Expert Systems with Applications 38 (2011) 14208–14217

Contents lists available at ScienceDirect

Expert Systems with Applications

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

The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms

K.P. Anagnostopoulos [1], G. Mamanis *

Department of Production and Management Engineering, Democritus University of Thrace, Kimmeria Campus, 67100 Xanthi, Greece

ARTICLE INFO

Keywords:
Cardinality constraints
Efficient frontier
Evolutionary algorithms
Mean–variance optimization
Multiobjective optimization
Portfolio selection

ABSTRACT

This paper compares the effectiveness of five state-of-the-art multiobjective evolutionary algorithms (MOEAs) together with a steady state evolutionary algorithm on the mean–variance cardinality constrained portfolio optimization problem (MVCCPO). The main computational challenges of the model are due to the presence of a nonlinear objective function and the discrete constraints. The MOEAs considered are the Niched Pareto genetic algorithm 2 (NPGA2), non-dominated sorting genetic algorithm II (NSGA-II), Pareto envelope-based selection algorithm (PESA), strength Pareto evolutionary algorithm 2 (SPEA2), and e-multiobjective evolutionary algorithm (e-MOEA). The computational comparison was performed using formal metrics proposed by the evolutionary multiobjective optimization community on publicly available data sets which contain up to 2196 assets.

1. Introduction

A fundamental answer to the problem of portfolio selection was given by Markowitz who introduced the mean–variance model (Markowitz, 1952, 1990). The Markowitz model is a problem of multiobjective nonlinear optimization where the best trade-offs between maximizing mean and minimizing variance is sought. This trade-offs form a special set of particular interest in portfolio theory, which is known as the set of efficient portfolios and its image in mean–variance space as the efficient frontier (Elton, Gruber, Brown, & Goetzmann, 2007). Calculating the efficient frontier for the Markowitz model several computationally efficient algorithms exist, which are able to solve even large instances of the problem with more than 2000 assets (Hirschberger, Qi, & Steuer, 2010; Markowitz, 1987; Stein, Branke, & Schmeck, 2007).

However, through the years the Markowitz model has been expanded by introducing additional real-world constraints, such as the cardinality constraint which imposes a limit on the number of assets in the portfolio and the quantity constraints which restrict the proportion of each asset in the portfolio to lie between lower and upper bounds (Chang, Meade, & Beasley, 2000; Mitra, Kyriakis, Lucas, & Pirbhai, 2003). This model, which we call *mean–variance cardinality constrained portfolio optimization model* (MVCCPO), presents many optimization and financial challenges and concerns.

Financially the constraints reflect the real portfolio decision making process as managers desire to hold a portfolio with a rather small number of assets compared to those available in the financial markets, in order to facilitate each management and to control transaction costs. Quantity constraints are generally used in order to avoid very small, unrealistic holdings and excessively investing in a single asset's risk. From the computational point of view the constraints lead to the introduction of integer variables and transform the feasible space into a non-convex region (Crama & Schyns, 2003; Jobst, Horniman, Lucas, & Mitra, 2001; Mitra et al., 2003). The problem is transformed into a discrete multiobjective nonlinear optimization problem which is computationally very difficult to be solved; and it has been shown that belongs to the class of NP-hard problems (Bienstock, 1995;Shaw, Liu, & Kopman, 2008).

The significance of the MVCCPO model has motivated many researchers to investigate various algorithms from exact methods (Bertsimas & Shioda, 2009; Bienstock, 1995; Li, Sun, & Wang, 2006; Shaw et al., 2008) to metaheuristics (Chang et al., 2000; Crama & Schyns, 2003; Cura, 2009; Fernández & Gómez, 2007; Maringer & Kellerer, 2003; Schaerf, 2002; Soleimani, Golmakani, & Salimi, 2009). The two approaches have their own merits and demerits. Depending on the input data and the constraint parameters, exact methods may fail to provide an optimal solution in reasonable time; and the computation time grows rapidly with the problem size. On the other hand, metaheuristics are not guaranteed to find the optimal solutions; however, they are able to find a good solution (sometimes even the real optimum) within reasonable computation time.

Recently, researchers have tried to tackle the MVCCPO problem in its multiobjective form trying to optimize both objectives

* Corresponding author. Tel.: +30 2321065328.
E-mail addresses: kanagn@civil.duth.gr (K.P. Anagnostopoulos), gmamanis@pme.duth.gr (G. Mamanis).
[1] Tel.: +30 2541079676.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.233

simultaneously using so-called multiobjective metaheuristics. The main advantage of these methods, comparing them to single objective metaheuristics, is that they find the efficient frontier in a single run of the algorithm. Streichert, Ulmer, and Zell (2003, 2004a, 2004b) experimented with various data structures for chromosome representation and crossover operators within the context of a MOEA and show that a hybrid encoding (containing both binary and real values) is the best suited for the MVCCPO model. Armananzas & Lozano, 2005 have compared and adapted three optimization heuristics namely greedy local search, simulated annealing (SA) and ant colony optimization (ACO). The results indicated that ACO and SA were the best optimizers. Chiam, Tan, and Mamum (2008) proposed an ordered based solution representation to handle quantity and cardinality constraints. Branke, Scheckenbach, Stein, Deb, and Schmeck (2009) proposed a hybrid algorithm combining NSGA-II with the critical line algorithm. Skolpadungket, Dahal, and Harnpornchai (2007) have performed an experimental study using well known MOEAs on the MVCCPO problem and they found that SPEA2 is the best algorithm for the metrics considered. However, they perform experiments with only one data set containing 31 assets – an instance clearly too small both to take into account the size of real world portfolios and the computational behavior of algorithms as the problem size grows. Anagnostopoulos and Mamanis (2010) and Fieldsend, Matatko, and Peng (2004) consider another form of the MVCCPO model where the cardinality constraint becomes an objective function to be minimized. For the interested readers a survey of applications of MOEAs in finance and economics and especially in portfolio selection can be found in Schlottmann and Seese (2004), Tapia and Coello (2007).

There are various additional studies applying metaheuristics (either single or multiobjective) to portfolio optimization but rely on different portfolio models usually by considering more complex risk measures and/or additional constraints. For the sake of completeness we refer the studies of Dueck and Winker (1992), Gilli, Këllezi, and Hysi (2006), Hochreiter (2007), Anagnostopoulos and Mamanis (2009) and Chang, Yang, and Chang (2009). A comprehensive survey of metaheuristics in portfolio selection can be found in Di Tollo and Rolli (2008).

In this study, we compare the effectiveness of the state-of-art multiobjective evolutionary algorithms (MOEAs) on the MVCCPO problem. On the one hand, we consider the MVCCPO as the generic model of the discrete mean–variance portfolio selection problems. On the other hand, the algorithms are chosen in such a way as to capture the main MOEAs algorithmic design components as are identified in a recent book (Coello, Lamont, & Van Veldhuizen, 2007). For comparison purposes, we also include in the study a variant of the single objective evolutionary algorithm (SOEA) proposed in Chang et al. (2000).

Our experimental investigation aims: (a) to answer the question if solving the bi-objective cardinality constrained portfolio optimization problem with MOEAs provides any advantage over its single objective counterpart which is the most often applied, (b) to compare the effectiveness of the current state-of-the-art of MOEAs on the MVCCPO model, and (c) to investigate the effectiveness of MOEAs to solve large scale instances of the MVCCPO problem in reasonable time. According to our knowledge, there is no other study in literature that performs such an exhaustive computational analysis with the state-of-art MOEAs in MVCCPO model with real-world data from major stock markets. Furthermore, this is the first study that gives evidence that MOEAs can actually solve large scale instances of the MVCCPO model with more than 2000 assets in reasonable time and with very good approximation.

The rest of the paper has as follows. In Section 2, the mean–variance cardinality constrained portfolio optimization model is described. A description and an implementation of the state of the art MOEAs for solving the problem is presented in Section 3. Section 4 is devoted to numerical results, and Section 5 concludes.

2. The mean–variance cardinality constrained portfolio optimization model

In its multiobjective form, the mean–variance cardinality constrained portfolio optimization model (MVCCPO) is formulated as follows (Model 1):

$$\begin{aligned}
\min \quad & \rho(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij}, \\
\max \quad & \mu(x) = \sum_{i=1}^{n} x_i \mu_i, \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_i = 1, && (1)\\
& \sum_{i=1}^{n} \delta_i \le K, && (2)\\
& l_i \delta_i \le x_i \le u_i \delta_i, \quad i = 1,\dots,n, && (3)\\
& \delta_i \in \{0,1\}, \quad i = 1,\dots,n. && (4)
\end{aligned}$$

where n is the number of assets considered for inclusion in the portfolio; x_i is the decision variable which denotes the proportion held of asset i; μ_i is the expected return of asset i; σ_{ij} is the covariance between the returns of assets i and j; $\rho(x)$ is the variance of portfolio x; $\mu(x)$ is the expected return of portfolio x; δ_i is a binary variable which is 1 if an asset $i = 1,\dots,n$ is held and 0 otherwise; K is the maximum number of assets allowed in a portfolio; l_i, u_i are the lower and upper proportion of capital invested in holding securities.

The feasible space X is defined by Eqs. (1), (2), (3), (4). Eq. (1) describes the standard budget constraint which requires that the portfolio weights, being proportions, must sum to one. Eq. (2) describes the cardinality constraint which limits the number of securities held in the portfolio. The cardinality constraint is introduced into the model using a binary variable δ which is 1 if a security is held and zero otherwise. The next Eq. (3) (quantity constraints) limits the amount of capital invested in holding securities to lie between lower and upper bounds.

As the above model is a multiobjective optimization problem, the aim is to find all non-dominated or efficient portfolios, i.e., every portfolio which we cannot improve upon an objective function without deteriorating another. In the particular problem at hand we say that a portfolio $x' \in X$ dominates another portfolio $x \in X (x' \succ x)$ if $\mu(x') \geqslant \mu(x), \rho(x') \leqslant \rho(x)$ with at least one strict inequality. A portfolio which is not dominated by any other portfolio is an efficient portfolio.

The bi-objective optimization problem (Model 1) is usually solved with single objective solution techniques. The most popular approach uses a trade-off coefficient λ to combine the two objectives into a scalar to be minimized (Model 2):

$$\begin{aligned}
\min \quad & \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij} - (1-\lambda)\sum_{i=1}^{n} x_i \mu_i \\
\text{s.t.} \quad & x \in X
\end{aligned}$$

By repeatedly varying the parameter value λ and solving a sequence of optimization problems (for each λ) the efficient portfolios from the minimum variance portfolio ($\lambda = 1$) to the maximum return portfolio ($\lambda = 0$) can be found. This approach, however, has a shortfall since it cannot find all efficient points as shown in Chang et al. (2000).

An alternative approach is the one that tries to minimize variance while return is constrained to have a lower bound (Model 3):

$$\begin{aligned} \min \quad & \rho(x) \\ \text{s.t.} \quad & \mu(x) \geq d \\ & x \in X \end{aligned}$$

with the above formulation, solving for different return levels of $d \in [d_{\min}, d_{\max}]$ the set of the efficient portfolios can be found.

Models 2 and 3 are used by researchers who apply single objective metaheuristics. Their main advantage is that they are reduced to a scalar optimization problem. However, solving the MVCCPO problem with these methods requires the repeated use of an optimization technique to find one point on the efficient frontier per run and this may be time consuming. Furthermore, a uniform set of λ parameters or return levels d does not guaranty a uniformly distributed set of efficient points (Mukerjee, Biswas, Deb, & Mathur, 2002). The diversity of points along the efficient frontier is crucial for the portfolio manager since certain trade-off portfolios of interest may be missed if they are concentrated in a small area of the underlying efficient frontier.

3. The tested algorithms

In this study we compare the effectiveness of the current state of MOEAs together with a simple steady state evolutionary algorithm on the MVCCPO problem formulated in Section 2. The following five MOEAs are tested: Strength Pareto evolutionary algorithm 2 (SPEA2) (Zitzler, Laumanns, & Thiele, 2001), non-dominated sorting genetic algorithm II (NSGA-II) (Deb, Ptratap, Agarwal, & Meyarivan, 2002), Pareto envelope-based evolutionary algorithm (PESA) (Corne, Knowles, & Oates, 2000), Niched Pareto genetic algorithm 2 (NPGA2) (Erikson, Mayer, & Horn, 2001), e-multiobjective evolutionary algorithm (e-MOEA) (Hanne, 2007). Except their popularity, these algorithms were chosen in such a way as to represent the main algorithmic aspects proposed up to know in the MOEA literature and are these aspects that are actually compared.

3.1. Description of algorithms

All algorithms implemented in this study fit in some way the general framework represented in Table 1, and apply in a different manner the operators. One difference is that PESA and e-MOEA apply the evaluate operator after the update operator in the archive of the next generation A^{t+1}.

All evolutionary algorithms can be seen as that they involve two populations of individuals. The first population, usually called archive or external population in MOEAs literature, is used to retain the "best" solutions found during the search; the second is the usual population of individuals, sometimes used to simply store the offspring population and some other times it takes part on the reproduction process as well.

At first, the archive A^0, which has a user-specified maximum size N^{arc}, is set to the empty set and the population B^0 (of size N^{pop}) to a random sample of the solution space through the *initialize* operator. Note that in single objective evolutionary algorithm (SOEA) the archive is the population that is randomly initialized using this framework. At each generation the *evaluate* operator assigns fitness to individuals from both the archive and the normal population. Here is the main difference between the SOEA and MOEAs. The SOEA assigns fitness to individuals based on a scalar fitness function; in our case, like the paper of Chang et al. (2000), the fitness function is formed by the weighted sum method described in Section 2 (Model 2). Solving for the efficient frontier using this model requires the application of the evolutionary algorithm several times in order to obtain one efficient point (portfolio) per run.

Table 1
Structure of evolutionary algorithms.

```
t = 0
(A^0, B^0) = initialize()
while (termination = false) do
  evaluate(A^t, B^t)
  A^{t+1} = update(A^t, B^t)
  B^{t+1} = vary(sample(A^{t+1}))
  t = t + 1
end while
return best(A^t, B^t)
```

On the other hand, MOEAs do not combine the two objectives, rather they use the *evaluate* operator in such a way as to guide the population towards non-dominated regions as well as to preserve the diversity. In the following step, the archive is updated by the "best" individuals based on information from both the archive A and the normal population B. In this way elitism is ensured. The *update* operator may be deterministic as well as stochastic. The *sample* and *vary* operators specify the particular selection and reproduction scheme and are the same as in traditional evolutionary algorithms. At the last stage the "best" solutions from the archive and the final offspring population is returned by the algorithm.

As discussed above, the main difference among the algorithms lies in their fitness assignment technique (*evaluate* operator). In MOEAs, the *evaluate* operator should serve two goals: population convergence towards the efficient frontier and preservation of diversity of solutions along the efficient frontier. Both goals are achieved by assigning a rank and a density value to every solution. All MOEAs give first priority to non-dominance and second priority to diversity. However, the particular method each MOEA employs for achieving the two fundamental goals differs. Methods for convergence towards the efficient frontier and diversity preservation have been summarized in the recent book of Coello et al. (2007, Chapter 2, pp. 79–85).

3.1.1. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

To guide the individuals towards the efficient frontier, NSGA-II adopts the dominance depth method which classifies the solutions in several layers based on which front an individual is located. To preserve the diversity of solutions a crowding mechanism is employed which calculates the volume of the hyper-rectangle defined by the two nearest neighbors. Based on these values *update* operator returns the best N^{arc} individuals from the union of the archive and the population. Individuals with the lower rank have the priority to survive. If a number of solutions that have the same rank does not all fit the archive, then the less crowded individuals from the particular rank are selected to enter the archive.

3.1.2. Strength Pareto evolutionary algorithm 2 (SPEA2)

SPEA2 emphasizes non-dominated individuals by using a fine-grained technique, which combines the dominance count and dominance rank method. Each individual is given a raw fitness value based on the number of individuals it dominates and is dominated by. The density information is incorporated by adding to the raw fitness a value that is equal to the inverse of the k-th smallest Euclidean distance (measured in objective space) to the k-th nearest neighbor plus two. Next, the update operator returns all non-dominated individuals from the combined set $A \cup B$. However, there are two possibilities with this process. Either the individuals returned by the update operator exceeds the maximum archive size N^{arc} or they are not enough. In the later case the best dominated solutions according to their fitness values are selected. In the former case, a truncate operator is applied that recursively deletes surplus solutions based on the nearest neighbor Euclidean

distance. At each stage, if there is more than one solution with the same minimum distance, the decision is done considering the second nearest neighbor and so forth.

3.1.3. Niched Pareto genetic algorithm 2 (NPGA2)

The third tested algorithm is the Niched Pareto genetic algorithm 2 proposed by Erikson et al. (2001). Its initial version has been proposed in Horn, Nafpliotis, and Goldberg (1994). The additional component we have added in NPGA2 is the elitist update operator in order to ensure a fair comparison with the other strategies. Note that elitism plays an important role in MOEAs and its inclusion is crucial to achieve good convergence and diversity characteristics towards the exact efficient frontier (Laumanns, Zitzler, & Thiele, 2000). Furthermore, in order not to destroy NPGA's characteristics the update operator is a stochastic strategy based on tournament selections. First the algorithm assigns rank on every individual in the combined population $A \cup B$ based on the number of individuals a solution is dominated by (dominance rank method). For preserving diversity a niche count is calculated for each competitor i using the following equation:

$$m_i = \begin{cases} \sum_{j \in A^{t+1}} \left(1 - \frac{d_{ij}}{\sigma_{sh}}\right) & \text{if } d_{ij} < \sigma_{sh}, \\ 0 & \text{if } d_{ij} \geq \sigma_{sh}, \end{cases}$$

where d_{ij} is the distance between competitor i and individual j and σ_{sh} the user-specified niche radius. For measuring the distance following the authors suggestions we use the metropolitan metric. The distance is calculated using normalized objective values (see Section 4.1). Elitism is introduced in NPGA2 using the update operator as follows. A small number of individuals (t_{dom}) are randomly selected from the composite population $A \cup B$. Based on the rank values, the individual with the smallest rank is added (if not already inside) to the archive of the next generation. If there are more individuals with identical ranks then the niche count is calculated for those solutions and the one with the smallest value is added on the archive. Note that the niche count is calculated using only solutions from the archive of the next generation A^{t+1}. This strategy ensures some elitism, and furthermore we think that it does not change the initial character of the algorithm. Although we do not claim that it is a new algorithm, we abbreviate the name E-NPGA2 for this technique in order to highlight that some form of elitism has been introduced.

3.1.4. Pareto envelope-based selection algorithm (PESA)

PESA, first, applies the update operator which is performed iteratively. The newly generated solutions B^t are incorporated into the archive one by one. A candidate child enters the archive when it is non-dominated within B^t, or it is not dominated by any current member of the archive. If the addition of a solution renders the archive over-full, then a current solution with the maximum fitness value is deleted. The fitness of an individual is based on a recursively division of the objective space into hyperboxes. The number of solutions that reside on the same hyperbox is the fitness of an individual.

3.1.5. e-MOEA

The last tested algorithm e-MOEA is based on the notion of e-dominance for preserving diversity (Hanne, 2007). Like PESA, e-MOEA applies the update operator before the evaluation process and is used in such a way as to emphasize non-dominated solutions, as well as, to preserve diversity. Like PESA, every individual in B^t is checked for an inclusion in the archive A^{t+1} of the next generation iteratively. An efficient offspring is inserted into the archive if at least one of the following expressions is true:

(1) Dominates any current archive member.
(2) Is non-dominated within the archive and its smallest distance to every individual in archive of the next generation is bigger than e.
(3) Improves the optimum of a single objective function.

After the inclusion of a solution, any dominated solutions are deleted from the archive. If expression three is true, solutions with distance to the newly added solution smaller than the user-specified e-value are deleted from the archive. The distance between individuals is measured by the Euclidian distance applied in the normalized objective vectors (see Section 4.1). After the *update* operator, the algorithm assigns fitness to every individual based on the k-th smallest Euclidean distance like SPEA2. This was introduced in order to increase the probability of distant solutions to be selected.

3.1.6. Single objective evolutionary algorithm (SOEA)

The SOEA used is a simple steady state algorithm similar to the one used in Chang et al. (2000). In each iteration, a single child is generated using two parents from the archive. The *update* operator then replaces the worst archive member with this newly created solution. The fitness of an individual is measured by the weighted function identified in Section 2 (Model 2). Thus, smaller values are preferable.

3.2. Implementation of algorithms

3.2.1. Solution representation and encoding

In order to allow for a fair comparison, we have chosen all algorithms to have the same solution representation. We have implemented the hybrid representation proposed by Streichert et al. (2003), Streichert et al. (2004a, Streichert et al. (2004b) which seems to be more appropriate for portfolio optimization.

In hybrid representation two vectors are used for defining a portfolio: a binary vector that specifies whether a particular asset participates in the portfolio, and a real-valued vector used to compute the proportions of the budget invested in the assets:

$$\Delta = \{\delta_1, \ldots, \delta_n\}, \quad \delta_i = 0 - 1, \ i = 1, \ldots, n,$$
$$W = \{w_1, \ldots, w_n\}, \quad 0 \leq w_i \leq 1, \ i = 1, \ldots, n.$$

3.2.2. Constraints satisfaction

Before the objective values were computed the following repair algorithm was performed in order to find the portfolio x associated with the above encoding. First, if the number of assets in the portfolio (i.e., the number of 1's in Δ), overcomes the maximum allowed, we delete (by changing its value from 1 to 0 in Δ) those assets that have the minimum weight in W. In this way the portfolio satisfies cardinality constraint.

The next step in the repair process is used to satisfy the remaining constraints (budget, lower proportion). The real portfolio weights are computed using the following equation:

$$x_i = l_i \cdot \delta_i + \frac{w_i \cdot \delta_i}{\sum_{i=1}^{n} w_i \cdot \delta_i} \left(1 - \sum_{i=1}^{n} l_i \cdot \delta_i\right), \quad i = 1, \ldots, n.$$

To account with upper quantity bounds, we propose the repair algorithm outlined in Chang et al. (2000). However, we do not describe it here because no such bounds are used in this study.

3.2.3. Variation and sample procedure

Sample and *vary* operators are the same as in usual evolutionary algorithms. We use identical schemes for all tested algorithms in order to ensure a fair comparison. For selecting the parents we

use binary tournament selection for all MOEAs, except E-NPGA2 which uses a different tournament size (t_{dom}).

For reproducing the offspring population, we have used the uniform crossover operator in each string of the chromosome. In uniform crossover two selected individuals generates a single child and its value for each array is selected with equal probability from one or another parent.

The children were considered also for mutation. We have used different mutation probabilities for each string (p_W, p_Δ). In real-valued arrays the Gaussian random mutation was used with standard deviation 0.15, while in the binary string bit flip mutation in a randomly defined position was applied.

4. Computational analysis

In this section, we present the computational results obtained by performing experiments on a public available data set, which contains five test problem instances and can be downloaded from: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. The data set provides the necessary input data for groups of assets in different stock market indices: the Hong Kong Hang Seng with 31 assets (P1), the German Dax 100 with 85 assets (P2), the British FTSE 100 with 89 assets (P3), the US S&P 100 with 98 assets (P4), and the Japanese Nikkei 225 with 225 assets (P5).

We have also performed a number of experiments for testing the effectiveness of MOEAs to solve large scale instances of the MVCCPO problem using a public benchmark data set available in the web page: http://w.3.uniroma1.it/Tardella/homepage.html. This data set contains weekly stock prices and the required expected returns and covariance matrices for five major capital market indices. We have chosen the two largest data sets, namely S&P 500 which contains 476 assets and NASDAQ which includes 2196 assets, to perform experiments with the two best algorithms observed from the above experimental comparison.

All algorithms have been implemented in Visual C++ and run on a personal computer Core 2 Duo at 2.1 GHz. The maximum number of assets allowed in the portfolio was fixed to be ten ($K = 10$) for all test problems. For the stock weights x_i lower and upper bounds were 1% and 100% respectively, i.e., $l_i = 0.01$, $u_i = 1$, $i = 1, \ldots, n$. These are the usual constraint parameters utilized by most studies.

4.1. Performance indicators

Up to now there are two approaches for comparing the quality of different MOEAs: (a) by measuring each qualitative characteristic of an approximation set separately, (b) by identifying whether one approximation set is better than another based on certain set preference relations (Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fonseca, 2003). In this study we follow the second approach and we use the ε-indicator and the hypervolume metric proposed by Zitzler and Thiele (1999), Zitzler et al. (2003). When two algorithms generate conflicting preferences between the two metrics they are incomparable (Knowles, Thiele, and Zitzler, 2006).

In order to allow the objectives to contribute approximately equally to each metric we have used the following linear normalization technique (Knowles et al., 2006). Each objective function value i was transformed according to the following equation:

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

where $f_i^{\min}$ and $f_i^{\max}$ are the minimum and maximum values, that the ith objective can take. In order to obtain $f_i^{\min}, f_i^{\max}$, we have run the single objective evolutionary algorithm a number of times for each instance to get the single objective approximate optimum value. The value of $f_i^{\min}, f_i^{\max}$ was chosen as the 10% difference of the optimum values obtained. Table 2 shows these values for each problem instance.

Table 2
Values of objective functions for each problem.

Problem	$f_1^{\min}$	$f_1^{\max}$	$f_2^{\min}$	$f_2^{\max}$
P1	0.000578	0.005253	0.00234	0.011950
P2	0.000130	0.003120	0.00140	0.010800
P3	0.000185	0.001668	0.00211	0.009030
P4	0.000120	0.003233	0.00156	0.010000
P5	0.000270	0.001800	−0.00034	0.004370

The reference point required to compute the hypervolume metric was $z^{\text{ref}} = \{1, 0\}$ normalized for all problem instances.

4.2. Parameter settings

Before the experiments were performed, some tuning for all algorithms was done using a moderate size problem instance, namely P2, which involves 85 assets. The best parameter values obtained for all algorithms are shown in Table 3 and are kept constant for all instances in an effort to test the developed techniques robustness and scaling.

In order to ensure a fair comparison we have used identical population and archive sizes for all MOEAs which were allotted to run for 400 generations (i.e., 100,000 generated solutions). For SOEA we believe that it is not trivial to run the algorithm for 100,000 generated solutions because this would mean that either we would obtain only five efficient portfolios (given that it requires, for each λ, 20,000 generated solutions in order to converge near the optimum) or (in order to get roughly 250 efficient points) we would run each of the 250 optimization problems only for 400 generations (i.e., 400 function evaluations) leading us to a local rather a global optimum. For this reason, we have chosen to run the algorithm to its own stopping criteria even though this requires a considerable higher number of solutions to be generated.

Apart from those values, PESA, E-NPGA2, and e-MOEA require some additional parameters to be set. After trial and error analysis the division of the search space for PESA was set equal to 100, the niche radius σ_{sh} for E-NPGA2 to 0.07 and the size of the comparison set t_{dom} to 8 for all the experiments. For e-MOEA we have performed a number of experiments for each instance to choose the appropriate e value in order to obtain roughly 250 efficient solutions on average at the end of all runs of the algorithm. These values were 0.00432, 0.0046, 0.00442, 0.004675, and 0.00458 for each problem instance respectively. In order to cope with the stochastic nature of the tested algorithms, 30 different optimization runs have been carried out in all test problems considered.

4.3. Experimental results

4.3.1. Comparison of the algorithms

The results, as can been seen in Figs. 1–5 that shows the median and the standard deviation for each problem instance, demonstrate a superiority of SPEA2, since it wins in four out of the five test

Table 3
Best parameter values for the algorithms.

Parameters	NSGA-II	SPEA2	E-NPGA2	PESA	e-MOEA	SOEA
N^{pop}	250	250	250	250	250	1
N^{arc}	250	250	250	250	–	100
$p_{\text{crossover}}$	0.9	0.9	0.9	0.8	0.8	0.9
p_W	1.0	1.0	1.0	1.0	1.0	1.0
p_Δ	1/n	1/n	1/n	1/n	1/n	1/n

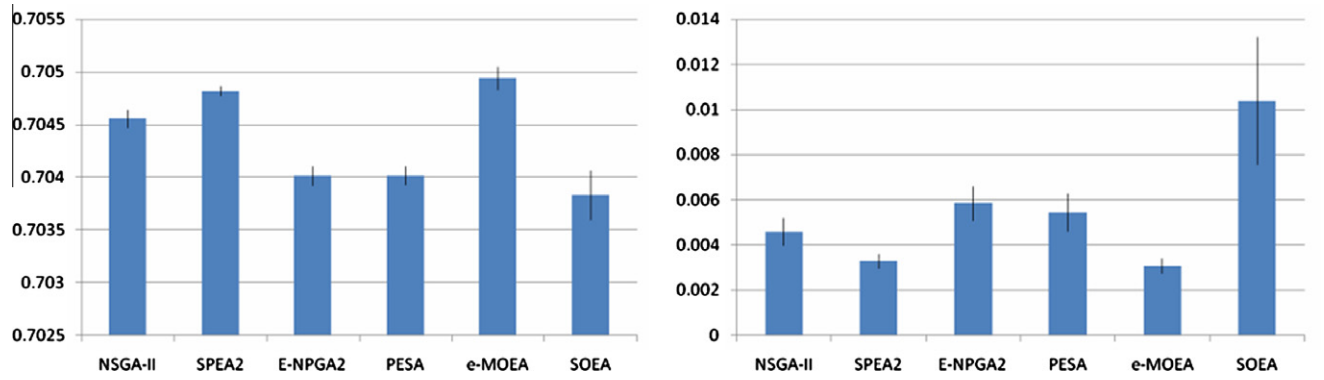

Fig. 1. Hypervolume indicator (left) and e-indicator (right) for Hong Kong Hang Seng data set.

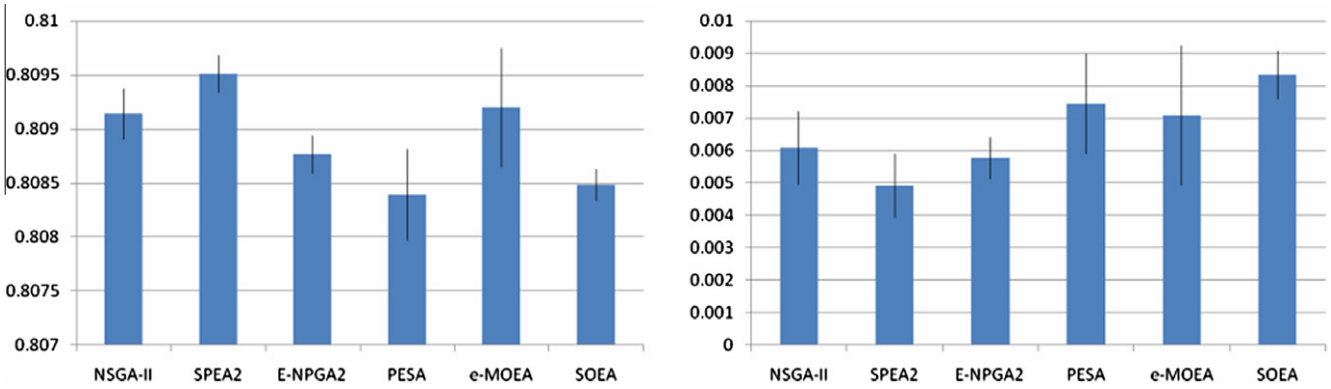

Fig. 2. Hypervolume indicator (left) and e-indicator (right) for German Dax 100 data set.

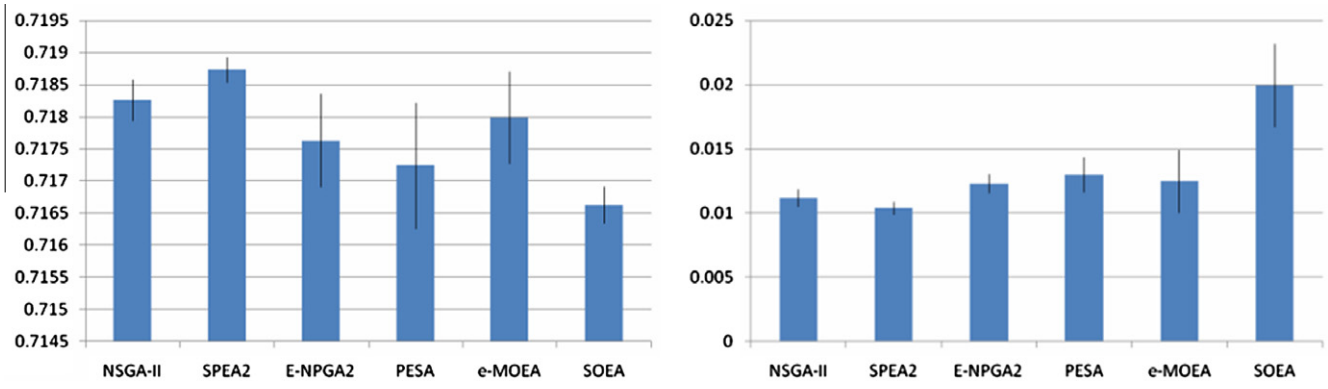

Fig. 3. Hypervolume indicator (left) and e-indicator (right) for British FTSE 100 data set.

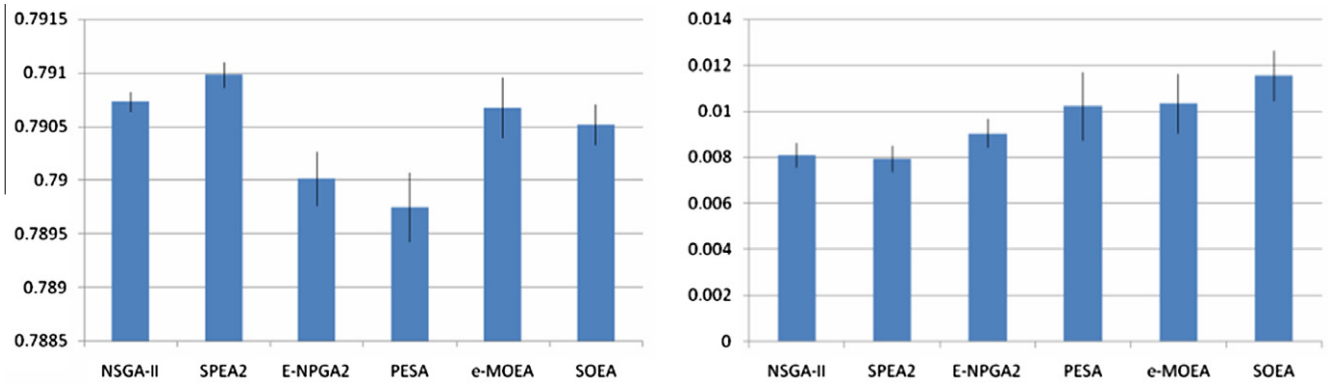

Fig. 4. Hypervolume indicator (left) and e-indicator (right) for US S&P data set.

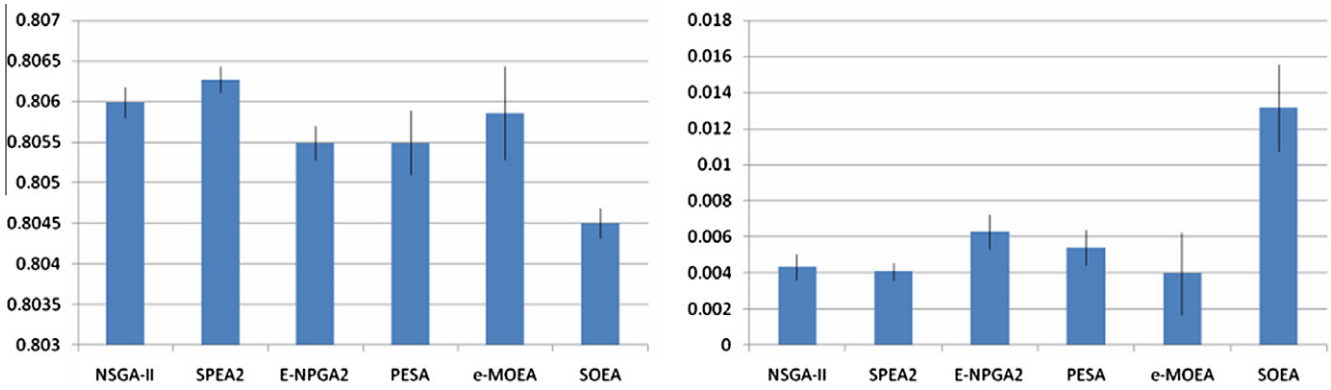

Fig. 5. Hypervolume indicator (left) and e-indicator (right) for Japanese Nikkei 225 data set.

problems. However, we should note that this superiority comes at the cost of greatest computational effort since SPEA2 is the worst algorithm in terms of the CPU time (except SOEA). NSGA-II and e-MOEA comes at the second place and seems to have almost comparable performance. NSGA-II wins two instances, e-MOEA one, and for two instances no one dominates the other since they generate conflicting results in terms of the hypervolume and ε-indicator. However, e-MOEA is faster than NSGA-II requiring, on average, only half of its run-time.

We have also performed a Kruskal–Wallis test to judge the statistical significance of the results and it was found that the differences are indeed significant in all instances at the 0.01 significance level.

Based on this analysis we conclude that SPEA2, NSGA-II and e-MOEA can be considered as the best techniques for solving the MVCCPO problem, since E-NPGA2 and PESA the best that can achieve is incomparability in three problem instances (E-NPGA2 vs e-MOEA). SPEA2 dominates (i.e., wins in both metrics) both E-NPGA2 and PESA in all problem instances while NSGA-II dominates E-NPGA2 in four instances and PESA in all instances. However, PESA is the fastest approach among all algorithms.

A remarkable result of the experiments is that all MOEAs are better than the single objective evolutionary algorithm and they doing this in less CPU time and much less solutions generation. The best that SOEA can achieve is incomparability with PESA on two instances and with E-NPGA2 on one instance. An interesting result is that SOEA has the worst performance in terms of ε-indicator for all problem instances. Furthermore, the small standard deviation for all algorithms reveals that are reliable algorithms for solving the MVCCPO model. Again, SPEA2 provides the most reliable results for all algorithms.

For illustrative purposes we give the efficient frontier of the algorithms for the largest problem instance, the Japanese Nikkei 225 (P5), along with the true cardinality constrained efficient frontier (TCCEF) (Figs. 6–11). The TCCEF is obtained from the web page of Professor F. Tardella: http://w3.uniroma1.it/Tardella/homepage.html, and contains 500 efficient points generated using the method described in Cesarone, Scozzari, and Tardella (2008). Due to space limitations we do not provide the efficient frontiers for the other instances. Similar conclusions are drawn by examining these graphs. We see the good performance of MOEAs both in terms of convergence and coverage of the efficient frontier and, on the other hand, the bad performance of SOEA mainly in terms of diversity of efficient points and the coverage of the efficient frontier.

4.3.2. Computational experiments on large scale problem instances

In this section we report the results obtained by running the NSGA-II and SPEA2, the best algorithms observed from the previous experimental comparison, on the large data sets S&P 500 (476 assets) and NASDAQ (2196 assets).

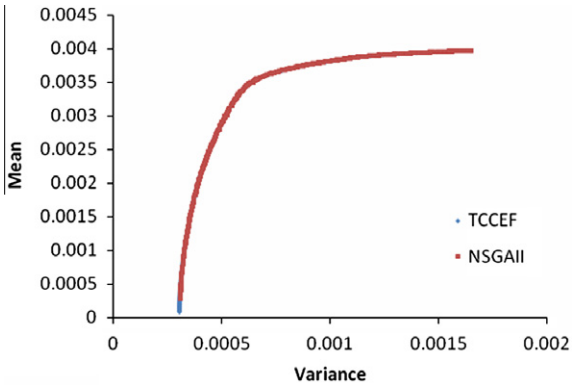

Fig. 6. Efficient frontiers for Japanese Nikkei 225data set, $K = 10$.

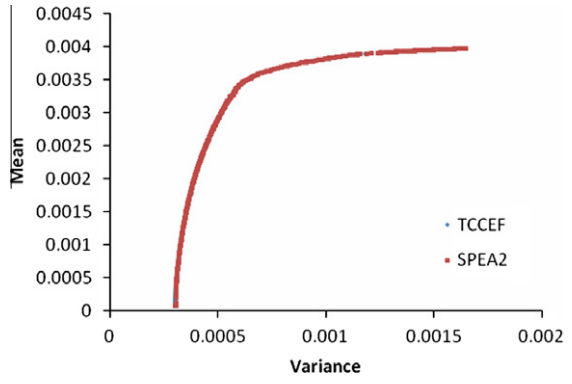

Fig. 7. Efficient frontiers for Japanese Nikkei 225 data set, $K = 10$.

For the S&P 500 we have solved the problem with the same constraint parameters as defined in Section 4.3.1 (i.e., $K = 10$ and $l_i = 0.01$, $u_i = 1$, $i = 1, \ldots, n$). Figs. 12 and 13 illustrate the approximate efficient frontiers obtained by the proposed techniques along with the true cardinality constrained efficient frontier (TCCEF). The TCCEF is given along with the data set and contains 500 efficient points computed using Model 3 and applying the algorithm described in Cesarone et al. (2008).

It is seen that MOEAs generate efficient frontiers very close to the true efficient points with good coverage characteristics (see Fig. 13).

To provide a numerical evidence for the effectiveness of MOEAs and to compare their results with exact efficient points, an

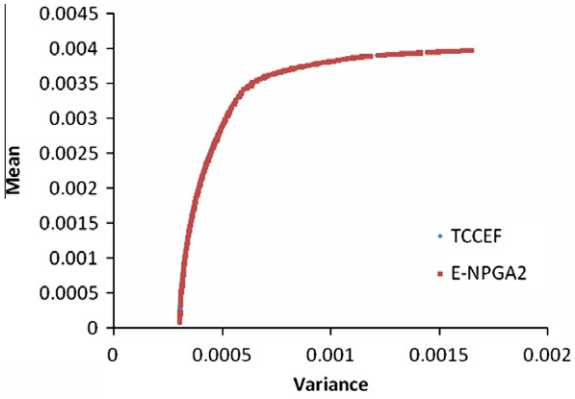

Fig. 8. Efficient frontiers for Japanese Nikkei 225 data set, $K = 10$.

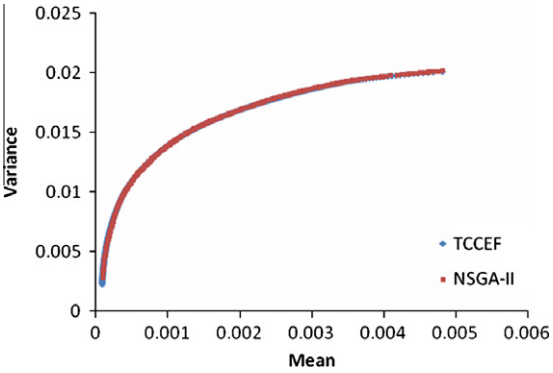

Fig. 12. Efficient frontiers for S&P 500 data set, $K = 10$.

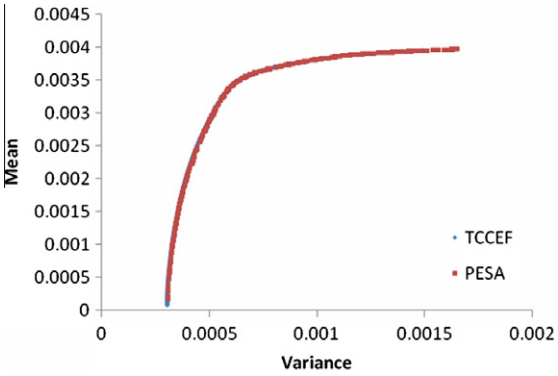

Fig. 9. Efficient frontiers for Japanese Nikkei 225 data set, $K = 10$.

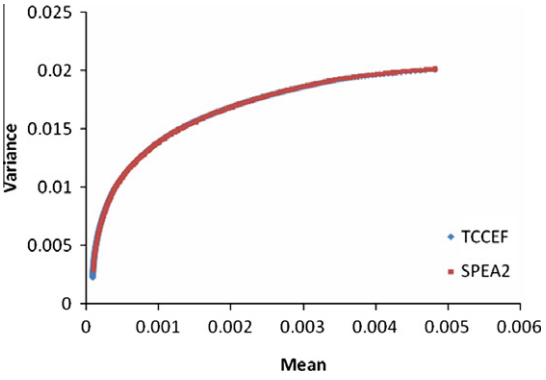

Fig. 13. Efficient frontiers for S&P 500 data set, $K = 10$.

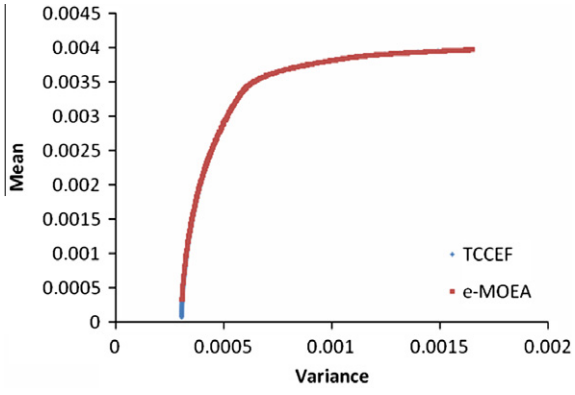

Fig. 10. Efficient frontiers for Japanese Nikkei 225 data set, $K = 10$.

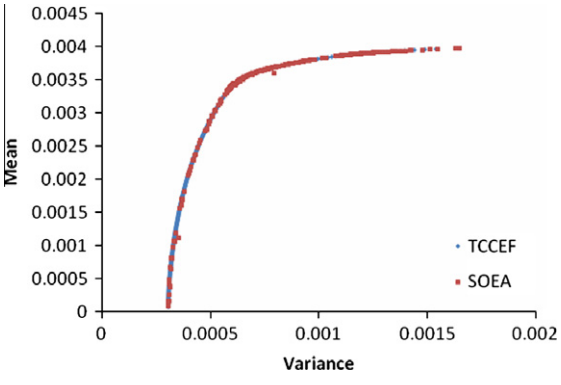

Fig. 11. Efficient frontiers for Japanese Nikkei 225 data set, $K = 10$.

approximation error was computed as follows: For each solution x^a generated by a MOEA, the percent error was calculated as: $\eta = \frac{\|F(x^a) - F(x^r)\|_2}{\|F(x^r)\|_2} 100\%$, where x^r is the solution from the reference efficient frontier that gives the smallest Euclidean distance to x^a and $F(x) = [\sqrt{\rho(x)}, \mu(x)]^T$. The approximation error is the average of all percent errors. Note that in $F(x)$ we have used the standard deviation instead of variance in order to have the same scale for both objective functions. As reference efficient frontier we have used the TCCEF when available or otherwise the efficient frontier of the Markowitz model (TMEF).

The average approximation error for SPEA2 and NSGA-II is 0.358% and 0.405%, respectively showing also numerically that the algorithms almost found the exact efficient points with SPEA2 generating a slightly better value. This was achieved, generating only 100,000 solutions and within 457 s for NSGA-II and 607 s for SPEA2 on average.

Applied the Mann–Whitney test for comparing the samples of quality metrics values for NSGA-II and SPEA2 it was observed that, unlike the experiments with the OR-Library (Section 4.3.1), there are no significant differences between the algorithms for both the hypervolume and ε-indicator at the 5% significance level.

For the largest problem instance NASDAQ with 2196 assets the authors give the exact cardinality constrained efficient frontier only when the cardinality constraint parameter $K = 5$. With $K = 10$ their algorithm was unable to provide optimal solutions. We have performed experiments for both instances to show the effectiveness of MOEAs.

The output of MOEAs for $K = 5$ are shown in Figs. 14 and 15. It is clear that both achieve a good coverage of the exact efficient frontier. The approximation error for this instance is 0.905% for SPEA2 and 1.827% for NSGA-II showing also numerically the convergence of the algorithms towards the TCCEF. The computational times for

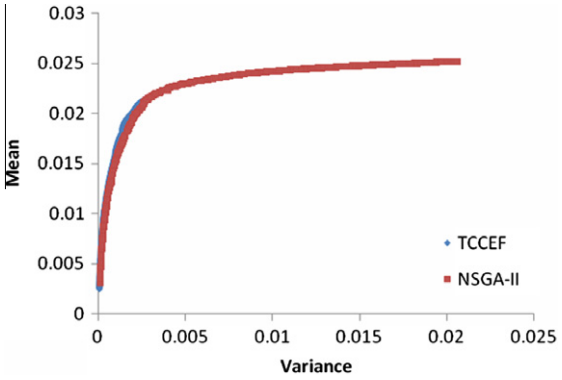

Fig. 14. Efficient frontiers for NASDAQ data set, $K = 5$.

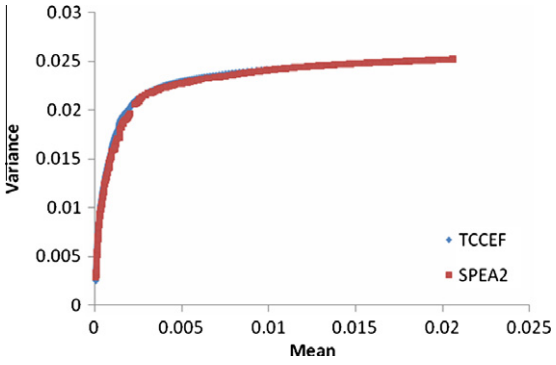

Fig. 15. Efficient frontiers for NASDAQ data set, $K = 5$.

NSGA-II and SPEA2 were 4797 and 4914 s respectively. Compared with the 738,094 s of the Cesarone et al. (2008) algorithm, the performance of MOEAs is rather satisfactory.

For the $K = 10$ instance, as we have already discussed, the authors do not provide any exact efficient points for the MVCCPO problem. For this reason in Figs. 16 and 17 the MOEAs' efficient frontiers are compared with the exact efficient frontier of the standard Markowitz model (TMEF). It is easily seen that MOEAs attain a very good coverage of the TMEF and this reveals that they provide an even better approximation of the (unknown) cardinality constrained efficient frontier. The TMEF provides an upper bound border for any constrained efficient frontier as any additional constraint imposed on the model results in points below the TMEF (see also the analysis in Steuer, Qi, and Hirschberger (2007), Section 6). The approximation error was computed at 6.675% for SPEA2 and 5.79% for NSGA-II. This approximation error is only an

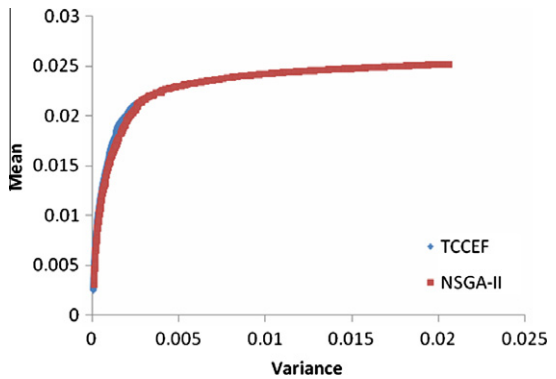

Fig. 16. Efficient frontiers for NASDAQ data set, $K = 10$.

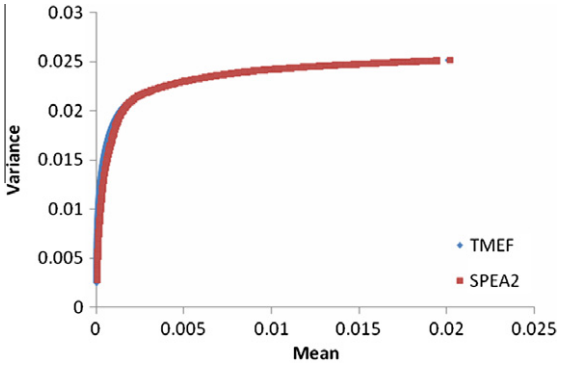

Fig. 17. Efficient frontiers for NASDAQ data set, $K = 10$.

upper bound of the true approximation error to the unknown exact cardinality constrained efficient frontier.

The comparison of the sample of quality metrics values shows that the algorithms are equivalent for this instance since there were no significant differences found for both quality metrics (hypervolume, ε-indicator) using the Mann–Whitney test. The computational times for this instance were 4883 and 4821 s for SPEA2 and NSGA-II, respectively.

5. Conclusion

In this paper, we have presented a computational comparison of the current state-of-the-art multiobjective evolutionary algorithms (MOEAs) on the mean–variance cardinality constrained portfolio optimization problem. Except their popularity, the particular MOEAs were selected in such a way as to capture the main algorithmic operators used in MOEAs design. Furthermore, we have included in the comparison test a variant of a single objective evolutionary algorithm (SOEA). The experiments, which were performed on a public available benchmark data set containing instances that range from 31 to 225 assets, showed a clear superiority of SPEA2. In the second place NSGA-II and e-MOEA seems to have comparable performance. A very interesting result of the study is that all MOEAs outperform the SOEA in all problem instances and this can be demonstrated with visual comparisons as well. This result of MOEAs comes with much fewer solution generation and computational time than SOEA.

Furthermore, the ability of the best performed MOEAs, i.e., NSGA-II and SPEA2 to solve large-scale instances of the MVCCPO problem was shown graphically, as well as, numerically, comparing them with exact efficient points, and using data sets which contain up to 2196 assets.

References

Anagnostopoulos, K. P., & Mamanis, G. (2009). Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science*. doi:10.1007/s10287-009-0113-8.

Anagnostopoulos, K. P., & Mamanis, G. (2010). A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research, 37*, 1285–1297.

Armananzas, R., & Lozano, J. A. (2005). A multiobjective approach to the portfolio optimization problem. In *IEEE congress on evolutionary computation* (pp. 1388–1395).

Bertsimas, D., & Shioda, R. (2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications, 43*, 1–22.

Bienstock, D. (1995). Computational study of a family of mixed integer quadratic programming problems. *Mathematical Programming, 74*, 121–140.

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

Cesarone, F., Scozzari, A., & Tardella, F. (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*.

Chang, T. J., Meade, N., & Beasley, J. E. (2000). Heuristics for cardinality constrained portfolio optimization. *Computers & Operations Research, 27*, 1271–1302.

Chang, T. J., Yang, S. C., & Chang, K. J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications, 36*, 10529–10537.

Chiam, S. C., Tan, K. C., & Mamum, A. Al. (2008). Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing, 5*, 67–80.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2nd ed.). Springer.

Corne, D. W., Knowles, J. D., & Oates, M. J. (2000). The Pareto envelope-based selection algorithm for multi-objective optimization. In *Proceedings of the parallel problem solving from nature (PPSN VI), Lecture notes of computer science* (Vol. 1917, pp. 839–848).

Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research, 150*, 546–571.

Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications, 10*, 2396–2406.

Deb, K., Ptratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA II. *IEEE Transactions on Evolutionary Computation, 6*, 182–197.

Di Tollo, G., & Rolli, A. (2008). Metaheuristics for the portfolio selection problem. *International Journal of Operations Research, 5*, 13–35.

Dueck, G., & Winker, P. (1992). New concepts and algorithms for portfolio choice. *Applied Stochastic Models and Data Analysis, 8*, 159–178.

Elton, E. J., Gruber, M. J., Brown, S. J., & Goetzmann, W. N. (2007). *Modern portfolio theory and investment analysis* (7th ed.). Wiley.

Erikson, M., Mayer, A., & Horn, J. (2001). The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In E. Zitzler, L. Thiele, K. Deb, C.A.C. Coello, & D. Corne (Eds.), *Lecture notes in computer science* (Vol. 1917, pp. 681–695).

Fernández, A., & Gómez, S. (2007). Portfolio selection using neural networks. *Computers & Operations Research, 34*, 1177–1191.

Fieldsend, J. E., Matatko, J., & Peng, M. (2004). Cardinality constrained portfolio optimisation. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 3177, pp. 788–793).

Gilli, M., Këllezi, E., & Hysi, H. (2006). A data-driven optimization heuristic for downside risk minimization. *Journal of Risk, 8*, 1–18.

Hanne, T. (2007). A multiobjective evolutionary algorithm for approximating the efficient set. *European Journal of Operational Research, 176*, 1723–1734.

Hirschberger, M., Qi, Y., & Steuer, R. E. (2010). Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming. *European Journal of Operational Research, 204*, 581–588.

Hochreiter, R. (2007). *An evolutionary computation approach to scenario-based risk-return portfolio optimization for general risk measures. Lecture notes in computer science* (Vol. 4448). Springer, pp. 199–207.

Horn, J., Nafpliotis, N., & Goldberg, G. E. (1994). *A niched Pareto genetic algorithm for multiobjective optimization*. Piscataway, NJ: IEEE Service Center, pp. 82–87.

Jobst, N. J., Horniman, M. D., Lucas, C. A., & Mitra, G. (2001). Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance, 1*, 489–501.

Knowles, J., Thiele, L., & Zitzler, E. (2006). *A tutorial on the performance assessment of stochastic multiobjective optimizers*. TIK Report 214. ETH Zurich: Computer Engineering and Networks Laboratory (TIK).

Laumanns, M., Zitzler, E., & Thiele, L. (2000). A unified model for multi-objective evolutionary algorithms with elitism. In *IEEE congress on evolutionary computation* (pp. 46–53).

Li, D., Sun, X., & Wang, J. (2006). Optimal lot solution to cardinality constrained mean–variance formulation for portfolio selection. *Mathematical Finance, 16*, 83–101.

Maringer, D. G., & Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *OR Spectrum, 25*, 481–495.

Markowitz, H. M. (1952). Portfolio selection. *Journal of Finance, 7*, 77–91.

Markowitz, H. M. (1987). *Mean–variance analysis in portfolio choice and capital markets*. Blackwell Publishers.

Markowitz, H. M. (1990). *Portfolio selection, efficient diversification of investments*. Cambridge MA and Oxford UK: Blackwell.

Mitra, G., Kyriakis, T., Lucas, C., & Pirbhai, M. (2003). A review of portfolio planning: Models and systems. In S. Satchell, & A. Scowcroft (Eds.), *Advances in portfolio construction and implementation* (pp. 1–39).

Mukerjee, A., Biswas, R., Deb, K., & Mathur, A. P. (2002). Multi-objective evolutionary algorithms for the risk-return trade-off in bank loan management. *International Transactions in Operational Research, 9*, 583–597.

Schaerf, A. (2002). Local search techniques for constrained portfolio selection problems. *Computational Economics, 20*, 177–190.

Schlottmann, F., & Seese, D. (2004). Financial applications of multi-objective evolutionary algorithms: Recent development and future research directions. In C. Coello-Coello & G. Lamont (Eds.), *Applications of multi-objective evolutionary algorithms* (pp. 627–652). World Scientific.

Shaw, D. X., Liu, S., & Kopman, L. (2008). Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optimization Methods and Software, 23*, 411–420.

Skolpadungket, P., Dahal, K., & Harnpornchai, N. (2007). Portfolio optimization using multi-objective genetic algorithms. In *IEEE congress on evolutionary computation, CEC 2007* (pp. 516–523).

Soleimani, H., Golmakani, H. R., & Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications, 36*, 5058–5063.

Stein, M., Branke, J., & Schmeck, H. (2007). Efficient implementation of an active set algorithm for large-scale portfolio selection. *Computers & Operations Research, 35*, 3945–3961.

Steuer, R. E., Qi, Y., & Hirschberger, M. (2007). Suitable-portfolio investors, nondominated frontier sensitivity, and the effect of multiple objectives on standard portfolio selection. *Annals of Operations Research, 152*, 297–317.

Streichert, F., Ulmer, H., & Zell, A. (2003). Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In: *Selected papers of the international conference on operations research* (pp. 253–260).

Streichert, F., Ulmer, H., & Zell, A. (2004a). Comparing discrete and continuous genotypes on the constrained portfolio selection problem. In *Proceedings of conference on genetic and evolutionary computation. Lecture notes in computer science* (Vol. 3103, pp. 1239–1250).

Streichert, F., Ulmer, H., & Zell, A. (2004b). Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. *Proceedings of congress on evolutionary computation* (Vol. 1, pp. 932–939). Portland, USA: IEEE Press.

Tapia, M. G. C., & Coello, C. A. C. (2007). Applications of multi-objective evolutionary algorithms in economics and finance. In *IEEE congress on evolutionary computation* (pp. 532–539).

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: *Improving the strength Pareto evolutionary algorithm, TIK-103*. Zurich, Switzerland: Department of Electrical Engineering, Swiss Federal Institute of Technology.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*, 257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation, 7*, 117–132.