Applied Soft Computing 24 (2014) 757–772

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization

Khin Lwin*, Rong Qu, Graham Kendall

ASAP Group, School of Computer Science, University of Nottingham, UK

ARTICLE INFO

Article history:
Received 2 August 2013
Received in revised form 30 March 2014
Accepted 11 August 2014
Available online 27 August 2014

ABSTRACT

Portfolio optimization involves the optimal assignment of limited capital to different available financial assets to achieve a reasonable trade-off between profit and risk objectives. In this paper, we studied the extended Markowitz's mean-variance portfolio optimization model. We considered the cardinality, quantity, pre-assignment and round lot constraints in the extended model. These four real-world constraints limit the number of assets in a portfolio, restrict the minimum and maximum proportions of assets held in the portfolio, require some specific assets to be included in the portfolio and require to invest the assets in units of a certain size respectively. An efficient learning-guided hybrid multi-objective evolutionary algorithm is proposed to solve the constrained portfolio optimization problem in the extended mean-variance framework. A learning-guided solution generation strategy is incorporated into the multi-objective optimization process to promote the efficient convergence by guiding the evolutionary search towards the promising regions of the search space. The proposed algorithm is compared against four existing state-of-the-art multi-objective evolutionary algorithms, namely Non-dominated Sorting Genetic Algorithm (NSGA-II), Strength Pareto Evolutionary Algorithm (SPEA-2), Pareto Envelope-based Selection Algorithm (PESA-II) and Pareto Archived Evolution Strategy (PAES). Computational results are reported for publicly available OR-library datasets from seven market indices involving up to 1318 assets. Experimental results on the constrained portfolio optimization problem demonstrate that the proposed algorithm significantly outperforms the four well-known multi-objective evolutionary algorithms with respect to the quality of obtained efficient frontier in the conducted experiments.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

Portfolio selection problem is a well-studied topic in finance and it is concerned with the optimal allocation of a limited capital among a finite number of available risky assets, such as stocks, bonds, and derivatives in order to gain the possible highest future wealth. Markowitz's mean-variance model [40,41] is considered to play an important role in the development of Modern Portfolio Theory. The mean-variance (MV) model assumes that the future market of the assets can be correctly reflected by the historical market of the assets. It considers the trade-off between risk and reward in selecting *efficient* portfolios. A portfolio is considered to be *efficient* if it provides the highest possible reward for a given risk or alternatively, if it presents the least possible risk for a given level of profit. The reward (profit) of the portfolio is measured by the average expected return of those individual assets in the portfolio whereas the risk is measured by its combined total variance.

While investing the capital within the MV framework, investors have *two* objectives: maximizing the total profit and minimizing the total risk of their portfolios. With these two conflicting objectives to be optimized simultaneously, the portfolio selection problem can be classified as a multi-objective optimization problem. A single solution that optimizes all the conflicting objectives simultaneously hardly exists in practice. Instead, there exists a set of acceptable 'compromise' solutions which are optimal in such a way that no other solutions are superior to them when all objectives are considered simultaneously. Such solutions are referred to as *efficient* solutions, *non-dominated* solutions or *Pareto-optimal* solutions.

The collection of such efficient portfolios conveying the compromise between risk and return is called the *efficient frontier* or *Pareto-optimal front*. The efficient frontier helps investors to visualize the risk and return trade-off curve in a two-dimensional graph with risk on the horizontal axis and expected return on the vertical axis (see Fig. 13).

* Corresponding author at: C87, School of Computer Science, University of Nottingham, Nottingham NG8 1BB, United Kingdom. Tel.: +44 0115 84 66525.
E-mail address: psxktl@nottingham.ac.uk (K. Lwin).

http://dx.doi.org/10.1016/j.asoc.2014.08.026
1568-4946/© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

Since the Markowitz's pioneering work, many researchers have pursued studies for efficient algorithms [27,29,43,52] to compute the efficient frontier of the MV model. However, the classic MV model assumes a perfect market where short sales are disallowed, securities can be traded in any (non-negative) fractions, no limitation on the number of assets in the portfolio, investors have no preferences over assets and they do not care about different assets types in their portfolios. In practice, these assumptions are unrealistic. As a result, several extensions and modifications have been proposed to address the real-world constraints. In this paper, we extended the basic MV model to include four practical constraints as follows:

Cardinality constraint

Cardinality constraint limits the number of assets (K) that compose the portfolio. Very often in practice, investors prefer to have a limited number of assets included in their portfolio since the management of many assets in the portfolio is tedious and hard to monitor. They also intend to reduce transaction costs and/or to assure a certain degree of diversification by limiting the maximum number of assets in their portfolios.

Floor and ceiling constraints

The floor and ceiling constraints specify the minimum and maximum limits on the proportion of each asset that can be held in a portfolio. In practice, investors prefer to avoid excessive administrative costs for very small holdings of assets in the portfolio and/or some institutional policies require to model their policies on the lower and upper bounds of each asset in the portfolio. The floor and ceiling constraint is also known as bounding or quantity constraints.

Pre-assignment constraint

The pre-assignment constraint is usually used to model the investor's subjective preferences. An investor may intuitively wish some specific assets to be included in the portfolio, with its proportion fixed or to be determined.

Round lot constraint

Round lot constraint requires the number of any asset in the portfolio to be in exact multiple of the normal trading lots. In practice, several market securities are traded as multiples of minimum lots.

These four constraints stated above are *hard* in the sense that they have to be satisfied at any time. In practice, portfolios are composed of markets with hundreds to thousands of available assets, and the calculation of risk measures grows quickly in relation to the number of assets. By introducing the cardinality constraint alone already transforms the classic quadratic optimization model into a mixed-integer quadratic programming problem which is an NP-hard problem [6,47]. There are several exact approaches proposed in the literature for cardinality constrained portfolio optimization problem [5,6,35,47]. However, all these works relaxed the cardinality constraint as an inequality constraint allowing the number of assets in the portfolio to vary with maximum bound (K) and the results showed that they are able to handle the test problems with limited size (up to 500 assets). On the other hand, Gulpinar et al. [26] considered the strict cardinality constraint and computational results are performed on a small test problem involving 98 assets.

When additional constraints are added to the basic MV model, the problem thus becomes more complex and the exact optimization approaches run into difficulties to deliver solutions within reasonable time for large problem size. As a result, this motivates the investigation of approximate algorithms such as meta-heuristics [33] and hybrid meta-heuristics [56,45]. In general, meta-heuristics cannot guarantee the optimality of the solution, but they are efficient in finding the optimal or near optimal solutions in a reasonable amount of time.

There exist many studies which applied meta-heuristics or other techniques to solve portfolio optimization problem [21,39]. The recent research in portfolio optimization problem is widely carried out by incorporation of constraints in the problem model and/or handling the problem as a multi-objective one. Although the portfolio optimization problem involves two conflicting objectives, many studies in the literature [11,17,20,37] have been performed as single objective meta-heuristics approaches with aggregating function that combines two objectives into a single scale objective, and in which the weights are varied to generate the set of efficient solutions for portfolio selection problems with cardinality and quantity constraints. Mansini and Speranza [38] showed that the portfolio selection problem with round lot constraint is an NP-complete problem and proposed three mixed integer linear programming heuristic algorithms to solve the problem. Lin and Liu [36] proposed a genetic algorithm with three different models for portfolio selection problems with round lots. Chang et al. [11] and Gaspero et al. [25] discussed the pre-assignment briefly but had not addressed the constraint in their experiments.

In recent years, many publications had discussed the portfolio optimization problems with multi-objective evolutionary algorithms by considering a subset of the real-world constraints. Diosan [22] and Mishra et al. [42] applied several well-known multi-objective evolutionary algorithms to solve the unconstrained portfolio optimization problem. Recently, Krink et al. [34] also proposed an algorithm called DEMPO inspired by the NSGA-II algorithm [19]. The difference between NSGA-II and DEMPO is that Differential Evolution (DE) is used instead of Genetic Algorithm (GA) to generate new candidate solutions during the evolution. DEMPO is applied to solve the basic portfolio optimization problem based on Value-at-Risk risk measure and experimental results show that DEMPO outperforms NSGA-II. Armananzas and Lozano [3] studied greedy search, simulated annealing (SA) and ant colony optimization (ACO) algorithms in a multi-objective framework to solve the portfolio selection problem with cardinality constraints.

Anagnostopoulos and Mamanis [2] considered the extended MV model with cardinality and quantity constraints and tested five advanced MOEAs to investigate the performance. The cardinality constraint considered in their work is relaxed and as a result a portfolio can be composed of any number of assets with maximum bound (K). The experimental results confirmed that all multi-objective algorithms considered outperformed the single objective evolutionary algorithm. The results also concluded that SPEA-II [60] performed the best among those algorithms tested. Branke et al. [7] also presented an envelope-based MOEA integrating the NSGA-II [18] and the critical line algorithm. Chaim et al. [12] proposed an order-based solution representation and considered the cardinality constraint as a soft constraint and quantity constraint as a hard constraint. In their work, the cardinality constraint was relaxed and hence it was allowed to vary the number of assets in the portfolio from the minimum limit to the maximum limit.

Streichert et al. [55,54] applied a multi-objective evolutionary algorithm (MOEA) to solve the portfolio selection problems with cardinality, floor and round lot constraints. These works studied various crossover operators adopting hybrid chromosome representation with binary and real values. This hybrid encoding enhances the performance of the algorithm significantly regardless of the choice of crossover operators. Skolpadungket et al. [50] also studied the portfolio selection problems with cardinality, floor and round lot constraints and tested them with various MOEAs. They adopted the same hybrid encoding as Streichert et al. [55,54].

Experiments are performed on the small dataset containing 31 assets and the performance metrics showed that SPEA-II [60] is the best algorithm among those tested. In their work, the cardinality constraint was relaxed and only the maximum cardinality constraint was considered.

Fieldsend et al. [23] and Anagnostopoulos and Mamanis [1] considered the cardinality constraint as an additional objective to be minimized. Brito and Vicente [8] reformulated the cardinality constrained MV model as a bi-objective problem, allowing the investors to analyze the efficient trade-off between mean-variance and cardinality. The detailed reviews of the multi-objective evolutionary algorithms in portfolio optimization can be found in [10,13,44,49].

In this work, we propose a new learning-guided hybrid evolutionary algorithm for the mean-variance portfolio optimization problem within the context of the multi-objective optimization. We extended the MV model to consider the strict cardinality, quantity, pre-assignment and round lot constraints.

We for the first time investigate the performance of the learning-guided multi-objective evolutionary algorithm with external archive (MODEwAwL) on the extended MV model with four constraints considered. Randomly generating a new candidate solution is very unlikely to achieve a good-quality practical solution for the constrained portfolio optimization problem. Instead, a learning-guided solution generation scheme incorporating additional problem-specific heuristics is proposed to generate a good-quality solution. The proposed algorithm contributes to enhance an efficient convergence of the search algorithm by concentrating on the promising areas of the search space.

In this study, we consider four existing well-known multi-objective evolutionary algorithms (MOEAs), the Non-dominated Sorting Genetic Algorithm (NSGA-II) [19], the Strength Pareto Evolutionary Algorithm (SPEA2) [60], Pareto Envelope-based Selection Algorithm (PESA-II) [16] and Pareto Archived Evolution Strategy (PAES) [30]. A large set of simulation experiments have been conducted over a number of instances. Results demonstrate that the proposed algorithm is highly efficient in terms of both finding solutions close to the true Pareto-front and good distribution along the Pareto-front.

The rest of the paper is organized as follows. Section 2 describes the generic multi-objective portfolio optimization, followed by the extended MV model considering the cardinality, quantity, pre-assignment and round lot constraints. Section 3 introduces the proposed algorithm outlining the main differences from the existing approaches. Section 4 provides the detailed structure of the proposed algorithm. Section 5 discusses the analysis of the simulation results. In Section 6, conclusion and future work are presented.

2. Multi-objective portfolio optimization

Multi-objective optimization generally involves balancing all conflicting objectives and searches for a *set* of compromise solutions between the objectives while satisfying the various constraints. In such context, this set of solutions is known as *Pareto-optimal* solutions [18].

In multi-criteria variant of portfolio optimization problem, the MV model can be formalized as a bi-objective optimization problem. The objective is to find a set of *efficient* portfolios that maximize return and minimize risk simultaneously. In this work, four real-world constraints, cardinality, quantity, pre-assignment and round lot, are considered (see Section 1). Mathematically, the problem with considered constraints can be formulated as follows:

$$\min \quad f_1 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{1}$$

$$\max \quad f_2 = \sum_{i=1}^{N} w_i \mu_i \tag{2}$$

$$\textit{subject to} \quad \sum_{i=1}^{N} w_i = 1 \tag{3}$$

$$\sum_{i=1}^{N} s_i = K, \tag{4}$$

$$w_i = y_i.\upsilon_i, \qquad i = 1, \ldots, N, \quad y_i \in \mathbb{Z}_+ \tag{5}$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, \qquad i = 1, \ldots, N, \tag{6}$$

$$s_i \geq z_i, \qquad i = 1, \ldots, N \tag{7}$$

$$s_i, z_i \in \{0, 1\}, \qquad i = 1, \ldots, N \tag{8}$$

where N is the number of available assets, μ_i is the expected return of asset i ($i = 1, \ldots, N$), σ_{ij} is the covariance between assets i and j ($i = 1, \ldots, N; j = 1, \ldots, N$), and w_i ($0 \leq w_i \leq 1$) is the decision variable which represents the proportion held of asset i. Eq. (3) defines the *budget constraint* (all the money available should be invested) for a feasible portfolio.

Eq. (4) defines the cardinality constraint where K is the number of invested assets in the portfolio and s_i denotes whether asset i is invested or not. If s_i equals to one, asset i is chosen to be invested and the proportion of capital w_i lies in $[\epsilon_i, \delta_i]$, where $0 \leq \epsilon_i \leq \delta_i \leq 1$. Otherwise, asset i is not invested and w_i equals to zero.

In this study, we adopted the *strict* cardinality constraint [3,11,37,39,55,54] and thus require to select fixed K number of assets. Experimental results from the literature [11,55] showed that when the cardinality constraint with high K value is imposed, the approximation of the constrained efficient frontier tends to approach towards the unconstrained efficient frontier (UCEF). The cardinality constraint has been relaxed in several related works [2,7,50], where the equality constraint is replaced by inequality constraint (i.e. up to K assets can be included in the portfolio). In some works [12,25], the cardinality constraint is alternatively relaxed by specifying the maximum and minimum number of assets that a portfolio can hold.

Eq. (7) defines the pre-assignment constraint to fulfil the investors' subjective requirements where the binary vector z_i denotes if asset i is in the pre-assigned set that has to be included in the portfolio or not. Eq. (5) defines the round lot constraint where y_i is a positive integer variable and υ_i is the minimum lot that can be purchased for each asset. The inclusion of round lot constraint may make it impossible to exactly satisfy the budget constraint (see Eq. (3)) as the total capital might not be the exact multiples of the required trading lot for various assets.

The above stated model could be solved by obtaining a set of efficient portfolios. These obtained solutions are optimal in the sense that there are no other solutions in the solution domain or search space that are superior to them when all objectives are considered simultaneously [18]. The complete set of these efficient portfolios forms the *efficient frontier* that represents the best trade-offs between the mean return and the variance (risk). In practice, when more real-world constraints are considered, the efficient frontier reduces to a smaller curve.

In a two-dimensional space of risk and return, a solution a is said to be *efficient* (i.e., Pareto-optimal) if there does not exist any solution b such that b dominates a [24]. Solution a is considered to dominate solution b if and only if:

$$f_1(a) \leq f_1(b) \quad \text{AND} \quad f_2(a) > f_2(b)$$
$$\text{OR}$$
$$f_2(a) \geq f_2(b) \quad \text{AND} \quad f_1(a) < f_1(b)$$

The ultimate goals in a multi-objective portfolio optimization problem are to find a set of solutions as close as possible to the Pareto-optimal front and to find a good distribution of solutions along the Pareto front. Once the efficient frontier is obtained, the decision maker determines the portfolio based on the investor's risk preference. Hence, the diversity of the solutions along the efficient frontier is important for the decision maker not to miss certain trade-off portfolios which he/she might be interested.

3. Learning-guided multi-objective evolutionary algorithm (MODEwAwL)

The multi-objective portfolio optimization problem becomes too complex to solve by numerical methods when those practical constraints reflecting investors' preferences and/or institutional trading rules are considered. Over the last two decades, multi-objective evolutionary algorithms (MOEAs) have received a significant amount of attention and demonstrated their effectiveness and efficiency in approximating the Pareto-optimal front [13].

DEMO [46] is one of the recent algorithms which combines the advantages of DE [53] with the mechanisms of Pareto-based sorting and crowding distance sorting [19]. It had been successfully tested on the carefully designed test functions (ZDT) introduced in [59]. The procedure of the DEMO is described in Fig. 1. DEMO maintains a population of individuals, where each represents a potential solution to the optimization problem. During the evolution, it allows its population capacity expand in order to add newly found non-dominated solutions (see Fig. 1, lines 3–9). Hence, it enables the newly found non-dominated solutions to immediately take part in the generation of the subsequent candidate solutions. This feature of DEMO promotes fast convergence towards the true Pareto front. In each generation, if the population exceeds the size limit, it is sorted based on the non-domination and crowding distance metrics [19] in order to identify those individuals to be truncated. It thus aims to maintain a good distribution of non-dominated portfolios.

In this work, we propose a learning-guided multi-objective evolutionary algorithm (MODEwAwL) for the constrained portfolio optimization. The proposed algorithm adopts a new approach to extend generic DEMO scheme to solve the constrained portfolio optimization problem. The main differences of our approach with respect to the DEMO scheme in the literature can be outlined as follows:

1. A secondary population (i.e. an external archive) is introduced to store the well spread non-dominated solutions found throughout the evolution (see Section 4.9).
2. A learning mechanism is proposed to extract the important features from the efficient solutions found throughout the evolution (see Section 4.4).
3. An efficient solution generation scheme utilizing the learning mechanism, problem specific heuristics and effective direction-based search methods is proposed to guide the search towards the promising search space (see Section 4.5).

The proposed MODEwAwL use the archive to extract the important features of non-dominated solutions. Incorporating learning mechanism and prior problem-specific knowledge exploitation in the evolution process allows the proposed MODEwAwL to generate promising offspring solutions. The proposed MODEwAwL thus aims to promote convergence by concentrating on the promising regions of the search space. The pseudocode of the proposed algorithm is described in Fig. 2.

Differential Evolution for Multi-objective Optimization

1\. evaluate the initial population P of random individuals.
2\. **while** stopping criterion not met:
3\.   **for** each individual $p_i (i = 1, ..., P_{Size})$
4\.   create a candidate p' from parent p_i
5\.   evaluate p'.
6\.   **if** p' dominates p_i, p' replaces p_i.
8\.   **else if** p_i dominates p', discard p'.
9\.   **else** add p' to P.
10\.   **if** $\mid P \mid \geq P_{Size}$, truncate it.
11\.   randomly enumerate the individuals in P.

Fig. 1. The procedure of DEMO [46].

Pseudocode: MODEwAwL

INITIALIZATION:
  randomly create initial population P.
  maintain the archive A with non-dominated solutions from P.
while stopping criterion not met:
  LEARNING MECHANISM:
    learn from the archive A to identify the promising asset(s)
  EVOLVE:
    for each individual $p_i (i = 1, ..., NP)$ in P
      CANDIDATE GENERATION:
        create new candidate p' from P and learning mechanism.
      REPAIR:
        repair p' **if** constraints are violated.
        evaluate the candidate p' by f_1 and f_2 (see Eq.1,2)
      SELECT:
        if p' dominates p_i, p' replaces p_i.
        else if p_i dominates p', discard p'.
        else add p' to the current population P.
  TRUNCATE:
    if $\mid P \mid \geq NP$
      maintain P with best NP solutions, ranked by non-domination
      and crowding distance metrics
  ARCHIVE:
    maintain the archive A with non-dominated solutions from P
    if $\mid A \mid \geq M$
      maintain A with M least crowded non-dominated solutions
  randomly enumerate the individuals in P
Output: the non-dominated solutions in the archive.

Fig. 2. The procedure of the proposed MODEwAwL.

4. The proposed MODEwAwL

4.1. Notation

Let

A	the archive maintaining the set of non-dominated portfolio(s)
CR	the crossover probability for differential evolution
F	the scaling factor for differential evolution
K	the number of assets in a portfolio, i.e. the cardinality
L	the number of assets in the pre-assignment set
M	the maximum size of the archive
N	the number of available assets
NP	the number of individuals in the population
P	list of portfolios in the population
c_i	the concentration of ith asset in the archive
p_i	the ith portfolio in the population

w_i the proportion of capital invested in the ith asset
υ_i the minimum trading lot of the ith asset
ϵ_i the lower bound on the proportion of the ith asset
δ_i the upper bound on the proportion of the ith asset
$r[x_1, x_2]$ random real value between x_1 and x_2, both inclusive
$R[x_1, x_2]$ random integer value between x_1 and x_2, both inclusive

$$s_i = \begin{cases} 1 & \textbf{if} \text{ the } i\text{th } (i = 1, \ldots, N) \text{ asset is chosen} \\ 0 & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1 & \textbf{if } i\text{th asset is in pre} - \text{assigned set} \\ 0 & \text{otherwise} \end{cases}$$

4.2. Solution representation and encoding

In our solution representation, two vectors of size N are used to define a portfolio p: a binary vector s_i, $i = 1, \ldots, N$ denoting whether asset i is included in the portfolio, and a real-value vector w_i, $i = 1, \ldots, N$ representing the proportions of the capital invested in the assets. Some existing researches [2,50,55,54] adopt similar encoding to define a portfolio. When the cardinality and pre-assignment constraints are considered, the introduction of binary variables s_i in the multi-objective portfolio model enhances the evaluation of the algorithm.

4.3. Initial population generation

To generate an initial population, K different indexes (including all assets in the pre-assignment subset) are randomly selected and proportions are assigned to those selected assets randomly. If the generated portfolio violates the budget and/or quantity constraints, such solution is corrected by the constraint handling techniques provided in Section 4.6. Hence, all generated solutions in the trial population are feasible.

4.4. Learning mechanism

At each generation, the distribution of assets from non-dominated solutions in the external archive is observed to identify the promising assets. The concentration score of each asset c_i is calculated by counting its occurrences in the archive divided by archive size.

$$c_i = \frac{\sum_{j=1}^{|A|} s_{i,j}}{|A|}.$$

The new solutions to be generated are encouraged to compose with those assets by exploiting the knowledge obtained throughout the evolution to direct the search towards the promising search space. The proposed learning mechanism is computationally cheap as it only uses a single update at each generation. A similar form of scoring function has been used as one of the components in the trade-off studies by Smith et al. [51].

4.5. Candidate generation

One of the factors to consider in designing the portfolio model in the proposed MODEwAwL is to find an effective way to generate offspring. We aim to find effective and efficient scheme with a good balance between the exploitation and exploration. The new solution is generated by two phases: the selection of assets from a universe of N available assets and the allocation of capital to those selected assets. The idea presented here is to use DE for exploring the real decision variables and exploit learning mechanism and problem specific heuristics described below to select the promising assets in the new solution.

The information about the concentration of the assets in the non-dominated portfolios in the archive is exploited in selecting the promising assets for the new candidate portfolio. Hence the assets are ranked according to their concentrations in the archive non-dominated solutions. The assets which score greater than zero are considered to be promising ones. The higher the score of the asset, the higher its chances to be included in the new candidate portfolio (see Section 4.4).

In finance literature, it is considered to be a fundamental premise to utilize assets that have low correlation with each other. Hence the assets which are less correlated to each other are preferable to the heavily correlated assets. It is also commonly believed that it is beneficial to reduce the portfolio's standard deviation of return. Intuitively, investors prefer higher return assets with less risk [28].

In order to generate a new candidate solution, the L assets are firstly selected if the pre-assignment constraint is considered. By taking into account of the above stated intuitive learning, in this work, the proposed MODEwAwL then alternatively uses the following selection schemes to fill the remaining assets:

S1 The $(K - L)$ assets are selected by roulette wheel selection based on the concentration score c_i.
S2 The $(K - L)$ assets which have the highest concentration score c_i are selected.
S3 The $(K - L)$ assets which have the highest expected return values are selected.
S4 The random n assets (where $n = R[0, K - L]$) which have the highest concentration score c_i are selected. The remaining $(K - n)$ assets are filled by selecting one of the following methods.

- Select those assets which have the lowest risk values.
- Select those assets which have the highest return values (i.e. **S3**).
- Select those assets which have the least correlation from those n assets already chosen.

By adopting the above stated selection scheme, the new candidate solution satisfies the pre-assignment and cardinality constraints. The proportions of those selected assets for the new candidate solution are assigned by using a direction-based offspring generation scheme where $p1$, $p2$ and $p3$ are randomly selected portfolios from the current population P as follows:

W1 $w'_i := w3_i + r[0, 1] \times (w1_i - w2_i)$
W2 $w'_i := w3_i + F \times (w1_i - w2_i)$
W3 rank $p1$, $p2$ and $p3$ by dominance and crowding distance measure (i.e. $p1$ is the best portfolio and $p3$ is the worst portfolio among three portfolios) and generate weight allocations of candidate portfolio by directing away from $p3$ and towards the middle between $p1$ and $p2$ as follows:

$$w'_i := \frac{w1_i + w2_i}{2}$$

The detailed procedure of the candidate generation is provided in Fig. 3. The proposed candidate generation mechanism intends to guide the search toward promising direction by learning from the reference assets from the archive and reference proportions from the current population. In this way, the proposed algorithm converges efficiently. The new candidate portfolio is repaired if the quantity and round lot constraints are violated (see the repair mechanism in Section 4.6).

Candidate Generation

```
Input: concentration score of assets c_i (i = 1, ..., N)
  randomly select K assets by S1, S2, S3 or S4.
  randomly select an index i from those K selected and assign i to j and χ
  for each selected asset
      randomly select three different portfolios: p1, p2, p3 ∈ P
      if   r(0,1) < CR   ||   j == χ
          allocate weight w' by W1, W2 or W3
      else
          allocate weight w' randomly.
      randomly select an index i from those K selected and assign i to j
Output: candidate solution p'.
```

Fig. 3. The procedure of generating a candidate solution.

4.6. Constraint handling

When using an evolutionary algorithm to solve constrained optimization problems, there are various methods proposed in the literature [15] for handling constraints in evolutionary optimization, such as penalty function method, special representations and operators, repair methods and multi-objective methods. Among those methods, repair method is one of the effective approaches to locate feasible solutions.

During the population sampling, each constructed individual portfolio is repaired if it does not satisfy all considered constraints. As described in Section 4.5, the new solution generated by our proposed MODEwAwL already satisfies the cardinality and pre-assignment constraints.

Hence, the following repair mechanism stated in [50,54] is applied:

1. All weights of selected asset in the candidate solution are adjusted by setting $w'_i = \epsilon_i + (w'_i - \epsilon_i)/\sum(w'_i - \epsilon_i)$.
2. The weights are then adjusted to the nearest round lot level by setting $w'_i = w'_i - (w'_i \mod \upsilon_i)$. The remaining amount of capital is redistributed in such a way that the largest amount of $(w'_i \mod \upsilon_i)$ is added in lot of υ_i until all the capital is spent.

4.7. Selection scheme

The proposed MODEwAwL applies the elitist selection scheme based on Pareto optimality (see Fig. 2). During the evolution, the population is extended by adding the newly found non-dominated solutions. Hence, at each generation, the number of portfolios in the current population will be between *NP* and 2*NP*.

4.8. Truncate population

In each generation, if the number of portfolios in the current population exceeds its limit *NP*, it needs to identify those which need to be removed. The individuals in the population are sorted based on the non-dominance and crowding distance metrics. Then the current population is truncated by keeping the best *NP* individuals for the next generation.

4.9. Maintaining the external archive

The main objective of the external archive *A* is to keep all the non-dominated solutions encountered along the search process. This approach is adopted in order to save and update all well spread non-dominated solutions generated by the algorithm during the search.

In each generation, the archive *A* is updated with the non-dominated solutions from the trial population. The computational time of maintaining the archive increases with the archive size [14,32,60]. The size of the archive is therefore restricted to a pre-specified value. When the external archive has reached its maximum capacity *M*, the crowding distances of the solutions are calculated to determine the most crowded archive members which need to be discarded.

5. Performance evaluation

In this section, we first introduce the test problems and performance metrics used for evaluating the proposed MODEwAwL. We then study the effectiveness of the two components extended for MODEwAwL, i.e. the external archive and the learning-guided solution generation scheme, respectively. Finally, we compare the proposed MODEwAwL with four state-of-the-art multi-objective evolutionary algorithms in terms of the performance metrics.

5.1. Dataset

Seven test problems based on well-known major market indices for the portfolio optimization problems from the publicly available OR-library [4] is used to evaluate the performance of the algorithms. Table 1 shows the details of these benchmark indices and their sizes. The first five datasets (D1–D5) built from weekly price data from March 1992 to September 1997 are available at: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. They were first introduced by Chang et al. [11].

The remaining two datasets were built based on the index tracking problem and they were first introduced by Canakgoz and Beasley [9]. These two datasets (D6 and D7) are available at: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/indtrackinfo.html.

These problem instances have been used for portfolio optimization with cardinality, quantity, pre-assignment and round lot constraints in order to study the performances. All algorithms have been implemented in C# and run on a personal computer Intel(R) Core(TM)2 Duo CPU E8400 3.16 GHz. The experimental results obtained for each algorithm are the average of 20 runs.

5.2. Quality indicators

To evaluate the performance of the multi-objective evolutionary algorithms from various aspects, several performance metrics have been proposed in the literature which mainly consider proximity, diversity and distribution. In this study we use four widely adopted performance evaluation metrics namely generational distance, inverted generational distance, diversity and hypervolume.

5.2.1. Inverted generational distance (IGD)

The inverted generational distance [48] uses the true Pareto front as a reference and measures the distance of each of its elements from the true Pareto front to the non-dominated front obtained by an algorithm. It is mathematically defined as:

$$\mathrm{IGD} = \frac{\sqrt{\sum_{i=1}^{Q} d_i^2}}{Q}$$

Table 1
The Benchmark instances from OR-library.

Instance	Origin	Name	Number of assets
D1	Hong Kong	Hang Seng	31
D2	Germany	DAX100	85
D3	UK	FTSE 100	89
D4	US	S&P 100	98
D5	Japan	Nikkei	225
D6	US	S&P 500	457
D7	US	Russell 2000	1318

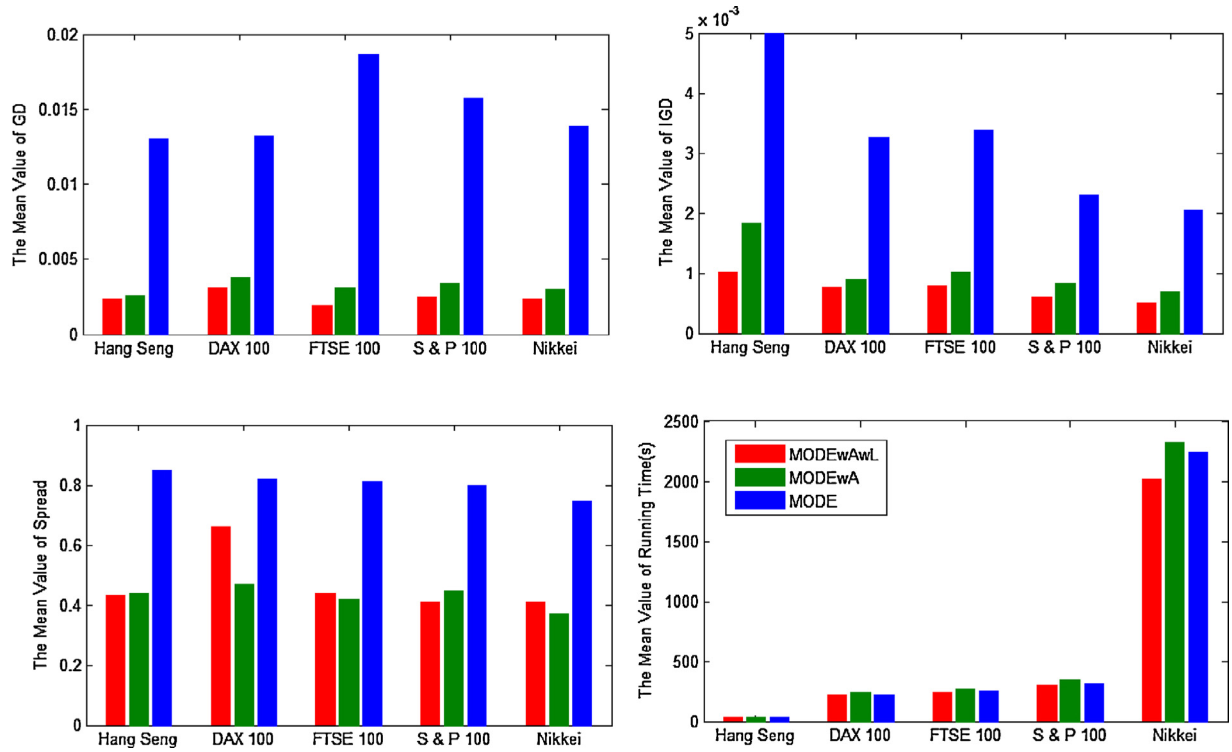

Fig. 4. Effectiveness of the learning-guided solution generation scheme and archive. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

where Q is the number of solutions in the true Pareto front and d_i is the Euclidean distance between each of the solution and the nearest member from the set of non-dominated solutions found by the algorithm. This metric measures both the diversity and the convergence of an obtained non-dominated solution set. The smaller the value of this metric, the closer the obtained front is to the true Pareto front.

The true Pareto front for highly constrained multi-objective portfolio optimization problem considered in this work is unknown. We use the best known unconstrained efficient

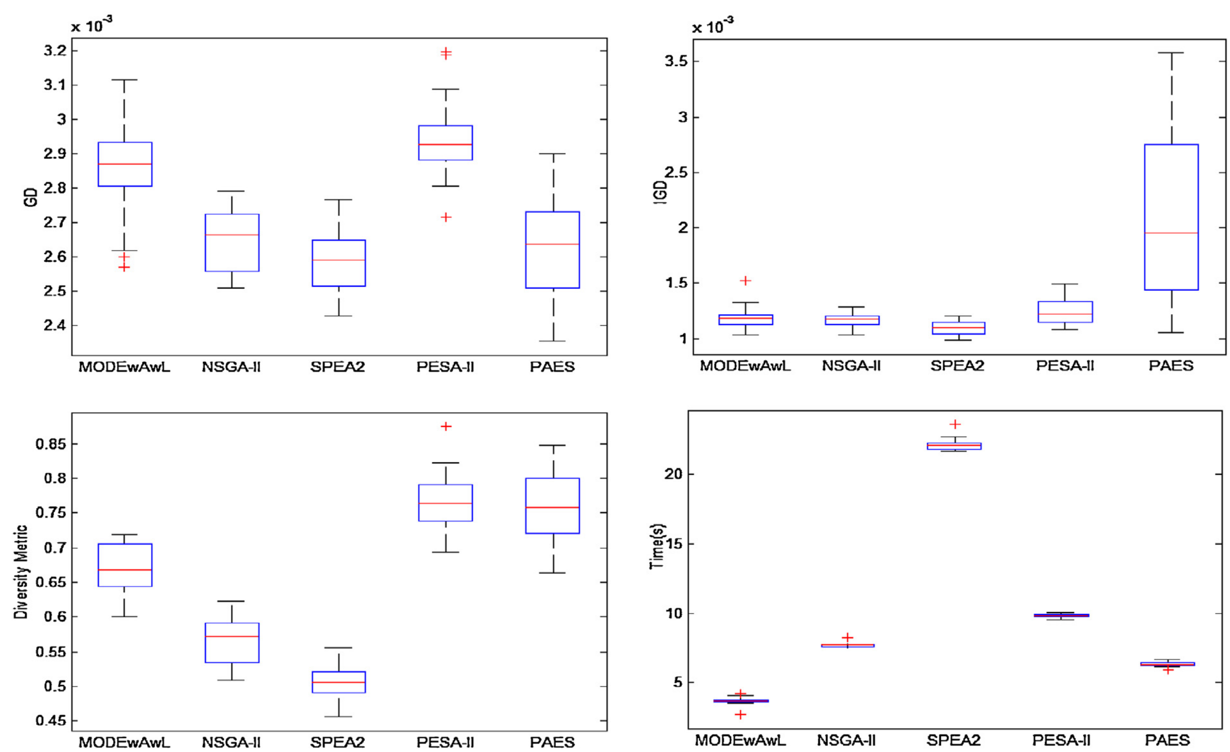

Fig. 5. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for Hang Seng dataset.

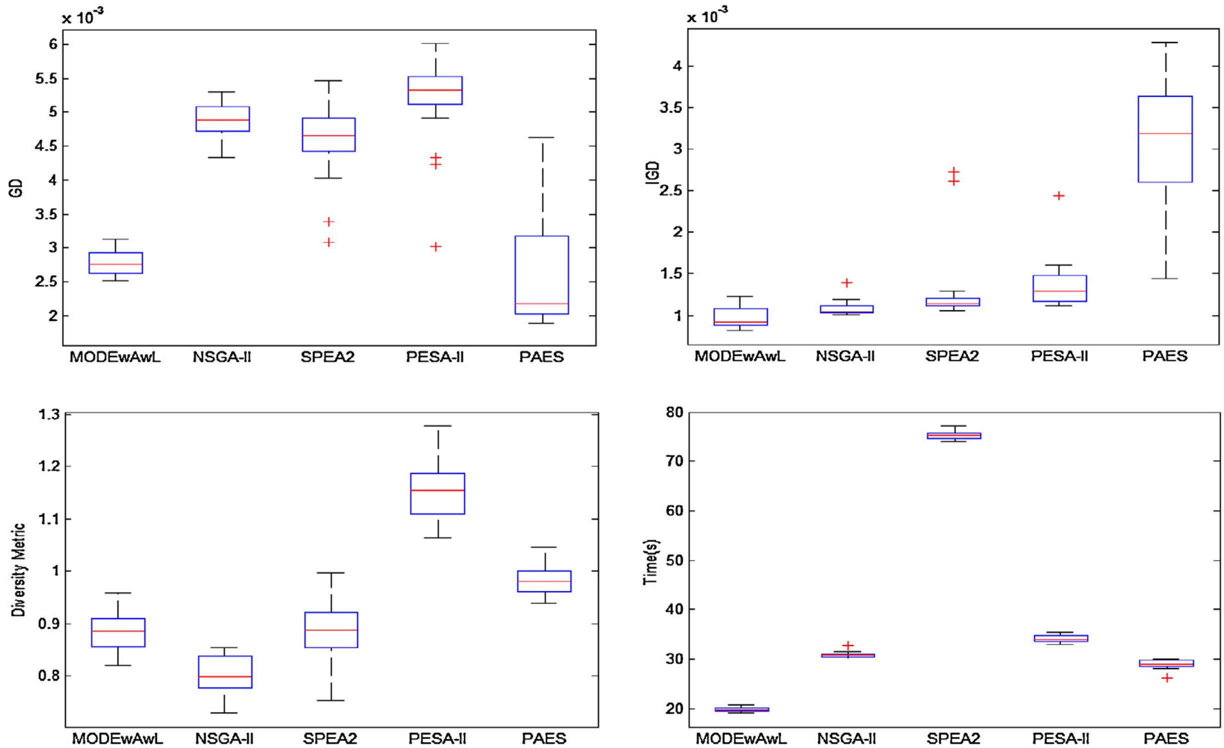

Fig. 6. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for DAX 100 dataset.

frontier (UCEF) provided by the OR-library [4] as the true Pareto front reference set. This has been widely adopted in the literature.

5.2.2. Generational distance (GD)

The generational distance [57] is a variant of IGD. It measures how far the solutions of the computed Pareto front obtained by an algorithm are from those at the true Pareto front. The smaller value indicates that all the generated solutions are on the true Pareto front.

5.2.3. Diversity metric (Δ)

The diversity metric (Δ) [19] measures the performance indices of distribution and spread simultaneously for two-objective optimization problems. The diversity metric (Δ) is defined as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|Q|-1} \mid d_i - \bar{d} \mid}{d_f + d_l + (\mid Q \mid -1)\bar{d}}$$

where d_i is the Euclidean distance in the objective space between consecutive solutions in the obtained non-dominated front Q, and $\bar{d}$ is the average of these distances. The parameters d_f and d_l are the Euclidean distance between the extreme solutions and the boundary solutions of the obtained non-dominated front Q. The lower value of the spread (Δ) indicates a better diversity.

5.2.4. Hypervolume (HV)

Hypervolume metric [61], also known as S-metric or Lebesgue measure, is widely recognized as an unary value which is able to measure both convergence and diversity. This metric calculates the normalized volume of the objective space covered by the obtained Pareto set Q bounded by a reference point r. Therefore, higher values are preferable. For each solution $i \in Q$, a hypercube c_i from solution i and the reference point r is measured. The hypervolume HV is calculated as:

$$\mathrm{HV} = volume(\cup_{i=1}^{|Q|} c_i)$$

An accurate calculation of HV requires a normalized objective space and we used the linear normalization technique proposed by Knowles et al. [31] as follows:

$$f_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}}$$

where $f_i^{\min}$ and $f_i^{\max}$ are the minimum and maximum value of the ith objective. The value of $f_i^{\min}$ and $f_i^{\max}$ are set as the minimum and maximum value obtained from running all algorithms. The reference point was chosen as r={1,0}.

5.3. Effectiveness of the learning-guided solution generation and archive

In this section, our experiments focus on the impact of the learning-guided solution generation mechanism. In order to evaluate the performance of the MODEwAwL, we compare it with two variants of the algorithm: the multi-objective differential evolution (MODE) and the multi-objective differential evolution with archive (MODEwA). Fig. 4 shows the comparisons of the three algorithms in terms of IGD, GD and Δ. The experimental results distinctly show that the proposed algorithm with the learning-guided solution generation mechanism outperforms MODE and MODEwA in most instances.

5.4. The overall performance evaluation

In order to evaluate the overall performance of the proposed MODEwAwL, we compare it with four state-of-the-art multi-objective evolutionary algorithms in the literature.

- NSGA-II: the Non-dominated Sorting Genetic Algorithm II was proposed by Deb et al. [19]. The algorithm uses binary tournament selection based on the crowding distance. It performs

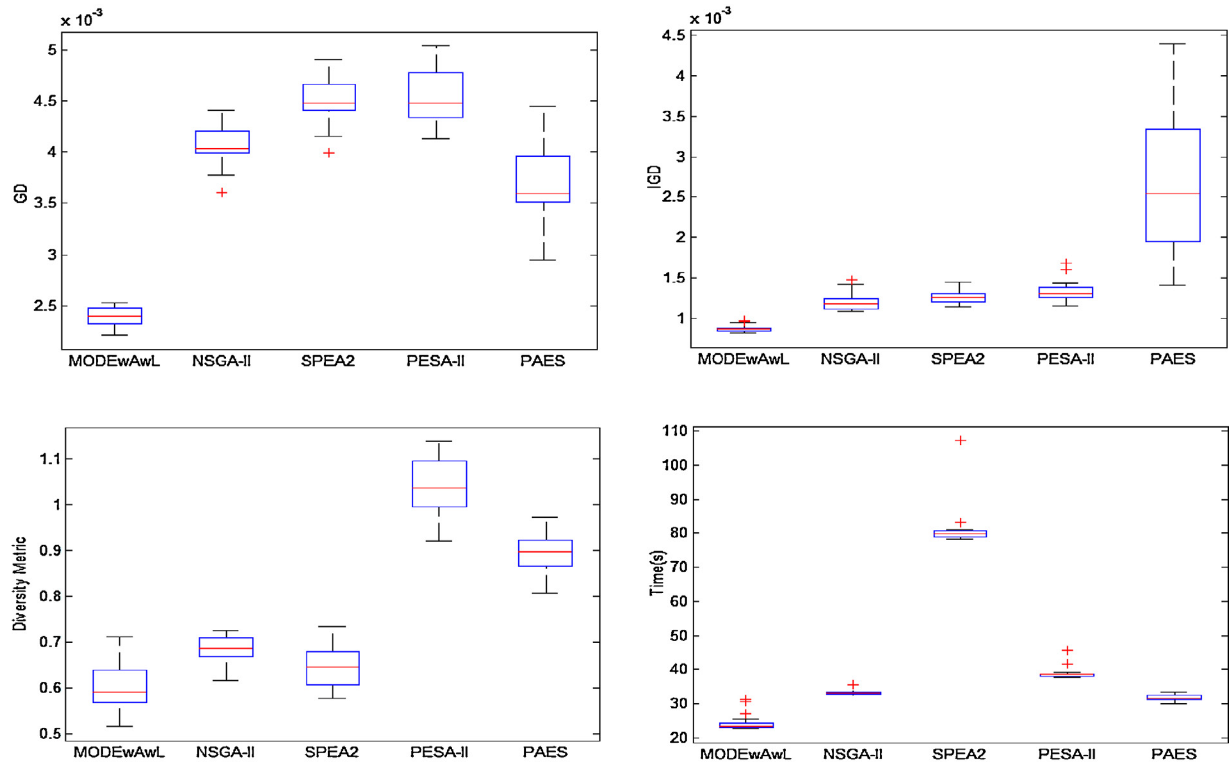

Fig. 7. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for FTSE 100 dataset.

crossover and mutation by simulated binary crossover and polynomial mutation operators.

- SPEA2: the Strength Pareto Evolutionary Algorithm was proposed by Zitzler et al. [60]. The algorithm employs fine-grained fitness assignment, density estimation techniques and archive truncation methods. Like NSGA-II, it uses binary tournament selection, simulated binary crossover and polynomial mutation evolutionary operators.

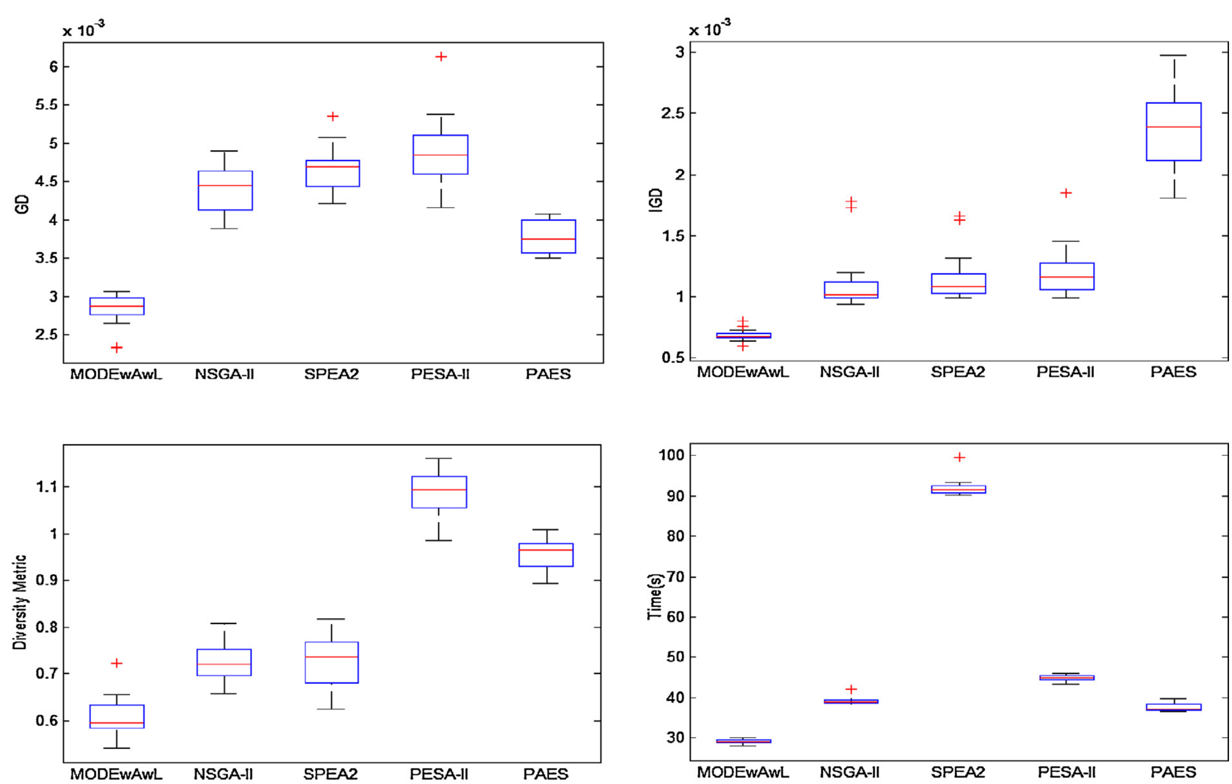

Fig. 8. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for S&P 100 dataset.

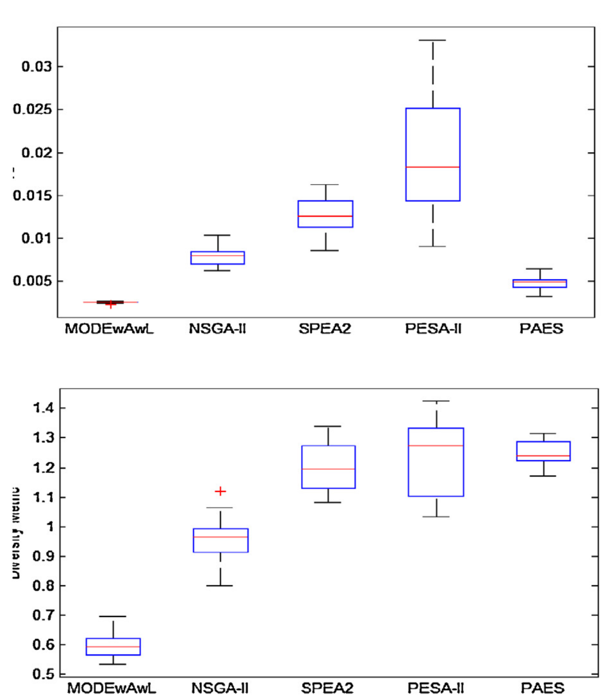

Fig. 9. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for Nikkei dataset.

- PESA2: the Pareto Envelope-based Evolutionary Algorithm was proposed by Corne et al. [16]. The algorithm uses hyper-boxes to assign fitness and employs the simulated binary crossover and polynomial mutation operations.
- PAES: the Pareto Archived Evolution Strategy was proposed by Knowles and Corne [30]. The algorithm uses a simple (1 + 1) local search evolution strategy. It maintains an archive of non-dominated solutions and it exploits those Pareto solutions to estimate the quality of new solutions.

In order to ensure a fair comparison, we have used the same population size and archive size (if applicable) for all the algorithms tested in this work. We have chosen to run all the algorithms run for the same stopping criteria (i.e. the same number of evaluations) to

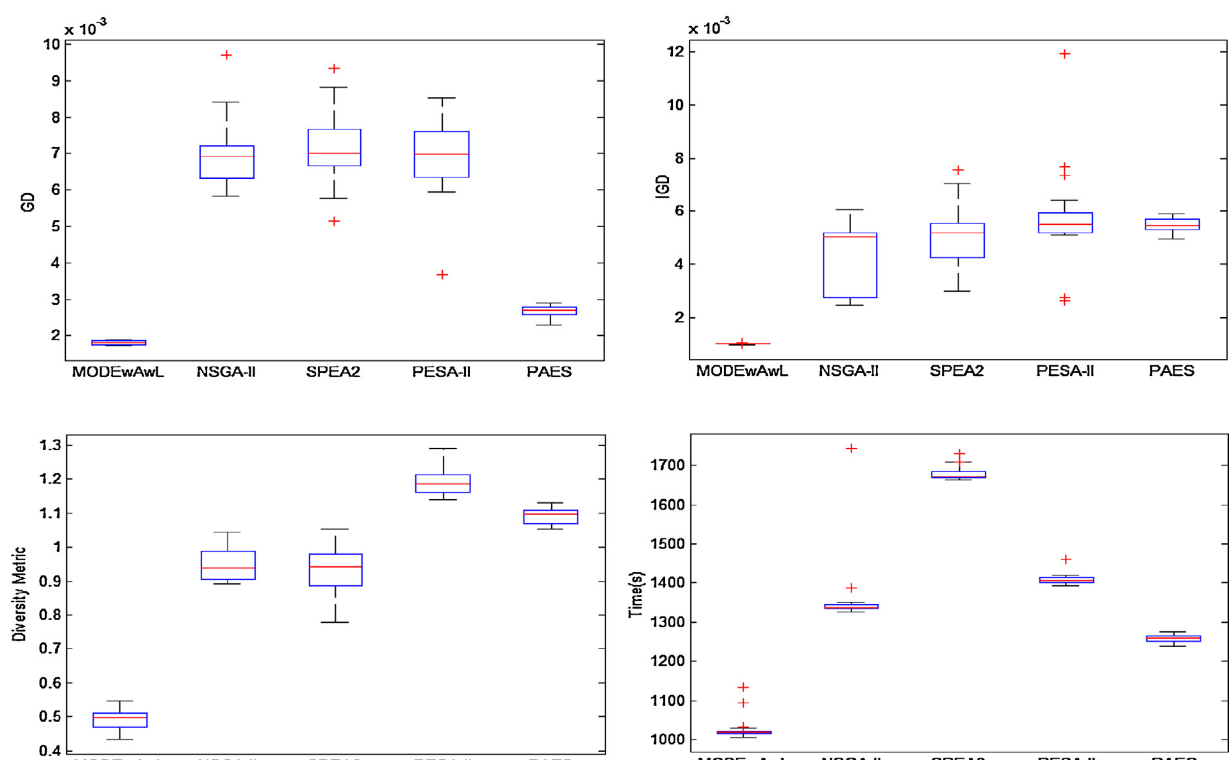

Fig. 10. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for S&P 500 dataset.

Table 2
Parameter setting of five algorithms.

Parameters	MODEwAwL	NSGA-II	SPEA2	PESA-II	PAES
Number of population (*NP*)	100	100	100	100	100
Number of generation	1000 N	1000 N	1000 N	1000 N	1000 N
Scaling factor (*F*)	0.3	–	–	–	–
Crossover probability (*CR*)	0.9	0.9	0.9	0.9	–
Crossover distribution index	–	20	20	20	–
Mutation probability	–	1/N	1/N	1/N	1/N
Mutation distribution index	–	20	20	20	20
Tournament round	–	–	1	–	–
Number of bisection	–	–	–	5	5
Archive size (*M*)	100	–	100	100	100

generate the Pareto front. Each algorithm also uses the same encodings (see Section 4.2) and repair mechanism (see Section 4.6) when a newly constructed portfolio violates the considered constraints. Before the experiments were performed, parameters are tuned for all algorithms using the smallest problem instance, i.e. Hang Seng. Table 2 shows the best parameter values of the algorithms.

5.5. Comparisons of the algorithms

In this section, we have performed a number of experiments. The results of GD, IGD and Δ and running time of the five algorithms performed on seven datasets from OR-library are shown in Figs. 5–11. For example in Fig. 11, top left boxplot represents the performance of each algorithm considered in terms of GD metric, top right boxplot represents the performance of each algorithm considered in terms of IGD metric, bottom left shows the performance of each algorithm in terms of Diversity metric and bottom right boxplot displays the computational time for each algorithm considered. These results are obtained for the constrained portfolio optimization problem with cardinality $K = 10$, floor $\epsilon_i = 0.01$, ceiling $\delta_i = 1.0$, pre-assignment $z_{30} = 1$ and round lot $\upsilon_i = 0.008$.

The results show that for most of the problem instances, the MODEwAwL obtains the smallest mean values for GD, IGD and Δ, compared with the other four algorithms, demonstrating the best performance among the five algorithms. NSGA-II comes at the second and SPEA2 comes at the third places. NSGA-II and SPEA2 seem to have almost comparable results for most problem instances. However, SPEA2 is the most computationally expensive algorithm in terms of CPU time. The results also confirm that PAES is the worst algorithm for the portfolio optimization with considered constraints. However, PAES is the second fastest algorithm after MODEwAwL. For most of the problem instances, the proposed algorithm MODEwAwL is also computationally efficient compared to the others. Fig. 12 shows the hypervolume (HV) calculation performed on seven datasets and for each problem instance, the results reconfirm the superiority of MODEwAwL since it outperforms in six out of seven datasets.

For illustrative purpose, the obtained efficient frontiers of the algorithms for seven instances along with the true unconstrained efficient frontier (UCEF) are provided in Fig. 13. When the problem sizes are small, the Pareto set obtained by the considered algorithms is very competitive to each other such that it would be hard to differentiate visually. As the problem sizes increase, the proposed algorithm obtained significantly better efficient frontier than those obtained by other MOEAs considered in this work. Based on the analysis, we conclude that the proposed MODEwAwL is able to solve large-scale real-world portfolio optimization efficiently. The results also demonstrate that NSGAII

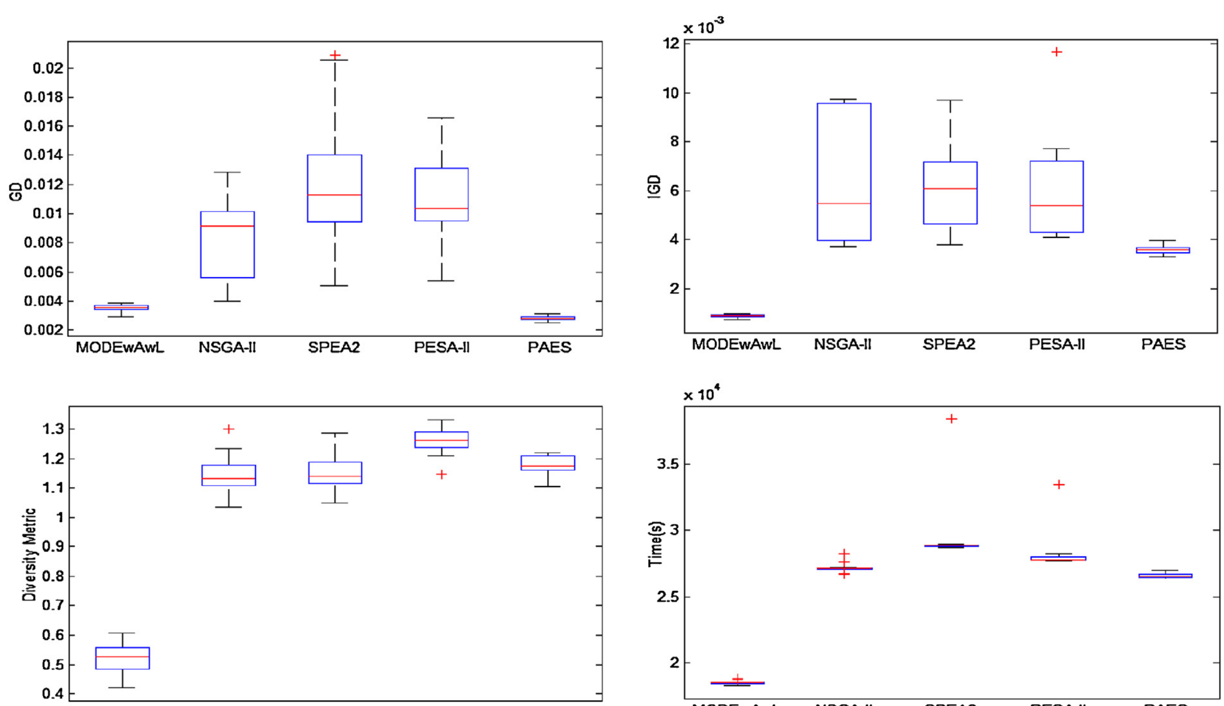

Fig. 11. Performance comparisons of five algorithms in term of GD, IGD and Δ metrics for Russell 2000 dataset.

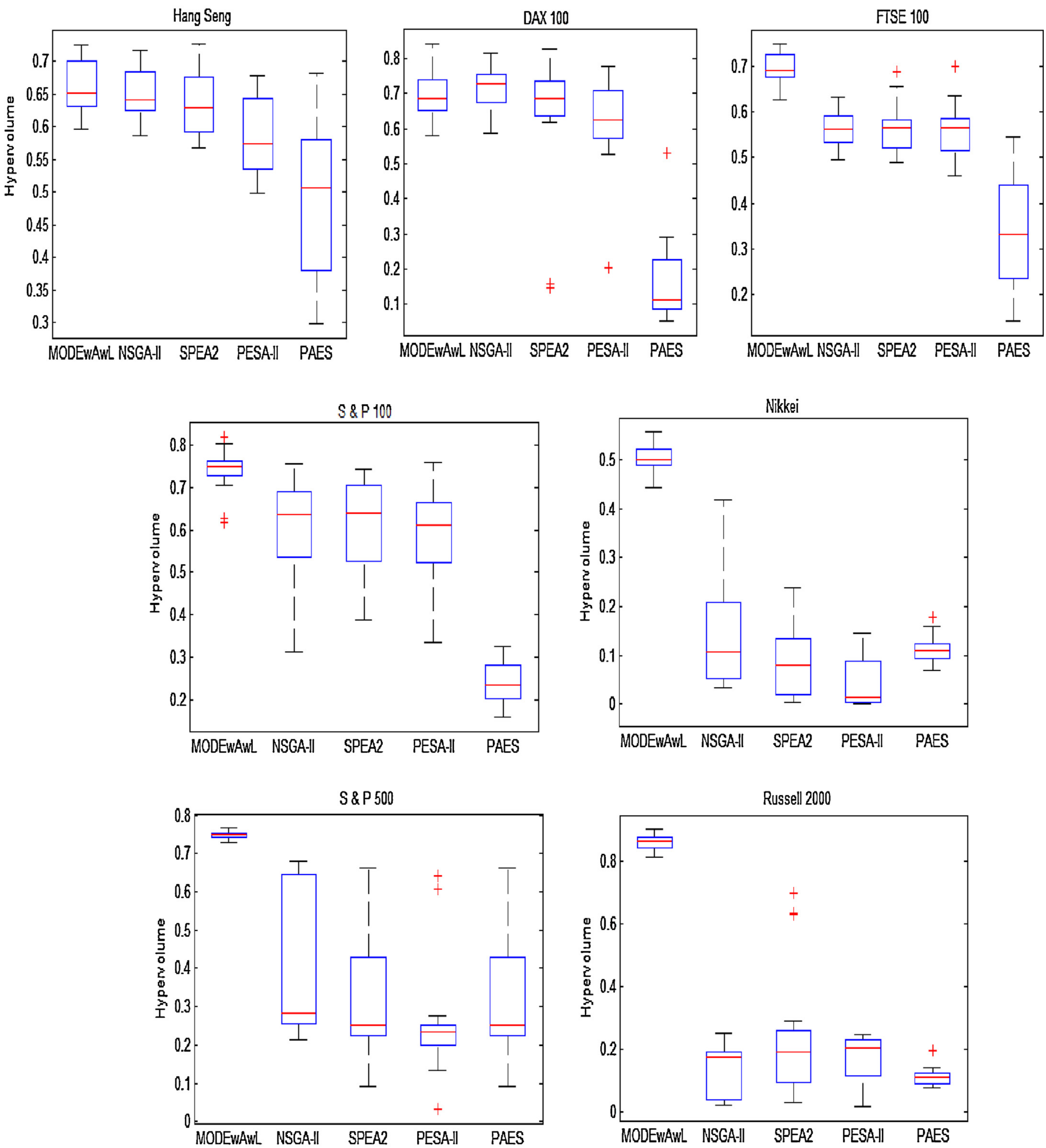

Fig. 12. Performance comparisons of five algorithms in term of HV metric.

and SPEA2 loose their effectiveness when the problem dimension increases.

To gain an intuitive view of the five algorithms over generations, we plot the GD, IGD and Δ metrics over generations on five selected instances in Fig. 14 where the results are averaged over 20 runs. The results confirm that all algorithms considered are able to converge and MODEwAwL is able to converge the fastest in most problem instances.

Experiments are also performed for different cardinality values with $K = 15$ and $K = 5$. The results are made publicly accessible at: http://cs.nott.ac.uk/~ktl/results/MODEwAwL-results.pdf. On the majority of datasets, MODEwAwL is significantly better than the other compared MOEAs. The experimental results have further demonstrated that the proposed algorithm is efficient for various search spaces with different values of K. The proposed MODEwAwL is thus more robust than the compared MOEAs.

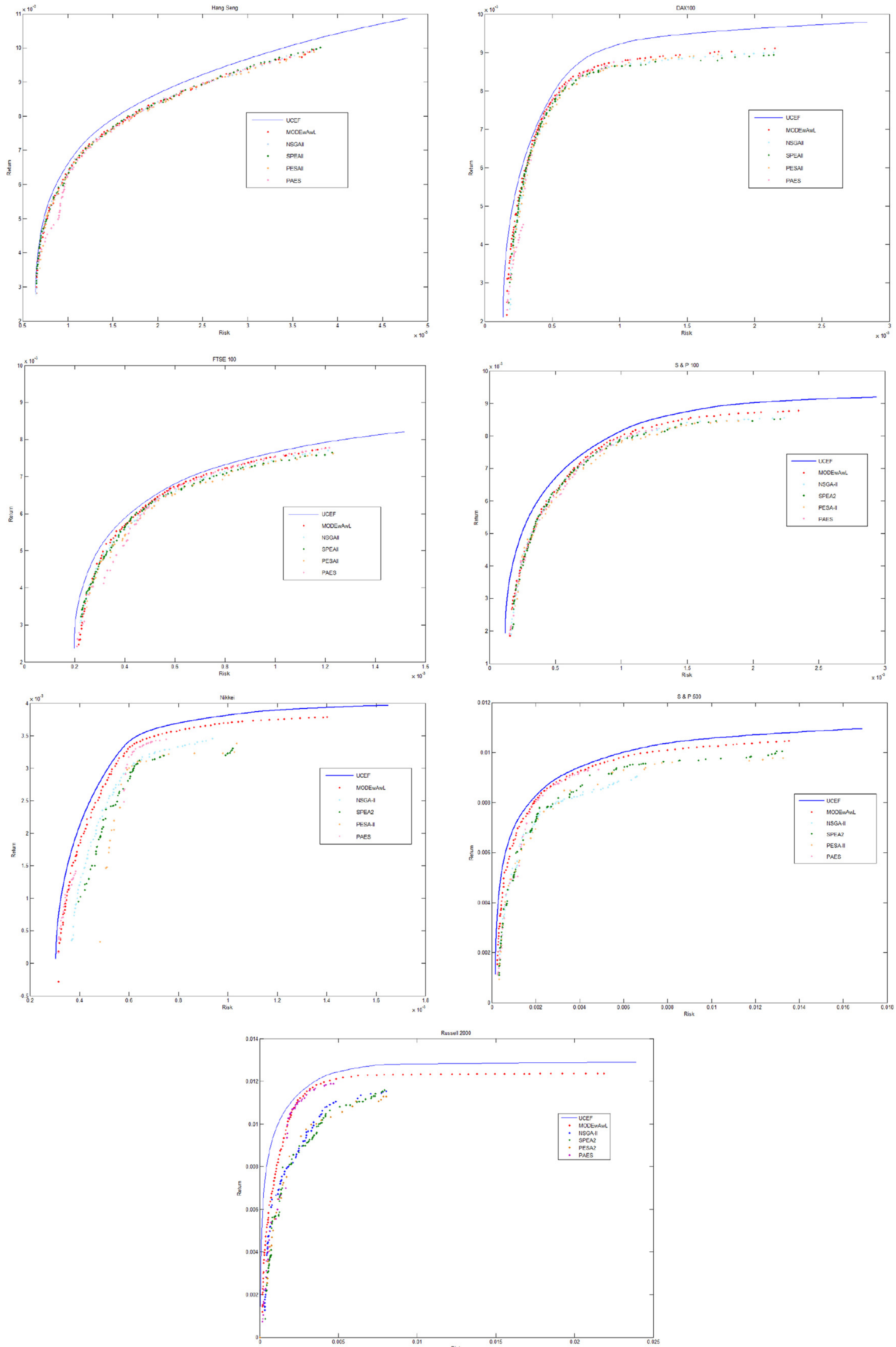

Fig. 13. Comparison of efficient frontiers for seven datasets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

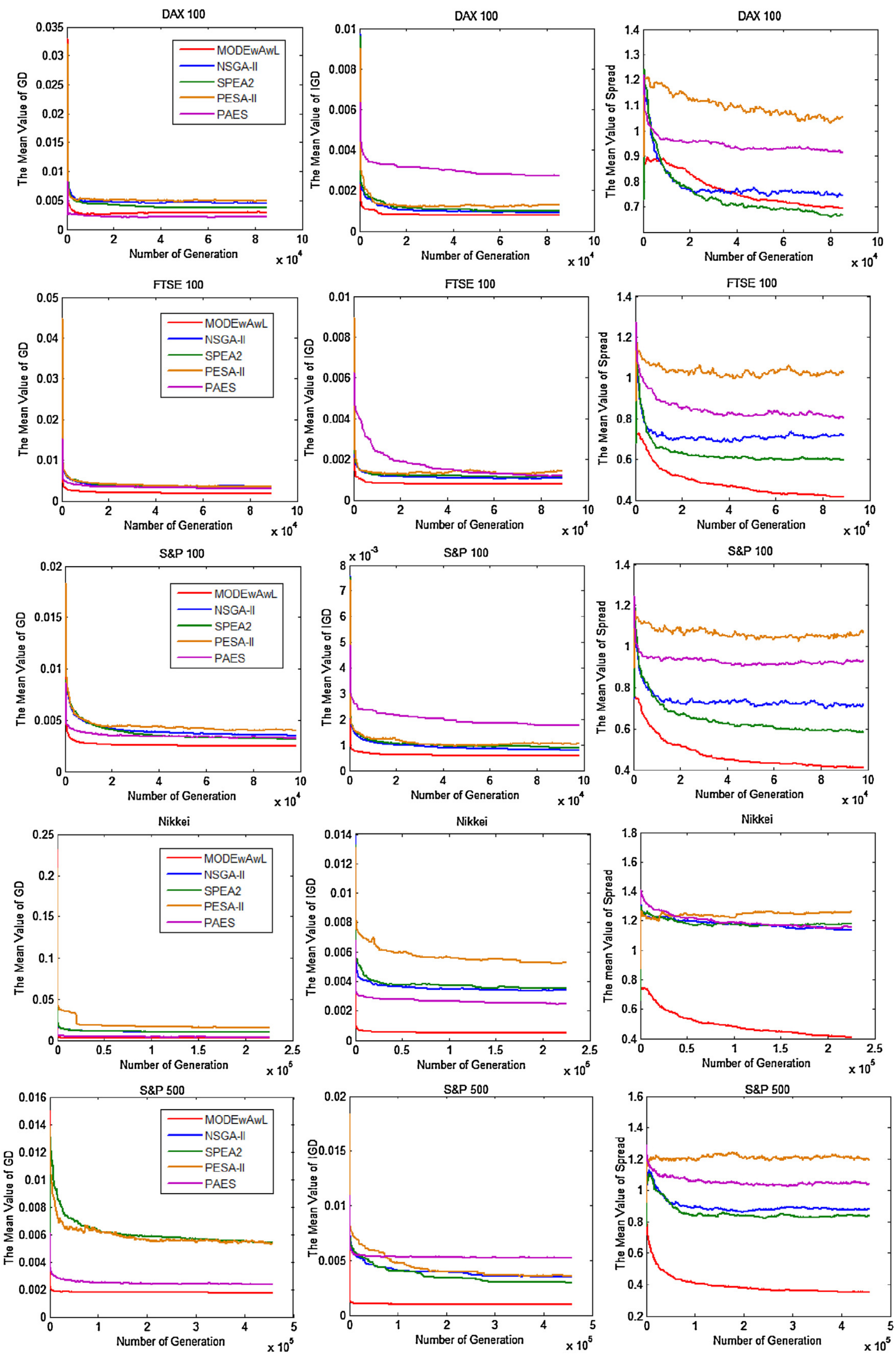

Fig. 14. Comparisons of convergence of five algorithms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Table 3
Student's t-test results of different algorithms on seven problem instances with $K = 10$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z_{30} = 1$ and $\upsilon_i = 0.008$.

Algorithm1 ↔ Algorithm2	Hang Seng	DAX 100	FTSE 100	S&P 100	Nikkei	S&P 500	Russell 2000
MODEwAwL ↔ NSGA-II	∼	+	+	+	+	+	+
MODEwAwL ↔ SPEA2	–	+	+	+	+	+	+
MODEwAwL ↔ PESA-II	∼	+	+	+	+	+	+
MODEwAwL ↔ PAES	+	+	+	+	+	+	+
NSGA-II ↔ SPEA2	–	+	∼	∼	+	+	∼
NSGA-II ↔ PESA-II	+	+	+	∼	+	+	∼
NSGA-II ↔ PAES	+	+	+	+	∼	+	–
SPEA2 ↔ PESA-II	+	∼	∼	∼	+	∼	∼
SPEA2 ↔ PAES	+	+	+	+	–	∼	–
PESA-II ↔ PAES	+	+	+	+	–	∼	–

Table 4
Student's t-test results of different algorithms on 5 problem instances with $K = 15$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z_{30} = 1$ and $\upsilon_i = 0.008$.

Algorithm1 ↔ Algorithm2	Hang Seng	DAX 100	FTSE 100	S&P 100	Nikkei	S&P 500	Russell 2000
MODEwAwL ↔ NSGA-II	∼	+	+	+	+	+	+
MODEwAwL ↔ SPEA2	∼	+	+	+	+	+	+
MODEwAwL ↔ PESA-II	+	+	+	+	+	+	+
MODEwAwL ↔ PAES	+	+	+	+	+	+	+
NSGA-II ↔ SPEA2	+	∼	+	+	+	∼	∼
NSGA-II ↔ PESA-II	+	+	+	+	+	∼	∼
NSGA-II ↔ PAES	+	+	+	+	+	+	∼
SPEA2 ↔ PESA-II	+	∼	∼	∼	+	∼	∼
SPEA2 ↔ PAES	+	+	+	+	–	∼	∼
PESA-II ↔ PAES	+	+	+	+	–	∼	∼

Table 5
Student's t-test results of different algorithms on five problem instances with $K = 5$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z_{30} = 1$ and $\upsilon_i = 0.008$.

Algorithm1 ↔ Algorithm2	Hang Seng	DAX 100	FTSE 100	S&P 100	Nikkei	S&P 500	Russell 2000
MODEwAwL ↔ NSGA-II	+	+	+	+	+	+	+
MODEwAwL ↔ SPEA2	+	+	+	+	+	+	+
MODEwAwL ↔ PESA-II	+	+	+	+	+	+	+
MODEwAwL ↔ PAES	+	+	+	+	+	+	+
NSGA-II ↔ SPEA2	–	+	∼	+	+	∼	∼
NSGA-II ↔ PESA-II	+	+	∼	+	+	∼	∼
NSGA-II ↔ PAES	+	+	+	+	–	–	–
SPEA2 ↔ PESA-II	+	∼	∼	∼	+	∼	∼
SPEA2 ↔ PAES	+	+	∼	∼	–	–	–
PESA-II ↔ PAES	∼	+	∼	+	–	–	–

As stated in Section 5.2.1, IGD can provide the overall performance of an algorithm, measuring its convergence and diversity simultaneously. We compare the IGD values of the five algorithms by using Student's t-test [58]. The statistical results obtained by a two-tailed t-test with 38 degrees of freedom at a 0.05 level of significance are given in Tables 3–5. The result of Algorithm-1 ↔ Algorithm-2 is shown as "+", "–", or "∼" when Algorithm-1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm-2, respectively. Results show that MODEwAwL outperforms other algorithms in most of the problem instances except Hang Seng dataset. For Hang Seng test problem, the performance of SPEA2 outperforms MODEwAwL when $K = 10$. We therefore can conclude that the proposed MODEwAwL has the best optimization performance for the portfolio optimization problem with considered constraints.

6. Conclusion and future work

In this work, we investigated the portfolio selection problem with four practical constraints which limit the number of assets in a portfolio, restrict the minimum and maximum proportions of assets held in the portfolio, require some specific assets to be included in the portfolio and require to invest the assets in units of a certain size respectively.

We have demonstrated that maintaining a secondary population of solution set in combination with learning-guided candidate solution generation scheme contribute to better performance over four existing well-known MOEAs, NSGA-II, SPEA2, PEAS-II and PAES. The experimental results not only show that the quality of the generated Pareto set approximations significantly improved, but also that the overall computation time can be reduced. As to the Pareto set approximation, the proposed solution generation scheme embedding learning mechanism, problem specific heuristics and direction-based search methods plays a major role, while the efficiency is mainly because the proposed algorithm is computationally cheap as it only uses a single update at each generation. Performance wise, the proposed MODEwAwL algorithm is not only capable to deliver high-quality portfolios enriched with additional constraints but also able to efficiently solve a reasonable size of asset up to 1318. The proposed algorithm could be applied to other practical applications such as knapsack problem with relevant constraints. For future work, the proposed algorithm can be extended to include constraints such as transaction cost and short selling.

Acknowledgements

This research was supported by the School of Computer Science, The University of Nottingham. The authors would also like to thank anonymous reviewers for their helpful comments.

References

[1] K. Anagnostopoulos, G. Mamanis, A portfolio optimization model with three objectives and discrete variables, Comput. Oper. Res. 37 (7) (2010) 1285–1297.
[2] K. Anagnostopoulos, G. Mamanis, The mean-variance cardinality constrained portfolio optimization problem: an experimental evaluation of five multiobjective evolutionary algorithms, Expert Syst. Appl. 38 (11) (2011) 14208–14217.
[3] R. Armananzas, J.A. Lozano, A multiobjective approach to the portfolio optimization problem, in: The 2005 IEEE Congress on Evolutionary Computation, 2005, vol. 2, IEEE, 2005, pp. 1388–1395.
[4] J. Beasley, Distributing test problems by electronic mail, J. Oper. Res. Soc. 41 (11) (1990) 1069–1072.
[5] D. Bertsimas, R. Shioda, Algorithm for cardinality-constrained quadratic optimization, Comput. Optim. Appl. 43 (1) (2009) 1–22.
[6] D. Bienstock, Computational study of a family of mixed-integer quadratic programming problems, Math. Program. 74 (2) (1996) 121–140.
[7] J. Branke, B. Scheckenbach, M. Stein, K. Deb, H. Schmeck, Portfolio optimization with an envelope-based multi-objective evolutionary algorithm, Eur. J. Oper. Res. 199 (3) (2009) 684–693.
[8] R. Brito, L. Vicente, Efficient Cardinality/Mean-Variance Portfolios, 2012.
[9] N. Canakgoz, J.E. Beasley, Mixed-integer programming approaches for index tracking and enhanced indexation, Eur. J. Oper. Res. 196 (1) (2009) 384–399.
[10] M. Castillo Tapia, C. Coello, Applications of multi-objective evolutionary algorithms in economics and finance: a survey, in: IEEE Congress on Evolutionary Computation, 2007. CEC 2007, IEEE, 2007, pp. 532–539.
[11] T. Chang, N. Meade, J. Beasley, Y. Sharaiha, Heuristics for cardinality constrained portfolio optimisation, Comput. Oper. Res. 27 (13) (2000) 1271–1302.
[12] S. Chiam, K. Tan, A. Al, Mamum, Evolutionary multi-objective portfolio optimization in practical context, Int. J. Autom. Comput. 5 (1) (2008) 67–80.
[13] C. Coello, Evolutionary multi-objective optimization and its use in finance, in: Handbook of Research on Nature Inspired Computing for Economy and Management, Idea Group Publishing, 2006.
[14] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evolut. Comput. 8 (3) (2004) 256–279.
[15] C.A. Coello, Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (11) (2002) 1245–1287.
[16] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, et al., PESA-II: region-based selection in evolutionary multiobjective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), Citeseer, 2001.
[17] T. Cura, Particle swarm optimization approach to portfolio optimization, Nonlinear Anal.: Real World Appl. 10 (4) (2009) 2396–2406.
[18] K. Deb, Multi-objective optimization, in: Multi-Objective Optimization Using Evolutionary Algorithms, 2001, pp. 3–46.
[19] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182–197.
[20] G. Deng, W. Lin, C. Lo, Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization, Expert Syst. Appl. (2011).
[21] G. Di Tollo, A. Roli, Metaheuristics for the portfolio selection problem, Int. J. Oper. Res. 5 (1) (2008) 13–35.
[22] L. Diosan, A multi-objective evolutionary approach to the portfolio optimization problem, in: International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 2, IEEE, 2005, pp. 183–187.
[23] J.E. Fieldsend, J. Matatko, M. Peng, Cardinality constrained portfolio optimisation, in: Intelligent Data Engineering and Automated Learning—IDEAL 2004, Springer, 2004, pp. 788–793.
[24] C. Fonseca, P. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput. 3 (1) (1995) 1–16.
[25] L.D. Gaspero, G.D. Tollo, A. Roli, A. Schaerf, Hybrid metaheuristics for constrained portfolio selection problems, Quant. Finance 11 (10) (2011) 1473–1487.
[26] N. Gulpinar, L.T.H. An, M. Moeini, Robust investment strategies with discrete asset choice constraints using dc programming, Optimization 59 (1) (2010) 45–62.
[27] M. Hirschberger, Y. Qi, R.E. Steuer, Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming, Eur. J. Oper. Res. 204 (3) (2010) 581–588.
[28] C. Israelsen, The benefits of low correlation, J. Index. 10 (6) (2007) 18–26.
[29] M.V. Jahan, Extremal optimization vs. learning automata: strategies for spin selection in portfolio selection problems, Appl. Soft Comput. 12 (10) (2012) 3276–3284.
[30] J. Knowles, D. Corne, The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC99, vol. 1, IEEE, 1999.
[31] J. Knowles, L. Thiele, E. Zitzler, A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, No. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006, February (revised version).
[32] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evol. Comput. 8 (2) (2000) 149–172.
[33] G.A. Kochenberger, et al., Handbook of Metaheuristics, Springer, 2003.
[34] T. Krink, S. Paterlini, Multiobjective optimization using differential evolution for real-world portfolio optimization, Comput. Manage. Sci. 8 (1-2) (2011) 157–179.
[35] D. Li, X. Sun, J. Wang, Optimal lot solution to cardinality constrained mean-variance formulation for portfolio selection, Math. Finance 16 (1) (2006) 83–101.
[36] C.-C. Lin, Y.-T. Liu, Genetic algorithms for portfolio selection problems with minimum transaction lots, Eur. J. Oper. Res. 185 (1) (2008) 393–404.
[37] K. Lwin, R. Qu, A hybrid algorithm for constrained portfolio selection problems, Appl. Intell. (2013), http://dx.doi.org/10.1007/s10489-012-0411-7.
[38] R. Mansini, M.G. Speranza, Heuristic algorithms for the portfolio selection problem with minimum transaction lots, Eur. J. Oper. Res. 114 (2) (1999) 219–233.
[39] D. Maringer, Portfolio Management with Heuristic Optimization, vol. 8, Springer Verlag, 2005.
[40] H. Markowitz, Portfolio selection, J. Finance 7 (1) (1952) 77–91.
[41] H. Markowitz, Portfolio Selection: Efficient Diversification of Investments, John Wiley and Sons, New York, 1959.
[42] S. Mishra, G. Panda, S. Meher, R. Majhi, M. Singh, Portfolio management assessment by four multiobjective optimization algorithm, in: Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE, IEEE, 2011, pp. 326–331.
[43] J.-S. Pang, A new and efficient algorithm for a class of portfolio selection problems, Oper. Res. 28 (3-Part-II) (1980) 754–767.
[44] A. Ponsich, A. Jaimes, C. Coello, A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications, IEEE Trans. Evolut. Comput. 17 (3) (2013) 321–344.
[45] G.R. Raidl, A unified view on hybrid metaheuristics, in: Hybrid Metaheuristics, Springer, 2006, pp. 1–12.
[46] T. Robič, Demo: differential evolution for multiobjective optimization, in: Evolutionary Multi-Criterion Optimization, Springer, 2005, pp. 520–533.
[47] D.X. Shaw, S. Liu, L. Kopman, Lagrangian relaxation procedure for cardinality-constrained portfolio optimization, Optim. Methods Softw. 23 (3) (2008) 411–420.
[48] M.R. Sierra, C.A.C. Coello, Improving PSO-based multi-objective optimization using crowding, mutation and epsilon-dominance, in: EMO'05, 2005, pp. 505–519.
[49] P. Skolpadungket, K. Dahal, A survey on portfolio optimisation with metaheuristics, in: SKIMA 2006, 2006, p. 103.
[50] P. Skolpadungket, K. Dahal, N. Harnpornchai, Portfolio optimization using multi-objective genetic algorithms, in: IEEE Congress on Evolutionary Computation, 2007. CEC 2007, IEEE, 2007, pp. 516–523.
[51] E.D. Smith, Y.J. Son, M. Piattelli-Palmarini, A.Terry Bahill, Ameliorating mental mistakes in tradeoff studies, Syst. Eng. 10 (3) (2007) 222–240.
[52] M. Stein, J. Branke, H. Schmeck, Efficient implementation of an active set algorithm for large-scale portfolio selection, Comput. Oper. Res. 35 (12) (2008) 3945–3961.
[53] R. Storn, K. Price, Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012. Berkeley, CA, 1995.
[54] F. Streichert, H. Ulmer, A. Zell, Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem, in: Congress on Evolutionary Computation, 2004. CEC2004, vol. 1, IEEE, 2004, pp. 932–939.
[55] F. Streichert, H. Ulmer, A. Zell, Evolutionary algorithms and the cardinality constrained portfolio optimization problem, in: Operations Research Proceedings 2003, Springer, 2004, pp. 253–260.
[56] E.-G. Talbi, A taxonomy of hybrid metaheuristics, J. Heuristics 8 (5) (2002) 541–564.
[57] D.A. Van Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithm research: a history and analysis. Technical Report. Citeseer, 1998.
[58] R.E. Walpole, R.H. Myers, S.L. Myers, K. Ye, Probability and Statistics for Engineers and Scientists, vol. 8, Prentice Hall, Upper Saddle River, NJ, 1998.
[59] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[60] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, L. Thiele, Spea2: Improving the Strength Pareto Evolutionary Algorithm, 2001.
[61] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evolut. Comput. 3 (4) (1999) 257–271.