Comput Manag Sci (2011) 8:259–279
DOI 10.1007/s10287-009-0113-8

ORIGINAL PAPER

# Multiobjective evolutionary algorithms for complex portfolio optimization problems

**Konstantinos P. Anagnostopoulos · Georgios Mamanis**

Received: 23 October 2009 / Accepted: 4 November 2009 / Published online: 18 November 2009


**Abstract** This paper investigates the ability of Multiobjective Evolutionary Algorithms (MOEAs), namely the Non-dominated Sorting Genetic Algorithm II (NSGA-II), Pareto Envelope-based Selection Algorithm (PESA) and Strength Pareto Evolutionary Algorithm 2 (SPEA2), for solving complex portfolio optimization problems. The portfolio optimization problem is a typical bi-objective optimization problem with objectives the reward that should be maximized and the risk that should be minimized. While reward is commonly measured by the portfolio's expected return, various risk measures have been proposed that try to better reflect a portfolio's riskiness or to simplify the problem to be solved with exact optimization techniques efficiently. However, some risk measures generate additional complexities, since they are non-convex, non-differentiable functions. In addition, constraints imposed by the practitioners introduce further difficulties since they transform the search space into a non-convex region. The results show that MOEAs, in general, are efficient and reliable strategies for this kind of problems, and their performance is independent of the risk function used.

**Keywords** Multiobjective optimization · NSGA-II · PESA · Portfolio selection · SPEA2

K. P. Anagnostopoulos · G. Mamanis (✉)
Department of Production and Management Engineering,
Democritus University of Thrace, Xanthi, Greece
e-mail: gmamanis@pme.duth.gr

K. P. Anagnostopoulos
e-mail: kanagn@civil.duth.gr

## 1 Introduction

Portfolio optimization, that is, the optimal allocation of wealth among a finite number of assets, is one of the most important issues in finance. Since the pioneering work of Markowitz the problem is formulated as a bi-criterion optimization problem, where the reward (measured by the mean of return) of a portfolio should be maximized, while the risk (measured by the variance of return) should be minimized (Markowitz 1952, 1990).

Despite its wide use even today, the Markowitz model has often been criticized, mainly for the validity of variance as a measure of risk. Consequently, several risk measures have been proposed related to possible losses of the portfolio, the so-called quantile-based risk measures. These measures replace variance in the standard mean-variance model, thus leading to an entirely family of mean-risk portfolio selection models (Mansini et al. 2003).

Probably the most popular among the quantile-based risk measures, Value-at-Risk (VaR) describes the maximum loss of a portfolio that will not be exceeded during a specified period with a given probability. An advantage of VaR is that its application does not require any distribution assumption of returns to be made. However, it is a very difficult function to optimize since it is non-linear, non-convex, non-differentiable, and its graph can have many local minima and maxima (Gaivoronski and Pflug 2004); and it fails to be a coherent measure of risk (Artzner et al. 1999). For this reason an alternative risk measure related to VaR has been suggested, namely Expected Shortfall (ES, or Conditional Value-at-Risk) which indicates the expected loss in the worst cases (Acerbi and Tasche 2002; Rockafellar and Uryasev 2000). In this study variance in the standard mean-variance model is replaced with these two quantile-based risk measures.

Apart from the criticism about variance, practitioners have proposed several new constraints that are faced in real financial decision making in order, for example, to avoid very small holdings, to restrict the total number of holdings and/or to take into consideration the roundlot of assets that can be bought or sold in a bunch (Mitra et al. 2003). In any case the portfolio selection problem becomes combinatorial as the additional constraints require the introduction of integer variables. Whatever risk measure is used, the resulting mixed integer optimization problem becomes quite complex as it exhibits multiple local extrema and discontinuities (Chang et al. 2000; Crama and Schyns 2003; Gilli and Kellezi 2002; Jobst et al. 2001).

Replacing variance with quantile-based risk measures and including additional constraints to the portfolio selection model prevents classical optimization methods from being applicable, and heuristic techniques are the only alternatives for finding optimal or near-optimal solutions in a reasonable amount of time. One of the first attempts for the use of heuristics to portfolio selection was made by Dueck and Winker (1992). They use a local search technique, called Threshold Accepting, in a portfolio selection problem with semi-variance as a risk measure. Chang et al. (2000) extended the standard mean–variance model to include cardinality constraints, as well as upper and lower bounds on the proportion of the portfolio invested in each asset; and applied three heuristic algorithms based upon genetic algorithms, tabu search and simulated annealing. Various algorithms have also been proposed for solving the constrained portfolio selection problem: a branch-and-bound algorithm combined with heuristics (Jobst et al. 2001); hill climbing, simulated annealing and tabu search (Schaerf 2002); a simulated annealing algorithm applied to an extended version of the model with trading and turnover constraints (Crama and Schyns 2003); a hybrid local search algorithm which combines principles of simulated annealing and evolutionary strategies (Kellerer and Maringer 2003); a GRASP algorithm enhanced by a learning mechanism and a bias function (Anagnostopoulos et al. 2004); a threshold accepting heuristic for minimizing value-at-risk and expected shortfall (Gilli et al. 2006); a genetic algorithm to solve three different models for portfolio selection with minimum transaction lots (Lin and Liu 2008).

Most of the literature transforms the bi-objective optimization problem into a single objective one usually by minimizing some risk measure subject to a return constraint. Tracing out the efficient frontier with these methods requires the problem to be solved several times for various values of return, and this may be time consuming.

This study tries to optimize return and risk simultaneously, without specifying any lower or upper limit in either return or risk. For doing this state-of-the-art Multiobjective Evolutionary Algorithms (MOEAs) are used, namely Non-dominated Sorting Genetic Algorithm II (NSGA-II), Pareto Envelope-based Selection Algorithm (PESA) and Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Corne et al. 2000; Deb et al. 2002; Zitzler et al. 2001). Our intent is to investigate if MOEAs can tackle complex portfolio optimization problems efficiently and reliably, as well as to make a cross-algorithm performance comparison in order to identify if there is an apparent dominant technique.

The structure of the paper is as follows. In Sect. 2, after a short review of the standard mean–variance model, the constrained mean-risk portfolio selection is defined as a bi-objective combinatorial problem. An implementation of the state of the art MOEAs for solving the problem is presented in Sect. 3. Section 4 is devoted to numerical results, and some concluding remarks are presented in Sect. 5.

## 2 The portfolio optimization problem

### 2.1 The standard portfolio optimization problem

We consider a standard portfolio optimization problem in the sense that there is a single investment period and $n$ assets available for investment. At the beginning of the period the investor determines the proportion weights $x = (x_1, \ldots, x_n)$ of the initial budget that will be allocated in the available assets. The future portfolio return is a random variable denoted by $R(x) = \sum_{i=1}^{n} x_i R_i$ where the $R_i$'s are the random returns on $n$ assets to be realized at the end of the holding period. The goal of the investor is to maximize the random portfolio return under the natural constraint that the sum of the weights, being proportions, must sum to one.

Since Markowitz's seminal paper this stochastic optimization problem is formulated as a mean-risk bi-objective optimization problem, where the expected portfolio return is maximized and a risk measure is minimized subject to constraints that define the feasible portfolios.

$$\begin{aligned}&\max \ \mu(x)\\&\min \ \rho(x)\\&s.t.\ x \in X.\end{aligned}$$

Since there are two criteria, the aim is not to find a single optimal solution (portfolio), but rather a set of optimal portfolios, the so-called Pareto-optimal or efficient portfolios. The set of Pareto-optimal portfolios is constituted by all feasible portfolios that are not dominated by any other portfolio in the feasible set.

$$P = \{x \in X | \neg\exists\, y \in X : y \succ x\}.$$

The symbol $\succ$ stands for the Pareto dominance relation, i.e. a solution $y$ is said to dominate a solution $x$ if $\mu(y) > \mu(x)$ and $\rho(y) \leq \rho(x)$ or $\mu(y) \geq \mu(x)$ and $\rho(y) < \rho(x)$.

Formulated by Markowitz, the *standard mean–variance* model uses variance as a risk measure thus resulting in the following nonlinear bi-criterion optimization problem.

$$\begin{aligned}&\max \ \mu(x) = E[R(x)] = \sum_{i=1}^{n} x_i \mu_i\\&\min \ \rho(x) = V[R(x)] = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij}\\&s.t.\ x \in X = \left\{ x \in \Re^n \left| \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \right.\right\},\end{aligned}$$

where $V$ stands for variance, and the means $\mu_i$, variances $\sigma_{ii}$ and covariances $\sigma_{ij}$ are assumed to be known. The budget constraint requires that all the available capital is invested, and the non-negativity of the weights means that no short sales are allowed.

## 2.2 Practical portfolio optimization problems

The Markowitz model has been criticized not only for the main assumptions it is based upon, such as returns are normally distributed, but also because it neglects some important aspects of portfolio performance in real life situations. These lead to new research directions on portfolio models.

One way in this direction is the replacement of variance with other risk measures, which are more appropriate to quantify the losses that might be incurred. In this work variance is replaced with two quantile-based risk measures, i.e. Value-at-Risk (VaR) and Expected Shortfall (ES). Their main advantage is that they do not assume the returns follow any particular distribution. Accordingly, we rely on the non-parametric methods to estimate these quantities (Benati and Rizzi 2007).

The second way of improvement is the introduction of practical constraints that are taken into account by portfolio managers. For example, it is useful to avoid very small holdings, and to restrict the total number of assets held. These requirements can be modeled as quantity and cardinality constraints. Furthermore, class constraints can be added to the model in order to limit the proportion of the portfolio invested in assets with common characteristics (bank assets, insurance assets, etc). In general, however, these constraints lead to sets of discrete variables and transform the feasible space $X$ into a non-convex set.

In this study the following *constrained mean-risk* portfolio optimization model is considered.

$$\min \quad \rho(x)$$

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

$$s.t. \quad \sum_{i=1}^{n} x_i = 1 \tag{1}$$

$$\sum_{i=1}^{n} \delta_i = K \tag{2}$$

$$l_i \delta_i \le x_i \le u_i \delta_i, \quad i = 1, \ldots, n \tag{3}$$

$$L_m \le \sum_{i \in C_m} \le U_m, \quad m = 1, \ldots, M \tag{4}$$

$$\delta_i \in \{0, 1\}, \quad i = 1, \ldots, n. \tag{5}$$

The additional constraints are described through Eqs. 2–5. Equations 2 and 3 describe cardinality and quantity constraints. Both constraints are computed using a binary variable $\delta_i$ which is equal to 1 if asset $i = 1, \ldots, n$ is held in the portfolio and 0 otherwise. Equation 4 describes class constraints. Introduced by Chang et al. (2000), class constraints are used to limit the total proportion invested in assets with common characteristics leading to a more diversified and safe portfolio. Classes are considered to be mutually exclusive sets of assets, i.e. $C_i \cap C_j = \oslash$ for all $i \neq j$. In this study it is assumed that $L_m > 0$ for every class $C_m$, $m = 1, \ldots, M$. Since $L_m > 0$ for every $m = 1, \ldots, M$ at least one asset from each class should be selected. Consequently, the number of assets in the portfolio should be more than the number of classes, i.e. $K \ge M$.

The risk function $\rho(x)$ is either described by VaR or ES. Next we show the equations utilized for computing the risk of a portfolio using scenarios, since it is not assumed that returns follow any particular distribution (order estimators). Let $r_{it}$ be the observed return of asset $i$ at period $t$. We suppose that each period defines a different scenario with an associated probability $p_t$. All scenarios are considered equally likely, thus $p_t = 1/T$, where $T$ is the total number of scenarios.

For a portfolio $x$ its realization under scenario $t$ is given by the following equation:

$$z_t(x) = \sum_{i=1}^{n} r_{it} x_i, \quad t = 1, \ldots, T.$$

The expected return of the portfolio is therefore given by $\mu(x) = \sum_{t=1}^{T} z_t(x) p_t$.

The VaR at a given confidence level $\alpha$ is the maximum level of loss that the portfolio will not exceed with a probability $\alpha$. Probability $\alpha$ is a user specified parameter and is usually set at a very small number (e.g. 0.01, 0.05 or 0.1) in order to account only for extreme losses. The negative sign is used next in order to describe the loss since $z_t(x)$ describes return.

$$VaR_\alpha(x) = -inf \left\{ z_{(t_\alpha)} \middle| \sum_{j=1}^{t_\alpha} p_{(j)} \geq \alpha \right\},$$

where $z_{(j)}$ are the ordered returns such that $z_{(1)}(x) \leq z_{(2)}(x) \leq \cdots \leq z_{(T)}(x)$ and $p_{(j)}$ their corresponding probabilities of occurrence.

On the other hand, ES is the average loss conditioned that exceeds VaR.

$$ES_\alpha(x) = -E\left\{z_t(x) \,|z_t(x) < -VaR_\alpha(x)\right\}$$

$$ES_\alpha(x) = \frac{\sum_{t=1}^{T} z_t(x) \mathbf{1}_{\{z_t(x) < -VaR_\alpha(x)\}}}{\sum_{t=1}^{T} \mathbf{1}_{\{z_t(x) < -VaR_\alpha(x)\}}},$$

where $\mathbf{1}_{\{z_t(x) < -VaR_\alpha(x)\}}$ is 1 if the expression in brackets is true and zero otherwise.

## 3 Evolutionary multiobjective optimization

Unlike traditional evolutionary algorithms, MOEAs use non-dominated ranking and selection to guide the population towards the Pareto front, and diversity preserving techniques to avoid convergence to a single point on the front. All other topics, i.e. population, recombination and mutation operators, remain the same.

Evolutionary multiobjective optimization has received great attention over the past two decades and several techniques have been proposed (Coello 2000; Coello et al. 2007). NSGA-II, PAES, PESA and SPEA2 are among the most cited MOEAs (Coello et al. 2007). The main advantage of MOEAs is their ability to find multiple Pareto-optimal solutions in a single run.

There are few studies that adopt a proper multiobjective approach to portfolio selection, most of them using a mean-variance formulation: MOEAs with local search for feasible solutions (Streichert et al. 2003); a MOEA to solve the multi-objective quadratic programming problem (Ong et al. 2005); greedy search, simulated annealing and ant colony optimization (Armananzas and Lozano 2005); NSGA-II, PESA and SPEA2 for solving the standard mean-variance portfolio optimization problem (Diosan 2005); a hybrid multiobjective optimization approach combining evolutionary computation with linear programming (Subbu et al. 2005); a MOEA with an ordered based representation (Chiam et al. 2008); a hybrid algorithm combining NSGA-II with the critical line algorithm (Branke et al. 2009); a differential evolution algorithm for multiobjective portfolio optimization (Krink and Paterlini 2009).

## 3.1 Tested algorithms

All MOEAs applied here fit the following general framework outlined in Laumanns et al. (2000), and apply in a different manner the operators. One difference is that PESA applies the evaluate operator after truncation in the archive of the next generation.

$t = 0$
$(A^0, B^0) = initialize()$
**while** $(termination = \mathit{false})$ **do**
  $evaluate(A^t, B^t)$
  $A^{t+1} = truncate(update(A^t, B^t))$
  $B^{t+1} = variation(sample(A^{t+1}))$
  $t = t + 1$
**end while**
**return** $truncate(update(A^t, B^t))$

The main concept of the algorithms is that they involve two populations of individuals. Population $A$, usually called archive or external population, is used to retain the "best" solutions found so far; while $B$ is the usual population of individuals, sometimes used to simply store the offspring population and some other times it takes part in the reproduction process. At first, the archive $A^0$ is set to the empty set and the population $B^0$ to a random sample of the search space through the *initialize* operator. At each generation the *evaluate* operator assigns fitness to individuals from both the archive and the normal population $B$, 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 *update* operator returns the "best" individuals from both the archive and population. If the number of solutions exceeds the user-specified maximum archive size $N^{arc}$, then a *truncate* operator deletes the surplus solutions. *Sample* and *variation* operators specify the particular selection and reproduction scheme, and are the same as in traditional evolutionary algorithms. At the last stage the "best" $N^{arc}$ solutions from the archive and the final offspring population are returned by the algorithm.

The main difference among the algorithms lies in their fitness assignment technique (*evaluate* operator). All MOEAs calculate two values to discriminate between"good" and "bad" individuals. The first one is used to emphasize non-dominated solutions in order to guide the search towards the efficient frontier. The second value serves to discriminate between individuals which are equivalent in terms of the dominance relation in order to preserve diversity. However, the particular method each MOEA employs for achieving the two fundamental goals differs.

For achieving the first target, NSGA-II implements the dominance depth method which classifies solutions in several layers based on which front an individual is located. On the other hand, SPEA2 does this by using a finer grained technique, which combines the dominance count and dominance rank method. With this technique, an individual is assigned a value based on the number of solutions it dominates and is dominated by. PESA achieves that goal by retaining only non-dominated individuals in the archive through the *update* operator.

For preserving diversity, NSGA-II and SPEA2 compute a density value based on the nearest neighbor. NSGA-II calculates the volume of the hyperrectangle defined by the nearest neighbors; while SPEA2 uses a clustering approach which computes 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. On the other hand, PESA uses a histogram approach which divides the objective space into hyper-boxes and calculates for each solution the number of other solutions in the archive that reside in the same box. This process for PESA can be seen as secondary information, since it is applied only to non-dominated individuals.

These methods for generating non-dominated solutions and preserving diversity represent the main theory in evolutionary multiobjective optimization literature. For more information on MOEAs the reader is referred to the recent book of Coello et al. (2007) and the original studies. Next we present only those problem specific components which are required for solving the portfolio optimization models.

*Sample* and *variation* operators are the same as in usual evolutionary algorithms. Identical schemes for all tested algorithms are used in order to ensure a fair comparison. For selecting the parents, binary tournament selection is used for all algorithms. *Variation* operators are described in Sect. 3.3.

### 3.2 Quality metrics

Evaluating the quality of solution sets produced by different algorithms or different algorithm configurations is not as easy as in the single objective case where the obvious quality measure is the objective function value. The quality of an approximate efficient frontier depends on various qualitative characteristics, such as the closeness to the true efficient frontier, the uniformity and spread of the solutions along the frontier.

For measuring the closeness of an approximate solution set, we use the $\epsilon$-indicator proposed by Zitzler et al. (2003). This metric gives the minimum value by which each member of a reference set must be multiplied, in order for the resulting set to be dominated by the approximation set. If the approximation set matches exactly the reference set then it takes the identical value of 1 (or 0 in the additive version). Low values for this metric reveal that the approximate solutions generated by a MOEA are very close to the reference set. The exact Pareto-optimal front is usually used as reference set when it is known; otherwise as reference set is used the Pareto-optimal solutions produced by all algorithms and all replicates for a particular problem instance.

For measuring diversity we use the hypervolume indicator (Zitzler and Thiele 1999). The hypervolume indicator measures how well the algorithms perform in identifying solutions along the full extent of the efficient frontier. In the two-dimensional case, it simply measures the surface of the objective space dominated by the solutions produced from the particular algorithm bounded by a reference point. Thus, higher values are preferable for this metric. For the standard mean–variance model the reference point used was (0.003, 0); and for the Mean-ES and Mean-VaR problems was set at (0.05, 0) and (0.03, 0), respectively, where the first coordinate describes risk while the later describes mean (see Sect. 4).

Both indicators are equally important for the quality of solution sets, and for a MOEA to be better than another algorithm it should provide better values for both metrics. In some cases, the two indicators may return opposite preference orderings for a pair of approximation sets. For such a result the two sets should be incomparable (Knowles et al. 2006).

## 3.3 Algorithms implementation

A crucial factor for the performance of evolutionary algorithms is how to incorporate problem-specific knowledge into them. Knowledge may be incorporated either implicitly, in the design of data structures, encoding and constraint representations, or explicitly, via the initialization of the first population and in the control of evolutionary algorithms parameters (Bonissone et al. 2006). For example, Streichert et al. (2003) observed that the optimal portfolios are composed only of a limited number of the available assets. Thus, a hybrid representation was proposed, where an additional binary string is included to reflect the existence of the assets in the portfolio. Such a scheme facilitates the removal and adding of assets to portfolios, resulting in better algorithm performance.

### *3.3.1 Solution representation and encoding*

All algorithms have the same solution representation. For representing a solution three arrays are used, which are symbolized with three Greek capital letters. The array $A$ consists of $M$ real numbers associated with each class; the array $B$ contains $K$ integer numbers, each one representing an asset in the portfolio; and the array $\Gamma$ includes $K$ real numbers associated with each asset. The main motivation for implementing such a representation scheme was to avoid using any penalty method for handling the constraints.

$$
\begin{aligned}
&A = \{\alpha_1, \ldots, \alpha_M\}, \quad 0 \leq \alpha_i \leq 1, \; m = 1, \ldots, M \\
&B = \{\beta_1, \ldots, \beta_M, \beta_{M+1}, \ldots, \beta_K\}, \quad \beta_i \in C_i, \; \beta_j \in \{1, \ldots, n\} - \{\beta_1, \ldots, \beta_M\}, \\
&\qquad i = 1, \ldots, M, \; j = M+1, \ldots, K \\
&\Gamma = \left\{\gamma_{\beta_1}, \ldots, \gamma_{\beta_K}\right\}, \quad 0 \leq \gamma_{\beta_i} \leq 1, \; i = 1, \ldots, K.
\end{aligned}
$$

With this solution representation we account for the cardinality constraint since the array $B$ has exactly the required size. Moreover, we ensure that each class has a "representative" in the portfolio since the first $M$ assets are selected from each class, while the remaining $K - M$ assets are selected from the universe except, of course, the ones that are already in.

For example, one chromosome might be the following. Suppose that $n = 96$, $M = 6$ and each class contains 16 assets indexed with increasing number (i.e. class 1 contains the assets 1 to 16, class 2 the assets 17 to 32, etc.). Suppose also that exactly $K = 10$ assets must be in the portfolio. A solution might therefore be:

$$\begin{aligned} A &= \{0.1, 0.2, 0.5, 0.4, 0.3, 0.6\} \\ B &= \{1, 20, 35, 48, 71, 80, 5, 25, 90, 60\} \\ \Gamma &= \{\gamma_1 = 0.1, \gamma_{20} = 0.3, \gamma_{35} = 0.8, \gamma_{48} = 0.5, \gamma_{71} = 0.7, \gamma_{80} = 0.9, \gamma_5 = 0.4, \\ &\quad \gamma_{25} = 0.3, \gamma_{90} = 0.2, \gamma_{60} = 0.65\}. \end{aligned}$$

Of course this solution representation does not tell anything about the real proportions invested in the assets of array $B$. To find the real portfolio $x$, the solution must be repaired. A repair mechanism is used in such a way as to the portfolio $x$ satisfies quantity and class constraints.

First array $A$ is normalized to find the real class proportion ($rcp$) associated with each class $m$

$$rcp(m) = L_m + \frac{\alpha_m}{\sum_{j=1}^{M} \alpha_j} \left(1 - \sum_{j=1}^{M} L_j\right), \quad m = 1, \ldots, M.$$

In this way class proportions both satisfy lower limits and the summation to one.

Using the above formula we can compute the real proportion that will be invested in each class for the solution defined above. Suppose that $L_m = 0.05$ for all $m = 1, \ldots, M$. We have $1 - \sum_{j=1}^{M} L_j = 0.7$ and $\sum_{j=1}^{M} \alpha_j = 2.1$. Thus, the real proportion $rcp$ of the portfolio that will be assigned in the classes 1 and 2 is $rcp(1) = 0.05 + 0.1 \times 0.7/2.1 = 0.083$, and $rcp(2) = 0.05 + 0.2 \times 0.7/2.1 = 0.117$. Using the above formula for each class, we calculate $rcp(3) = 0.217$, $rcp(4) = 0.183$, $rcp(5) = 0.15$, $rcp(6) = 0.25$. Note that class proportions add to one and they satisfy lower bounds.

Next, the real proportion of each class is shared in the corresponding assets of array B. The proportion associated with each asset in the portfolio is calculated by the following equation:

$$x_{\beta_i} = l_{\beta_i} + \frac{\gamma_{\beta_i}}{\sum_{j \in B \cap C_{class(\beta_i)}} \gamma_j} \left(rcp(class(\beta_i)) - \sum_{j \in B \cap C_{class(\beta_i)}} l_j\right), \quad i = 1, \ldots, K. \tag{6}$$

For every $i \notin B \Leftrightarrow x_i = 0$. $class(\beta_i)$ returns the class that the asset $\beta_i$ belongs to.

If we suppose that $l_i = 0.01$ for every $i = 1, \ldots, n$, we calculate for our example: $x_1 = 0.01 + (0.083 - 0.02) \times 0.1/0.5 = 0.0226$. Assets 1 and 5 belong to the same class. Using the above formula we calculate the real proportion invested in that asset $x_5 = 0.0604$. Note that the sum of the weights for these assets equals the $rcp$ of class 1 to which they both belong. Using the same formula for all assets in $B$, the real proportion weight invested in each asset can be calculated.

In this way the portfolio $x$ satisfies budget, lower quantity and lower class constraints. To account for upper quantity and class bounds, we propose the following repair algorithm similar with the one outlined in Chang et al. (2000). If a real class proportion $rcp(m)$ violates its upper limit $U_m$, then the real class proportion for the particular class $m$ takes this value and the remaining class values are normalized again according to Eq. 6. This process is continued until all upper limits are satisfied. The same process is used for the real asset proportions.

Having a feasible portfolio, the objective functions can be evaluated as described in Sect. 2. After this, each MOEA calculates the fitness and maintain the population based on their algorithmic specific components until a maximum number of generations is reached.

### *3.3.2 Variation operators*

For generating the offspring population, the uniform crossover is used. The operators are applied independently in each array. Regarding the first part (array $A$) the values associated with each class are selected with equal probability from one or another parent. With reference to the second and third part (arrays $B$ and $\Gamma$), we combine the assets from both parents along with their associated values in a new multi-set. The resulting multi-set thus consists of pairs of assets and their associated values. If an asset is present in both parents the associated value is chosen with equal probability from one or another parent. The child then is created by selecting the assets along with their associated values from the newly defined set. The decision which asset to choose is done at random, but in such a way as to every class has a representative (i.e. one asset from each class is always selected first, since it is assumed that $L_m > 0$ for every class $C_m, m = 1, \ldots, M$).

The following mutation operator was applied for the arrays $A$ and $\Gamma$ and for a randomly chosen $i$.

$$\alpha_i^{mut} = p(\alpha_i - L_i) - L_i, \quad i = 1, \ldots, M$$

$$\gamma_{\beta_i}^{mut} = p(\gamma_{\beta_i} - l_i) - l_i, \quad i = 1, \ldots, K,$$

where $p$ is the step value to be chosen. Considering array $B$'s mutation, one asset (selected randomly) from the portfolio is replaced with another asset from the entire set in such a way as to each class has a "representative" in the portfolio.

## 4 Computational experiments

### 4.1 The standard mean–variance model

A performance evaluation in the standard mean-variance model (see Sect. 2.1) was undertaken first. The evaluation was based on a public benchmark data set provided by Beasley (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html). The numerical results are referred to the German DAX 100 data set which contains 85 assets.

The true efficient frontier is given along with the data set. All algorithms have been implemented in Visual C++ and run on a personal computer Core 2 Duo at 2.1 GHz.

We have used the same solution representation and repair algorithm that have been described in Sect. 3.3 without the array $A$. Moreover, the size of array $B$ and $\Gamma$ is exactly the number of available assets, but now $B$ is a bit string that contains the value of 1 if the corresponding asset is in the portfolio and 0 otherwise. Crossover and mutation operators have been also used as described, but now without the restriction imposed by the constraints. For example, in the uniform crossover it is no longer required that any particular assets must be in the portfolio as previously done because of class constraints. The uniform crossover is performed as usual: for each bit in the offspring, the parent who contributes its bit of the corresponding position to the offspring is chosen randomly with equal probability. For handling the budget constraint, the real proportion $x_i$ is calculated by using the equation $x_i = \frac{\beta_i \gamma_i}{\sum \beta_i \gamma_i}, \quad i = 1, \ldots, n$.

Some experimentation with the parameters of the algorithms was done before the evaluation using trial and error analysis. For all algorithms identical population ($N^{pop} = 500$) and archive sizes ($N^{arc} = 500$) have been fixed to ensure a fair comparison. All algorithms were allotted 1000 generations. For NSGA-II and SPEA2 the same crossover and mutation probabilities have been used: $p_c = 1.0$, $p_\Gamma = 1.0$, $p_B = 0.1$, where $p_\Gamma$ and $p_B$ are the probabilities of mutation in arrays $\Gamma$ and $B$ respectively. The mutation operators have been applied independently in each array. Since PESA emphasizes only non-dominated individuals, different values of parameters are required in order to avoid premature convergence: $p_c = 0.7$, $p_\Gamma = 0.7$, $p_B = 0.01$. PESA requires an additional parameter to be set, which is the number of divisions of the search space. After trial and error analysis, that value was set equal to 130. The problem was solved 20 times with e ach algorithm, thus a total of 60 trial runs were carried out.

Figures 1, 2 and 3 represent the approximation set that is closest to the median hypervolume metric for each algorithm along with the true efficient frontier (TEF).

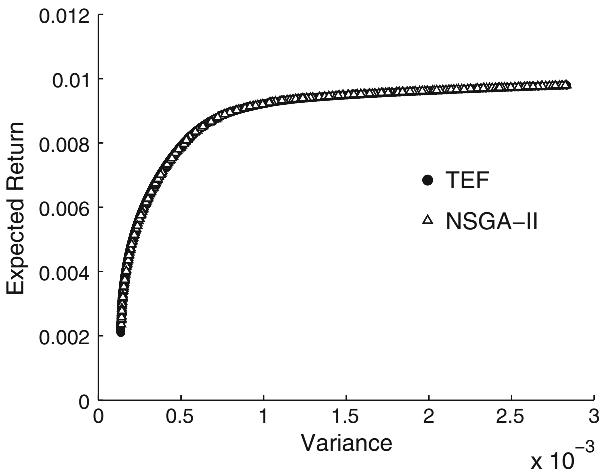


**Fig. 1** NSGA-II approximate and exact standard mean-variance efficient frontiers

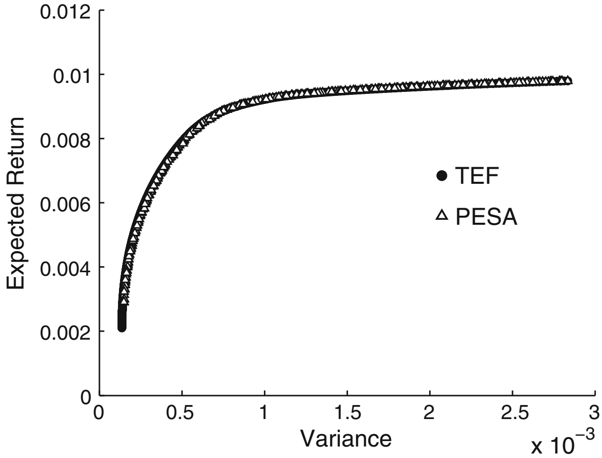


**Fig. 2** PESA approximate and exact standard mean–variance efficient frontiers

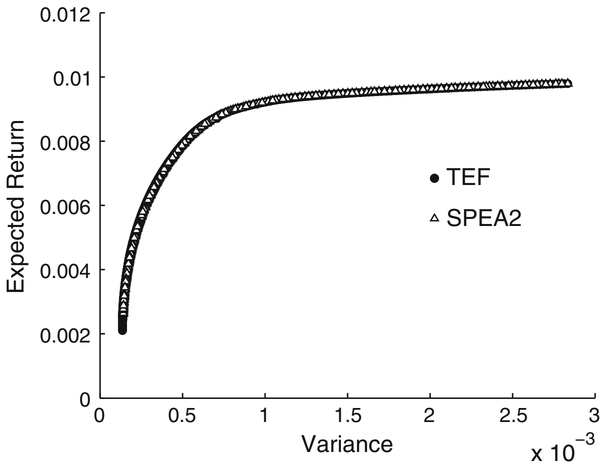


**Fig. 3** SPEA2 approximate and exact standard mean–variance efficient frontiers

These figures show that all algorithms attain a very close approximation of the TEF. Numerically, the $\epsilon$-indicator demonstrates that all algorithms achieve a very good value which is very close to the identical value 1 (Table 1).

As for cross-algorithm performance comparison, Table 1 shows that PESA performs best in terms of the closeness ($\epsilon$-indicator) to the true Pareto front, while SPEA2 has the best performance considering the diversity metric (hypervolume). SPEA2 outperforms NSGA-II because it has better average performance for both indicators, but their values are very close to each other. NSGA-II and PESA are incomparable since PESA performs poorly in terms of the hypervolume indicator. The relatively small standard deviations reveal that MOEAs are reliable strategies for solving the standard mean–variance optimization problem.

**Table 1** Results for the standard mean–variance portfolio problem

| Mean–variance | $\epsilon$-Indicator | | Hypervolume | | Time (s) | |
|---|---|---|---|---|---|---|
| | Median | SD | Median | SD | Median | SD |
| NSGA-II | 1.0596 | 0.0152 | 2.00759E-05 | 1.59475E-09 | **320.48** | 4.25 |
| PESA | **1.0304** | 0.0066 | 2.00627E-05 | 4.45052E-09 | 704.22 | 15.6 |
| SPEA2 | 1.0359 | 0.0105 | **2.00818E-05** | 2.91222E-09 | 1020.54 | 25.2 |

### 4.2 Complex portfolio optimization problems

We present here the numerical results obtained by applying MOEAs in the constrained mean-risk portfolio optimization model (see Sect. 2.2), as well as a cross-algorithm performance comparison. The data for this type of problems were observed from the yahoo finance webpage and they are referred to the S&P 100 index. Daily returns from 3 March 2004 to 20 February 2008 of 96 stocks have been used. It is assumed that each observation defines a different scenario, thus the number of scenarios considered are $T = 1000$. The 96 assets have been classified into six categories ($M = 6$) with 16 assets in each one. In order to test the robustness of the algorithms, a number of instances for different values of $K$ have been solved. The lower proportion allowed for investment in each class was 5%, thus $L_m = 0.05$ for each $m = 1, \ldots, 6$. Given that a lower class bound of 5% specifies an upper bound of 75% of investment in each class/asset, no upper limits have been specified. For the stock weights $x_i$ lower and upper bounds were 1 and 100% respectively, i.e. $l_i = 0.01$, $u_i = 1$, $\forall i \in B$.

The optimal parameters were obtained by solving the Mean-VaR portfolio optimization problem with $K = 10$ and $\alpha = 0.1$. The population and archive size was set again to contain 500 individuals. All algorithms have been terminated after 500 generations. For NSGA-II and SPEA2, the same crossover and mutation probabilities have been used, $p_c = 1.0$, $p_\Gamma = 0.1$, $p_B = 0.1$, whilst for PESA the same parameters were $p_c = 0.7$, $p_\Gamma = 0.005$, $p_B = 0.005$. The number of divisions of the search space was 100. In an effort to test the algorithms' robustness, these parameters and configurations have been utilized for all instances.

Due to inherent randomness of MOEAs, all problem instances were solved 20 times with 20 identical random seeds for all algorithms. Tables 2 and 3 show the median and standard deviation for each quality metric for each algorithm and for each instance.

Concerning the Mean-$ES_{0.1}$ problem with $K = 10$, the first conclusion is that no one algorithm can produce better results for both metrics than another algorithm. PESA generates the best results on average considering the $\epsilon$-indicator, but has the worst average performance in hypervolume metric. On the other hand, SPEA2 has the best average hypervolume, while it fails to generate good results for the proximity indicator ($\epsilon$-indicator). NSGA-II comes on the second place for both metrics. On the other hand, when six assets are allowed to be in the portfolio ($K = 6$), NSGA-II completely outperforms the other two since it wins in both quality indicators, while the other two are incomparable to each other. The standard deviations are very small

**Table 2** Results for Mean-$ES_{0.1}$ portfolio problem

| Instance | Algorithm | $\epsilon$-Indicator | | Hypervolume | | Time (s) | |
|---|---|---|---|---|---|---|---|
| | | Median | SD | Median | SD | Median | SD |
| $K = 10$ | NSGA-II | 1.018 | 0.003 | 6.415E-05 | 8.888E-08 | 213.60 | 0.58 |
| | PESA | **1.013** | 0.006 | 6.411E-05 | 8.762E-08 | **141.37** | 7.13 |
| | SPEA2 | 1.020 | 0.007 | **6.422E-05** | 3.597E-08 | 561.54 | 27.65 |
| $K = 6$ | NSGA-II | **1.008** | 0.007 | **6.205E-05** | 2.234E-08 | **207.07** | 6.34 |
| | PESA | 1.011 | 0.001 | 6.198E-05 | 2.029E-08 | 284.31 | 5.65 |
| | SPEA2 | 1.016 | 0.003 | 6.204E-05 | 1.696E-08 | 1129.5 | 30.68 |

**Table 3** Results for Mean-$VaR_{0.1}$ portfolio problem

| Instance | Algorithm | $\epsilon$-Indicator | | Hypervolume | | Time (s) | |
|---|---|---|---|---|---|---|---|
| | | Median | SD | Median | SD | Median | SD |
| $K = 10$ | NSGA-II | 1.044 | 0.008 | 3.823E-05 | 1.193E-07 | 210.09 | 0.55 |
| | PESA | 1.045 | 0.019 | **3.826E-05** | 1.110E-07 | **87.91** | 1.19 |
| | SPEA2 | **1.043** | 0.004 | 3.824E-05 | 3.936E-08 | 454.71 | 43.34 |
| $K = 6$ | NSGA-II | 1.041 | 0.010 | **3.713E-05** | 2.466E-08 | 204.59 | 1.60 |
| | PESA | **1.037** | 0.005 | 3.701E-05 | 1.150E-08 | **146.7** | 4.15 |
| | SPEA2 | 1.042 | 0.013 | 3.709E-05 | 4.096E-08 | 1838.5 | 128.12 |

for all algorithms meaning that MOEAs are reliable searching techniques for solving this class of problems.

In order to show the effectiveness of MOEAs, their solutions have been compared with results obtained using commercial software. Minimizing ES using scenarios is a linear programming problem as shown in Rockafellar and Uryasev (2000). Consequently, introducing the additional constraints defined in Sect. 2 the problem becomes a Mixed Integer Linear Programming (MILP) problem. Thus, by modeling portfolio return as constraint (which is also a linear function of $x$) and solving for different right hand side values, a number of efficient points can be found. Thirty seven efficient points from the minimum risk to the maximum return portfolio have been computed in this way.

Figures 4, 5 and 6 give the approximate efficient frontiers for the instance with $K = 10$ for each MOEA together with the exact efficient points generated using CPLEX software. The horizontal axis describes the average loss that might be incurred with probability $\alpha = 10\%$. The figures show that MOEAs provide a very good approximation of the true efficient frontier since they have generated a number of efficient points very close to the real ones with good diversity characteristics. Using the 37 exact efficient points as reference set, the $\epsilon$-indicator for NSGA-II, PESA and SPEA2, equal to 1.0088, 1.0082 and 1.012, respectively, reveals also numerically that the approximate portfolios are very close to the optimal ones.

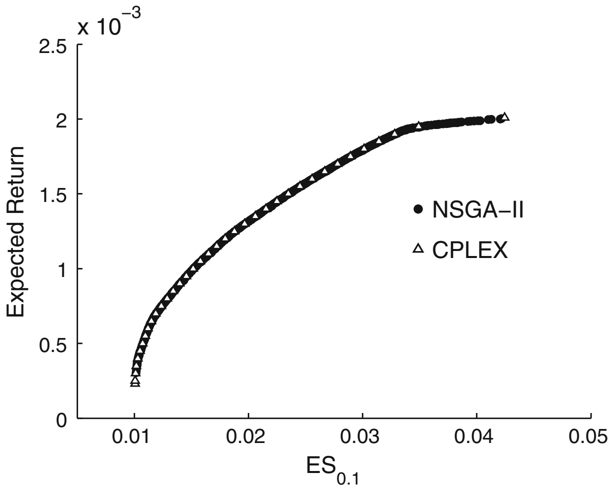


**Fig. 4** NSGA-II approximate efficient frontier together with exact efficient points obtained solving the MILP for the Mean-$ES_{0.1}$ with $K = 10$

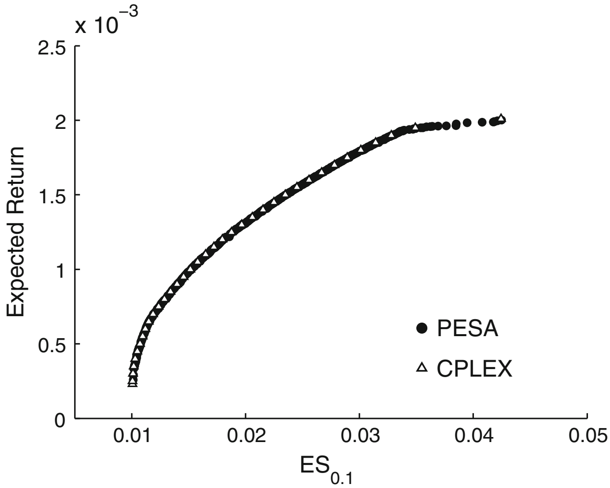


**Fig. 5** PESA approximate efficient frontier together with exact efficient points obtained solving the MILP for the Mean-$ES_{0.1}$ with $K = 10$

The same problem parameters were utilized for the Mean-VaR$_{0.1}$ portfolio problem, thus two instances have been also solved with the cardinality constrained to be ten and six respectively, and the shortfall probability $\alpha = 0.1$ for both instances. For the first instance ($K = 10$), it is concluded that SPEA2 can be no worse than NSGA-II since it generates better results for both $\epsilon$-indicator and hypervolume metric. Moreover, for this instance SPEA2 and PESA are incomparable since they generate conflicting values for each metric. SPEA2 wins in terms of the proximity indicator ($\epsilon$-indicator), while PESA produces the best results considering the hypervolume indicator. However, all

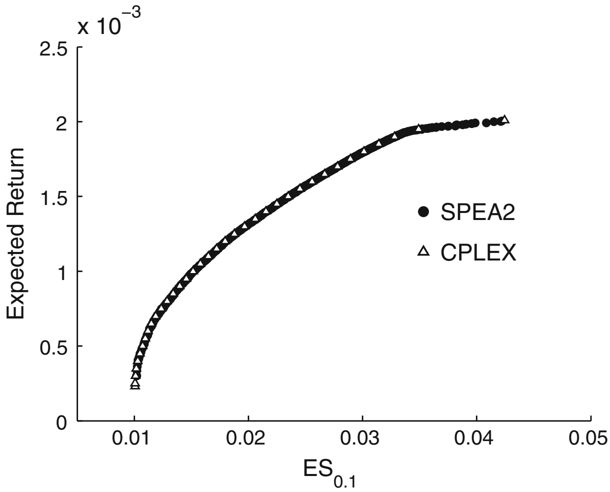


**Fig. 6** SPEA2 approximate efficient frontier together with exact efficient points obtained solving the MILP for the Mean-$ES_{0.1}$ with $K = 10$

algorithms seems to perform equally well in this problem, because their median values are very close to each other for both metrics.

For the second instance, where the cardinality constrained was fixed to be six (i.e. one asset for each class is selected), the results seem to be a little bit different. Here, NSGA-II outperforms SPEA2, since it generates better average values for both quality metrics. However, the values are very close to each other. NSGA-II has the best average quality concerning the diversity indicator, while PESA has the best value in terms of $\epsilon$-indicator, thus making the two algorithms incomparable for this problem instance.

In general, for the Mean-VaR portfolio problem all algorithms have the same quality on average. The difficulty to judge which one produces the best results is illustrated in Figs. 7, 8 and 9, where we provide a typical approximate efficient frontier generated by each MOEA. For this type of problem there is no procedure up to now which allows obtaining the true Pareto optimal points even without the additional constraints. However, Rockafellar and Uryasev (2000) state that minimizing ES (or CVaR) produces a minimum VaR also. On the same graph the 37 optimal portfolios generated from the minimization of ES have been plotted in Mean-VaR space. The graphs show that MOEAs dominate many of these points. The vertical axis here gives the expected return of the portfolio, while the horizontal axis describes the maximum loss that might be incurred with a probability $\alpha = 10\%$.

In terms of computational time, PESA is the fastest algorithm in general, but its running time is heavily influenced by the number of times the truncate operator becomes active. The same can be said for SPEA2, the worst algorithm in terms of run time. NSGA-II was in the second place, but its running time shows more stable performance.

In summary, PESA performs best in terms of $\epsilon$-indicator, since it wins three instances (together with the standard mean-variance problem), while SPEA2 and NSGA-II win one instance. However, PESA performs poorly in terms of the

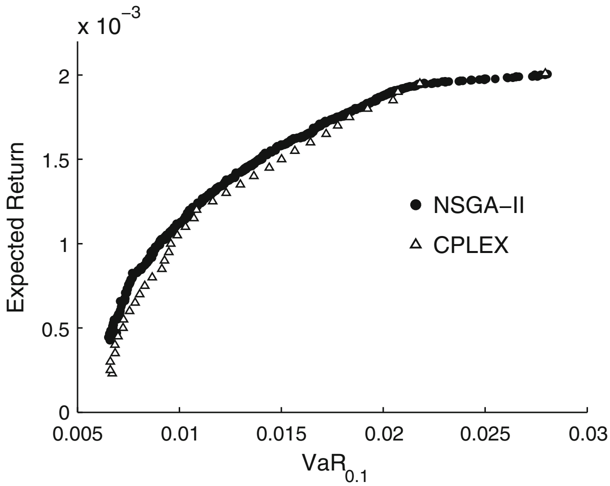


**Fig. 7** NSGA-II approximate efficient frontier together with the approximate efficient points obtained solving the MILP for the Mean-$VaR_{0.1}$ with $K = 10$

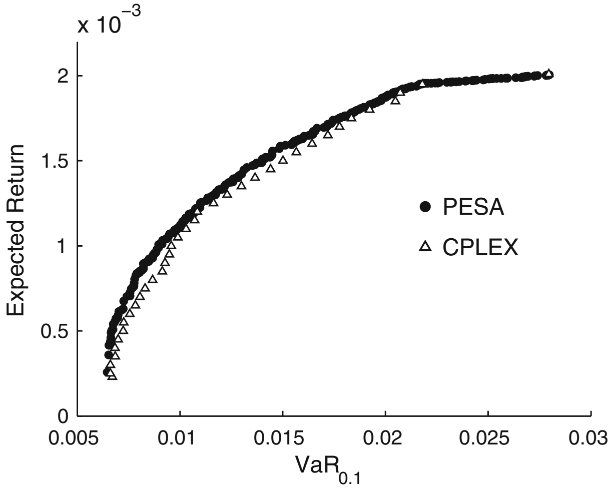


**Fig. 8** PESA approximate efficient frontier together with the approximate efficient points obtained solving the MILP for the Mean-$VaR_{0.1}$ with $K = 10$

hypervolume indicator. NSGA-II and SPEA2, on the other hand, are the best algorithms on average when the hypervolume indicator is considered. Furthermore, NSGA-II completely wins an instance (Mean-$ES_{0.1}$, $K = 6$) while the other two fail to win anyone. In general, NSGA-II and SPEA2 seem to generate very close average results for both metrics. On the other hand, PESA fails to produce better results simultaneously for both metrics than any other optimizer.

We conclude that, in general, all MOEAs were able to generate a very good approximation to the constrained efficient frontier for the problems considered. The small

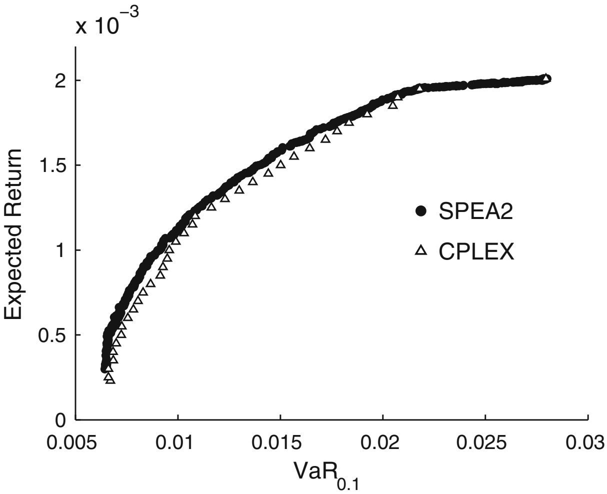


**Fig. 9** SPEA2 approximate efficient frontier together with the approximate efficient points obtained solving the MILP for the Mean-$VaR_{0.1}$ with $K = 10$

standard deviation for all metrics also reveals that are reliable strategies for solving complex portfolio optimization problems. Furthermore, they are easily adapted in any change of the risk function and/or the addition of constraints. The results show that there is no apparent dominant technique. One can use any of the algorithms and would obtain satisfactory results for such problems. However, a better strategy would probably be to run all algorithms for some replicates and then to obtain the efficient portfolios from these replicates.

## 5 Conclusion

In this paper state of the art Multiobjective Evolutionary Algorithms have been applied for solving portfolio optimization problems with complex constraints and risk functions. Two portfolio problems in the sense of Markowitz have been considered but by replacing variance with quantile based risk measures, namely Value at Risk and Expected Shortfall. Furthermore, the model has been enriched with additional constraints that derive from real financial decision making such as quantity constraints, cardinality constraints and class constraints. All algorithms were able to generate very good approximation sets with a number of diverse solutions well extended in the objective space. The techniques have also been evaluated with results provided by exact methods and have shown very good performance. As for cross-algorithm performance comparison, PESA performs the best in terms of the closeness to the Pareto optimal front, while NSGA-II and SPEA2 have the best average performance in t erms of hypervolume indicator. Another advantage of the methods is their flexibility to adapt in any addition of new constraints and/or replacement of the risk function.

## References


Acerbi C, Tasche D (2002) Expected shortfall: a natural coherent alternative to value at risk. Economic Notes by Banca Monte dei Paschi di Siena spA 31(2):379–388

Anagnostopoulos KP, Chatzoglou PD, Katsavounis S (2004) A reactive Greedy Randomized Adaptive Search Procedure for a mixed integer portfolio optimization problem. In: Chatzoglou PD (eds) Proceedings of the 2nd international conference on accounting and finance in transition (CD)

Armananzas R, Lozano JA (2005) A multiobjective approach to the portfolio optimization problem. In: IEEE congress on evolutionary computation, vol 2, pp 1388–1395

Artzner P, Delbaen F, Eber JM, Heath D (1999) Coherent measures of risk. Math Finance 9(3):203–228

Benati S, Rizzi R (2007) A mixed integer linear programming formulation of the optimal mean/Value-at-Risk portfolio problem. Eur J Oper Res 176:423–434

Bonissone PP, Subbu R, Eklund N, Kiehl TR (2006) Evolutionary Algorithms + Domain Knowledge = Real-World Evolutionary Computation. IEEE Trans Evol Comput 10(3):256–280

Branke J, Scheckenbach B, Stein M, Deb K, Schmeck H (2009) Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. Eur J Oper Res 199(3):684–693

Chang TJ, Meade N, Beasley JE, Sharaiha YM (2000) Heuristics for cardinality constrained portfolio optimization. Comput Oper Res 27:1271–1302

Chiam SC, Tan KC, Mamum AAl (2008) Evolutionary multi-objective portfolio optimization in practical context. Int J Autom Comput 5(1):67–80

Coello CAC (2000) An updated survey of GA-based multiobjective optimization techniques. ACM Comput Surv 32(2):109–143

Coello CAC, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, Berlin

Corne DW, Knowles JD, Oates MJ (2000) The pareto envelope-based selection algorithm for multiobjective optimization. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel H-P (eds) Proceedings of the parallel problem solving from nature. VI Conference, Paris, France. Springer. Lecture Notes in Computer Science, vol 1917, pp 839–848

Crama Y, Schyns M (2003) Simulated annealing for complex portfolio selection problems. Eur J Oper Res 150:546–571

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Diosan L (2005) A multi-objective evolutionary approach to the portfolio optimization problem. In: Proceedings of the international conference on computational intelligence for modelling, control and automation, pp 183–188

Dueck G, Winker P (1992) New concepts and algorithms for portfolio choice. App Stoch Models Data Anal 8:159–178

Gaivoronski AA, Pflug G (2004) Value at risk in portfolio optimization: properties and computational approach. J Risk 7:1–31

Gilli M, Kellezi E (2002) A global optimization heuristic for portfolio choice with VaR and expected shortfall. In: Kontoghiorghes EJ, Rustem B, Siokos S (eds) Computational methods in decision-making, economics and finance, Applied Optimization Series. Kluwer Academic Publishers, Dordrecht, pp 167–183

Gilli M, Kellezi E, Hysi H (2006) A data-driven optimization heuristic for downside risk minimization. J Risk 8(3):1–18

Jobst NJ, Horniman MD, Lucas CA, Mitra G (2001) Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. Quant Finance 1:1–13

Kellerer H, Maringer DG (2003) Optimization of cardinality constrained portfolios with a hybrid local search algorithm. OR Spectrum 25(4):481–495

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

Krink T, Paterlini S (2009) Multiobjective optimization using differential evolution for real-world portfolio optimization. Comput Manage Sci. doi:10.1007/s10287-009-0107-6

Laumanns M, Zitzler E, Thiele L (2000) A unified model for multi-objective evolutionary algorithms with elitism. In: Proceedings of the 2000 congress on evolutionary computation. IEEE Press, Piscataway, NJ, pp 46–53

Lin CC, Liu YT (2008) Genetic algorithms for portfolio selection problems with minimum transaction lots. Eur J Oper Res 185:393–404

Mansini R, Ogryczak W, Speranza GM (2003) LP solvable models for portfolio optimization: a classification and computational comparison. IMA J Manage Math 14:187–220

Markowitz HM (1952) Portfolio selection. J Finance 7:77–91

Markowitz HM (1990) Portfolio selection, efficient diversification of investments. Blackwell, Cambridge MA and Oxford UK

Mitra G, Kriakis T, Lucas C, Pirbhai M (2003) A review of portfolio planning: models and systems. In: Stachell SE, Scowcroft A (eds) Advances in portfolio construction and implementation. Butterworth-Heinemann, Oxford

Ong CS, Huang JJ, Tzeng GH (2005) A novel hybrid model for portfolio selection. Appl Math Comput 169:1195–1210

Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. J Risk 2(3):21–41

Schaerf A (2002) Local search techniques for constrained portfolio selection problems. Comput Econ 20(3):177–190

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

Subbu R, Bonissone P, Eklund N, Bollapragada S, Chalermkraivuth K (2005) Multiobjective financial portfolio design: A hybrid evolutionary approach. In: IEEE congress on evolutionary computation, vol 2, pp 1722–1729

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-103. Department of Electrical Engineering, Swiss Federal Institute of Technology, Zurich, Switzerland

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271

Zitzler E, Thiele L, Laumanns M, Fonseca CM, Grunertda Fonseca V (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7:117–132