fernández and S. Gómez, "Portfolio selection using neural networks," *Computers & Operations Research*, vol. 34, no. 4, pp. 1177–1191, 2007.
- [27] H. Markowitz, W. F. Sharpe, and M. Miller, *Founders of Modern Finance: Their Prize Winning Concepts and 1990 Nobel Lectures*, CFA Institute, Charlottesville, Va, USA, 1991.
- [28] W. F. Sharpe, *Portfolio theory and Capital Markets*, McGraw-Hill, New York, NY, USA, 1970.
- [29] J. S. Pang, "A new and efficient algorithm for a class of portfolio selection problems," *Operations Research*, vol. 28, no. 3, part 2, pp. 754–767, 1980.
- [30] M. J. Best and J. Hlouskova, "The efficient frontier for bounded assets," *Mathematical Methods of Operations Research*, vol. 52, no. 2, pp. 195–212, 2000.
- [31] M. G. Speranza, "A heuristic algorithm for a portfolio optimization model applied to the Milan stock market," *Computers & Operations Research*, vol. 23, no. 5, pp. 433–441, 1996.
- [32] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for cardinality constrained portfolio optimisation," *Computers and Operations Research*, vol. 27, no. 13, pp. 1271–1302, 2000.
- [33] L. Zhang, C. Zhou, M. Ma, and C. Sun, "Multi-objective differential evolution algorithm based on max-min distance density," *Computer Research and Development*, vol. 44, no. 1, pp. 177–184, 2007.
- [34] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [35] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 203–208, Trondheim, Norway, May 2009.
- [36] G. Vedarajan, L. C. Chan, and D. Goldberg, "Investment portfolio optimization using genetic algorithms," in *Proceedings of the Late Breaking Papers at the Genetic Programming Conference*, pp. 255–263, 1997.
- [37] V. L. Huang, P. N. Suganthan, and J. J. Liang, "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems," *International Journal of Intelligent Systems*, vol. 21, no. 2, pp. 209–226, 2006.
- [38] M. A. Abido, "Environmental/economic power dispatch using multiobjective evolutionary algorithms," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1529–1537, 2003.
- [39] Q. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Tech. Rep. CES-887, University of Essex and Nanyang Technological University, 2008.