

The mean-variance cardinality constrained portfolio optimization problem using a local search-based multi-objective evolutionary algorithm

Bili Chen¹ · Yangbin Lin¹ · Wenhua Zeng² · Hang Xu² · Defu Zhang³

Published online: 10 April 2017
© Springer Science+Business Media New York 2017

Abstract Portfolio optimization problem is an important research topic in finance. The standard model of this problem, called Markowitz mean-variance model, has two conflicting criteria: expected returns and risks. In this paper, we consider a more realistic portfolio optimization problem, including both cardinality and quantity constraints, which is called Markowitz mean-variance cardinality constrained portfolio optimization problem (MVCCPO problem). We extend an algorithm which is based on a multi-objective evolutionary framework incorporating a local search schema and non-dominated sorting. To quantitatively analyze the effectiveness of the proposed algorithm, we compared our algorithm with the other five algorithms on public available data sets involving up to 225 assets. Several modifications based on the fundamental operators and procedures of the algorithm, namely, the boundary constraint handling strategy, the local search schema, the replacement strategy and the farthest-candidate approach, are proposed one-by-one. Success of this exercise is displayed via simulation results. The experimental results with different cardinality constraints illustrate that the proposed algorithm

outperforms the other algorithms in terms of proximity and diversity. In addition, the diversity maintenance strategy used in the algorithm is also studied in terms of a spread metric to evaluate the distribution of the obtained non-dominated solutions. The sensitivity of our algorithm has also been experimentally investigated in this paper.

Keywords Portfolio optimization · Multi-objective evolutionary framework · Local search · Non-dominated sorting

1 Introduction

The problem of portfolio optimization is an important research topic in finance, which has attracted much attention in recent years. Markowitz [1, 2] proposed a famous model called mean-variance (MV) model to solve this problem. This model has two conflicting criteria: expected returns and risks. The target of the problem is to assign a given capital to different available assets for the purpose of minimizing the risk and maximizing the expected return. Because of the two conflicting objectives, there is a set of optimal solutions which is named efficient portfolios rather than the single optimal portfolio in the MV model. In addition to these two objectives, constraints are always imposed on the model for more realistic management decisions, such as cardinality constraint, i.e., the size of holding assets is limited, and quantity constraint, i.e., a lower and an upper proportion of a particular asset is constrained [3]. MV model with cardinality and quantity constraints for portfolio optimization is usually called MVCCPO problem (Mean-Variance Cardinality Constrained Portfolio Optimization problem), which has aroused the interest of many researchers to design various algorithms. These algorithms are usually divided into

✉ Yangbin Lin
yblin@jmu.edu.cn

¹ Computer Engineering College, Jimei University, Xiamen 361021, China

² School of Software, Xiamen University, Xiamen 361005, China

³ School of Information Science and Engineering, Xiamen University, Xiamen 361005, China

two categories: exact algorithms and heuristic algorithms. Next, we will give a brief introduction about them.

1.1 Exact algorithms

Bienstock [4] proposed a branch-and-cut algorithm to find the exact solutions of the cardinality constrained portfolio optimization. Jobst et al. [5] studied the effects of the portfolio selection problem with buy-in thresholds, cardinality constraints and transaction round lot restrictions. Then, they adopted a branch-and-bound algorithm to solve them. Li et al. [6] presented a convergent Lagrangian and contour-domain cut method to get an optimal solution for the MVCCPO problem. Shaw et al. [7] proposed a Lagrangian relaxation procedure which is able to take advantage of the special structure of the objective function to solve the the MVCCPO problem. Vielma et al. [8] developed a linear programming based branch-and-bound algorithm for the exact solution of a portfolio optimization problem. Bertsimas and Shioda [9] proposed branch-and-bound based algorithms which use the special structure for the exact solution of a cardinality-constrained quadratic optimization problems. The computational results have indicated that these algorithms have practical advantages for some special problems. Gulpinar et al. [10] introduced a local deterministic optimization method which is based on the difference of convex function programming and developed a difference of convex algorithm to find the exact solution of the MVCCPO problem. Computational results have indicated that the algorithm is superior to the standard approach on historical data. The disadvantage of the exact algorithm is that it always has a high computation time and can only find an optimal solution within a specified time.

1.2 Heuristic algorithms

The heuristic algorithms try to find the optimal solutions in a reasonable time, but are not guaranteed to get the optimal solutions. There are many heuristic algorithms including: Evolutionary Algorithm (EA), tabu search, Differential Evolution (DE), Simulated Annealing (SA), and Particle Swarm Optimization (PSO), etc. Many researchers devoted themselves to improve these algorithms. For example, Padhye and Deb improve the DE through a unified approach in [11] and the PSO by an algorithmic link with genetic algorithms in [12]. Some heuristic algorithms called Multi-objective evolutionary algorithms (MOEAs) are designed for solving Multi-objective Optimization Problems (MOPs), namely, Non-dominated Sorting Genetic Algorithm II (NSGA-II) [13], Strength Pareto evolutionary algorithm 2 (SPEA2) [14], Non-dominated sorting and Local search based multi-objective evolutionary algorithm (NSLS) [15], Decomposition based MOEA (MOEA/D)

[16], Decomposition based MOEA with DE (MOEA/D-DE) [17], MOEA/D with a two-phase strategy and a niche-guided schema (MOEA-TPN) [18], and ND/DPP (a dual-population based algorithm which combines NSGA-II and MOEA/D-DE) [19]. Furthermore, researchers have proposed several novel general methods such as efficient non-dominated sort (ENS) [20], adaptive hybrid crossover operator (AHX) [21], and adaptive memetic computing [22] to improve these MOEAs. Chang et al. [3] employed EA, tabu search and SA to solve the MVCCPO problem. And a number of publications follows the work of Chang et al. [3]. Schaerf [23] adopted local search, tabu search and SA for the MVCCPO problem. Crama and Schyns [24] proposed an SA method for the mixed integer quadratic programming problem which is a MV model with additional realistic constraints. Derigs and Nickel [25] presented an SA based decision support system generator and a tracking error model with data stemming for measuring the performance of the portfolio. They continue the work, which can be found in Derigs and Nickel [26]. Ehrgott et al. [27] proposed a method based on multi-criteria decision making and utilized a customized local search, SA, tabu search and genetic algorithm heuristics to solve four test problems. Streichert and Tanaka-Yamawaki [28] adopted a hybrid multi-objective evolutionary algorithm combined with a local search schema on the constrained and unconstrained portfolio selection problem. Fernandez and Gomez [29] utilized an artificial neural networks based heuristic algorithm to find the efficient frontier of the portfolio selection problem and compare it with the previous heuristic algorithms. Chiam et al. [30] proposed a multi-objective based approach to find the efficient frontier of the classical MV model. Branke et al. [31] combined an active set algorithm with a MOEA which is called envelope-based multi-objective evolutionary algorithm and the computational results illustrate that the algorithm is superior to the existing MOEAs. Cura [32] presented a heuristic algorithm based on PSO to solve MVCCPO problem and compared it with the genetic algorithms, SA and tabu search. Experimental results have shown that the PSO based approach outperforms the other methods. Pai and Michel [33] adopted k-means clustering to eliminate the cardinality constraints and utilized a variant version of $(\mu+\lambda)$ evolution strategy to make the reproduction process converge rapidly. Soleimani et al. [34] used a corresponding genetic algorithm to solve the mixed-integer nonlinear programming on some test problems involving up to 2000 assets. Chang et al. [35] presented a heuristic method based on genetic algorithm for the portfolio optimization problems with three different risk measures and they have demonstrated the results on three test problems, involving up to 99 assets. Anagnostopoulos and Mamanis [36] considered the portfolio selection as a tri-objective optimization problem to find a

tradeoff between risk, return and the number of assets. They also considered the quantity and class constraints and used three wellknown MOEAs: NSGA-II [13], Pareto Envelope-based Selection Algorithm (PESA) [37] and SPEA2 [14] to solve the problem. Furthermore, they have compared five state-of-the-art MOEAs on the MVCCPO problem in [38]. They are Niche Pareto genetic algorithm 2 (NPGA2) [39], NSGA-II, PESA, SPEA2 [14], and e-multi-objective evolutionary algorithm (e-MOEA) [40], which are performed on the public benchmark data sets. Computational results have illustrated a superiority of SPEA2 and NSGA-II in the five MOEAs. Ruiz-Torrobiano and Suarez [41] presented a hybridization of evolutionary algorithms, quadratic programming and devised pruning heuristics for the MVC-CPO problem. Woodside-Oriakhi et al. [42] studied the application of genetic algorithm, tabu search and SA based meta-heuristic algorithms to find the cardinality constrained efficient frontier (CCFEF) of the MV model. They reported the computational results in terms of the quality of solution and the computation time on the publicly available data sets involving up to 1318 assets. Lwin and Qu [43] presented a hybrid algorithm which uses an elitist strategy and a partially guided mutation to improve the solution. Murray and Shek [44] suggested a local relaxation approach which is also suitable for large scale MVCCPO problem. Deng et al. [45] considered the MVCCPO problem as a single-objective problem and proposed an improved PSO to solve it. Liagkouras and Metaxiotis [46] proposed a new Probe Guided Mutation (PGM) operator, which is used in conjunction with MOEAs to solve the MVCCPO problem. Tuba and Bacanin [47] incorporated the Firefly Algorithm (FA) into the Artificial Bee Colony (ABC) algorithm to solve the MVCCPO problem. The search method from FA improves exploitation and convergence speed of the original ABC algorithm. The algorithm was compared with genetic algorithms, SA, tabu search and PSO. Experimental results show that the algorithm is better than the other algorithms. Cui et al. [48] hybridized PSO and a mathematical programming method to solve the MVCCPO problem. Experimental results show that the algorithm is useful. Baykasoğlu et al. [49] developed a greedy randomized adaptive search procedure (GRASP) in the asset selection level to choose exactly number of assets and used the quadratic programming model to deal with the other problem constraints. The computational results demonstrate that the proposed algorithm is competitive with the state of the art algorithms. Ruiz-Torrobiano and Suárez [50] consider the MVCCPO problem with transaction costs, which is an extension of the MV model. Because of transaction costs, the authors used a “rolling window” procedure to evaluate the out-of-sample performance of the portfolios. The authors presented a memetic algorithm combined with a genetic algorithm and quadratic programming. The method is tested on the

publicly available data with a various range of investment strategies and the results show that their algorithm can improve the performance of the portfolio model with transaction costs. For reason of space, the reader is referred to Metaxiotis and Liagkouras [51] for more information about the literature survey on the portfolio selection problem.

Recently, Chen et al. [15] proposed a new Non-dominated sorting and Local search based multi-objective evolutionary framework (NSLS) to solve MOPs. However, NSLS is only used to deal with some benchmark test problems with the MOPs without constraints. In this paper, we propose a method called *e*-NSLS which is an extension of NSLS for the purpose of effectively solving the MVCCPO problem with both cardinality and quantity constraints. Moreover, we tested *e*-NSLS on five public available benchmarks involving up to 225 assets in comparison with the state-of-the-art algorithms.

The rest of this paper is organized as follows: Section 2 details the MVCCPO problem. Section 3 briefly introduces three contrast algorithms. Details of the proposed algorithm for solving the MVCCPO problem are presented in Section 4. Section 5 provides the numerical comparison results. And Section 6 concludes.

2 Portfolio optimization

The MVCCPO problem can be formalized by:

$$\min \quad \rho(x) = \sum_{i=1}^n \sum_{j=1}^n x_i x_j r_{ij} \tag{1}$$

$$= \sum_{i=1}^n \sum_{j=1}^n x_i x_j c_{ij} \sigma_i \sigma_j$$

$$\max \quad \mu(x) = \sum_{i=1}^n x_i \mu_i \tag{2}$$

s.t.

$$\sum_{i=1}^n x_i = 1 \tag{3}$$

$$\sum_{i=1}^n \theta_i \leq K \tag{4}$$

$$l_i \theta_i \leq x_i \leq u_i \theta_i \quad i = 1, 2, \dots, n \tag{5}$$

$$\theta_i \in \{0, 1\}, \quad i = 1, 2, \dots, n \tag{6}$$

where n is the number of assets available in the portfolio optimization; x_i is the investment proportion of the i th asset; r_{ij} is the covariance between asset i and j ; c_{ij} is the correlation between the returns of assets i and j ; σ_i and σ_j are the standard deviation of the return of assets i and j , respectively; μ_i is the expected return of the i th asset. $\rho(x)$ is the variance of portfolio x ; $\mu(x)$ is the expected return of portfolio x ; θ_i is a binary variable which specifies whether asset i is held in the portfolio or not. If asset i is held, then $\theta_i = 1$,

otherwise, $\theta_i = 0$; K is the maximum number of assets allowed in the portfolio; l_i and u_i are the lower and upper proportion of the total investment of asset i , respectively.

Equation (1) minimizes the risk (variance) associated with the covariance matrix and the selected portfolio. Equation (2) maximizes the expected return. Equations (3), (4), (5), and (6) are the constraints. Equation (3) ensures that the sum of investment proportions of each asset is equal to one. Equation (4) is the cardinality constraint ensuring that there are no more than K assets are held. Equation (5) ensures that the investment proportion of each asset is bounded in the lower and upper proportion. Equation (6) determines whether asset i is held or not.

The model is obviously a MOP with two objectives: risk (variance) and the expected return (mean). The target of the model is to find all non-dominated portfolios which can be called efficient portfolios. Denote a portfolio x as $x = (x_1, x_2, \dots, x_i, x_n)$, where x_i is the investment proportion of the i th asset. Supposed two portfolios x_1 and x_2 , x_1 is said to dominate x_2 (denoted as $x_1 \succ x_2$), when $\mu_1 \geq \mu_2$, $\rho_1 \leq \rho_2$ and at least one is inequality. An efficient portfolio is defined when the portfolio is not dominated by any other portfolios.

3 The contrasting algorithms

Evolutionary algorithms (EAs) [52] are stochastic search heuristic that simulates the process of natural evolution. The EAs mainly contain three operators: selection, crossover and mutation. The general framework of EAs is illustrated as follows:

```

Input:  $T$ 
initialize  $P_0$  and set  $t = 0$ 
While ( $t < T$ )
    evaluate( $P_t$ )
     $Q_t = \text{select}(P_t)$ 
     $Q_t' = \text{crossover}(Q_t)$ 
     $Q_t'' = \text{mutation}(Q_t')$ 
    evaluate( $Q_t''$ )
     $P_{t+1} = g(P_t \cup Q_t'')$ 
     $t = t + 1$ 
End While
Output:  $P_{t+1}$ 

```

Where t and T are the iteration counter and the maximum number of iterations, respectively; P_t and Q_t are the population in iteration t and the external population, respectively; and g denotes the operator updating the next population in different EAs. After initialization, when the number of iterations t does not reach T , at each iteration, the current population is first evaluated using a fitness function that is usually directly related to the objective function. Next,

individuals with the best fitness value are selected to a mating pool and some new individuals are created by crossover and mutation operator. Finally, an updating mechanism replaces these older individuals with better individuals to participate in the next generation. The role of crossover operator is actually a recombination of two parents in the mating pool making, so that the child can share the characteristics of its parents. And the mutation operator is used to partially modify the newborn child. By constantly changing the population, these individuals become stronger than former through the evolutions.

EA has a popular application for solving MOP, which is called MOEA. In 2011, Anagnostopoulos and Mamanis [38] compared the effectiveness of five state-of-the-art MOEAs for the MVCCPO problem on the public benchmark data sets. Experimental results indicate that SPEA2 and NSGA-II are superior to the other three algorithms: NPGA2, PESA and e-MOEA. Therefore, the best two algorithms: NSGA-II and SPEA2 in [38] are used here for comparison. Furthermore, Zhang et al. [16] proposed MOEA/D, which is based on decomposition. Their research group further improved MOEA/D and proposed a better algorithm called MOEA/D-DE (a new version of MOEA/D) in [17]. Therefore, in this paper, *e*-NSLS is compared with the above three algorithms: SPEA2, NSGA-II and MOEA/D-DE. In this section, we briefly introduce the main ideas of these three algorithms. Furthermore, we also compare our algorithm with ABC-FC and GRASP-QUAD, which are the recent heuristic algorithms to solve the MVCCPO problem. ABC-FC [47] is an artificial bee colony (ABC) algorithm improved with the firefly algorithm (FA) applied to the MVCCPO problem. GRASP-QUAD [49] contains two procedures, namely, stock selection and proportion determination to solve the cardinality portfolio optimization problem, which has only one objective.

4 A local search-based heuristic algorithm for portfolio optimization *e*-NSLS

NSLS is based on the non-dominated sorting suggested in NSGA-II and presents a new local search schema for convergence. Furthermore, a farthest-candidate mechanism which is used in the sampling theory is combined for diversity preservation. NSLS is an efficient algorithm for solving MOPs. At each iteration, given a population S , NSLS adopt a local search schema to obtain a better population S' , and then non-dominated sorting is performed on $S \cup S'$ to get a new population for the next iteration. In addition, the farthest candidate approach is adopted to help the non-dominated sorting procedure to select the new population. NSLS maintains two populations: the current population S which is called the internal population and the population

S' produced by the local search schema which is called the external population in this paper. The former one is analogous to the population in EAs and the latter one is analogous to the archive. In this paper, the size of S is equal to the size of S' . Therefore, here, we denote the size of S as N and S' as N^{arc} for convenience. In this paper, we propose an extension of NSLS (e -NSLS) to solve the MVCCPO problem which has several constraints. In addition, a repair strategy is adopted in e -NSLS for the purpose of making the solution satisfy the constraints.

4.1 Portfolio representation and encoding

For the sake of using NSLS to solve the portfolio problem, we adopt the similar portfolio representation and encoding as [38]. The following two vectors are used to specify a portfolio:

$$\theta = \{\theta_1, \theta_2, \dots, \theta_n\}, \quad \theta_i \in \{0, 1\}, \quad i = 1, 2, \dots, n. \quad (7)$$

$$\mathbf{x} = \{x_1, x_2, \dots, x_n\}, \quad 0 \leq x_i \leq 1, i = 1, 2, \dots, n. \quad (8)$$

where n is the available number of assets; θ_i determines whether asset i is included in the portfolio. x_i is a real value storing the proportion of the investment in asset i .

4.2 Initialization

In the initialization phase of e -NSLS, a population with size of N is randomly generated. The variables of each solution in the first population are randomly valued between the bounds.

4.3 Local Search Schema

Local search is a heuristic method which is usually used for dealing with optimization problems. In this paper, the local search schema proposed in NSLS is utilized for producing some neighborhoods as candidates. Supposed that there is a portfolio $\mathbf{x}(x_1, x_2, \dots, x_n)$, the new neighborhoods of \mathbf{x} is calculated as follows:

$$x'_i = x_i \pm \tau \times (b_i - c_i) \quad i \in [1..n] \quad (9)$$

where, \mathbf{b} and \mathbf{c} are two portfolios randomly selected from population S ; τ , which follows a normal distribution $N(\mu, \sigma)$, plays the role as a perturbed factor to avoid the local search schema converging fast to the efficient front. The local search schema has the ability to exploit information about the search space in producing new neighborhoods. The distance between the neighbor portfolio and the current portfolio should neither be too wide nor too narrow. If the generated neighbor portfolios are far away from

the current portfolio, it will slow down the convergence to the optimal. On the contrary, if the neighbor portfolios are distributed too narrow from the current portfolio, the diversity may decrease and the convergence speed will be too fast. Therefore, the strategy of producing local search neighborhoods should be appropriate to satisfy a balance of exploration and exploitation. The local search schema proposed in NSLS can meet the above demands.

Assuming a population S , our algorithm repeatedly tries to enhance the portfolio in S by replacing it with the neighboring portfolios produced by the local search schema. Assuming a current portfolio \mathbf{x} , portfolios \mathbf{x}_1 and \mathbf{x}_2 are determined by \mathbf{x} according to (7). The principle of replacement strategy is as follows:

- (1) If portfolio \mathbf{x}_1 dominates \mathbf{x} and portfolio \mathbf{x}_2 dominates \mathbf{x} , replace \mathbf{x} by \mathbf{x}_1 or \mathbf{x}_2 randomly.
- (2) If portfolio \mathbf{x}_1 dominates \mathbf{x} only, replace \mathbf{x} by \mathbf{x}_1 ;
- (3) If portfolio \mathbf{x}_2 dominates \mathbf{x} only, replace \mathbf{x} by \mathbf{x}_2 ;
- (4) If portfolio \mathbf{x}_1 does not dominate \mathbf{x} and portfolio \mathbf{x}_2 does not dominate \mathbf{x}_1 , replace \mathbf{x} by \mathbf{x}_1 ;
- (5) If portfolio \mathbf{x}_2 does not dominate \mathbf{x} and portfolio \mathbf{x}_1 does not dominate \mathbf{x}_2 , replace \mathbf{x} by \mathbf{x}_2 ;

Note that in NSLS, the substitution operation occurs in each variable. This means that when the number of portfolio variables increases, the computational complex will increase too. Therefore, e -NSLS does not allow substitution in each variable. Only some randomly chosen variables require the substitution operation. Let e denote the number of variables that needs the operation. e is defined as follows:

$$e = 2\omega * \sqrt{n} \quad (10)$$

where n is the number of variables and ω is a control factor related to the number of variables. When the number of variables is high, ω is used to increase the number of substitution operations to achieve a better convergence. e -NSLS then randomly choose e variables to perform the substitution operation.

4.4 The fast nondominated sorting and the farthest-candidate approach

The fast non-dominated sorting is a key technology proposed in NSGA-II, which is used for selecting a new population. The fast non-dominated sorting ranks all individuals into some fronts with different rank value. At each iteration, the algorithm needs to combine the old and new population produced by the local search schema, and selects a new population of size N from the combined population for entering the next iteration. The algorithm chooses the N best solutions from the combined population according to

the ranking fronts. We denote the last front selected by the algorithm as F_e . Usually, the number of solutions in F_e will exceed N . Therefore, a technology needs to be developed for selecting the best required solutions in F_e . In NSGA-II, crowding-comparison mechanism was proposed to solve the problem. In NSLS, a farthest-candidate approach which has a good effect for maintaining the diversity is adopted. In the farthest-candidate approach, the solution which is unselected and farthest from the selected solutions is defined as the best solution. The best solution in F_e is chosen from the unselected solutions one by one until the number of selected solutions is equal to N . For the detailed information about these two technologies, the reader is referred to [13, 15], respectively.

4.5 The framework of e -NSLS

The proposed method is an iteration-based algorithm. Given a maximum generation denoted here as T and a current population of portfolios S_t , the algorithm constantly improves S_t by iteration for T times. At each iteration, a new population S'_t is constructed by performing local search schema on the current population S_t and the fast non-dominated sorting approach is adopted on $S_t \cup S'_t$, and then the farthest-candidate approach for diversity preservation is adopted to select the next population S_{t+1} . The overall algorithm is shown in the Pseudo Code of e -NSLS.

The computational complexity of one iteration of e -NSLS is as follows: the local search is $O(MN\sqrt{e})$; the fast non-dominated sorting is $O(M(2N)^2)$ and the farthest candidate approach is $O(MN^2)$. Where n is the number of variables, N is the size of the internal population and M is the number of the objectives. Usually, n is usually smaller than N , so the total computational complexity of e -NSLS is dominated by non-dominated sorting which has a computational complexity $O(MN^2)$.

Pseudo Code of e -NSLS for MVCCPO problem

```

Initialize  $\mu, \sigma, K$ , where  $\mu$  and  $\sigma$  are the mean and the variance value of the normal distribution used in the local search, respectively;  $K$  is the maximum number of assets allowed in a portfolio.
 $S_1 = \emptyset$ 
For  $i = 1$  to  $N$  do
    Initialize portfolio  $i$ 
    If constraints are violated
        Repair by constraint handling techniques (see Sect. 4.6)
     $S_1 = S_1 \cup i$ 
End For
For  $t = 1$  to  $T$  do
    Produce a new population by Eq. (9):  $S'_t = \text{localsearch}(S_t, \mu, \sigma)$  (see Sect. 4.3)
    Generate all non-dominated fronts of  $S'_t \cup S_t$ :  $F = (F_1, F_2, \dots)$  by the fast nondominated sorting
    Set  $S_{t+1} = \emptyset$  and  $i = 1$ 
    While  $|S_{t+1}| + |F_i| \leq N$ 
        Put  $i$ -th non-dominated front into the former population:  $S_{t+1} \cup F_i$ 
         $i = i + 1$ 
    End While
    Select  $N - |S_{t+1}|$  solutions defined as  $G$  from  $F_i$  by using the farthest-candidate approach and add them to  $S_{t+1}$ :  $S_{t+1} = S_{t+1} \cup G$  (see Sect. 4.4)
    For  $i = 1$  to  $N$  do
        If constraints are violated
            Repair by constraint handling techniques (see Sect. 4.6)
    End For
End For
Return the final population  $S_T$  as the result
    
```

4.6 Constraints handling method

Constraints are another difficulty in the portfolio problem, and all solutions used during the algorithm must satisfy these constraints. Therefore, a solution repair algorithm is essential after the objective values are computed. In this paper, we randomly select k ($k \leq K$) maximum assets if the number of assets in the portfolio is greater than the maximum allowed number K .

Another problem is to deal with the budget amount, i.e., the lower and upper proportion constraint. The following equation can sum to one and satisfies the lower proportion limit.

$$x'_i = l_i * \theta_i + \frac{x_i * \theta_i}{\sum_{k=1}^n x_k * \theta_k} * (1 - \sum_{k=1}^n l_k * \theta_k) \quad i = 1, 2, \dots, n. \tag{11}$$

An iterative procedure presented in Chang et al. [3] is adopted here to satisfy the upper proportion limit. In [3], the proportions which are larger than the upper proportion u_i are set as the value of u_i . There exists several methods to bring the violated solution into the required range. Padhye et al. [53] summarized the constraint-handling strategies for real parameter evolutionary optimization and proposed two new inverse parabolic constraint-handling methods. In this paper, four simple and popular methods from [53] are chosen to handle with the upper proportion constraints in the MVCCPO problem. These four methods are Random method, Periodic method, Setonboundary method, and Exponential Spread (EXP-S) method, respectively. In fact, the violated solution handling strategy of the iterative procedure used in [3] is exactly the Setonboundary method. In e -NSLS, a candidate solution is produced by three parents. The inverse parabolic constraint-handling methods and shrink approach in [53], which bring the violated solution into the allowed range through one parent solution are not suitable for the proposed algorithm here.

- (1) Random method is one of the common and useful approach for dealing with the lower and upper proportion constraints. The solution out of the range of the lower and upper proportion is replaced by a random chosen value in the range $[l_i, u_i]$ as follows:

$$x_i^{new} = \text{random} [l_i, u_i], x_i > u_i \text{ or } x_i < l_i \tag{12}$$

- (2) The periodic method assumes that there exists a periodic repetition of the objective function with a periodic $p = u_i - l_i$. A violated solution is mapped in the range $[l_i, u_i]$ by the following equation:

$$x_i^{new} = \begin{cases} l_i + (x_i - u_i) \% p, & x_i > u_i \\ u_i - (l_i - x_i) \% p, & x_i < l_i \end{cases} \tag{13}$$

- (3) The setonboundary method is carried out by resetting the violated solution on the bound of the range which it violates as follows:

$$x_i^{new} = \begin{cases} u_i, & x_i > u_i \\ l_i, & x_i < l_i \end{cases} \tag{14}$$

- (4) The EXP-S method uses an exponential probability distribution to set the violated solution inside the feasible range. The applied exponential probability distribution is spread over the entire feasible range. The probability increases from the lower bound to the upper bound in the direction toward the violated boundary. For any random number r within $[0,1]$, a violated solution is mapped in the range $[l_i, u_i]$ by the following equation:

$$x_i^{new} = \begin{cases} l_i + \ln(1 + r(\exp(u_i - l_i) - 1)), & x_i > u_i \\ u_i - \ln(1 + r(\exp(u_i - l_i) - 1)), & x_i < l_i \end{cases} \tag{15}$$

Finally, to make all these proportions sum to one, an iterative procedure is used to adjust them. More information about the procedure, the reader may refer to Chang et al [3]. In the experimental section, we test the effectiveness of these four boundary constraint handling strategies in Section 5.3.1, hereafter choose one of them as the boundary constraint handling strategy in the proposed algorithm.

5 Computational results

The behavior of e -NSLS is compared to NSGA-II, SPEA2, MOEA/D-DE, ABC-FC and GRASP-QUAD on the benchmark data which is publicly available at: <http://people.brunel.ac.uk/~mastjib/jeb/orlib/portinfo.html>. These data refer to weekly stock price between March 1992 and September 1997. The complete information of the assets is included in the data package and is summarized in Table 1. Each portfolio problem file is composed of n assets. For each asset i , the file provides the mean return μ_i , the standard deviation of return σ_i . In addition, the correlation $c_{i,j}$ for each pair of asset i and j is also included in the file. The risk between asset i and j can be computed as follows:

$$r_{i,j} = c_{i,j} * \sigma_i * \sigma_j \tag{16}$$

All algorithms have been implemented in C++ and executed on Microsoft Window 7 Intel core 2 Duo CPU E8400 3.00GHz, with 2GB RAM.

Table 1 Information about test problem instances

Test problem instances	Abbreviation	Number of assets
Hong Kong Hang Seng	P1	31
German Dax 100	P2	85
British FTSE 100	P3	89
US S&P 100	P4	98
Japanese Nikkei 225	P5	225

5.1 Performance indicators

5.1.1 The Inverted Generational Distance (IGD)

In this paper, the inverted generational distance (IGD) [54] is adopted to compare the quality of the proposed method with other algorithms. Let F^* be a set of the true unconstrained efficient frontier (TUEF) that contains 2000 efficient points or the true cardinality constrained efficient frontier (TCCEF) [55] that contains 500 efficient points. The former file can be downloaded from <http://people.brunel.ac.uk/~mastjib/jeb/orlib/portinfo.html> and the latter can be downloaded from the web page of Professor F. Tardella: <http://w3.uniroma1.it/Tardella/homepage.html> in the objective space. F is an approximation obtained by the algorithm. In this paper, the inverted generational distance (IGD) metric is described as follows:

$$IGD(F^*, F) = \frac{\sum_{x \in F^*} \min dis(x, F)}{|F^*|} \tag{17}$$

where $\min dis(x, F)$ is the minimum Euclidean distance between solution x and the solutions in F . The IGD-metric value measures both the diversity and convergence to some extent. The smaller the value of IGD-metric, the better both the diversity and convergence toward the efficient frontier.

5.1.2 The Spread (Δ)

In order to quantitatively analysis the distribution effectiveness of e -NSLS with the other algorithms on MVCCPO problems, a spread metric presented by Wang et al. [56] is adopted in this paper. The spread is denoted as Δ and calculated as follows:

$$d(x, \Phi) = \|f(x) - f(y)\| \tag{18}$$

$$\bar{d} = \frac{1}{|\Phi|} \sum_{x \in \Phi} d(x, \Phi) \tag{19}$$

$$\Delta = \frac{\sum_{x \in \Phi} |d(x, \Phi) - \bar{d}| + \sum_{i=1}^m d(E_i, \Phi)}{(|\Phi| - m) \bar{d} + \sum_{i=1}^m d(E_i, \Phi)} \tag{20}$$

where (E_1, E_2, \dots, E_m) are the extreme solutions in the true Pareto front (PF), and Φ is a set of the obtained non-dominated solutions. The extreme solution $E_i (i \in m)$ is the point which has the best value in the i th objective. $f(x) =$

Table 2 Parameter settings in different experiments

	Population size	T	runs	l_i	u_i	K
Experiment 1	100	250	30	0.00	0.8	Assert number
Experiment 2	100	250	30	0.01	1.0	8
Experiment 3	100	250	30	0.01	0.8	2, 5, 10

Table 3 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems where $l_i = 0.01$, $u_i = 0.8$ and $K = 10$

Instance mean (std)	P1	P2	P3	P4	P5
e -NSLS ^{ran} (e -NSLS with random method)	2.68693e-04 (3.90572e-05)	4.31577e-04 (6.63579e-05)	2.16408e-04 (4.25815e-05)	2.74982e-04 (5.71395e-05)	1.53901e-04 (6.68723e-05)
e -NSLS ^{per} (e -NSLS with periodic method)	2.72354 e-04 (4.26817e-05)	4.44892e-04 (8.69088e-05)	2.10118e-04 (4.52755e-05)	2.71717e-04 (5.74615e-05)	1.30763e-04 (6.55065e-05)
e -NSLS ^{sob} (e -NSLS with setonboundary method)	2.66178 e-04 (4.45832e-05)	4.46525e-04 (9.89781e-05)	2.12372e-04 (3.80192e-05)	2.02461e-04 (4.06889e-05)	1.76195e-04 (4.85833e-05)
e -NSLS ^{exp-s} (e -NSLS with Exp-S method)	2.64134e-04 (4.10591e-05)	4.41455e-04 (7.00725e-05)	2.07387e-04 (4.22315e-05)	2.87793e-04 (4.05102e-05)	1.83235e-04 (3.96746e-05)

Table 4 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems where $l_i = 0.01$, $u_i = 0.8$ and $K = 10$

Instance Mean (std)	P1	P2	P3	P4	P5
e -NSLS with ES Strategy	3.70215e-04 (8.36921e-05)	5.65952e-04 (7.81291e-05)	5.13074e-04 (7.52905e-05)	4.65930e-04 (5.19328e-05)	3.59974e-04 (7.57163e-05)
e -NSLS with DE Strategy	2.66178e-04 (4.45832e-05)	4.46525e-04 (9.89781e-05)	2.12372e-04 (3.80192e-05)	2.02461e-04 (4.06889e-05)	1.76195e-04 (4.85833e-05)

Table 5 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems where $l_i = 0.01$, $u_i = 0.8$ and $K = 2$

Instance mean (std)	P1	P2	P3	P4	P5
e -NSLS with One Candidate Strategy	1.51734e-04 (3.81881e-05)	1.1794e-04 (3.55100e-05)	1.32576e-04 (134825e-04)	9.47719e-05 (3.80414e-05)	2.47162e-04 (6.54705e-05)
e -NSLS with Two Candidates Strategy	1.16568e-04 (2.07732e-05)	1.05562e-04 (3.38895e-05)	1.04006e-04 (5.26723e-06)	9.10271e-05 (3.70775e-05)	2.23156e-04 (4.77321e-05)

Table 6 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems where $l_i = 0.01$, $u_i = 0.8$ and $K = 2$

Instance mean (std)	P1	P2	P3	P4	P5
<i>e</i> -NSLS removing the replacement strategy	1.22273e-04 (2.96922e-05)	1.13084e-04 (3.52611e-05)	1.12996e-04 (5.41515e-05)	9.2836e-05 (4.42844e-05)	2.18170e-04 (9.99449e-05)
<i>e</i> -NSLS	1.16568e-04 (2.07732e-05)	1.05562e-04 (3.38895e-05)	1.04006e-04 (5.26723e-06)	9.10271e-05 (3.70775e-05)	2.17225e-04 (4.98846e-05)

$(\rho(x) \cdot \mu(x))$, which is a vector containing $\rho(x)$ and $\mu(x)$. A smaller value of Δ reveals a better distribution of these non-dominated solutions. Therefore, a smaller value of Δ is desirable.

5.1.3 Wilcoxon’s signed ranks test

We also adopt a non-parametric statistical hypothesis test known as Wilcoxon’s signed ranks test [57] to statistically analyze *e*-NSLS and the other algorithms. Wilcoxon’s signed ranks test is a common tool for detecting the difference between two algorithms. In our experiment, *p*-value [57] is calculated by the SPSS software packages. The significance level α is set to 0.05 as suggested in [57]. $\alpha < 0.05$ indicates that there is a significant difference between the two algorithms.

5.2 Parameter settings

The performance of *e*-NSLS is compared with other five algorithms including: NSGA-II, SPEA2, MOEA/D-DE, ABC-FC and GRASP-QUAD. Experiments with different parameter settings are conducted. The common parameters include the population size and the maximum number of generations. They are set to 100 and 250 for each algorithm, respectively.

Apart from the above parameters, NSGA-II, SPEA2, MOEA/D-DE and *e*-NSLS require some additional parameters. For NSGA-II, SPEA2 and MOEA/D-DE, the parameter settings are the same as in their original researches [13, 14]. For *e*-NSLS, only three additional parameters are set.

In the normal distribution $N(\mu, \sigma)$, μ and σ are set to 0.5 and 0.1, respectively. ω is set to 1 for P1-P4 and 3 for P5. In addition, each algorithm runs 30 times independently for each test problem. These parameter settings are summarized in Table 2. The parameter settings about the comparison with ABC-FC and GRASP-QUAD, the reader can refer to Section 5.3.8.

5.3 Experimental results

5.3.1 Experiment 1 : Anatomy of *e*-NSLS with simulation results

(1) Simulation results of different boundary constraint handling strategies In this section, in order to verify the effectiveness of the boundary constraint handling strategies in the proposed algorithm, four versions of the proposed algorithm with different boundary constraint handling strategies in Section 4.6 are executed using different parameter settings. These four algorithm are named as *e*-NSLS^{ran}, *e*-NSLS^{per}, *e*-NSLS^{sob}, and *e*-NSLS^{exp-s}, respectively. In this particular test we set population Size $N = 100$, $T = 250$, $runs = 30$, $l_i = 0.01$, $u_i = 0.8$ and $K = 10$ for all test instances. Table 3 gives the IGD-metric values of the non-dominated solutions found by each algorithm on all five test data in 30 independent runs. It is obvious from Table 3 that the proposed algorithm is not very sensitive to the boundary constraint handling strategy because that all four algorithms achieve almost the same effect. The probable reason is that the local search schema of the proposed algorithm plays a more important role in the searching process.

Table 7 The Spread-metric values of the non-dominated solutions found by each algorithm on all five test problems where $l_i = 0.01$, $u_i = 0.8$, and $K = 5$

Instance mean (std)	P1	P2	P3	P4	P5
NSLS	4.37926e-01 (4.40526e-02)	4.60142e-01 (7.58853e-02)	6.31231e-01 (4.65906e-02)	5.66789e-01 (5.50770e-02)	6.25287e-01 (4.65694e-02)
<i>e</i> -NSLS	2.56228e-01 (6.91804e-02)	2.97449e-01 (7.0191e-02)	4.08663e-01 (2.85545e-02)	3.41233e-01 (6.97148e-02)	5.02889e-01 (8.79155e-02)

Table 8 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems where $l_i = 0.00$, $u_i = 0.8$

Instance mean (std)	P1 ($K = 31$)	P2 ($K = 85$)	P3 ($K = 89$)	P4 ($K = 98$)	P5 ($K = 225$)
NSGA-II	1.34875e-04 (1.74884e-05)	4.51960e-04 (1.11108e-04)	4.11015e-04 (8.13094e-05)	4.06491e-04 (8.97948e-05)	3.67625e-04 (6.05015e-05)
SPEA2	2.19255e-04 (8.24058e-05)	5.28810e-04 (9.98237e-05)	4.31634e-04 (8.47299e-05)	4.38279e-04 (7.08541e-05)	3.45099e-04 (4.31618e-05)
MOEA/D-DE	1.82442e-04 (2.04428e-05)	2.51261e-04 (3.79443e-05)	1.90736e-04 (3.10247e-05)	1.84176e-04 (2.17102e-05)	1.29301e-04 (2.98856e-05)
<i>e</i> -NSLS	1.00527e-04 (3.50325e-06)	1.15016e-04 (3.42765e-05)	1.00179e-04 (1.53592e-05)	1.71902e-04 (6.42923e-05)	2.67251e-04 (4.82984e-05)

In the following experiments, we select the setonboundary method as our boundary constraint handling strategy.

(2) Simulation results of DE operator in *e*-NSLS The proposed local search schema designed for the MVCCO problem has two main characteristics. One is that it uses a mutation operation similar to DE operator for the purpose of producing some neighborhoods as candidates. The other one is that it generates two candidates by seeking two directions for the current individual, we shall refer to this strategy

as **Two Candidates Strategy**. In this section, we make the first attempt in understanding the importance of the local search schema.

First, we replace the DE operator by the evolution strategies (ES) operator which was presented in [58] and [59]. The mutation mechanism of ES is to change each variable of the solution by adding random noise derived from a Gaussian distribution as: $x'_i = x_i \pm N(0, \sigma^2)$, $i \in [1..n]$. An experiment is conducted to compare the effects of changing DE operator to ES operator on the proposed algorithm. Here

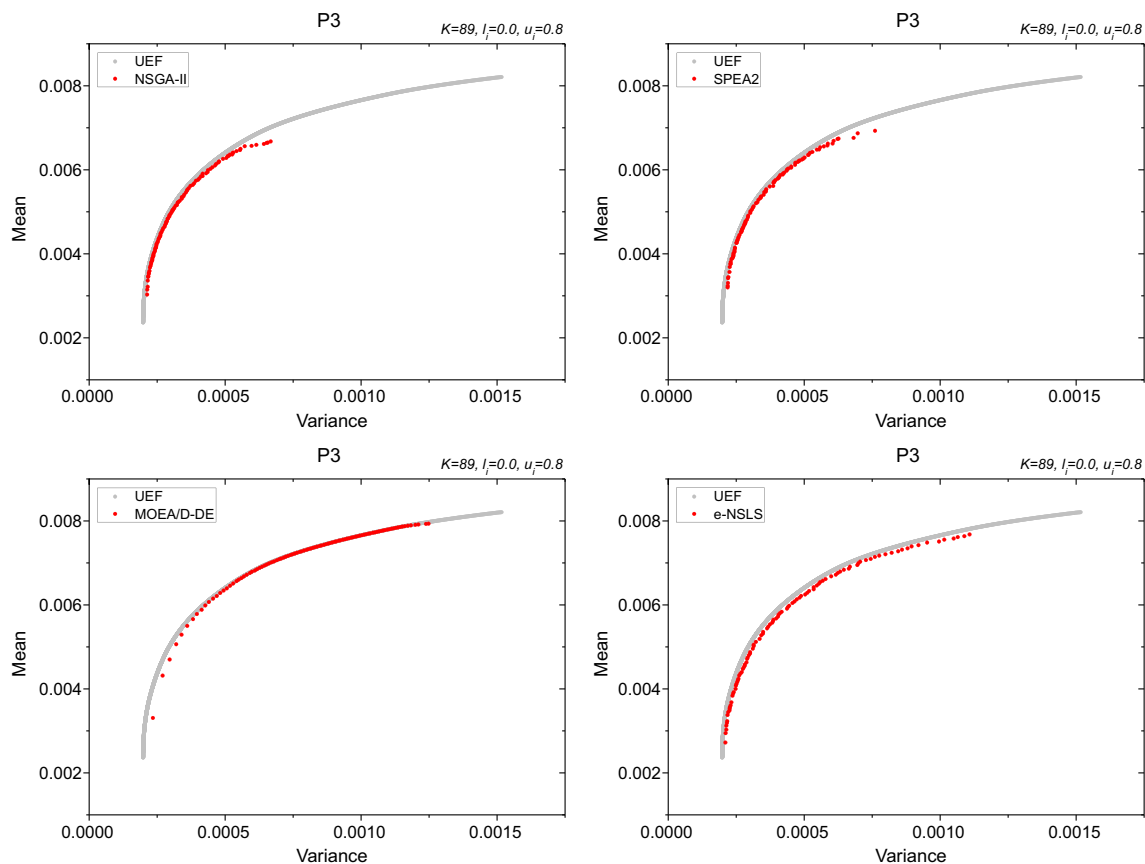
**Fig. 1** Efficient frontiers for British FTSE 100 data set (P3) of $l_i = 0.0$, $u_i = 0.8$

Table 9 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems with cardinality constraint ($K = 8$) and $l_i = 0.01, u_i = 1.0$

Instance mean (std)	P1	P2	P3	P4	P5
NSGA-II	2.78777e-04 (5.7247e-05)	3.21132e-04 (8.02728e-05)	2.54870e-04 (7.18994e-05)	2.10318e-04 (4.18929e-05)	7.16553e-05 (1.09698e-05)
SPEA2	4.98116e-04 (1.33880e-04)	4.20396e-04 (8.00619e-05)	2.89113e-04 (5.72844e-05)	2.43755e-04 (4.69655e-05)	9.89304e-05 (1.93536e-05)
MOEA/D-DE	2.10069e-04 (6.4112e-05)	2.74866e-04 (4.30625e-05)	1.91436e-04 (3.80357e-05)	1.90286e-04 (3.97725e-05)	1.39907e-04 (5.53971e-05)
<i>e</i> -NSLS	1.46640e-04 (5.06999e-05)	2.27097e-04 (6.69567e-05)	1.33261e-04 (2.6731e-05)	1.30626e-04 (3.58381e-05)	1.30581e-04 (5.35568e-05)

we set $\sigma = 0.2$. Table 4 presents the mean and standard deviation results based on 30 independent runs on all test problems with the IGD metric. It is obvious from Table 4 that *e*-NSLS with DE shows a minor improvement in all cases, which reveals that the DE operator is a key feature in the algorithm for dealing with the MVCCPO problems.

Second, to further verify the effectiveness of **Two Candidates Strategy**, another experiment is conducted to compare the **Two Candidates Strategy** and **One Candidate Strategy**. Table 5 reveals that **Two Candidates Strategy** is a little better than **One Candidate Strategy** on all test

problems in terms of IGD metric, which reveals that **Two Candidates Strategy** is really better than **One Candidate Strategy** to a certain degree.

(3) Simulation results of the replacement strategy in *e*-NSLS In this section, we investigate the role of the replacement strategy in *e*-NSLS. The replacement strategy of *e*-NSLS is modified by always accepting a candidate solution (unlike before where the candidate is accepted only if it was better or non-dominated). Table 6 systematically

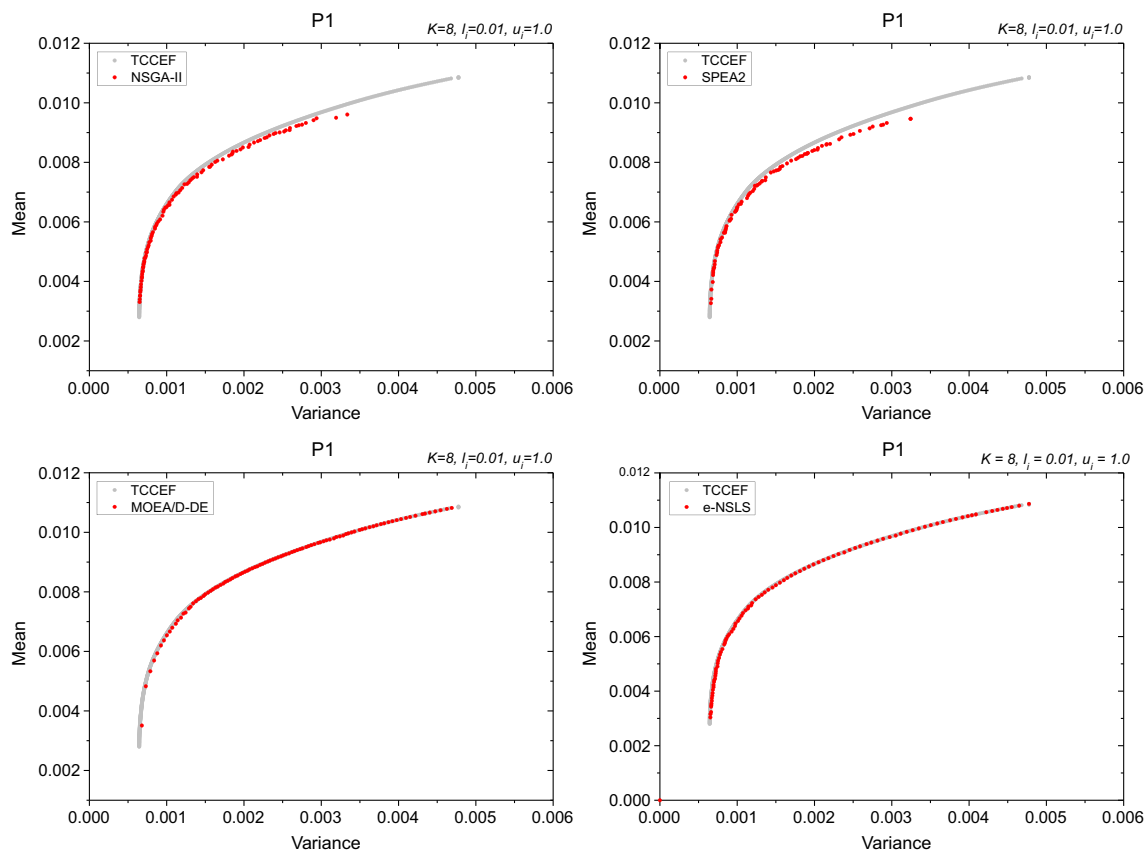


Fig. 2 Efficient frontiers for Hong Kong Hang Seng data set (P1) of $K = 8$, and $l_i = 0.01, u_i = 1.0$

Table 10 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems with cardinality constraint ($K = 2$) and $l_i = 0.01, u_i = 0.8$

Instance mean (std)	P1	P2	P3	P4	P5
NSGA-II	1.33377e-04 (1.93122e-05)	1.48038e-04 (2.35707e-05)	1.43465e-04 (9.13701e-05)	5.18289e-05 (7.36003e-06)	1.96553e-04 (3.77761e-05)
SPEA2	2.32734e-04 (1.80286e-05)	1.32511e-04 (1.59675e-05)	3.20497e-04 (1.05807e-04)	9.31555e-05 (4.35602e-05)	2.73735e-04 (3.54402e-05)
MOEA/D-DE	2.08286e-04 (3.40791e-05)	1.75816e-04 (1.15787e-05)	1.91508e-04 (5.81048e-05)	6.42845e-05 (5.26237e-06)	2.57464e-04 (3.37446e-05)
<i>e</i> -NSLS	1.16568e-04 (2.07732e-05)	1.05562e-04 (3.38895e-05)	1.04006e-04 (5.26723e-06)	9.10271e-05 (3.70775e-05)	2.17225e-04 (4.98846e-05)

compares the effects of the original algorithm and removing the replacement strategy. It is clear from Table 6 that the replacement strategy is effective.

(4) Simulation results of the farthest-candidate approach

The farthest-candidate approach plays an important role in *e*-NSLS to maintain the diversity of the solutions. In this section, to investigate the effectiveness of the farthest-candidate approach, we present simulation results and show comparisons based on the spread metric (Δ), which measures the distribution of these obtained non-dominated solutions. In this experiment, the farthest-candidate approach is substituted with the crowding distance approach in [13]. The experimental results are demonstrated in Table 7, from which we can see *e*-NSLS has a better distribution than NSLS. Thus, we can conclude that the farthest-candidate approach is an effective skill to improve the performance of the algorithm.

5.3.2 Experiment 2: Different parameter setting with $l_i = 0.0, u_i = 0.8$ and different cardinality constraint

This section provides the computational results obtained by each algorithm when $l_i = 0.0, u_i = 0.8$ and the cardinality constraint is the assert number of each instance.

Table 8 shows the mean and standard deviation of the IGD-metric values of the non-dominated solutions found by NSGA-II, SPEA2, MOEA/D-DE and *e*-NSLS on the five test problems in 30 independent runs. The best values of the mean IGD among these four algorithms are noted with bold font. As can be seen in Table 8, *e*-NSLS has a better IGD-metric value than the other three algorithms for all test problems. This indicates that *e*-NSLS is suitable for solving the MVCCPO problem. The best obtained efficient surfaces of British FTSE 100 data set (P3) in the 30 runs of each algorithm are represented in Fig. 1. According to the parameter setting, the cardinality constraint is ignored, which means that it allows that a solution can hold 89 asserts (P3 has 89 variables) at the same time. We can observe from Fig. 1 that *e*-NSLS outperforms NSGA-II, SPEA2 and MOEA/D-DE in term of the diversity and convergence. MOEA/D-DE cannot search the area with low variance and low return while *e*-NSLS can.

5.3.3 Experiment 3: Different parameter setting with $l_i = 0.01, u_i = 1.0$ and $K = 8$

In this experiment, the lower proportion and upper proportion of the total investment of asset i in each portfolio are set at 1% and 100% (i.e., $l_i = 0.01, u_i = 1.0, i = 1, 2, \dots, n$),

Table 11 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems with cardinality constraint ($K = 5$) and $l_i = 0.01, u_i = 0.8$

Instance mean (std)	P1	P2	P3	P4	P5
NSGA-II	1.48900e-04 (1.31849e-05)	1.24768e-04 (3.65273e-05)	1.01812e-04 (3.22848e-05)	8.92138e-05 (3.08637e-05)	4.34399e-05 (7.20878e-06)
SPEA2	2.2859e-04 (3.00743e-05)	1.88044e-04 (3.71782e-05)	1.6594e-04 (5.4563e-05)	1.24588e-04 (4.94274e-05)	6.99569e-05 (1.19805e-05)
MOEA/D-DE	1.73138e-04 (7.31571e-06)	2.28178e-04 (5.81107e-05)	1.97975e-04 (3.69499e-05)	1.56806e-04 (3.44817e-05)	1.27920e-04 (4.53111e-05)
<i>e</i> -NSLS	1.00057e-04 (9.48537e-06)	5.93127e-05 (2.03598e-05)	7.6096e-05 (1.2992e-05)	7.30457e-05 (4.24315e-05)	1.22930e-04 (5.45395e-05)

Table 12 The IGD-metric values of the non-dominated solutions found by each algorithm on all five test problems with cardinality constraint ($K = 10$) and $l_i = 0.01, u_i = 0.8$

Instance mean (std)	P1	P2	P3	P4	P5
NSGA-II	6.86043e-04 (1.71002e-04)	5.32154e-04 (1.06619e-04)	3.95745e-04 (7.0063e-05)	3.08198e-04 (4.85185e-05)	1.18127e-04 (2.29422e-05)
SPEA2	8.22846e-04 (1.52676e-04)	5.88174 e-04 (1.06209e-04)	4.22696e-04 (6.34234e-05)	3.32589e-04 (6.59574e-05)	1.42996e-04 (2.58666e-05)
MOEA/D-DE	2.79394e-04 (6.95866e-05)	2.94487e-04 (4.85814e-05)	2.21992e-04 (5.89064e-05)	2.30933e-04 (5.07679e-05)	1.56151e-04 (5.04274e-05)
<i>e</i> -NSLS	2.66178e-04 (4.45832e-05)	4.46525e-04 (9.89781e-05)	2.12372e-04 (3.80192e-05)	2.02461e-04 (4.06889e-05)	1.76195e-04 (4.85833e-05)

and the cardinality constraint is set to 8, which means that the number of assets a portfolio can hold is not greater than 8. Table 9 shows the mean and standard deviation of the IGD-metric values of the non-dominated solutions found by the four algorithms in 30 independent runs. The best values of the mean IGD metric among these four algorithms are noted with bold font as before. Table 9 shows that *e*-NSLS is better than the other algorithm on four test problems out of five test problems. This reveals that *e*-NSLS has a

better convergence and diversity than the other three algorithms. Fig. 2 shows the efficient frontiers for Hong Kong Hang Seng data set P1 of these four algorithms. We can observe from Fig. 2 that *e*-NSLS has a uniform distribution of solutions while the other algorithms may not. Due to the good convergence ability of the local search schema and the effective diversity maintenance ability of the farthest-candidate approach, *e*-NSLS achieves the best result in these algorithms on most of the test problems.

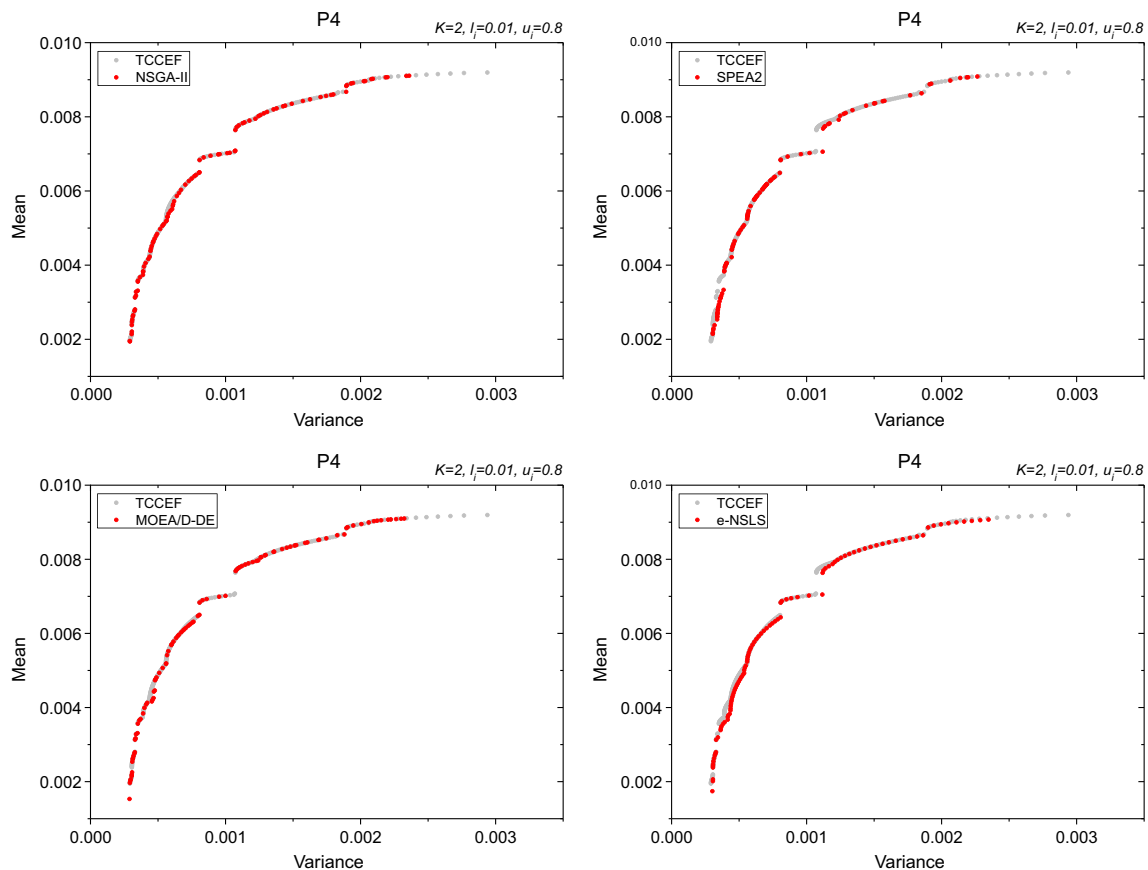


Fig. 3 Efficient frontiers for US S&P 100 data set (P4) of $K = 2$, and $l_i = 0.01, u_i = 0.8$

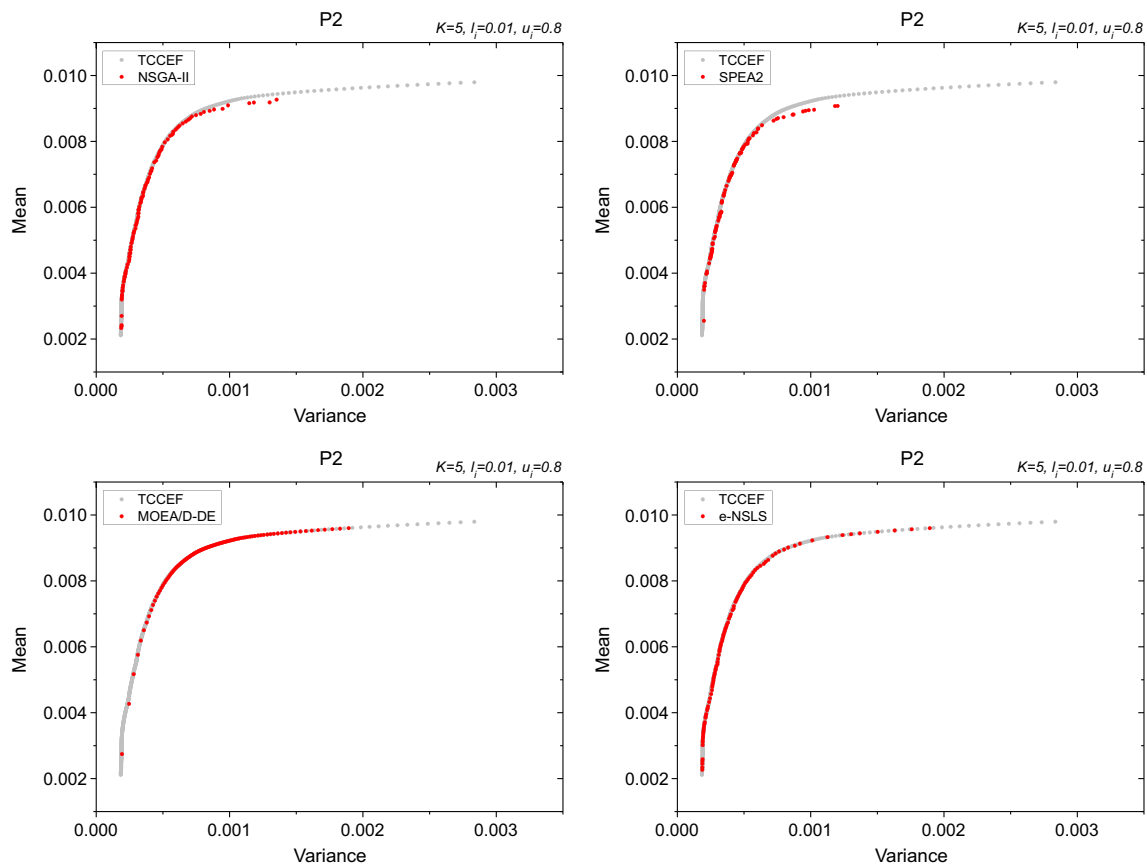


Fig. 4 Efficient frontiers for German Dax 100 data set (P2) of $K = 5$, and $l_i = 0.01$, $u_i = 0.8$

5.3.4 Experiment 4: Different parameter setting with $l_i = 0.01$, $u_i = 0.8$ and $K = 2, 5, 10$

This section provides three experiments with different values of K . The lower proportion and upper proportion of the total investment of asset i in each portfolio are set to 1% and 80% (i.e., $l_i = 0.01$, $u_i = 0.8$, $i = 1, 2, \dots, n$), and the cardinality constraint is set as 2, 5, and 10, respectively. The corresponding results are shown in Tables 10, 11, 12. The experimental results presented in these tables prove that none of the three algorithms which we took for comparisons has distinct advantages. When $K = 5$, e -NSLS performs better than the other algorithms on all test problems. When $K = 2$, e -NSLS performs better than the other three algorithms on P1, P2, P3 and worse on P4 and P5. When $K = 10$, e -NSLS performs better than the other three algorithms on P1, P3, P4 and worse on P2 and P5. The probable reason for this is that e -NSLS is based on local search and is imposed on each variable. When the number of cardinality constraint is changed, it needs a different parameter setting. Therefore, it may demand a different parameter set-

ting on e -NSLS. Improving the algorithm on P2 and P5 is left as our future study.

Figures 3, 4, 5 show the non-dominated solutions found by each algorithm under these parameter settings on P4 and P2, respectively. We can see e -NSLS achieves a better convergence and diversity over the true efficient frontier than the other three algorithms on every test instance. Moreover, P5, which has the highest number of variables in these five test problems, is chosen to illustrate the performance of these algorithms in Fig. 6. We can see from Fig. 6 that NSGA-II, MOEA/D-DE, e -NSLS can converge to the TCCEF and have a uniform distribution along the TCCEF when $K = 2$. MOEA/D-DE has a better convergence and distribution than the other four algorithms. When $K = 5$, e -NSLS has a better performance than the other three algorithms on P5, which is consistent with the results of IGD-metric. When $K = 10$, e -NSLS is good at finding the lower risk and return portfolio while MOEA/D-DE has a uniform distribution along the TCCEF. However, it can be seen that on average, e -NSLS is better approach than other three algorithms for tackling the MVCCPO problem.

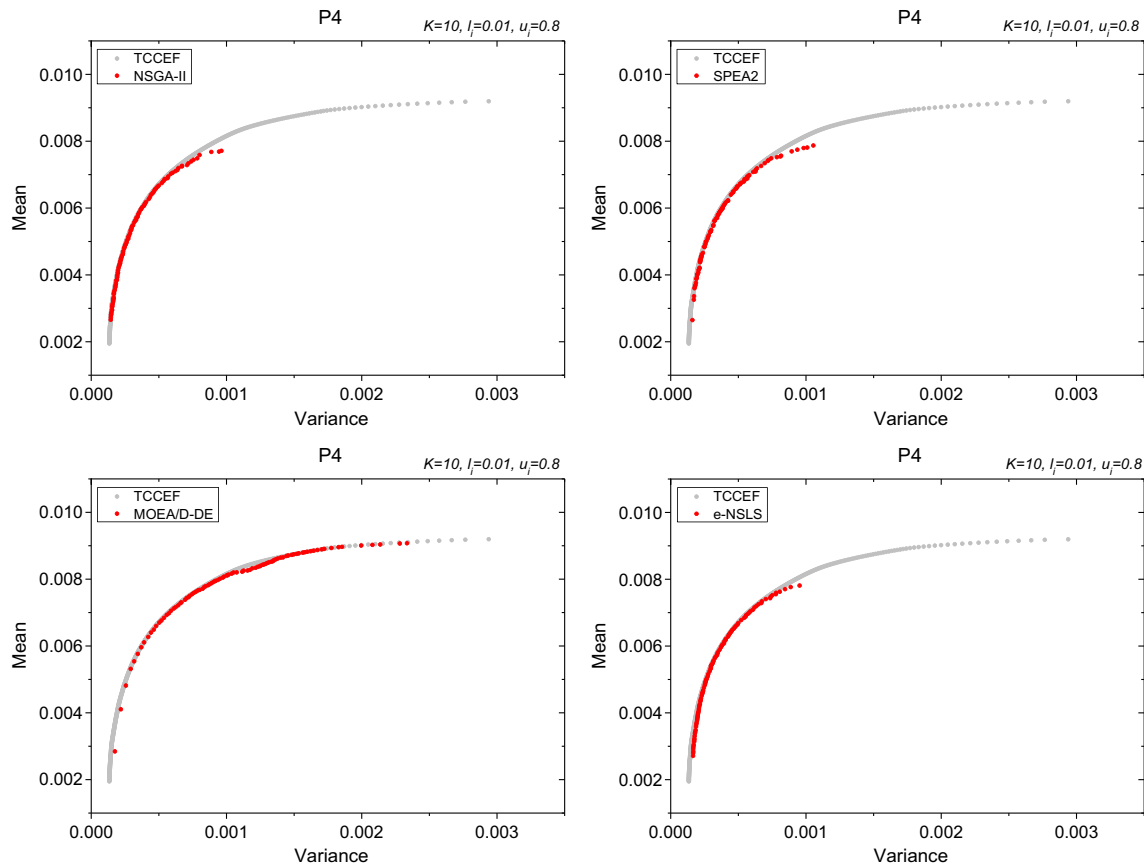


Fig. 5 Efficient frontiers for US S&P 100 data set (P4) of $K = 8$, and $l_i = 0.01, u_i = 1.0$

5.3.5 Results of the Wilconox’s signed rank test

The computational results are also compared statistically using the Wilcoxon’s signed rank test [57] and the corresponding results are given in Table 13. It is clear from this table that all p -values are smaller than $\alpha = 0.05$, which further indicates that the proposed e -NSLS performs better than the other algorithms.

5.3.6 Effectiveness of the farthest-candidate approach for the portfolio optimization problems

In this section, to further demonstrate the effectiveness of the farthest-candidate approach in maintaining the diversity of the obtained non-dominated solutions, the spread metric (Δ) is used to measure the distribution of these obtained non-dominated solutions. For reason of paper space, we use the experimental results in Section 5.3.2. The experimental results are demonstrated in Table 14. It is clear from the table that e -NSLS has a better distribution than the other algorithms, which shows the effectiveness of the farthest-candidate approach in e -NSLS.

5.3.7 Sensitivity

In this section, to verify the sensitivities of the performance of μ and σ in the Gaussian distribution $N(\mu\sigma^2)$, the different settings of μ and σ in e -NSLS on P2 are evaluated. We use similar parameter settings as Section 5.3.4 ($K = 5, l_i = 0.01$, and $u_i = 0.8$) except the settings of μ and σ .

Figure 7 illustrates the values of the mean IGD metric in 30 independent runs versus the values of μ and σ , respectively. For μ , it is clear from Fig. 7 that e -NSLS is getting better and better from 0 to 0.5, and performs well for all values of μ except very small ones between 0.5 and 1.0 on P2. For σ , we can observe that e -NSLS works well for all values of σ from 0.0 to 0.2. Therefore, we can claim that e -NSLS is not very sensitive to the setting of μ and σ .

There is another parameter ω , which is a control factor in e -NSLS. To study the sensitivity of e -NSLS to ω , we repeat the experiment in Section 5.3.4 with $K = 5, l_i = 0.01$, and $u_i = 0.8$ for $\omega \in [1, 7]$ with a step size one. Figure 8 illustrates the IGD values for different values of ω on the problem P5. It is obvious from the figure that the IGD value varies a little from 3 to 6. Therefore, it can be concluded that

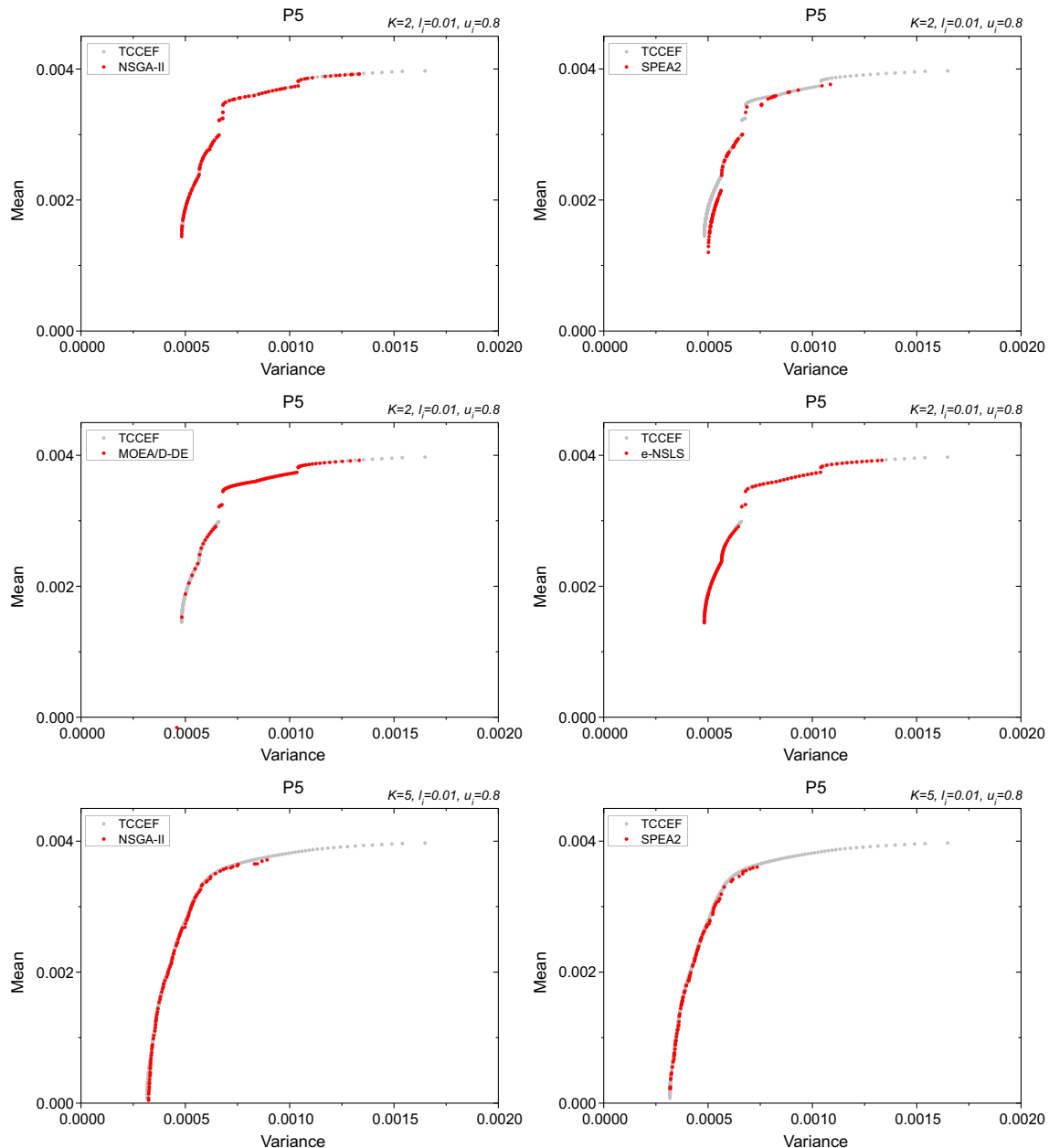


Fig. 6 Efficient frontiers for Japanese Nikkei 225 data set (P5) of $K = 2, 5, 10$

e-NSLS is not very sensitive to the setting of ω , and it has a better performance when ω is set as 3 or 4.

5.3.8 Experiment 5 : *e*-NSLS VS. ABC-FS and GRASP-QUAD

In this section, we extend the comparison with the results of ABC-FS [47] and GRASP-QUAD [49], which are two recent algorithms applied to solve the MVCCOP problem. It is important to note that there are two main differences between ABC-FS, GRASP-QUAD and our work. One is that the MVCCPO problem ABC-FC and GRASP-QUAD

solved has a different constraint. In this paper, the proposed algorithm is used to solve the MVCCPO problem applied in [31] and [38] with the constraint “ $\sum_{i=1}^n \theta_i \leq K$ ”, which means that the number of asserts a portfolio can hold is within range $[0, K]$, while in [47] and [49], the constraint is “ $\sum_{i=1}^n \theta_i = K$ ”, which means that a portfolio must hold exactly K asserts. The other one is that they consider the problem as a single objective optimization problem while in our paper, it is a bi-objective optimization problem.

In this section, we directly refer to the empirical results from [47] and [49]. The proposed algorithm *e*-NSLS is tuned to solve the MVCCPO problem with cardinality

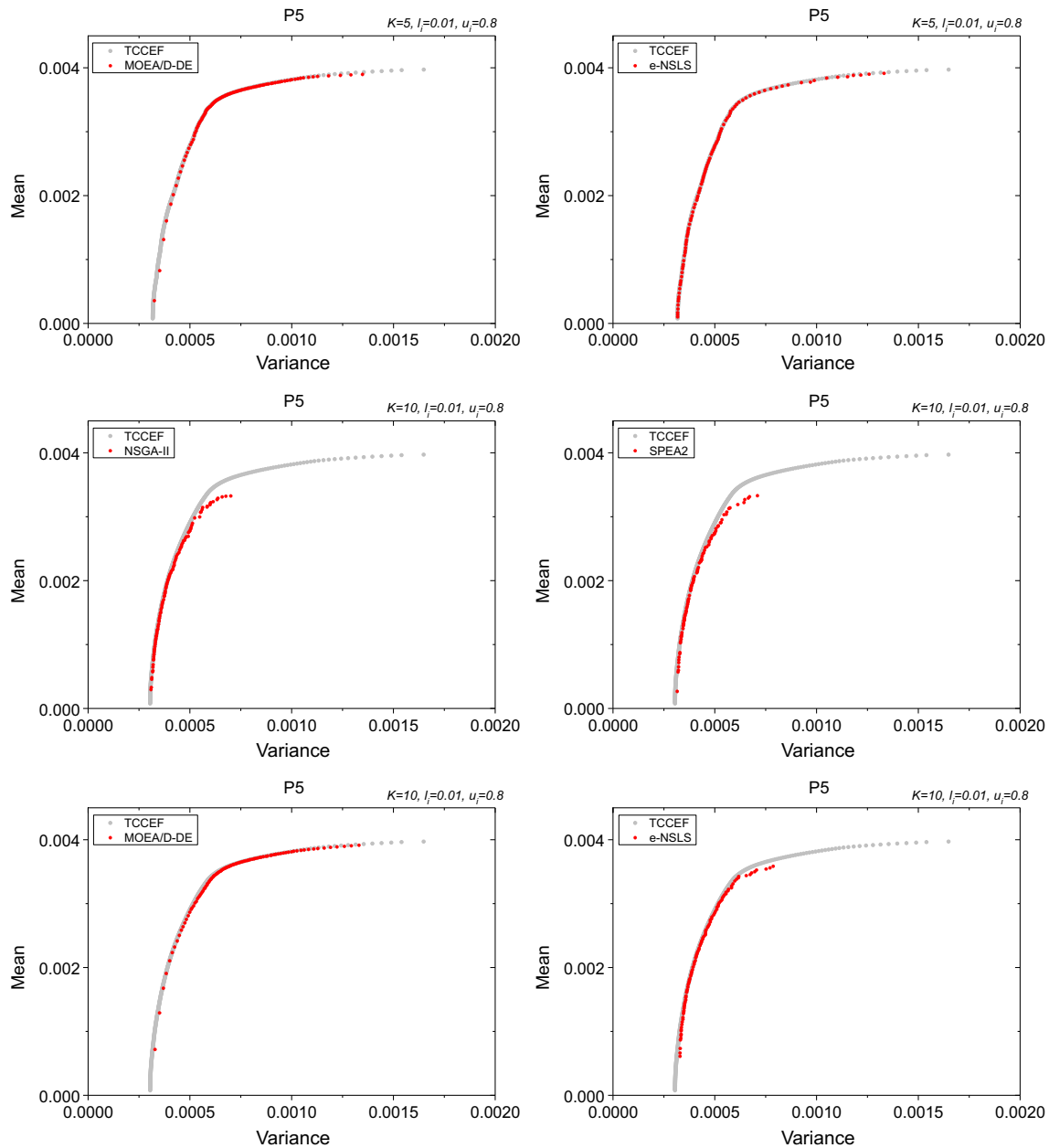


Fig. 6 (continued)

constraint “ $\sum_{i=1}^n \theta_i = K$ ”. To satisfy the cardinality constraint, when the number of asserts which are held by a portfolio holds is less than K , we choose the first assert (denoted as p) from the selected asserts and randomly select an assert (denoted as q) from the unselected asserts in the candidate list. Then, the proportion of q is allocated a half proportion of p . It is worthy mention that p and q need the additional procedure to satisfy the lower and upper limit when generated. The tuned e -NSLS to solve the problem in [49] is called as Ke -NSLS in brief.

Here, we use mean Euclidean distance in [32] to compare the true and the obtained efficiency frontier. In the

experiment, the MVCCPO problems are solved by using the parameter set in [49], namely, $N = 51, K = 10, l_i = 0.01$, and $u_i = 1.0$.

Table 13 Statistical results by Wilcoxon’s signed rank test for the IGD-metric considering all test problems

e -NSLS vs.	R^+	R^-	p -value
NSGA-II	281	44	0.001430282
SPEA2	311	14	0.000064510
MOEA/D-DE	265	60	0.005816359

Table 14 The spread-metric values on all five test problems with $l_i = 0.00, u_i = 0.8$

Instance mean	P1 ($K = 31$)	P2 ($K = 85$)	P3 ($K = 89$)	P4 ($K = 98$)	P5 ($K = 225$)
NSGA-II	4.59735e-01 (3.7586e-02)	5.82689e-01 (4.8150e-02)	6.2721e-01 (4.8763e-02)	5.96600e-01 (4.06910e-02)	9.0968e-01 (6.24810e-02)
SPEA2	0.63295e-01 (5.5065e-02)	7.08400e-01 (3.22080e-02)	7.39911e-01 (4.42450e-02)	6.88512e-01 (3.80210e-02)	8.75003e-01 (6.88650e-02)
MOEA/D-DE	7.51914e-01 (6.5118e-03)	1.008100e-00 (2.62130e-02)	9.94052e-01 (1.64380e-02)	8.99219e-01 (1.57960e-02)	9.85263e-01 (1.87320e-02)
<i>e</i> -NSLS	2.62023e-01 (3.2073e-02)	3.72841e-01 (2.4595e-02)	4.23952e-01 (4.02750e-02)	4.09717e-01 (3.69190e-02)	6.62137e-01 (2.93220e-02)

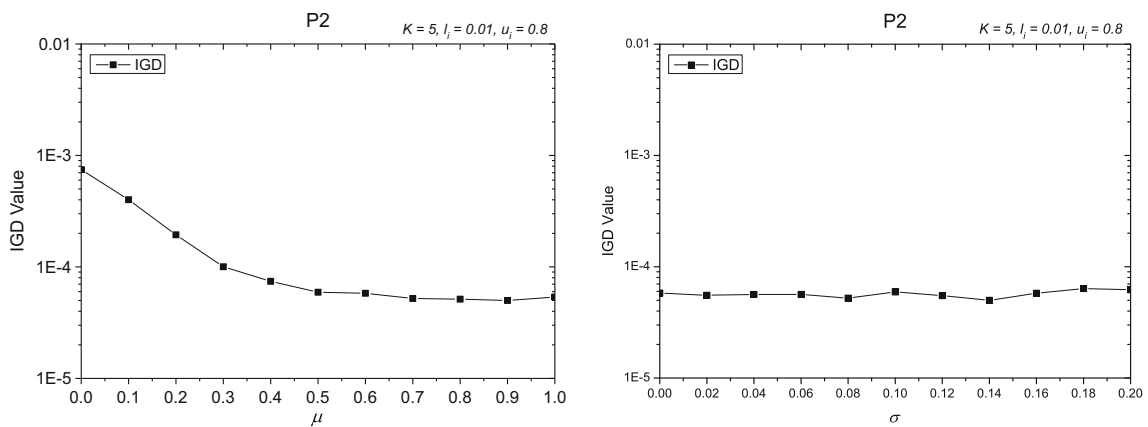


Fig. 7 The mean IGD-metric values versus the value of μ and σ in *e*-NSLS for P2

Fig. 8 The mean IGD-metric values versus the value of ω in *e*-NSLS for P5

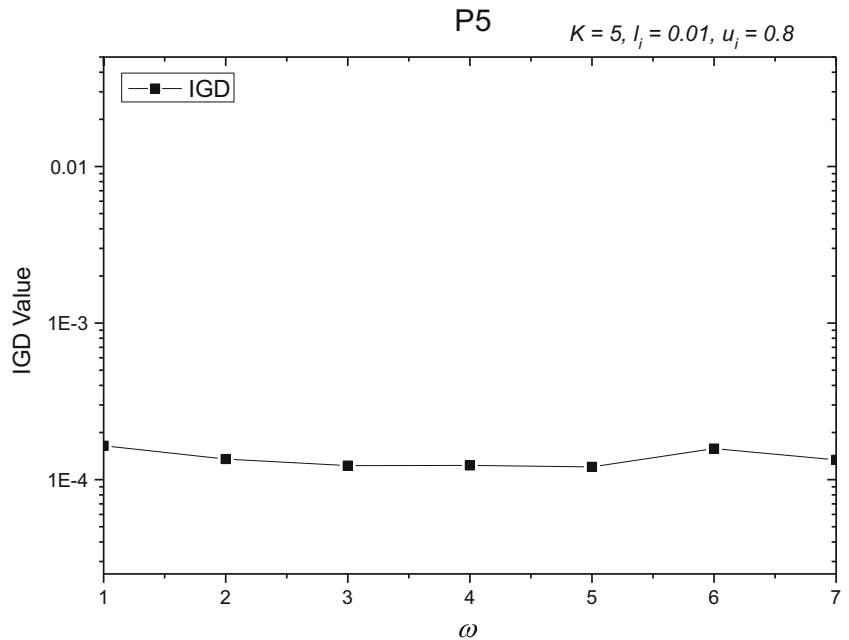


Table 15 The mean Euclidean distance-metric values of the non-dominated solutions found by each algorithm on all five test problems

Instance	n	ABC-FS	GRASP-QUAD	Ke -NSLS
P1	31	0.0004	0.0001	0.0001
P2	85	0.0009	0.0001	0.0000
P3	89	0.0003	0.0000	0.0000
P4	98	0.0001	0.0001	0.0000
P5	225	0.0000	0.0000	0.0000

The mean Euclidean distance is calculated as follows. Let the point (v_k^T, r_k^T) ($k = 1, 2, 3, \dots, \zeta$) denote the variance and the return of the point in the true efficiency frontier, and the point (v_i^A, r_i^A) ($i = 1, 2, 3, \dots, \xi$) represents the variance and return of the point in the efficient frontier obtained by the algorithm, where ζ is the number of the true efficient frontier points and ξ is the number of the efficient frontier points obtained by the algorithm. Denote the point $(v_{o_i}^T, r_{o_i}^T)$ be the closet point in the true efficient frontier to the point (v_i^A, r_i^A) .

$$o_i = \operatorname{agrm}in_{k=1,2,3,\dots,\zeta} (\sqrt{(v_k^T - v_i^A)^2 + (r_k^T - r_i^A)^2}) \tag{21}$$

The mean Euclidean distance is calculated as:

$$med = \frac{\sum_{i=1}^{\xi} \sqrt{(v_i^A - v_{o_i}^T)^2 + (r_i^A - r_{o_i}^T)^2}}{\xi} \tag{22}$$

Numerical results obtained by ABC-FS, GRASP-QUAD, and Ke -NSLS are given in Table 15, from which we can see the proposed algorithm obtains better (smaller) med values than ABC-FS and GRASP-QUAD for all five benchmark sets.

6 Conclusions

In this paper, we extend a local search based multi-objective optimization (NSLS), which is a combination of a new local search schema and a farthest-candidate approach to find the cardinality constrained efficient frontier on the MVCCPO problems. We have presented computational comparison of e -NSLS with the other five state-of-the-art algorithms on the public available benchmark data set containing the instances that range from 31 to 225 assets. The capabilities of these six algorithms are shown numerically, as well as, graphically, comparing them with the true efficient points. Computational results reveal a good performance of e -NSLS in the

six algorithms in term of the diversity, convergence towards the TCCEF, the coverage, and the stability.

Moreover, the Wilcoxon signed ranks test analysis has been used to statistically test the significant performance of e -NSLS with the other algorithms. The results reveal that e -NSLS has a better performance than the other three algorithms. In addition, the key steps of e -NSLS, namely, the boundary constraint handling strategy, DE operator, the replacement strategy and the farthest-candidate approach are modified one-by-one. The experimental results show that these modified algorithms failed to perform as efficiently as e -NSLS on the MVCCPO problems. Furthermore, The sensitivities to three parameters: μ, σ and ω of e -NSLS are also experimentally investigated. The results show that e -NSLS is not very sensitive to the settings of μ, σ and ω on the MVCCPO problem.

In the near further, improving e -NSLS for solving portfolio optimization problems with higher number of variables will be our work. It will also be meaningful to design a controlled rule to pursue the optimal number of assets. Meanwhile the investor’s decision is going to be considered.

Acknowledgments This work is supported by the National Nature Science Foundation of China under Grant No. 61272003, No. 60672018, No.40774065 and the Natural Science Foundation of Fujian Province, China under Grant No. 2013h0032, No. 2013j01243.

References

1. Markowitz HM (1952) Portfolio select. J Financ 7(1):7791
2. Markowitz HM (1956) The optimization of a quadratic function subject to linear constraints. Nav Res Log Quart 3(1-2):111133
3. Chang TJ, Meade N, Beasley JE, Sharaiha YM (2000) Heuristics for cardinality constrained portfolio optimisation. Comput Oper Res 27(11):1271–1302
4. Bienstock D (1996) Computational study of a family of mixed-integer quadratic programming problems. Math Program 74(2):121–140
5. Jobst NJ, Horniman MD, Lucas CA, Mitra G (2001) Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. Quant Financ 1(3):489–501
6. Li D, Sun X, Wang J (2006) Optimal lot solution to cardinality constrained meanvariance formulation for portfolio selection. Math Financ 16(1):83–101
7. Shaw DX, Liu S, Kopman L (2008) Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. Optim Methods Softw 23(1):411–420
8. Vielma JP, Ahmed S, Nemhauser GL (2008) A lifted linear programming branchand-bound algorithm for mixed-integer conic quadratic programs. INFORMS J Comput 20(1):438–450
9. Bertsimas D, Shioda R (2009) Algorithm for cardinality-constrained quadratic optimization. Comput Optim Appl 43(1):1–22
10. Gulpinar N, An LTH, Moeini M (2010) Robust investment strategies with discrete asset choice constraints using DC programming. Optimization 59(1):45–62

11. Padhye N, Bhardawaj P, Deb K (2013) Improving differential evolution through a unified approach. *J Global Optim* 55(2):771–799
12. Deb K, Padhye N (2014) Enhancing performance of particle swarm optimization through an algorithmic link with genetic algorithms. *Comput Optim Appl* 57(1):761–794
13. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA II. *IEEE Trans Evol Comput* 6(2):182–197
14. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-103. Department of Electrical Engineering, Swiss Federal Institute of Technology, Zurich, Switzerland
15. Chen BL, Zeng WH, Lin YB, Zhang DF (2015) A new local search-based multiobjective optimization algorithm. *IEEE trans Evol Comput* 19(1):50–73
16. Zhang QF, Li H (2007) MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(4):712–731
17. Li H, Zhang QF (2009) Multiobjective optimization problems with complicated pareto sets. MOEA/D NSGA-II *IEEE Trans Evol Comput* 13(2):284–302
18. Jiang S, Yang S (2016) An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts. *IEEE Trans Cybern* 46(2):421–437
19. Li K, Kwong S, Deb K (2015) A dual-population paradigm for evolutionary multiobjective optimization. *Inform Sci* 309:50–72
20. Zhang X, Tian Y, Cheng R, Jin Y (2015) An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Trans Evol Comput* 19(2):201–213
21. Zhu Q, Lin Q, Du Z, Liang Z, Wang W, Zhu Z, Chen J, Huang P, Ming Z (2016) A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm. *Inform Sci* 345:177–198
22. Shim VA, Tang TKC (2015) Adaptive memetic computing for evolutionary multiobjective optimization. *IEEE Trans Cybern* 45(2):610–621
23. Schaerf A (2002) Local search techniques for constrained portfolio selection problems. *Comput Econ* 20(1):170–190
24. Crama Y, Schyns M (2003) Simulated annealing for complex portfolio selection problems. *Eur J Oper Res* 150(1):546–571
25. Derigs U, Nickel NH (2003) Metaheuristic based decision support for portfolio optimisation with a case study on tracking error minimization in passive portfolio management. *OR Spectr* 25(25):345–378
26. Derigs U, Nickel N-H (2004) On a local-search heuristic for a class of tracking error minimization problems in portfolio management. *Ann Oper Res* 131(131):45–77
27. Ehrgott M, Klamroth K, Schwehm C (2004) An MCDM approach to portfolio optimization. *Eur J Oper Res* 155(1):752–770
28. Streichert F, Tanaka-Yamawaki M (2006) The effect of local search on the constrained portfolio selection problem. In: *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp 2368–2374
29. Fernández A, Gómez S (2007) Portfolio selection using neural networks. *Comput Oper Res* 34(2):1177–1191
30. Chiam SC, Tan KC, Al Mamun A (2008) Evolutionary multi-objective portfolio optimization in practical context. *Int J Autom Comput* 5(1):67–80
31. Branke J, Scheckenbach B, Stein M, Deb K, Schmeck H (2009) Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *Eur J Oper Res* 199(1):684–693
32. Cura T (2009) Particle swarm optimization approach to portfolio optimization. *Nonlinear Anal Real* 10(2):2396–2406
33. Pai GAV, Michel T (2009) Evolutionary optimization of constrained k-means clustered assets for diversification in small portfolios. *IEEE Trans Evol Comput* 13(3):1030–1053
34. Soleimani H, Golmakani HR, Salimi MH (2009) Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Syst Appl* 36(1):5058–5063
35. Chang TJ, Yang SC, Chang KJ (2009) Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Syst Appl* 36(5):10529–10537
36. Anagnostopoulos KP, Mamanis G (2010) A portfolio optimization model with three objectives and discrete variables. *Comput Oper Res* 37(5):1285–1297
37. Corne DW, Knowles JD, Oates MJ The Pareto envelope-based selection algorithm for multi-objective optimization. In: *Proceedings of the parallel problem solving from nature (PPSN VI)*, pp 839–848
38. Anagnostopoulos KP, Mamanis G (2011) The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Syst Appl* 38(9):14208–14217
39. Erickson MA, Maye HJ (1993) The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. *Lect Notes Comput Sci* 1993(1993):681–695
40. Hanne T (2007) A multiobjective evolutionary algorithm for approximating the efficient set. *Eur J Oper Res* 176(1):1723–1734
41. Ruiz-Torrubiano R, Suarez A (2010) Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints. *IEEE Comput Intel Mag* 5(2):92–107
42. Woodside-Oriakhi M, Lucas C, Beasley JE (2011) Heuristic algorithms for the cardinality constrained efficient frontier. *Eur J Oper Res* 213(1):538550
43. Lwin K Qu R (2013) A hybrid algorithm for constrained portfolio selection problems. *Appl Intel* 39(2):251–266
44. Murray W, Shek H (2012) A local relaxation method for the cardinality constrained portfolio optimization problem. *Comput Optim Appl* 53(1):681709
45. Deng GF, Lin WT, Lo CC (2012) Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Syst Appl* 39(2):4558–4566
46. Liagkouras K, Metaxiotis K (2014) A new probe guided mutation operator and its application for solving the cardinality constrained portfolio optimization problem. *Expert Syst Appl* 41(12):6274–6290
47. Tuba M, Bacanin N (2014) Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio selection problem. *Appl Math Inform Sci* 8(4):2831–2844
48. Cui T, Cheng S, Bai R (2014) A combinatorial algorithm for the cardinality constrained portfolio optimization problem
49. Baykasoğlu A, Yunusoglu MG, Özsoydan FB (2015) A grasp based solution approach to solve cardinality constrained portfolio optimization problems. *Comput Ind Eng* 90(C):339–351
50. Ruiz-Torrubiano R, Suárez A (2015) A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. *Appl Soft Comput* 36(C):125–142
51. Metaxiotis K, Liagkouras K (2012) Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature reviews. *Expert Syst Appl* 39(12):11685–11698
52. Holland JH (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, MI, USA
53. Padhye N, Mittal P, Deb K (2015) Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Comput Optim Appl* 62(1):851–890
54. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Da Fonseca VG (2003) Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans Evol Comput* 7(2):117–132

55. Cesarone F, Tardella SA (2008) Efficient algorithms for mean-variance portfolio optimization with real-world constraints. In: Proceedings of the 18th AFIR colloquium: Financial risk in a changing world, Rome
56. Wang YN, Wu LH, Yuan XF (2010) Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Comput* 14(1):193–209
57. Garcia S, Molina D, Lozano M, Herrera F (2009) A Study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *J Heuristics* 15(4):617–644
58. Rechenberg I (1971) *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Ph.D. Thesis, Technical University of Berlin, Department of Process Engineering
59. Rechenberg I (1973) *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart