Using Multiobjective Algorithms to Solve the Discrete Mean-Variance Portfolio Selection

K. P. Anagnostopoulos

Department of Production Engineering and Management, Democritus University of Thrace

Kimmeria Campus, Xanthi 671 00, Greece

Tel: 30-254-107-9676 E-mail: kanagn@civil.duth.gr

G. Mamanis

Department of Production Engineering and Management, Democritus University of Thrace

Kimmeria Campus, Xanthi 671 00, Greece

Tel: 30-232-106-5328 E-mail: gmamanis@pme.duth.gr

Abstract

In this paper we tackle the standard Markowitz mean-variance model extended to include complex constraints. We formulate the problem as a bi-objective mixed integer optimization problem, i.e. maximization of return and minimization of risk. To find the set of Pareto-optimal portfolios, we implement two multiobjective algorithms, a population based multiobjective optimizer and a multiobjective optimizer which uses a local search evolution strategy. Finally, we evaluate the performance of the two multiobjective evolutionary algorithms on a public benchmark data set and a data set constructed using a representative emerging market's index.

Keywords: Markowitz model, Multiobjective algorithm, Pareto front, Portfolio selection

1. Introduction

Portfolio selection consists in finding the best allocation of wealth among a finite number of assets. A fundamental answer to this problem was given by H. M. Markowitz who proposed the mean-variance model (Markowitz, 1952; Markowitz, 1990). Through this framework a portfolio is characterized by the reward, measured by the expected return, and the risk, measured by the variance of return. Since there are two criteria not a single but a set of optimal portfolios, the so-called Pareto-optimal or efficient portfolios, should be found. Most of the literature is concerned with minimization of the variance measure for given levels of return, in order to find the efficient front. Therefore, a quadratic optimization problem under linear constraints should be solved for which computationally effective algorithms exist.

Despite its wide use even today, Markowitz model has some limitations. A weakness is its high computational complexity and the problem of estimating the input data (expected returns, variances, covariances). Another point of criticism is on the validity of variance as a measure of risk. Variance is a symmetric measure of risk, i.e. penalizes negative as well as positive deviations from the mean, and is based on the controversial assumption that asset's returns are normally distributed (Benati and Rizzi, 2007; Konno and Yamazaki, 1991). Furthermore it rests on the assumption that investors' utility functions take on only the single argument of portfolio return (Steuer et al., 2005). Finally, practitioners find Markowitz model too simplistic as it neglects some important constraints that are faced in real portfolio decision making. (Mitra 2003). Introducing these constraints into the model the portfolio selection problem becomes a mixed integer quadratic optimization problem with a complex search space with multiple local extrema and discontinuities (Chang et al., 2000; Crama and Schyns, 2003; Gilli and Këllezi, 2002; Jobst et al., 2001).

Since heuristic techniques have been proved quite useful in many hard combinatorial problems, researchers took up this advantage to tackle difficult portfolio selection models. Mansini and Speranza (1999) have formulated the optimum portfolio choice with round lots as a mixed integer programming problem and they have proposed heuristics for its solution based upon the idea of constructing and solving mixed integer sub-problems. Chang et al. (2000) have extended the standard Markowitz model to deal with cardinality constraints as well as upper and lower bounds on the proportion of the portfolio invested in each asset. They show how under this type of constraints the efficient front may become discontinuous. For finding the cardinality constrained efficient front they have 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 evolution strategies (Kellerer and Maringer, 2003); a GRASP algorithm enhanced by a learning mechanism and a bias function (Anagnostopoulos et al., 2004); customized local search, simulated annealing, tabu search and genetic algorithm heuristics (Ehrgott et al., 2004); genetic algorithms to solve portfolio selection problems with minimum transaction lots (Lin and Liu, 2008).

However, all the above-mentioned methods transform the multiobjective optimization problem into a single objective one by using either a weighted sum of the objective functions or the ε-constraint method (usually by minimizing variance subject to return constraint). Computing the efficient front with these methods requires the problem to be solved several times for different values of return or weight parameters and this may be time consuming.

In this paper we confront the standard Markowitz model as a bi-objective optimization problem in order to find the efficient front in a single execution of the algorithm. Furthermore, we have enriched the model with class constraints as well as cardinality and quantity constraints. Class constraints, which have been proposed by Chang et al. (2000), but are not addressed in their paper, limit the proportion of the portfolio that can be invested in assets in each class. Such constraints represent the tendency of investors to limit the 'exposure' of their portfolio by investing in assets with a common characteristic, for example telecommunication stocks, health care stocks, computer software stocks, etc.

For solving the problem we propose an implementation of two well known Multiobjective Evolutionary Algorithms (MOEAs): the Pareto Archived Evolution Strategy (PAES) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II). Many multiobjective algorithms have been proposed in the literature. The reason for implementing these two algorithms is their different evolution philosophy: PAES is a multiobjective optimizer which uses a simple local search evolution strategy, while NSGA-II is a population based multiobjective optimizer.

The remainder of the paper has as follows. In Section 2, after a short review of the Markowitz model, the portfolio selection is defined as a multiobjective combinatorial problem. The proposed implementation of NSGA and PAES for solving the problem is presented in Section 3. Section 4 is devoted to numerical results, and some concluding remarks are presented in Section 5.

2. Problem formulation

2.1 The standard portfolio optimization problem

The problem of optimally selecting a portfolio among a finite number of assets was formulated by H.M. Markowitz in 1952. Markowitz work was based on the assumption that investors want maximum return and minimum risk from their portfolio. Considering asset's value as random variable, he defines return as the mean rate of return (per period) observed from historical data over a specific period of time, and risk as the variance or standard deviation of these rates. Since a portfolio is a collection of assets (thus a weighted sum of random variables) its value is also a random variable which can be described by expected return and variance of return.

Among all portfolios there are special ones for which it cannot be said that one is better than the other. All such portfolios that are Pareto-optimal (or non-dominated) offer the maximum level of return for a given level of risk, or equivalent, the minimum level of risk for a given level of return. Though bi-objective by definition, the standard portfolio selection problem is typically formulated as single-objective optimization problem as follows:

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \tag{1}$$

subject to

$$\sum_{i=1}^{N} x_i r_i = \rho \tag{2}$$

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

$$x_i \geq 0, \quad i = 1,\ldots,N \tag{4}$$

where

x_i : the decision variable which denotes the proportion held of asset *i*

r_i : the expected return of asset *i*

σ_{ij} : the covariance between assets *i* and *j*

ρ : the desired level of return for the portfolio

N : the number of assets available

The objective function (1) minimizes the total variance (risk) associated with the portfolio, equation (2) ensures that the portfolio has an expected return of ρ, and constraint (3) ensures that 100% of the budget is invested in the portfolio.

2.2 The constrained portfolio optimization problem

Simplifications to the basic problem have allowed the researchers to think for more realistic portfolio selection models. In real portfolio decision making, it is useful to avoid very small holdings, and to restrict the total number of assets. These requirements are modeled as threshold and cardinality constraints. Furthermore class constraints can be added to the basic model to limit the proportion of the portfolio invested in assets with common characteristics (bank assets, insurance assets, etc). In general, these constraints lead to sets of discrete variables, and the resulting mixed integer program becomes larger in size and computationally more complex than the standard mean-variance model.

Threshold and cardinality constraints can be added to the model using a binary variable z_i, which is equal to 1 if the asset i ($1 \leq i \leq N$) is held in the portfolio and 0 otherwise. Introducing finite upper and lower bounds ε_i, δ_i for the stock weight x_i, threshold constraints are represented by the following inequality:

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i, \quad i = 1,\ldots,N \tag{5}$$

To facilitate portfolio management or to control transaction costs, some investors may wish to limit the number of assets held in their portfolio. Cardinality constraint, which limits the portfolio to contain predetermined number of assets K, can be added to the model by counting the binary variables z_i. This constraint is expressed by the following equation:

$$\sum_{i=1}^{N} z_i = K \tag{6}$$

Moreover class constraints can be added by letting C_m, m=1, ... M, be M mutually exclusive sets of assets and L_m and U_m be the lower and upper proportion limit for class m. The constraints can be defined as:

$$L_m \leq \sum_{i \in C_m} x_i \leq U_m, \quad m = 1,\ldots,M,\ C_i \cap C_j = \varnothing \ \ for\, all\; i \neq j \tag{7}$$

In this paper we reformulate the quadratic optimization problem into a two-objective optimization problem. The problem to be solved is formulated as follows:

$$opt\ \boldsymbol{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})] = \left\{ \max \sum_{i=1}^{N} x_i r_i, \min \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j \sigma_{ij} \right\} \tag{8}$$

subject to

$$\sum_{i=1}^{N} x_i = 1 \tag{9}$$

$$\sum_{i=1}^{N} z_i = K \tag{10}$$

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i, \quad i = 1,\ldots,N \tag{11}$$

$$L_m \le \sum_{i \in C_m} x_i \le U_m, \quad m = 1,..M \tag{12}$$

$$z_i = 0-1 \tag{13}$$

$$x_i \ge 0, \quad i = 1,...,N \tag{14}$$

The objective function f_1 represents portfolio's return while objective function f_2 represents portfolio's variance of return. The N-vector **x** denotes the set of decision variables x_i.

Here we do not aim to minimize a single objective function like in the first problem but to optimize a vector objective function $\boldsymbol{f}$. We are trying to find those vectors **x** that are not dominated by any other in the feasible set. We say that a vector is non-dominated if its image in the objective space is non-dominated. For the specific problem we say that a solution $\mathbf{x}^1$ dominates the solution $\mathbf{x}^2$ iff:

$$[f_1(\mathbf{x}^1) \ge f_1(\mathbf{x}^2) \wedge f_2(\mathbf{x}^1) < f_2(\mathbf{x}^2)] \vee [f_1(\mathbf{x}^1) > f_1(\mathbf{x}^2) \wedge f_2(\mathbf{x}^1) \le f_2(\mathbf{x}^2)] \tag{15}$$

The Pareto-optimal front is constituted by all solutions that are not dominated by any other solution in the feasible set.

3. Multiobjective algorithms

Multiobjective Evolutionary Algorithms (MOEAs) have received great attention over the past two decades and several evolutionary-based multiobjective optimization techniques have been proposed. The primary reason for this development is their ability to find multiple Pareto-optimal solutions in a single run by using non-dominated ranking and selection to move a population towards the Pareto front, and specialized techniques to avoid convergence to a single point on the front. Based on this ground a number of independent Evolutionary Algorithm (EA) implementations have been proposed, for example the MultiObjective Genetic Algorithm (MOGA), the Niched-Pareto Genetic Algorithm (NPGA) and the Non-dominated Sorting Genetic Algorithm (NSGA) (Coello, 2000; 2006). A new generation of MOEAs started with the introduction of elitism (Zitzler and Thiele, 1999),. Most cited MOEAs of the second generation are the Strength Pareto Evolutionary Algorithm (SPEA, SPEA2) (Zitzler and Thiele, 1999), PAES, and NSGA II (Coello, 2006).

Recently, various attempts have been made for treating the portfolio selection as a multiobjective optimization problem: a MOEA is employed to solve the multiobjective quadratic programming problem (Ong et al., 2005); greedy search, simulated annealing and ant colony optimization (Armañanzas and Lozano, 2005); NSGA-II, PESA and SPEA2 for solving the standard portfolio optimization problem (Dioşan, 2005); MOEAs with local search for feasible solutions (Streichert et al., 2003); an hybrid multiobjective optimization approach combining evolutionary computation with linear programming (Subbu et al., 2005).

3.1 Implementation of PAES and NSGA II

The Pareto Archived Evolution Strategy (PAES) was proposed by Knowles and Corne (Knowles and Corne, 2003) in an attempt to adapt local search into multiobjective optimization problems. It uses a historical archive A to store non-dominated solutions, i.e., efficient portfolios, and a crowding procedure that recursively divides the objective space (specified by the return and variance of the portfolio) to maintain diversity. PAES consists of (1+1) evolution strategy where a single parent generates a single offspring by random mutation. In each iteration, the mutant solution is compared with each member in the archive for Pareto-dominance according the relation (15). If it is dominated by any member of the archive it is discarded from the process. On the other hand, if it dominates any archive members, these solutions are deleted and the mutant solution enters the archive. If neither dominates nor is dominated by any archive member, then it means that the candidate solution is non-dominated with the elements of the archive. In this case we distinguish two alternatives. If the archive is full, the decision is based on a crowding mechanism; otherwise the candidate solution is accepted as member of the archive.

The crowding procedure recursively divides the objective space occupied by non-dominated solutions into different grid regions. The number of divisions in each objective is given by the user and it is constant during the process while the grid is adapted to changes on the range in objective space of the current solutions in the archive so that it covers this range. For each solution its grid location is calculated and for each grid location the number of solutions resides there is computed. The mutant solution is accepted if increases the extent of the grid or its image in objective space is in less crowed region than any archive member. In Table 1 the PAES pseudocode is presented.

NSGA-II was introduced by Deb et al. (2002) as an improved version of NSGA. It uses a fast version of non-dominated sorting and selection to move the population along the Pareto-optimal front, and a crowding operator to guide the selection process toward a uniformly distributed Pareto front. The non-dominated sorting procedure classifies the individuals of each generation in layers based on Pareto non-domination. Firstly, the non-dominated individuals are identified so that to constitute the first non-dominated front and are assigned a specific rank. Next, these individuals are ignored temporarily and the second front of non-dominated individuals is identified. The process continues until all individuals in the population are classified. In Table 2 the NSGA-II pseudocode is presented.

The crowed-comparison operator is used to maintain the diversity of solutions in the population. Each individual is characterized by two attributes, that is its domination rank and its crowding distance. The crowding distance gives an estimate of the density of solutions surrounding a particular solution in the population. It is calculated separately in each layer. Initially the population is sorted according to each objective function value and the extreme solutions are assigned a large distance value so that they are always selected. The remaining solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions.

The crowding operator then selects individuals with the lowest rank. If two solutions have the same rank (i.e. they belong to the same non-dominated front) the procedure selects the solution that is located in a lesser crowed region.

After specifying the ranking procedure of individuals, the NSGA-II operates like a (μ+λ) evolution strategy using binary tournament selection, crossover and mutation operators in order to create an offspring population. Parent and offspring population are then shuffled together and the individuals are ranked according to the crowed-comparison operator. The best μ individuals form the next mating pool to be generated.

An important aspect when implement MOEAs to a specific problem is how to represent a solution and how to use the variation (recombination and mutation) operators in order to effectively explore the search space.

We choose both algorithms to have the same solution representation. For representing a solution we make use of three sets. The first set A consists of M real numbers associated with each class. The second set B, contains K integer numbers, each one representing an asset in the portfolio while set Γ includes K real numbers related with each asset.

$$\mathrm{A} = \{\alpha_1,..., \alpha_M\},\ 0 \le \alpha_i \le 1,\ i = 1,..M$$
$$\mathrm{B} = \{\beta_1,..., \beta_M, \beta_{M+1},..., \beta_K\},\quad \beta_i \in C_i,\ \beta_j \in \{1, N\} - \{\beta_i\},\ i = 1,..M\ and\ j = M+1,..., K$$
$$\Gamma = \{\gamma_{\beta_1},..., \gamma_{\beta_K}\},\ 0 \le \gamma_i \le 1,\ i \in B$$

In both algorithms the following procedure was implemented before every solution evaluation in order to find the real solution **x** (i.e. a portfolio) of the problem. In order to find the real proportion invested in each asset we apply the following procedure:

First we normalize the set A, so that to find the real class proportion (*rcp*) associated with each class *m*. We have:

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

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

Then the real proportion of each class is shared in the corresponding assets of set B. Thereafter, the proportion associated with each asset in the portfolio is calculated by the following equation:

$$x_{\beta_i} = \varepsilon_{\beta_i} + \frac{\gamma_{\beta_i}}{\sum_{\beta_i \in C_{class(\beta_i)}} \gamma_{\beta_i}}\left(rcp(class(\beta_i)) - \sum_{\beta_i \in C_{class(\beta_i)}} \varepsilon_{\beta_i}\right),\ i = 1,..K$$

$$x_i = 0\ \forall\, i \notin \mathrm{B}$$

where $class(\beta_i)$ returns the class that the asset β_i belongs.

In this way all the constraints are satisfied and there is no need for using penalty functions or repair mechanisms.

For generating the offspring population we use the uniform crossover operator where two selected individuals generates a single child. Regarding the first part (set 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 (sets B and Γ) we combine the assets from both parents in a new single set along with its values from set Γ in another set according these rules: If an asset is present in both parents it is copied only once and the associated value is chosen with equal probability from one or another parent. Next, we choose from the newly formed set M assets (with its weight value) each of one class with equal probability to ensure that every class has a "representative" in the new solution. The next K-M assets are chosen randomly from the remaining assets of both parents. Note that the crossover operator is used only by NSGA-II.

The same mutation operator was used in both algorithms (in PAES as the only generational force) and is described below.

$$\alpha_i^{mut} = p(\alpha_i + L_i) - L_i \;\; i = 1,..M$$
$$\gamma_{\beta_i}^{mut} = p(\gamma_{\beta_i} + \varepsilon_i) - \varepsilon_i \;\; i = 1,..K$$

where p is the is the step value to be chosen. If for any i the value γ becomes negative then it is set to zero and the corresponding asset changes with such a way so that all the constraints are satisfied. For example, if the corresponding asset is the only "representative" of a class then an asset from the same class is selected. If there is at least one asset in the portfolio belonging in this category then an asset from the entire set is selected except the ones that are currently in the portfolio. If for any i the value α become negative then it is simply set to zero.

4. Numerical results

We present here the numerical results obtained using a public benchmark data set that is available at the OR-Library retained by J.E. Beasley (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html) and a new one which was created using asset's prices from an emerging market's index. The numerical results from the OR-Library are referred to the German DAX 100 data set which contains 85 assets. As a representative emerging market's index the BOVESPA of Brazil was chosen. For estimating the required input parameters (expected returns, variances and covariances), daily returns for 55 securities were calculated using adjusted daily prices downloaded from the yahoo finance webpage for the period 2 January 2008 to 28 October 2009. Both algorithms have been implemented in Visual C++ and run on a personal computer Core 2 Duo at 2.1 GHz. For both indexes we have classified the assets into six categories (M = 6). For DAX 100 each class contains 15 assets except the last one which contains 10 assets, while for BOVESPA 9 assets are included in each class except one which contains 10 assets. The number of assets in the portfolio was fixed to be ten (K = 10) for both problems. The lower proportion allowed for investment in each class was 5%, thus $L_m = 0.05$ for each $m = 1,\ldots, 6$. We have not specified any upper limit here as the lower bound alone specifies an upper bound of 75% of investment in each class. For the stock weights x_i lower and upper bounds were 1% and 100% respectively, i.e., $\varepsilon_i = 0.01$, $\delta_i = 0.01$, $\forall i \in$ B.

For comparing the performance of MOEAs we use the following quantitative metrics: *Dist1*, *Dist2*, *Dist3* and the criterion *C* (Cardinality) (Czyzak and Jaszkiewicz 1998; Deb and Jain 2002). These metrics try to evaluate each desirable attribute of the Pareto-optimal set, i.e. the closeness to the true Pareto-optimal front, the uniformity and the cardinality of the generated set. Thus, evaluating the quality of approximation sets is also a multiobjective problem. For this reason several performance metrics have been suggested.

The fist two metrics *Dist1* and *Dist2* measures the closeness of the generated set to the reference set which is usually the true Pareto-optimal set. In our problem we do not know the exact Pareto-optimal front for the constrained problem, for this reason we use as a reference set the real Pareto front of the standard portfolio optimization problem. The two metrics that provide information about the average distance of all points in the reference set to the closest solution in the approximation set (*Dist1*) and the worst case (*Dist2*) are described by the following equations:

$$Dist1 = \frac{1}{|P|} \sum_{\mathbf{x}} \left\{ \min_{\mathbf{y}} c(\mathbf{x}, \mathbf{y}) \right\}$$

$$Dist2 = \max_{\mathbf{x}} \left\{ \min_{\mathbf{y}} c(\mathbf{x}, \mathbf{y}) \right\}$$

where |P| is the number of solutions generated by the algorithm, while the function $c(\mathbf{x}, \mathbf{y})$ measures the closeness of two solutions based on the achievement-scalarizing function:

$$c(\mathbf{x}, \mathbf{y}) = \max\{0, w_1(f_1(\mathbf{y}) - f_1(\mathbf{x})), -w_2(f_2(\mathbf{y}) - f_2(\mathbf{x}))\}$$

where $w_i = 1/\Delta_i$ and Δ_i is the difference between the minimal and maximal value of the objective function in the reference set. The solution obtained by an algorithm is described by **x**, while **y** represents the closest solution to **x** from the true Pareto-optimal set. The measure takes the value zero if on all objectives **x** reaches the value of solution **y**.

For measuring uniformity we use *Dist3* metric which is defined as the ratio *Dist2/Dist1*.

The criterion C measures the percentage of points generated by each algorithm in the combined Pareto-optimal front. To obtain the combined Pareto-optimal front, all non-dominated solutions (from both algorithms) are compared together with respect to the Pareto-dominance relation. For example, suppose that sets A and B have N_1 and N_2 elements in the combined Pareto-optimal set respectively. The ratios N_1/N_1+N_2 and N_2/N_1+N_2 are then used to indicate which algorithm is better in terms of solution quality.

Before the comparison both algorithms was tuned using the *Dist1* quality metric and DAX 100 data set. The best parameters found for both algorithms are shown in Table 3 for NSGA-II and in Table 4 for PAES.

Due to the randomness of MOEAs, the problem was solved 10 times for each algorithm. Table 5 shows the mean and variance for each quality metric and for each algorithm.

NSGA-II has better performance both in closeness to the reference set (*Dist1*, *Dist2*) and the uniformity of solutions (*Dist3*). Furthermore NSGA-II has more stable performance. The variance in ten runs is also small and smaller than that of PAES as shown in Tables 5 and 6.

Considering the C metric NSGA-II again outperforms PAES as it wins 8 out of 10 times for DAX 100 data set, with mean value equals 0.65, and best value 0.98.

In Figures 1-3 and 4-5 we provide the graphical representation for one of the ten runs of both algorithms together with the efficient front (UEF) that results from the standard portfolio optimization problem for DAX 100 and BOVESPA respectively.

The superiority of NSGA-II in both the convergence and uniformity is also shown by visual comparison, although PAES has also converged well especially in the middle and upper part of the efficient fronts. However, it is worth mentioning that PAES converges to the optimal set significantly faster than NSGA-II.

5. Conclusions

We have considered the classical mean-variance portfolio selection problem with additional practical constraints such as class constraints, as well as threshold and cardinality constraints. These constraints transform the standard Markowitz model in a mixed integer optimization problem and create discontinuities in the efficient front.

We have tried to solve the bi-objective version of Markowitz model rather than the single objective one which is the most commonly used, i.e. we have not specify any a priori preferences for objectives. We have applied two well known multiobjective evolutionary algorithms to solve the problem. The numerical results obtained using a public benchmark data set and a data set constructed using a representative emerging market's index shown that NSGA-II produces better results although PAES converges well. However, a further computational analysis is needed to strengthen these results. This analysis as well as the experimentation with other multiobjective algorithms is currently under investigation by the authors.

References

Anagnostopoulos, K.P., Chatzoglou, P.D. & Katsavounis, S. (2004). A reactive greedy randomized adaptive search procedure for a mixed integer portfolio optimization problem. In P.D. Chatzoglou (Ed.), *Proceedings of the 2nd International Conference on Accounting and Finance in Transition* (CD)

Armañanzas, R. & Lozano, J.A. (2005). A multiobjective approach to the portfolio optimization problem. In *IEEE Congress on Evolutionary Computation 2,* pp. 1388–1395

Benati, S. & Rizzi, R. (2007). A mixed integer linear programming formulation of the optimal mean/Value-at-Risk portfolio problem. *European Journal of Operational Research,* 176, 423–434

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

Coello, C.A.C. (2000). An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys,* 32(2), 109-143

Coello, C.A.C. (2006). Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine,* 1(1), 28-36

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

Czyzak, P. & Jaszkiewicz, A. (1998). Pareto simulated annealing-A metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis,* 7, 34-47

Deb, K. & Jain, S. (2002). *Running performance metrics for evolutionary multi-objective optimization*. Indian Institute of Technology, Kanpur, India

Deb, K., Ptratap, A., Agarwal, S. & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA II. *IEEE Transactions on Evolutionary Computation,* 6(2), 182-197

Dioşan, 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

Ehrgott, M., Klamroth, K. & Schwehm, C. (2004). An MCDM approach to portfolio optimization. *European Journal of Operational Research,* 155(3), 752–770

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

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, 1–13

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

Knowles, J. & Corne, D. (2003). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation,* 8(2), 149-172

Konno, H. & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science,* 37, 519-531

Lin, C.C. & Liu, Y.T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research 185,* pp. 393–404

Mansini, R. & Speranza, M.G. (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research 114(2),* pp. 219–233

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

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

Mitra, G. (2003). A review of portfolio planning: Models and systems. In S. Satchell, A. Scowcroft, *Advances in Portfolio Construction and Implementation, Butterworth-Heinemann, Amsterdam,* pp. 1-39

Ong, C.S., Huang, J.J. & Tzeng, G.H. (2005). A novel hybrid model for portfolio selection. *Applied Mathematics and Computation,* 169, 1195–1210

Schaerf, A. (2002). Local search techniques for constrained portfolio selection problems. *Computational Economics*, 20(3), 177-190

Steuer, R.E., Qi, Y. & Hirschberger, M. (2005). Multiple objectives in portfolio selection. *The Journal of Financial Decision Making,* 1, 11-26

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 2,* pp. 1722-1729

Zitzler, E. & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation,* 3(4), 257-271

Table 1. PAES pseudocode

```
create an initial solution s
add s to the archive A
while a termination criterion do not met
    mutate s to produce s'
    if (s' ≻ A)                  /* if s' dominates any archive member
        add s' to the archive A
        delete dominated solutions
        replace s with s'
    else if (s' ~ A)             /*if s' is non-dominated with the members of the archive
        apply test(s, s', A)     /* test determines which becomes the new current
                                     solution and whether to add s' to the archive A*/
end while
```

Table 2. NSGA-II pseudocode

```
create an initial population P of size N
generation ← 0
do while generation < maxgenerations
    evaluate P
    sort P according to Pareto non-domination
    assign crowding distance in each individual in P
    repeat
        choose the parents using binary tournament selection and the crowding operator
    recombine parents
    mutate offspring with probability p_m
    until N individuals are generated-population Q
    combine P and Q to form C
    sort C according to Pareto non-domination
    replace P with the best N solutions from C using the crowding operator
    generation ← generation + 1
end do
```

Table 3. Parameters for NSGA-II

Population size	1,000
Number of generations	1,000
Crossover probability	1
Mutation probability	0.1
Step value *p*	1.3 or 0.7 with equal probability

Table 4. Parameters for PAES

Archive size	1,000
Number of generations	50,000
Divisions	125
Mutation probability	1
Step value *p*	1.5 or 0.5 with equal probability

Table 5. Mean and variance for quality metrics and algorithms for DAX 100 data set

Metric	*Dist1*		*Dist2*		*Dist3*	
	Mean	Variance	Mean	Variance	Mean	Variance
NSGA II	0.031536	1.5×10^{-7}	0.10446	2.1×10^{-8}	3.314	0.00146
PAES	0.036	0.00017	0.10514	1.35×10^{-5}	3.912	5.25

Table 6. Mean and variance for quality metrics and algorithms for BOVESPA

Metric	*Dist1*		*Dist2*		*Dist3*	
	Mean	Variance	Mean	Variance	Mean	Variance
NSGA II	0.0164	3.1×10^{-7}	0.047	1.8×10^{-6}	2.86	0.0087
PAES	0.018	4.8×10^{-7}	0.049	1.1×10^{-5}	2.71	0.03

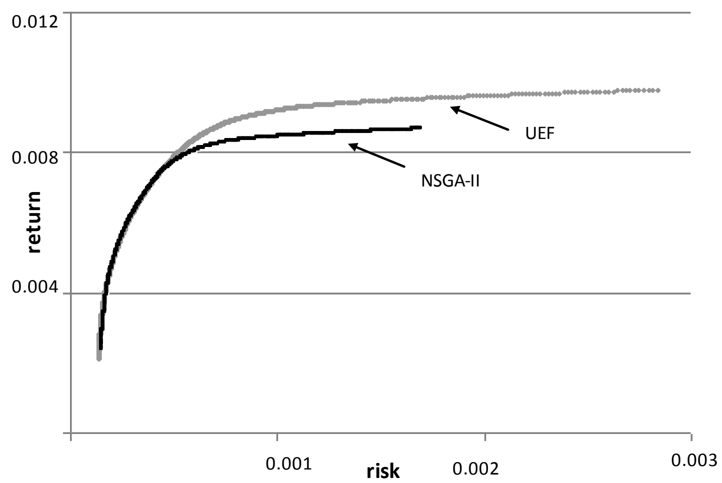

Figure 1. Standard efficient (UEF) and NSGA-II fronts (DAX 100)

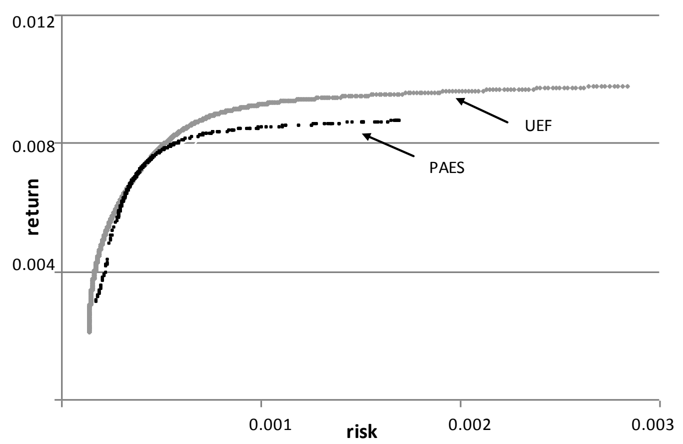

Figure 2. Standard efficient (UEF) and PAES fronts (DAX 100)

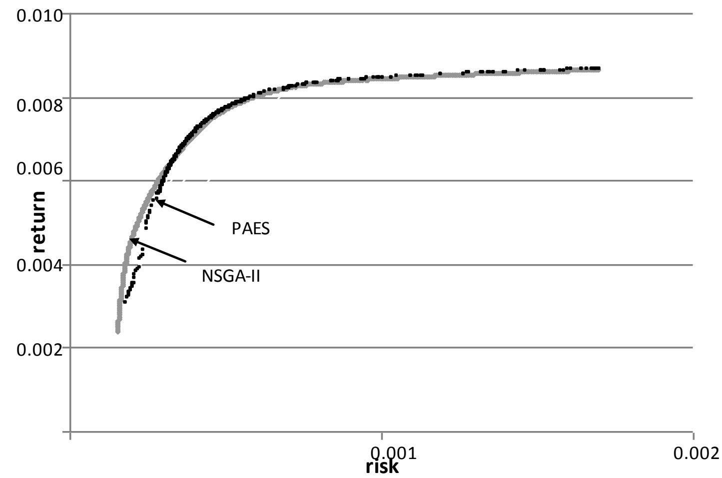

Figure 3. NSGA-II and PAES fronts (DAX 100)

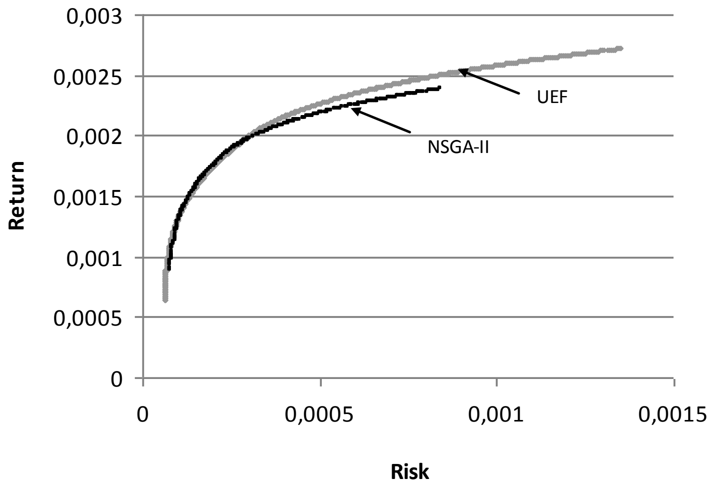

Figure 4. Standard efficient (UEF) and NSGA-II fronts (BOVESPA)

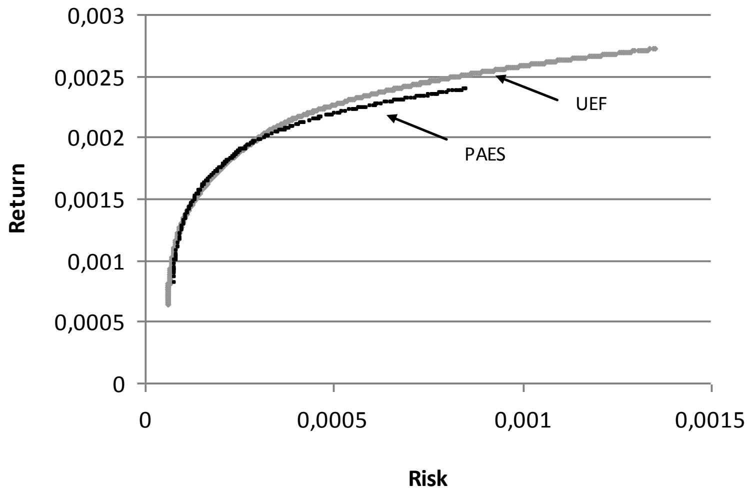

Figure 5. Standard efficient (UEF) and PAES fronts (BOVESPA)