

# Mean-VaR Portfolio Optimization: A Nonparametric Approach

Khin T. Lwin<sup>a,\*</sup>, Rong Qu<sup>b</sup>, Bart L. MacCarthy<sup>c</sup>

<sup>a</sup>*Anglia Ruskin IT Research Institute, Faculty of Science and Technology, Anglia Ruskin University, Chelmsford, Essex, CM11SQ, UK*

<sup>b</sup>*The Automated Scheduling, Optimisation and Planning (ASAP) Group, School of Computer Science, The University of Nottingham, Nottingham, NG8 1BB, UK*

<sup>c</sup>*Nottingham University Business School, The University of Nottingham, Nottingham, NG8 1BB, UK*

---

## Abstract

Portfolio optimization involves the optimal assignment of limited capital to different available financial assets to achieve a reasonable trade-off between profit and risk. We consider an alternative Markowitz's mean-variance model, in which the variance is replaced with an industry standard risk measure, Value-at-Risk (VaR), in order to better assess market risk exposure associated with financial and commodity asset price fluctuations. Realistic portfolio optimization in the mean-VaR framework is a challenging problem since optimizing VaR leads to a non-convex NP-hard problem which is computationally intractable. In this work, an efficient learning-guided hybrid multi-objective evolutionary algorithm (MODE-GL) is proposed to solve mean-VaR portfolio optimization problems with real-world constraints such as cardinality, quantity, pre-assignment, round-lot and class constraints. A learning-guided solution generation strategy is incorporated into the multi-objective optimization process to promote efficient convergence by guiding the evolutionary search towards promising regions of the search space. The proposed algorithm is compared against the Non-dominated Sorting Genetic Algorithm (NSGA-II) and the Strength Pareto Evolutionary Algorithm (SPEA2). Experimental results using historical daily financial market data from S &P 100 and S & P 500 indices are presented. Experimental results shows that MODE-GL outperforms two existing techniques for this important class of portfolio investment problems in terms of solution quality and computational time. The results highlight that the proposed algorithm is able to solve the complex portfolio optimization without simplifications while obtaining good solutions in reasonable time and has significant potential for use in practice.

---

\*Corresponding author. Address: Anglia Ruskin IT Research Institute, Faculty of Science and Technology, Anglia Ruskin University, Chelmsford, CM11SQ, UK

*Email addresses:* [khin.lwin@anglia.ac.uk](mailto:khin.lwin@anglia.ac.uk) (Khin T. Lwin), [rxq@cs.nott.ac.uk](mailto:rxq@cs.nott.ac.uk) (Rong Qu), [bart.maccarthy@nottingham.ac.uk](mailto:bart.maccarthy@nottingham.ac.uk) (Bart L. MacCarthy)

*Keywords:* Evolutionary computations, Multi-objective Constrained Portfolio Optimization, Value at Risk, Nonparametric Historical Simulation

---

## 1. Introduction

Portfolio optimization is concerned with the optimal allocation of limited capital to available financial assets to achieve a trade-off between reward and risk. The classical mean-variance (MV) model [53, 54] formulates the portfolio selection problem as a bi-criteria optimization problem with a tradeoff between minimum risk and maximum expected return. In the MV model, risk is defined by a dispersion parameter and it is assumed that returns are normally or elliptically distributed. However, the distributions of returns are asymmetric and usually have excess kurtosis in practice [6, 20, 28, 45, 58]. Variance as a risk measure has thus been widely criticized by practitioners due to its symmetrical measure which equally weights desirable positive returns against undesirable negative ones. In fact, Markowitz recognized the inefficiencies embedded in the mean-variance approach and suggested the semi-variance risk measure [54] in order to measure the variability of returns below the mean. In practice, many rational investors are more concerned with under-performance rather than over-performance in a portfolio.

These limitations have led to research directions where realistic risk measures are used to separate undesirable downside movements from desirable upside movements [14]. Among those various risk measures, Value-at-Risk (VaR) [55] and expected shortfall or conditional value-at-risk (CVAR) [61] are the widely accepted popular risk measures. Rockafellar and Uryasev [62] presented thorough discussions on VAR and CVaR with regular distributions. The choice between these two risk measures is based on many factors such as the differences in mathematical properties, stability of statistical estimation, simplicity of optimization procedures, and importantly, acceptance by financial industry and regulators [66]. Despite its acknowledged limitations [62], VaR has been widely adopted in the financial industry [8, 9, 10] but its non-linear and non-tractable properties [56] challenge its application. This paper presents an alternative bi-criterion Markowitz portfolio optimization model, in which the variance has been replaced with the VaR.

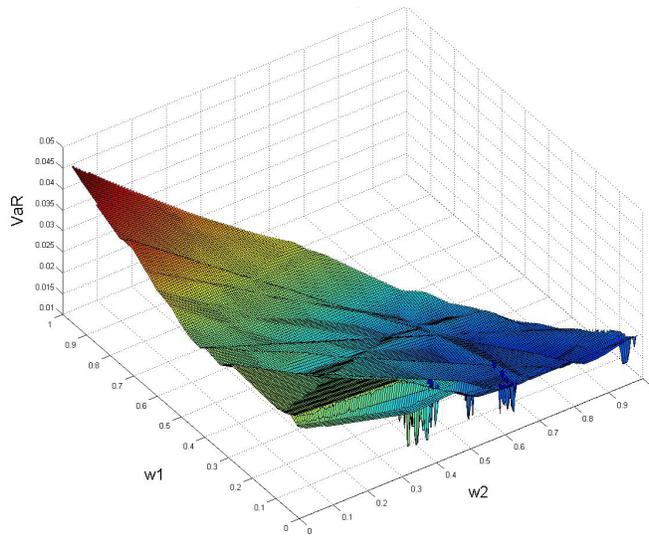
From a practical point of view, investors commonly face many real-world trading restrictions, which requires that the constructed portfolios have to meet trading constraints. As a result, several extensions and modifications of the basic Markowitz model reflecting real-world constraints have been developed. Since the additional trading constraints representing relevant practical issues lead to sets of discrete variables and constraints, the resulting optimization problem becomes very complex. In this work, we consider a single period multi-objective portfolio optimization problem with practical trading constraints, namely, cardinality, quantity, pre-assignment, round lot, class and class limit constraints.

Value-at-Risk (VaR) is used as a risk measure and a nonparametric historical simulation approach is adopted to calculate VaR.

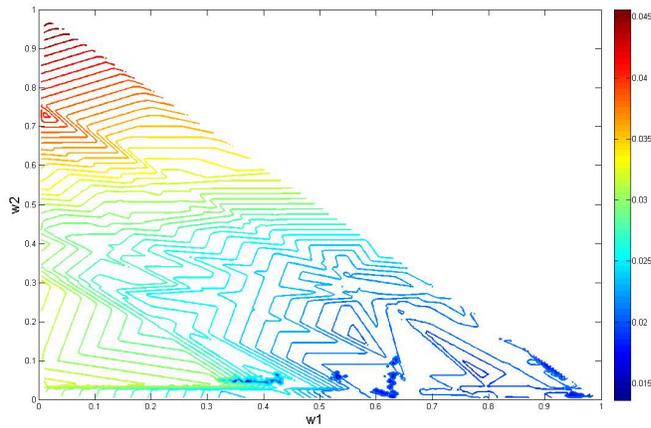
Minimizing a nonparametric VaR measure is a complex task due to the non-smooth objective function landscape with many local minima. Figure-1 shows the surface and contour plots of the Value-at-Risk (VaR) of feasible portfolios in a three assets universe, displaying the existence of non-smooth and non-convex surface with several local minima. In practice, portfolios are composed of markets with potentially hundreds to thousands of available assets, and the calculation of risk measures grows quickly in relation to the number of assets. When more dimensions and trading constraints are added to the problem, the complexity of the problem increases. Optimal exponential algorithms for reasonable problem dimensions are still not available. Approximation approaches such as smoothing [32] and meta-heuristics are the known alternatives to find optimal or near-optimal portfolios in a reasonable amount of time.

In this work, we present a new learning-guided multi-objective evolutionary algorithm (MODE-GL) for the mean-VaR portfolio optimization problems with practical investment constraints. A learning-guided solution generation strategy is incorporated into the multi-objective optimization process to promote efficient convergence by guiding the evolutionary search towards promising regions of the search space. The performance of the proposed algorithm is compared against two existing well-known multi-objective evolutionary algorithms, the Non-dominated Sorting Genetic Algorithm (NSGA-II) [26] and the Strength Pareto Evolutionary Algorithm (SPEA2) [81]. Moreover, the proposed learning-guided solution generation mechanism is incorporated in NSGA-II and SPEA2 to investigate its effectiveness. The performances of the algorithms are tested on two real datasets retrieved from the S & P 100 and S & P 500 indices. Experimental results show that the proposed algorithm is able to solve the complex portfolio optimization without simplifications while obtaining acceptable solutions in a reasonable time. Moreover, results show that the learning-guided solution generation strategy contributes to enhancing the efficient convergence of the search algorithm by concentrating on the promising areas of the search space.

The rest of the paper is organized as follows. Section 2 presents a brief introduction to the Value-at-Risk risk measure and a selective review of literature on mean-VaR portfolio optimization. Section 3 describes the generic multi-objective portfolio optimization, followed by the mean-VaR model considering cardinality, quantity, pre-assignment, round lot, class and class limit constraints. Section 4 introduces a new algorithm (MODE-GL) to address highly constrained portfolio optimization problems, highlighting the main differences from existing approaches and presenting the detailed structure of the algorithm. Section 5 applies the new approach to datasets generated from real market data and discusses the performance of the proposed algorithm in comparison to existing evolutionary optimization approaches. Finally conclusions



(a) Surface plot



(b) Contour plot

Figure 1: The historical VaR of feasible portfolios comprising of Coca-Cola Co., 3M Co. and Halliburton Co. with 3 years of data.  $w_1$  is the proportion of investment in Coca-Cola,  $w_2$  is the proportion of investment in Halliburton. The amount of investment in 3M is equal to  $1 - w_1 - w_2$ . Short selling is not allowed for any of the assets.

and future work are presented in Section 6.

## 2. Value-at-Risk: An Overview

In the literature, researchers and practitioners replace variance by introducing various downside risk measures [37, 46] in order to capture realistic market risk exposure by focusing on return dispersions below a specified target. The Safety-First principle introduced by Roy [64] is considered to be fundamental in the development of downside risk measures in the finance literature. Roy's Safety-First criterion suggests selecting a portfolio which minimizes probability of returns falling below some predefined disaster level. Consequently, a growing number of downside risk measures based on lower partial moments have been proposed by academics and practitioners [11, 12, 30].

The most popularly embraced technique for measuring downside risk among financial institutions is Value-at-Risk (VaR) [27, 29, 40, 47]. VaR measures the maximum likely loss of a portfolio from market risk with a given confidence level  $(1 - \alpha)$  over a fixed horizon. For instance, if a daily VaR is valued as 100,000 with 95% confidence level, this means that during the next trading day there is only a 5% chance that the loss will be greater than 100,000. The higher the confidence level, the better the chances that the actual loss will be within the VaR measure. Therefore, the confidence level  $(1 - \alpha)$  is usually high, typically 95% or 99%.

There are three main techniques commonly employed to measure VaR: the parametric approach (variance-covariance), nonparametric approach (historical simulation) and Monte Carlo simulation methods [40, 47]. The choice of the VaR method is critical since the results yielded from each method can be different from each other [51, 59]. Each method has its own strengths and weaknesses. The parametric method assumes financial returns follow a normal or known distribution function whereas the nonparametric (historical simulation) method makes no assumption regarding the distribution. The third method simulates several random scenarios, which can be computationally challenging. The analysis conducted by Perignon and Smith [57] shows that the most commonly used approach for computing VaR among investment firms that disclose their methodology is historical simulation.

Due to its conceptual simplicity, VaR has been widely recognized by financial regulators and investment practitioners. The Basel Committee for Banking Supervision of the Bank of International Settlements [8, 9, 10, 38] allows financial institutions to use VaR models to set aside regulatory capital amounts that would cover potential operational losses. In addition, the Securities and Exchange Commission (SEC) requires financial service firms to provide quantitative information about market risk using the VaR measure [1]. Cont et al [21] shows that nonparametric VaR has better robustness than CVaR, expected shortfall and Sharpe ratio. Rossello [63] also shows that nonparametric VaR is less sensitive to outliers than Sharpe ratio and Gain-Loss ratio.

Despite its wide use, VaR has limitations. When VaR is used as the objective function it leads to a non-convex and non-differential risk-return portfolio optimization problem where the number of local optima increases exponentially with the number of assets [23, 32, 44, 75]. In fact, Benati and Rizzi [13] show that optimization of the mean-VaR portfolio problem leads to a non-convex NP-hard problem which is computationally intractable. Moreover, the non-convexity of VaR discourages diversification. In other words, the VaR of a portfolio with two securities may be greater than the combination of VaRs of each security in the portfolio. Artzner et al [4] show that VaR fails to satisfy the subadditivity property for some distributions of asset returns. Many studies have applied alternative subadditive risk measures such as Conditional Value at Risk (CVaR) [61] and Partitioned Value-at-Risk (PVaR) [36] with corresponding operational consequences.

However, given that VaR is the predominantly used quantile-based, industry-standard risk measure, there is a need for efficient algorithms that minimize VaR while obtaining maximum return. In the literature, there are different approaches to measure VaR to investigate portfolio optimization [16, 33, 36, 56]. Although there has been considerable work related to portfolio optimization with various risk measures [7, 44, 46, 52], it is noticeable that the number of studies of non-parametric historical VaR in the context of mean-VaR remains relatively small. Gilli and Kellezi [34] and Gilli et al [35] propose a threshold accepting method to maximize a portfolio's return under VaR and expected shortfall constraints. Dallagnol et al [22] employ a genetic algorithm (GA) and particle swarm optimization (PSO) for a mean-VaR portfolio selection problem using historical simulation calculation. Alfaro-Cid et al [2] conduct a comparison between mean-variance and mean-VaR approach using a multi-objective genetic algorithm. However these models have often simplified the problem where practical constraints are not taken into account.

Baixauli-Soler et al [5] present a multi-objective GA for the mean-VaR portfolio optimization problem with minimum transaction units and transaction costs. Jevne et al [39] also study the mean-VaR portfolio optimization problem with minimum transaction units and transaction costs and investigate the effect of the initialization scheme on the results with multi-objective differential evolution and NSGA-II. Experimental results show that the refined initialization scheme improves the convergence of both algorithms. Anagnostopoulos and Mamanis replace the variance risk measure with VaR and expected shortfall (ES). Three multi-objective evolutionary algorithms are compared against exact methods to evaluate the portfolio selection problem with cardinality, quantity and class constraints [3].

### 3. Multi-objective Portfolio Optimization Problems

Multi-objective optimization generally involves balancing all conflicting objectives and searches for a set of compromise solutions between the objectives

while satisfying various constraints. In such contexts, this set of solutions is known as the set of Pareto-optimal solutions [25]. In the multi-criteria variant of the portfolio optimization problem, the objective is to find a set of *efficient* portfolios that minimize risk and maximize return simultaneously:

$$\begin{aligned}
\min \quad & f_1 = \psi(w) \\
\max \quad & f_2 = \mu(w) \\
\text{s.t.} \quad & w \in W.
\end{aligned} \tag{1}$$

In a two-dimensional space of risk and return, a solution  $a$  is said to be *efficient* (i.e., Pareto-optimal or non-dominated) if no solution  $b$  exists such that  $b$  dominates  $a$  [31]. Solution  $a$  is considered to dominate solution  $b$  if and only if  $C_1$  or  $C_2$  hold:

$$\begin{aligned}
C_1 : \quad & f_1(a) \leq f_1(b) \quad \wedge \quad f_2(a) > f_2(b) \\
C_2 : \quad & f_2(a) \geq f_2(b) \quad \wedge \quad f_1(a) < f_1(b)
\end{aligned} \tag{2}$$

The set of efficient portfolios forms the *efficient frontier* representing the best tradeoffs between return and the risk.

### 3.1. Mean-VaR Portfolio Optimization Problem

The mean-VaR model is based on Markowitz's MV model[53, 54]. It is obtained by replacing variance with VaR as a risk measure and is formulated as a multi-objective optimization problem where expected return is maximized and VaR is minimized. In this work, we compute VaR using a historical simulation (nonparametric) method. This method calculates VaR from the  $\alpha$  quantile of the empirical distribution of the historical data and therefore it does not assume that returns follow any particular distribution.

Let  $r_{it}$  be the observed return of asset  $i$  at time  $t$  using historical data over the time horizon  $T$ . Let  $w_i$  be the proportion of the budget invested in asset  $i$ . Given a set of  $N$  assets, the portfolio's return under scenario  $t$  is estimated by:

$$\kappa_t(w) = \sum_{i=1}^N r_{it}w_i, \quad t = 1, \dots, T. \tag{3}$$

Let  $\rho_t$  be the probability of scenario occurrence and assume all scenarios are considered to have equal probability:  $\rho_t = 1/T$ . The expected return of the portfolio over time  $T$  is obtained by:

$$\mu(w) = \sum_{t=1}^T \kappa_t(w) \rho_t \quad (4)$$

The VaR at a given confidence level  $(1- \alpha)$  is the maximum expected loss that the portfolio will not exceed with a probability  $\alpha$ .

$$\psi(w) = VaR_\alpha(w) = -inf \left\{ \kappa_t(w) \mid \sum_{t=1}^T \rho_t \geq \alpha \right\} \quad (5)$$

where returns  $\kappa_j(w)$  are placed in an ascending order such that  $\kappa_{(1)}(w) \leq \kappa_{(2)}(w) \leq \dots \leq \kappa_{(T)}(w)$  [3]. The mean-VaR portfolio selection problem is summarized as follows:

$$min \quad \psi(w) \quad (6)$$

$$max \quad \mu(w) \quad (7)$$

$$s.t \quad \sum_{i=1}^N w_i = 1, \quad 0 \leq w_i \leq 1 \quad (8)$$

where  $N$  is the number of available assets and  $w_i$  ( $0 \leq w_i \leq 1$ ) is the decision variable representing the proportion held of asset  $i$ . Eq(8) defines the *budget constraint* (all the money available should be invested) for a feasible portfolio and the non-negative property of  $w_i$  denotes that no short sales are allowed.

### 3.2. Practical Constraints

The basic model assumes a perfect market where securities are traded in any (non-negative) fractions, there is no limitation on the number of assets in the portfolio, investors have no preference over assets and they do not care about different asset types in their portfolios [48, 54]. In practice, however, these assumptions are unrealistic. As a result, several extensions and modifications have been proposed to address the real-world constraints [49, 71, 74]. The basic model can be extended (not exhaustively) with a number of real-world constraints to better reflect practical portfolio optimization:

#### Cardinality Constraint

Cardinality constraints limit the number of assets ( $K$ ) that compose the portfolio. Very often in practice, investors prefer to have a limited number of assets in their portfolio since the management of many assets in a

portfolio is tedious and may be hard to monitor. They may also seek to reduce transaction costs and/or to assure a certain degree of diversification by limiting the maximum number of assets in their portfolios.

### **Floor and Ceiling Constraints**

The floor and ceiling constraints specify the minimum and maximum limits on the proportion of each asset that can be held in a portfolio. In practice, investors prefer to avoid excessive administrative costs for very small holding of assets in the portfolio and/or some institutional policies may stipulate lower and upper bounds of some or each asset in the portfolio. These are also referred to as bounding or quantity constraints.

### **Pre-assignment Constraint**

The pre-assignment constraint is usually used to model the investor's subjective preferences. An investor may intuitively prefer a specific set of securities ( $Z$ ) to be included in the portfolio, with either its proportion fixed or to be determined.

### **Round Lot Constraints**

Round lot constraints require the amount of any assets in the portfolio to be in exact multiples of defined normal trading lots. In practice, several market securities are traded as multiples of minimum lots.

### **Class Constraints**

In practice, investors may ideally want to partition the available assets into mutually exclusive sets (classes). Each set may be grouped with common features or types such as health care assets, energy assets, etc or grouped by an investor's own intuition or preference. Investors may prefer to select at least one asset from each class to construct a diversified and safe portfolio. Class constraints were first introduced by Chang et al [15] and Anagnostopoulos and Mamanis [3] later considered class constraints in their work.

### **Class Limit Constraints**

In practice, investors may also want to restrict the total proportion invested in each class to a defined class limit.

### *3.3. Practical Mean-VaR Portfolio Optimization Problem*

The constraints described in Section-3.2 are *hard* in the sense that they have to be satisfied at any time. The extended mean-VaR model may then be formulated as follows:

$$\min \quad \psi(w) \quad (9)$$

$$\max \quad \mu(w) \quad (10)$$

$$\text{subject to } \sum_{i=1}^N w_i = 1, \quad 0 \leq w_i \leq 1 \quad (11)$$

$$\sum_{i=1}^N s_i = K, \quad (12)$$

$$w_i = y_i \cdot v_i, \quad i = 1, \dots, N, \quad y_i \in \mathbb{Z}_+ \quad (13)$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, \quad 0 \leq \epsilon_i \leq \delta_i \leq 1, \quad i = 1, \dots, N \quad (14)$$

$$L_m \leq \sum_{s_i \in C_m} w_i \leq U_m, \quad m = 1, \dots, M, \quad (15)$$

$$s_i \geq z_i, \quad i = 1, \dots, N \quad (16)$$

$$s_i, z_i \in \{0, 1\}, \quad \{z_i \in Z \mid z_i = 1\}, \quad i = 1, \dots, N \quad (17)$$

The additional constraints are described through Eqs (12 - 17). Eq-(12) defines the cardinality constraint where  $K$  is the number of invested assets in the portfolio and the binary variable  $s_i$  denotes whether asset  $i$  is invested or not. Eq-(14) defines the quantity constraint. If asset  $i$  is invested, the proportion of capital  $w_i$  lies in  $[\epsilon_i, \delta_i]$ .

Eq-(16) defines the pre-assignment constraint to fulfil the investor's subjective requirements where the binary vector  $z_i$  denotes if asset  $i$  is in the pre-assigned set  $Z$  that has to be included in the portfolio or not. Eq-(13) defines the round lot constraint where  $y_i$  is a positive integer variable and  $v_i$  is the minimum lot that can be purchased for each asset. Eq-(15) defines class and class limit constraints where  $C_m, m = 1, \dots, M$ , represents  $M$  mutually exclusive sets of assets and  $L_m$  and  $U_m$  are the lower and upper proportion limit for class  $m$ . In this work, it is assumed that  $L_m > 0$  for every class  $C_m$  and  $K \geq M + |Z| - |Z^{cl}|$  where  $Z^{cl}$  is the set of classes represented by the

preassigned set  $Z$  (i.e.,  $Z^{cl} = \{m : z_i \in C_m\}$ ,  $z_i \in Z$ ).

#### 4. A Learning-guided Multi-objective Evolutionary Algorithm

The portfolio optimization problem becomes too complex to solve by conventional optimization approaches as it typically exhibits multiple local extrema and discontinuities especially when variance is replaced by the VaR risk measure (see Figure-1) and constraints reflecting investor’s preferences and/or institutional trading rules are considered. Meta-heuristics and hybrid algorithms provide alternatives for finding optimal or near-optimal solutions in a reasonable amount of time. Over the last two decades, multi-objective evolutionary algorithms (MOEAs) have received a significant amount of attention and demonstrated their effectiveness and efficiency in approximating the Pareto-optimal front [17] for a wide range of problems [50, 78].

DEMO [60] is one of the recent algorithms which combines the advantages of Differential Evolution (DE) [69] with the mechanisms of Pareto-based sorting and crowding distance sorting [26]. It has been successfully tested on the carefully designed test functions (ZDT) introduced in [80]. The procedure of the DEMO is described in Algorithm 1. DEMO maintains a population of individuals, where each represents a potential solution to the optimization problem. During the evolution, it allows its population capacity to expand in order to add newly found non-dominated solutions (see Algorithm 1, line 3-11). Hence, it enables the newly found non-dominated solutions to immediately take part in the generation of the subsequent candidate solutions. This feature of DEMO promotes fast convergence towards the true Pareto front.

In each generation, if the population exceeds the size limit, it is sorted based on the non-domination and crowding distance metrics in order to identify those individuals to be truncated. The crowding distance value of a solution is an estimate of the density of solutions surrounding that solution [26]. It is estimated by calculating the average distance of two nearest neighbors on either side of a particular solution along each of the objectives. If solutions are non-dominated then the one with larger crowding distance is preferred. It thus aims to maintain a good distribution of non-dominated portfolios.

In this work, we propose a learning-guided multi-objective evolutionary algorithm (MODE-GL) for constrained portfolio optimization. The proposed algorithm adopts a new approach to extend the generic DEMO scheme to solve the constrained portfolio optimization problem. The main differences of our approach with respect to the DEMO scheme are:

1. A secondary population (i.e. an external archive) is introduced to store the well spread non-dominated solutions found throughout the evolution (see Section-4.1.8).

---

**Algorithm 1:** Procedure of MODE [60]

---

```
1 evaluate the initial population  $P$  of random individuals ;
2 while stopping criterion not met do
3   for each individual  $p_i (i = 1, \dots, P_{size})$  do
4     create a candidate  $p'$  from parent  $p_i$  ;
5     evaluate  $p'$  ;
6     if  $p'$  dominates  $p_i$  then
7       |  $p'$  replaces  $p_i$ 
8     else if  $p_i$  dominates  $p'$  then
9       | discard  $p'$ 
10    else
11      | add  $p'$  to  $P$ 
12  if  $|P| \geq P_{size}$  then
13    | truncate  $P$ 
14  randomly enumerate the individuals in  $P$ 
```

---

2. A learning mechanism is proposed to extract the important features from the efficient solutions found throughout the evolution (see Section-4.1.3).
3. Two extended variants of differential evolution mutation schemes are proposed (see Section-4.1.4).
4. An efficient solution generation scheme that utilizes the learning mechanism, problem specific heuristics and effective differential evolution mutation schemes is proposed to guide the search towards the promising regions of the search space (see Section-4.1.4).

The proposed MODE-GL extracts the important features of non-dominated solutions throughout the evolution. Incorporating a learning mechanism and prior problem-specific knowledge exploitation in the evolution process allows MODE-GL to generate promising offspring solutions. The approach thus aims to promote convergence by concentrating on the promising regions of the search space. On the other hand, adopting the two extended variants of differential evolution mutation schemes promotes the exploration of the search towards the least crowded region of the solution space.

The pseudocode of the proposed algorithm is described in Algorithm 2.

---

**Algorithm 2:** Procedure of MODE-GL

---

**Input:**  
     $A_{max}$  = the size of the archive  $A$  ;  
     $D_{max}$  = the size of the archive  $D$  ;  
     $NP$  = the number of individuals in the population  $P$  ;

1 randomly create an initial population  $P$  ;                   // Initialization  
2 **while** *stopping criterion not met* **do**  
3     maintain the archive  $A$  with non-dominated solutions from  $P$  ;  
4     **if**  $|A| \geq A_{max}$  **then**  
5         maintain archive  $A$  with  $A_{max}$  least crowded non-dominated  
           solutions ;  
6     maintain archive  $D$  with  $D_{max}$  least crowded solutions from  $P$  ;  
7     learn from the archive  $A$  to identify the promising asset(s) ;  
8     **for** *each individual*  $p_i (i = 1, \dots, N) \in P$  **do**  
9         create new candidate  $p'$  ;                                 // see Algorithm 3  
10        **if** *constraints are violated* **then**  
11            repair  $p'$   
12            evaluate  $p'$  by  $f_1$  and  $f_2$  ;                                 // see Eq 4, 5  
13            **if**  $p'$  *dominates*  $p_i$  **then**  
14                 $p'$  replaces  $p_i$  ;  
15            **else if**  $p_i$  *dominates*  $p'$  **then**  
16                discard  $p'$   
17            **else**  
18                add  $p'$  to the current population  $P$   
19     **if**  $|P| \geq NP$  **then**  
20         maintain  $P$  with the best  $NP$  solutions, ranked by  
           non-domination and crowding distance metrics ;  
21     randomly enumerate the individuals in  $P$

**Output:** archive  $A$

---

#### 4.1. The proposed MODE-GL

##### 4.1.1. Solution Representation and Encoding

Assume an array  $\Gamma$  consists of  $M$  real values each one representing the total proportion invested in each class; an array  $S$  contains  $K$  integer numbers, each representing a selected asset in the portfolio; and an array  $W$  includes  $K$  real values, representing the allocation of each selected asset in the portfolio. We present the following representation scheme to handle several considered

constraints:

$$\begin{aligned}
\Gamma &= \{\theta_1, \dots, \theta_M\}, \quad 0 \leq \theta_m \leq 1, \quad m = 1, \dots, M \\
S &= \{s_1, \dots, s_M, \dots, s_Q, s_{Q+1}, \dots, s_K\}, \\
Q &= M + |Z| - |Z^{cl}|, \\
s_b &\in C + Z, \quad s_j \in \{1, \dots, N\} - \{s_1, \dots, s_Q\} \\
b &= 1, \dots, Q, \quad j = Q + 1, \dots, K \\
W &= \{w_i, \dots, w_K\}, \quad 0 \leq w_i \leq 1, \quad i = 1, \dots, K.
\end{aligned}$$

With this solution representation, the cardinality constraint is satisfied by the array  $S$  which has specified size  $K$ . The preassignment constraint is satisfied by including all preassigned assets in  $S$ . The class represented by the preassigned assets in  $Z$  are identified and denoted by  $Z^{cl}$ . We then ensure that  $M - |Z^{cl}|$  assets are selected from each remaining class. As stated in Section-3.3, in this work, it is assumed that  $K \geq M + |Z| - |Z^{cl}|$ . If  $K > Q$ , then the remaining  $K - Q$  assets are randomly selected from the available unselected assets. In the literature, Anagnostopoulos and Mamanis [3] use a similar encoding scheme but their model does not consider either pre-assignment or round lot constraints.

To understand the way the problem is structured, consider an illustrative portfolio problem where  $N = 94$ ,  $M = 6$ ,  $C_1 \in \{1, \dots, 15\}$ ,  $C_2 \in \{16, \dots, 30\}$ ,  $C_3 \in \{31, \dots, 45\}$ ,  $C_4 \in \{46, \dots, 60\}$ ,  $C_5 \in \{61, \dots, 75\}$  and  $C_6 \in \{76, \dots, 94\}$ . An example portfolio with  $K = 10$  would be represented as described below:

$$\begin{aligned}
Z &= \{30\}, \quad Z^{cl} = \{C_2\}, \quad |Z^{cl}| = 1, \\
\Gamma &= \{0.05, 0.05, 0.05, 0.05, 0.05, 0.05\}, \\
S &= \{8, 17, 30, 47, 62, 85, 31, 92, 37, 69\}, \\
W &= \{w_8 = 0.112, w_{17} = 0.048, w_{30} = 0.024, w_{47} = 0.376, w_{62} = 0.024, \\
&\quad w_{85} = 0.136, w_{31} = 0.12, w_{92} = 0.064, w_{37} = 0.064, w_{69} = 0.032\}.
\end{aligned}$$

#### 4.1.2. Initial Population Generation

To generate an initial population, all assets in the pre-assignment set  $Z$  are included first and the remaining  $K - |Z|$  are randomly selected by making sure at least one asset from each class of  $M$  is included. The proportions (with exact lots) are assigned to those  $K$  selected assets randomly. If the generated portfolio violates the budget, quantity and/or class limit constraints, such a solution is corrected by the constraint handling techniques detailed in Section-4.1.5. This ensures that all generated solutions in the population are feasible.

#### 4.1.3. Learning Mechanism

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

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

The new solutions to be generated are encouraged to compose with those assets by exploiting the knowledge obtained throughout the evolution to direct the search towards the most promising regions of the search space. The larger the concentration scores the higher its chances are to be included in the new generated solution (See Section-4.1.4). The proposed learning mechanism is computationally cheap as it only uses a single update at each generation.

#### 4.1.4. Candidate Generation

One of the factors to consider in designing the portfolio model in MODE-GL is to find an effective way to generate offspring. In this section, an effective and efficient candidate generation scheme with a good balance between exploitation and exploration is proposed. A new solution is generated in two phases: the selection of assets from a universe of  $N$  available assets and the allocation of capital to those selected assets. In the first phase, learning mechanism (see Section-4.1.3) together with problem specific heuristics (**S1** - **S4**) described below are exploited to identify promising assets while directing the search towards the most promising regions of the search space.

In order to generate a new candidate solution, the  $|Z|$  pre-assigned assets are first selected. By taking into account the above stated intuitive learning, in this work, MODE-GL then randomly uses the following selection schemes until the remaining assets ( $K - |Z|$ ) have been selected, while making sure at least one asset from each class of  $M$  is included. By adopting the selection schemes stated below, it is ensured that the new candidate solution satisfies the pre-assignment, class and cardinality constraints.

- S1:** Roulette wheel selection based on the concentration score  $c_i$ .
- S2:** Select asset with the highest concentration score  $c_i$ .
- S3:** Select asset with the highest mean return values.
- S4:** Select asset with the least standard deviation of return values.

In the second phase, the proportions of those selected assets for the new candidate solution are assigned by using two extended variants of DE mutation schemes as follows:

$$\mathbf{W1:} \quad w'_i := best_i + r[0, 1] \times (w1_i - w2_i)$$

$$\mathbf{W2:} \quad w'_i := w_i + F \times (best_i - w_i) + F \times (w1_i - w2_i)$$

where  $F$  is the scaling factor for differential evolution. The two portfolios with  $w1_i$  and  $w2_i$  allocations are randomly selected from the least crowded portfolio archive  $D$  and  $best$  is the best solution randomly selected from the best 10% of archive  $A$ . These two DE mutation schemes are extended from similar variants of *DE/best/1* [24] and *DE/current-to-pbest/1* [77]. In our extended version the difference is that  $w1_i$  and  $w2_i$  are randomly selected portfolios from archive  $D$  to direct the search towards promising unexplored directions. The detailed procedure of the candidate generation is provided in Algorithm-3. The proposed candidate generation mechanism intends to guide the search towards promising directions by learning from the best found solutions from the archive  $A$ . In this way, the proposed algorithm converges efficiently. The new candidate portfolio is repaired if the quantity and round lot constraints are violated (see the repair mechanism in Section-4.1.5).

---

**Algorithm 3:** Procedure to Generate a Candidate Solution.

---

**Input:** concentration scores of assets  $c_i (i = 1, \dots, N)$  and  $\bar{p} \in P$

- 1 select  $|Z|$  assets from preassigned set  $Z$  and randomly select  $K - |Z|$  assets by **S1**, **S2**, **S3** and/or **S4** while ensuring class constraint;
- 2 randomly select three *different* portfolios:  $p1, p2, p3 \in \{P \setminus \bar{p}\}$ ;
- 3 randomly select an index  $i$  from those  $K$  assets and assign  $i$  to  $j$  and  $\gamma$ ;
- 4 **for** each selected asset **do**
- 5     **if**  $r(0, 1) < CR$  or  $j == \gamma$  **then**
- 6         | allocate weight  $w'$  by **W1** or **W2**;
- 7     **else**
- 8         | assign weight  $w'$  with corresponding  $\bar{w}$  of parent portfolio  $\bar{p}$ ;
- 9     | randomly select an index  $i$  from those  $K$  selected and assign  $i$  to  $j$ ;

**Output:** candidate solution  $p'$

---

#### 4.1.5. Constraint Handling

When using an evolutionary algorithm to solve constrained optimization problems, various methods have been proposed in the literature for handling constraints, such as penalty function methods, special representation and operator methods, repair methods, separation of objective and constraint methods, and hybrid methods [19]. Among those methods, the repair method is one of the commonly adopted approaches to locate feasible solutions for combinatorial optimization problems [19, 65].

During the population sampling, each constructed individual portfolio is

repaired if it does not satisfy all considered constraints. As described in Section-4.1.4, the new solution generated by MODE-GL already satisfies the cardinality, class and pre-assignment constraints. Hence, the following repair mechanism stated in [68, 70] is applied:

1. All weights of the selected asset in the candidate solution are adjusted by setting:

$$w'_i = \psi_i + \frac{w'_i - \psi_i}{\sum (w'_i - \psi_i)}$$

where the smallest divisible lot  $\psi_i = \text{inf} \{x_i : x_i \% v_i == 0 \text{ and } x_i \geq \epsilon_i\}$ .

2. The weights are then adjusted to the nearest round lot level by setting  $w'_i = w'_i - (w'_i \text{ mod } v_i)$ . The remaining amount of capital is redistributed in such a way that the largest amount of  $(w'_i \text{ mod } v_i)$  is added in multiples of  $v_i$  until all the capital is spent.
3. The weights are then adjusted (if the class limit constraints are violated). If  $\theta_m < L_m$ , insert  $y := L_m - \theta_m$  to the underflow class and subtract  $v_i$  from those classes where  $L_{m'} + v_i \leq \theta_{m'} \leq U_{m'}$  until  $\sum v_i \geq y$ . Similarly, the same for the overflow class. This process is repeated until all limits are satisfied.

#### 4.1.6. Selection Scheme

MODE-GL applies an elitist selection scheme based on Pareto optimality (see Algorithm 2, line 13-18). During the evolution, the population is extended by adding the newly found non-dominated solutions. Hence, at each generation, the number of portfolios in the current population will be between  $NP$  and  $2NP$ .

#### 4.1.7. Truncate Population

In each generation, if the number of portfolios in the current population exceeds its limit  $NP$ , the individuals in the population are sorted based on the non-dominance and crowding distance metrics [26]. The current population is then truncated by keeping the best  $NP$  individuals for the next generation.

#### 4.1.8. Maintaining Archives

The main objective of the external archive  $A$  is to maintain the well-spread non-dominated solutions encountered during the search. In each generation, archive  $A$  is updated with the non-dominated solutions from the trial population. The computational time to maintain the archive increases with the archive size [18, 43, 81]. The size of the archive is therefore restricted to a pre-specified value. When the external archive has reached its maximum capacity  $A_{max}$ , the most crowded non-dominated members are identified and discarded.

In addition, in each generation, a small number of the least crowded solutions are maintained in archive  $D$  and they are not required to be efficient. As noted, mean-VaR objective function landscapes are inclined to have many local minima (see Figure-1) and therefore the search needs to cover sufficient solution space to maximize the probability of discovering the global optimum. The least crowded solutions from archive  $D$  are exploited to promote the exploration of the search towards the least explored region of the solution space in order to achieve well-spread efficient solutions.

## 5. Performance Evaluation

In this section, we first introduce publicly available real datasets and performance metrics used for evaluating multi-objective evolutionary algorithms. The performance of MODE-GL is compared with four other evolutionary algorithms using commonly adopted performance evaluation metrics.

### 5.1. Datasets

For the empirical part of this work, two datasets based on historical daily financial market data have been retrieved from the Yahoo! Finance [76]. It was observed that historical time series downloaded from this site had some missing data points and hence those assets with missing data points were discarded. The first dataset (DS1) consists of 94 securities from the S & P 100 and covers daily financial time series data over a period of three years from 01/03/2005 to 20/02/2008, totalling 750 trading days. The second dataset (DS2) is composed of 475 securities from the S & P 500 and covers daily financial time series data over a period of one year from 11/04/2013 to 04/04/2014, totalling 250 trading days. The datasets are available to access online at <http://www.cs.nott.ac.uk/~kt1> These datasets have been used for portfolio optimization with cardinality, quantity, pre-assignment, round lot, class and class limit constraints in order to study the performance of the evolutionary algorithms considered in this work. All considered algorithms have been implemented in C# and run on a personal computer Intel(R) Core(TM)2 Duo CPU E8400 3.16GHz. The experimental results obtained for each algorithm are the average of 30 runs.

### 5.2. Quality Indicators

To evaluate the performance of the multi-objective evolutionary algorithms from various aspects, several performance metrics have been proposed in the literature [41, 72, 83], mainly defined by how close the obtained solutions are to the Pareto front and how evenly the solutions are distributed along the frontier [79, 80]. In this study we use two widely adopted performance evaluation metrics, namely the inverted generational distance and hypervolume metric.

### 5.2.1. Inverted Generational Distance (IGD)

The inverted generational distance [67] uses the true Pareto front as a reference and measures the distance of each of its elements from the true Pareto front to the non-dominated front obtained by an algorithm. It is defined as:

$$IGD = \frac{\sqrt{\sum_{i=1}^Q d_i^2}}{Q} \quad (18)$$

where  $Q$  is the number of solutions in the true Pareto front and  $d_i$  is the Euclidean distance between each of the solution and the nearest member from the set of non-dominated solutions found by the algorithm. This metric measures both the diversity and the convergence of an obtained non-dominated solution set. The smaller the value of this metric, the closer the obtained front is to the true Pareto front. A value of IGD equal to zero indicates that all obtained solutions lie on the true Pareto front and have the best possible spread. Figure 2, for example, shows that  $Q^* = 5$ ,  $d_1 = \sqrt{(1.5 - 2.5)^2 + (10 - 9)^2}$ ,  $d_2 = \sqrt{(2 - 2.5)^2 + (8 - 9)^2}$ ,  $d_3 = \sqrt{(3 - 3)^2 + (6 - 6)^2}$ ,  $d_4 = \sqrt{(4 - 5)^2 + (4 - 4)^2}$ ,  $d_5 = \sqrt{(6 - 5)^2 + (2 - 4)^2}$  and  $IGD = 0.6$ .

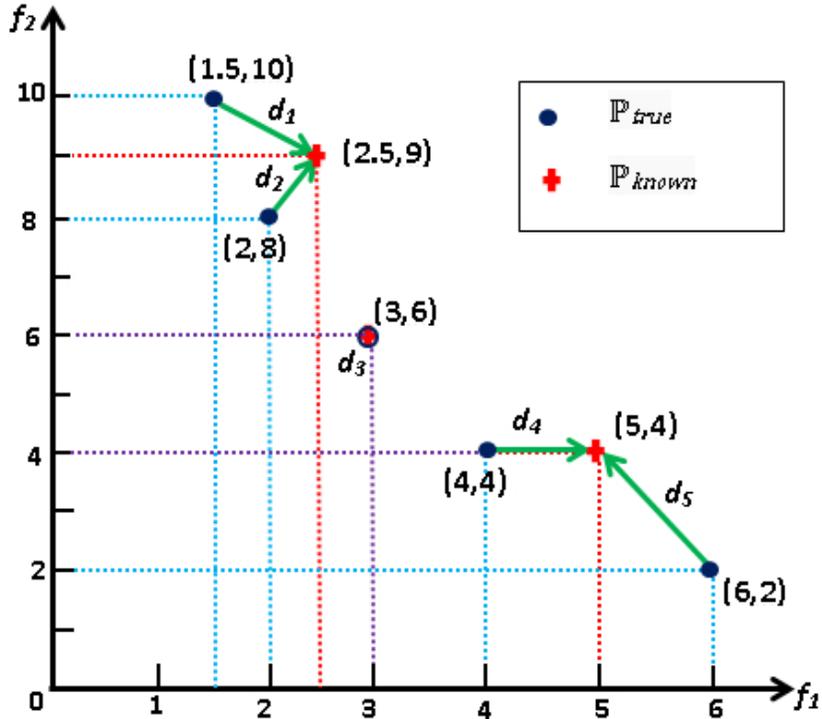


Figure 2: Example illustration of the inverted generational distance (IGD) metric.

The true Pareto front for this new highly constrained multi-objective portfolio optimization problem considered in this work is unknown. In this study, we use the best known efficient frontier obtained from all considered algorithms as the estimated Pareto front reference set.

### 5.2.2. The Hypervolume (HV) Metric

Hypervolume metric [82], also known as S-metric or Lebesgue measure, is widely recognized as a unary value which is able to measure both closeness of the solutions to the optimal set and diversity of the obtained solutions. The hypervolume metric calculates the volume of the objective space covered by members of an obtained Pareto set  $\mathbb{P}_{known}$  bounded by a reference point  $r$ . The reference point  $r$  is found by constructing a vector of worst objective function values.

Let  $\hat{Q}$  be the set of non-dominated solutions obtained by an algorithm. For each solution  $p \in \hat{Q}$ , a hypercube  $v_p$  from solution  $p$  and the reference point  $r$  is measured. The hypervolume (HV) value is calculated by summing all hypercubes  $v_i$ . The hypervolume (HV) is mathematically described as follows:

$$HV = volume\left(\bigcup_{p=1}^{|\hat{Q}|} \mathbb{V}_p\right)$$

When comparing two sets of non-dominated solutions, the set which conveys a larger HV value is considered to be better in terms of both proximity and diversity. The main advantage of the hypervolume metric is that it does not depend on the prior knowledge of the true Pareto front.

Figure 3 shows the graphical representation of the hypervolume metric for the minimization of two objectives:  $f_1$  and  $f_2$ . In this example, the hypervolume is represented by the grey area delimited by the non-dominated solutions ( $\hat{Q} = \{p_1, p_2, p_3, p_4, p_5\}$ ) and the reference point  $r$ .

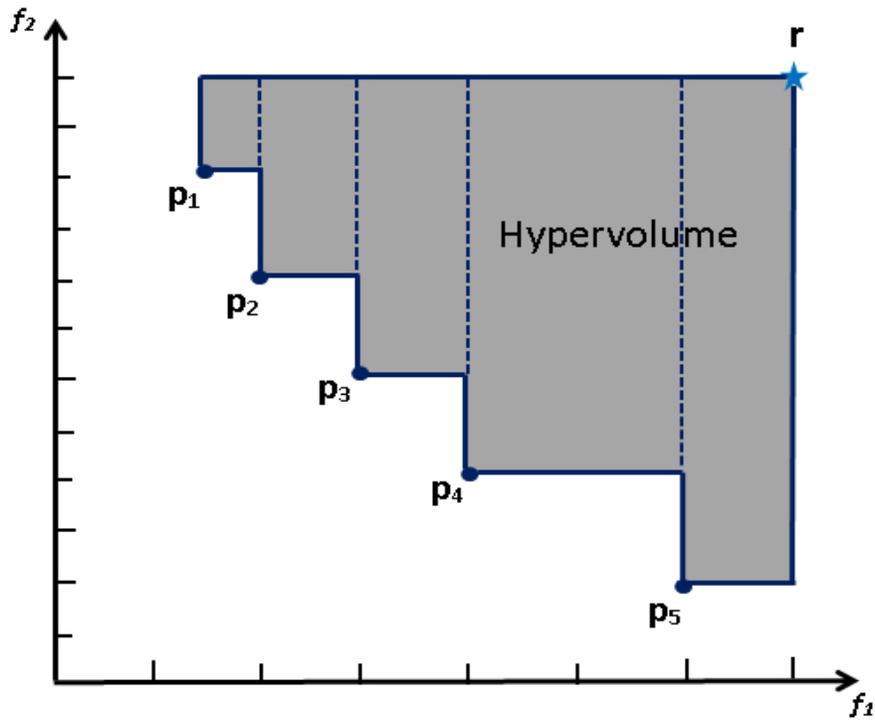


Figure 3: Graphical illustration of the hypervolume (HV) metric for a bi-objective minimization problem.

An accurate calculation of the hypervolume (HV) metric requires a normalized objective space and we use the linear normalization technique proposed by

Knowles et al. [42] as follows:

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

where  $f_i^{min}$  and  $f_i^{max}$  are the minimum and maximum value of the  $i^{th}$  objective. The value of  $f_i^{min}$  and  $f_i^{max}$  are set as the minimum and maximum value obtained from running all algorithms considered in this work.

### 5.3. Studied Approaches

In order to evaluate the overall performance of MODE-GL, we compare it with two well-known multi-objective evolutionary algorithms in the literature, namely NSGA-II and SPEA2. Moreover, learning-guided solution generation mechanism (see Section-4.1.3) has been incorporated into NSGA-II and SPEA2 in order to investigate the impact of the mechanism.

- NSGA-II: the Non-dominated Sorting Genetic Algorithm II was proposed by Deb et al [26]. The algorithm uses binary tournament selection based on the crowding distance. It performs crossover and mutation by simulated binary crossover and polynomial mutation operators.
- SPEA2: the Strength Pareto Evolutionary Algorithm was proposed by Zitzler et al [81]. The algorithm employs fine-grained fitness assignment, density estimation techniques and archive truncation methods. Like NSGA-II, it uses binary tournament selection, simulated binary crossover and polynomial mutation evolutionary operators.
- NSGA-II-GL: A learning mechanism is incorporated into the binary crossover scheme of NSGA-II.
- SPEA2-GL: A learning mechanism is incorporated into the binary crossover scheme of SPEA2.

Parameters	MODE-GL	NSGA-II	SPEA2	NSGA-II-GL	SPEA2-GL
Number of Population ( $NP$ )	100	100	100	100	100
Number of Generation	5,000N	5,000N	5,000N	5,000N	5,000N
Scaling Factor ( $F$ )	0.3	-	-	-	-
Crossover Probability ( $CR$ )	0.9	0.9	0.9	0.9	0.9
Crossover Distribution Index	-	20	20	20	20
Mutation Probability	-	1/N	1/N	1/N	1/N
Mutation Distribution Index	-	20	20	20	20
Tournament Round	-	-	1	-	1
$A_{max}$	100	-	100	-	100
$D_{max}$	10	-	-	-	-

Table 1: Parameter Setting of the Algorithms.

To conduct a fair comparison, we use the same population size and archive size (if applicable) for all the algorithms tested in this work. We have chosen to run all the algorithms with the same stopping criteria (i.e. the same number of evaluations) to generate the Pareto front. Each algorithm also uses the same encodings (see Section-4.1.1) and repair mechanism (see Section-4.1.5) when a newly constructed portfolio violates the considered constraints. Before the experiments were performed, parameters were tuned for all algorithms using DS1. Table-1 shows the best parameter settings used for each of the algorithms.

#### 5.4. Comparisons of the algorithms

In this section, we perform a set of experiments to investigate the potential of MODE-GL for multi-objective constrained portfolio optimization problems and compare it with four other algorithms, NSGA-II, SPEA2, NSGA-II-GL and SPEA2-GL.

The results for IGD, HV and running time of the five algorithms performed on first dataset (DS1) are shown in Figure-4. These results are obtained for the constrained portfolio optimization problem with cardinality  $K = 10$ , floor  $\epsilon_i = 0.01$  and ceiling  $\delta_i = 1.0$ , pre-assignment  $Z = \{30\}$ , round lot  $v_i = 0.008$ , class  $M = 6$  with 15, 15, 15, 15, 15, 19 assets in each class (i.e.,  $C_1 \in \{1, \dots, 15\}$ ,  $C_2 \in \{16, \dots, 30\}$ ,  $C_3 = \{31, \dots, 45\}$ ,  $C_4 \in \{46, \dots, 60\}$ ,  $C_5 \in \{61, \dots, 75\}$ ,  $C_6 \in \{76, \dots, 94\}$  and  $L_m = 0.05$  for each  $m = 1, \dots, 6$ . Given that the lower bound of 5% as class limit specifies an upper bound of 75% of investment in each class/category, no upper limits have been specified.

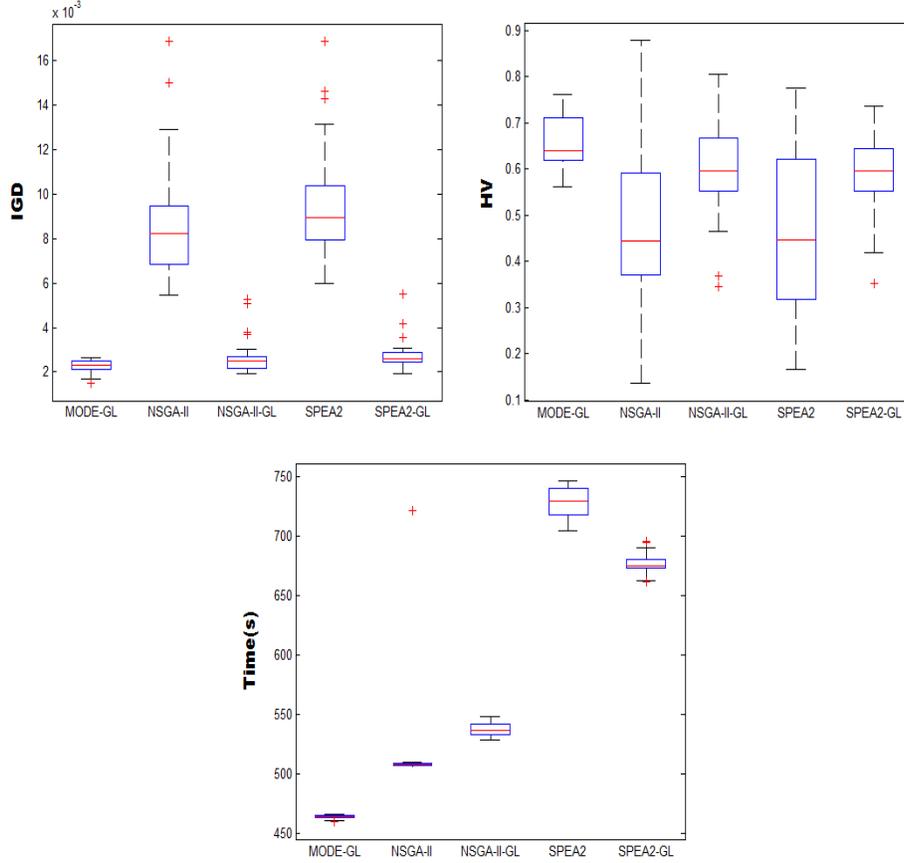


Figure 4: Performance of algorithms in terms of IGD, HV and computational time for S & P 100.

The results show that the proposed MODE-GL obtains the smallest mean values for inverted generational distance (IGD) and largest mean value for hypervolume (HV), compared with the other four algorithms, demonstrating the best performance among the five algorithms. NSGA-II and SPEA2 have similar performance and both have slow convergence compared to MODE-GL. SPEA2 and SPEA2-GL are the most computationally expensive algorithms in terms of CPU time. When the learning-guided solution generation mechanism is incorporated into NSGA-II and SPEA2, the performance of the algorithms improves significantly. Therefore, we conclude that the learning-guided solution generation mechanism promotes the effective convergence of the search.

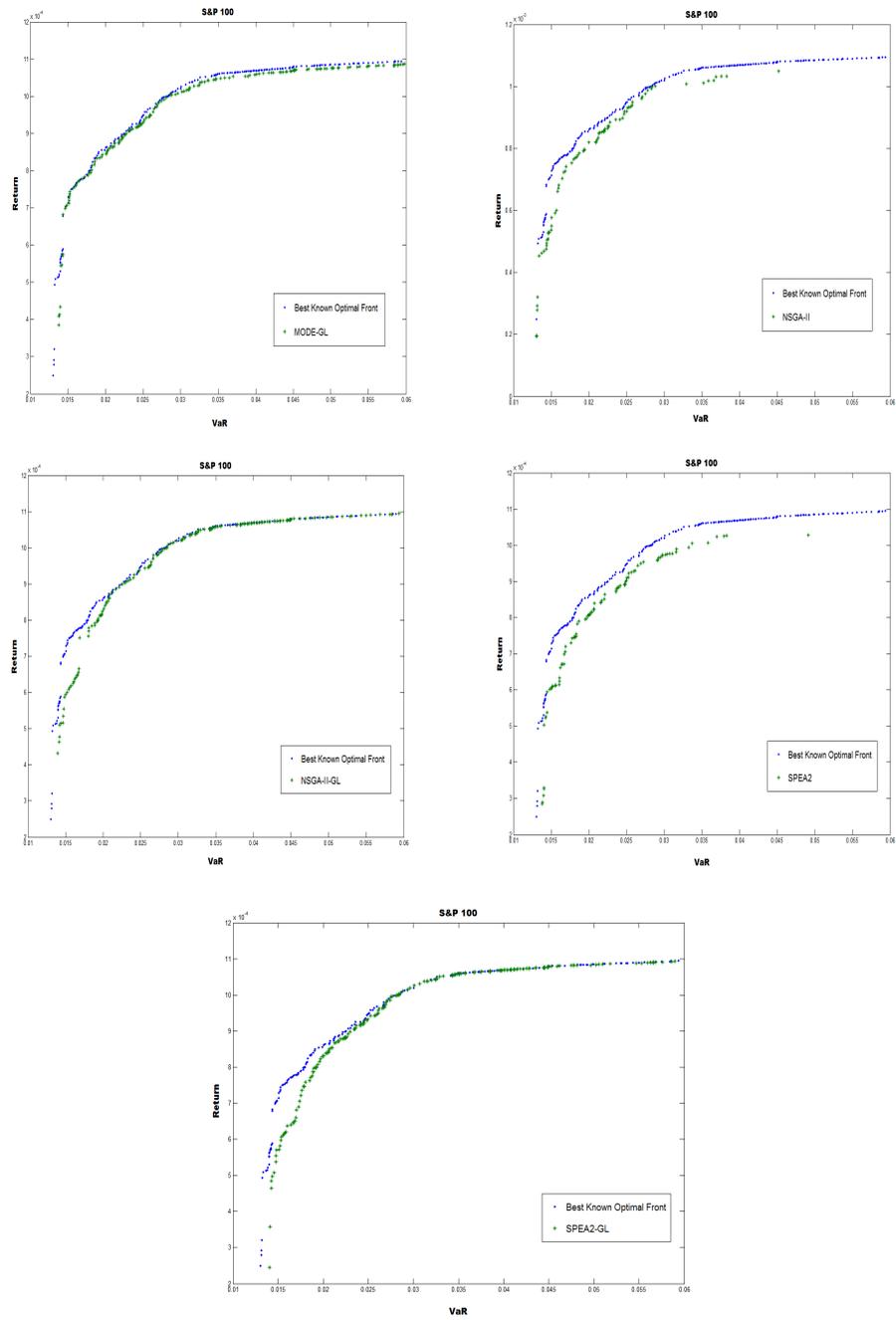


Figure 5: S & P 100: Comparison of efficient frontiers of each algorithm together with the best known optimal front obtained from all tested algorithms.

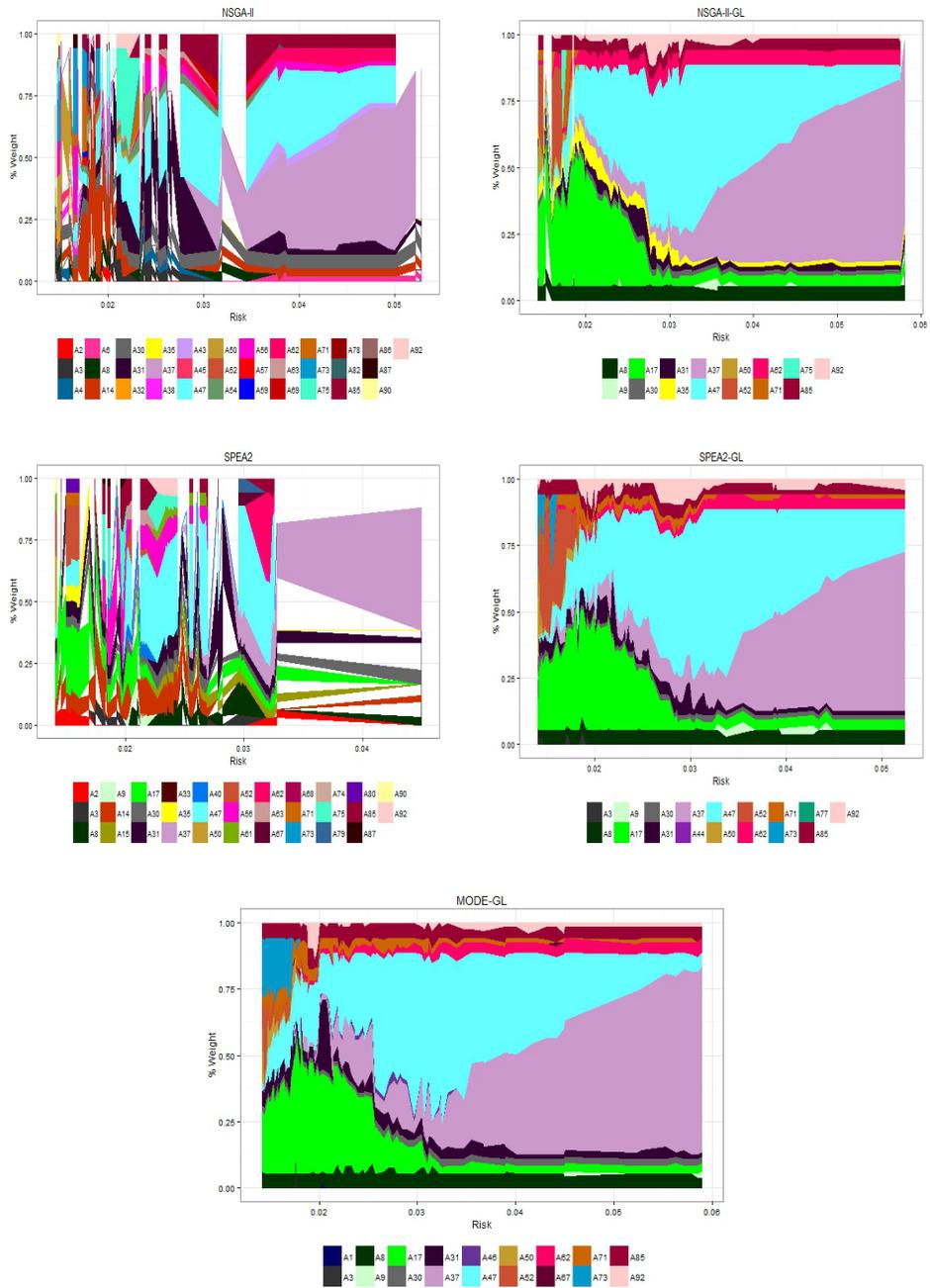


Figure 6: S & P 100: Transaction map for portfolio risk.

As noted, the optimal efficient frontier of the constrained portfolio optimization is not known for the tested datasets. The best estimated efficient frontier is thus obtained by collecting all the non-dominated portfolios produced from all the tested algorithms. For illustrative purpose, the obtained efficient frontiers of the tested algorithms for DS1 compared with the best known estimated efficient frontier are provided in Figure-5. The horizontal axis describes the loss that might be incurred with a probability  $\alpha = 0.01$ . Figure-5 shows that MODE-GL, NSGA-II-GL and SPEA2-GL provide a very good approximation of the efficient frontier. The performance of both NSGA-II and SPEA2 improves significantly when the learning-guided solution generation scheme is incorporated.

Figure-6 shows how the composition of the securities varies over the range of portfolio risk for dataset DS1. The results are generated from efficient solutions obtained from a single run of each algorithm and it shows that allocation to all asset classes is present and the preassigned constraint is also satisfied. In each case, the figure depicts how the obtained portfolio is allocated for an obtained level of risk. Each colour represents one of the assets selected in the obtained pareto set. A vertical strip through the bands (without white space) indicates the obtained portfolio allocations at that risk level. A vertical strip through the bands (with white space) indicates that no feasible solution can be found for a specific risk level. This discontinuity can also be seen in the obtained efficient frontier as depicted in Figure 5 of DS1 Dataset. When the learning mechanism is adopted, the obtained results indicate that the composition of the assets changes smoothly from one risk level to another.

The results for IGD, HV and running time of the five algorithms performed on the second dataset (DS2) are shown in Figure-7. These results are obtained for the constrained portfolio optimization problem with cardinality  $K = 20$ , floor  $\epsilon_i = 0.01$  and ceiling  $\delta_i = 1.0$ , pre-assignment  $Z = \{30\}$ , round lot  $v_i = 0.008$ , class  $M = 19$  with 25 assets in each class and  $L_m = 0.05$  for each  $m = 1, \dots, 19$ . Given that the lower bound of 5% as class limit specifies an upper bound of 10% of investment in each class/category, no upper limits have been specified.

Figure-8 provides the obtained efficient frontiers of the tested algorithms for DS2 compared with the best known estimated efficient frontier extracted from all considered algorithms. The horizontal axis describes the loss that might be incurred with a probability  $\alpha = 0.05$ . The results for DS2 are a little different but show the effectiveness of the incorporation of the learning mechanism in promoting solution quality. MODE-GL, NSGA-II-GL and SPEA2-GL all have similar quality on average. In terms of computational time, SPEA2-GL is the most computationally expensive algorithm in terms of CPU time whereas MODE-GL is the fastest.

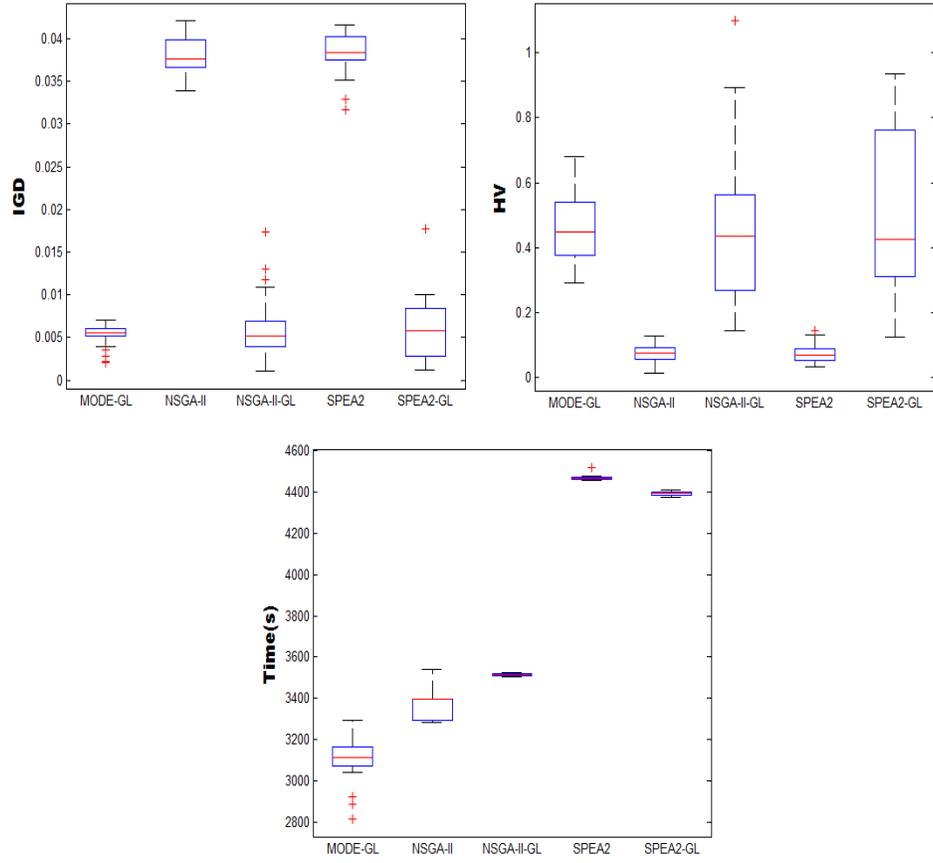


Figure 7: Performance of algorithms in terms of IGD, HV and computational time for S & P 500.

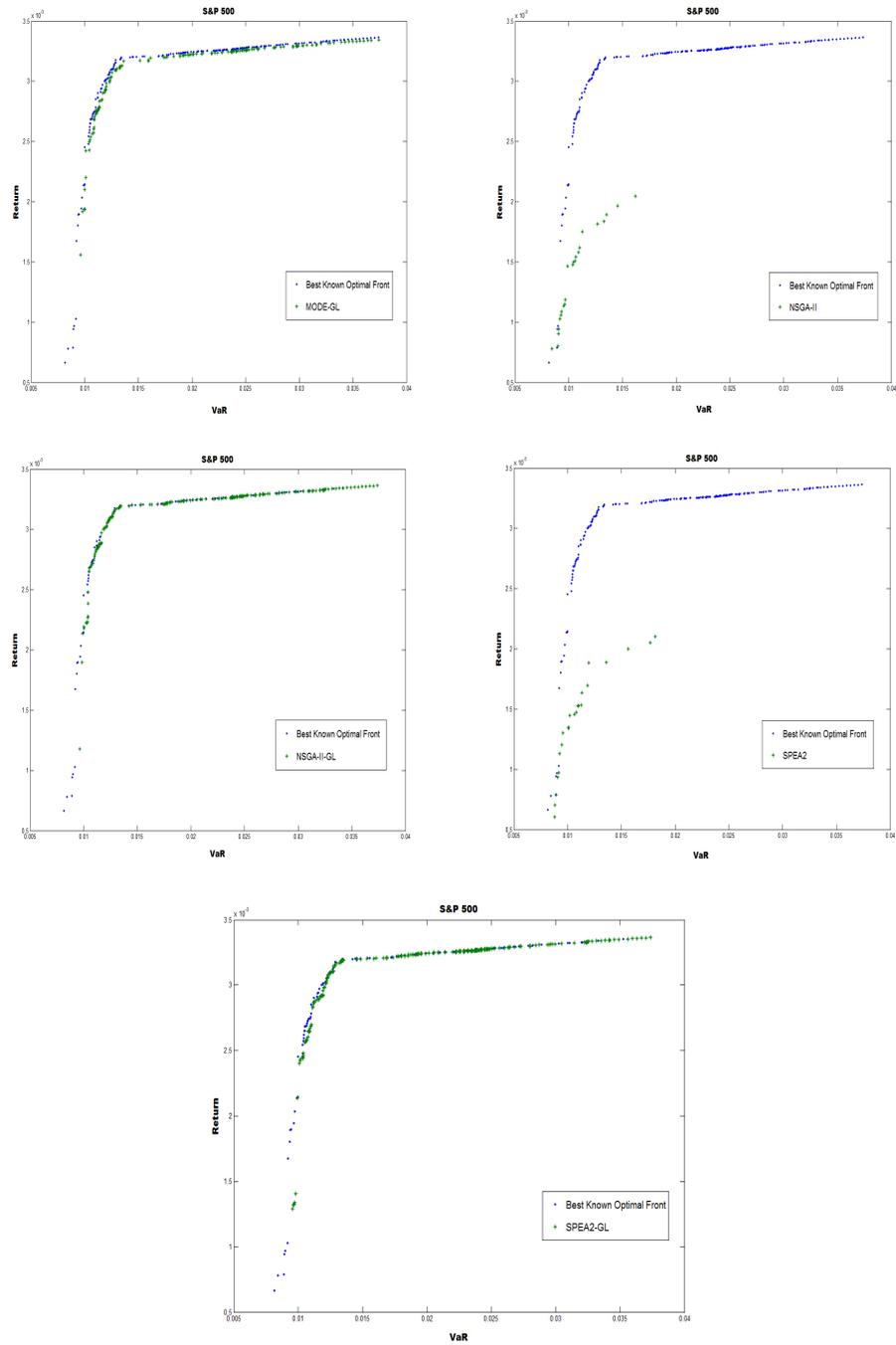


Figure 8: S & P 500: Comparison of efficient frontiers of each algorithm together with the best known optimal front from all tested algorithms.

We compare the IGD and HV values of the tested algorithms by using Student's t-test [73]. The statistical results obtained by a two-tailed t-test with 58 degrees of freedom at a 0.05 level of significance are given in Table-2 and Table-3. The results for Algorithm-1  $\leftrightarrow$  Algorithm-2 are shown as "+", "-", or "~" when Algorithm-1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm-2, respectively. The statistical results reconfirm the effectiveness of the proposed algorithm MODE-GL both in terms of solution quality and computational time. Moreover, the results also show that the performance of the NSGA-II and SPEA2 improves significantly when the learning-guided solution generation scheme is incorporated. Figure-9 plots the IGD metric over generation on S & P 100. The results confirm that all the algorithms considered are able to converge.

Algorithm1 $\leftrightarrow$ Algorithm2	IGD	HV
MODE-GL $\leftrightarrow$ NSGA-II	+	+
MODE-GL $\leftrightarrow$ NSGA-II-GL	+	+
MODE-GL $\leftrightarrow$ SPEA2	+	+
MODE-GL $\leftrightarrow$ SPEA2-GL	+	+
NSGA-II $\leftrightarrow$ NSGA-II-GL	-	-
NSGA-II $\leftrightarrow$ SPEA2	~	~
NSGA-II $\leftrightarrow$ SPEA2-GL	-	-
NSGA-II-GL $\leftrightarrow$ SPEA2	+	+
NSGA-II-GL $\leftrightarrow$ SPEA2-GL	~	~
SPEA2 $\leftrightarrow$ SPEA2-GL	-	-

Table 2: Student's t-Test Results of Different Algorithms on S & P100 dataset.

Algorithm1 $\leftrightarrow$ Algorithm2	IGD	HV
MODE-GL $\leftrightarrow$ NSGA-II	+	+
MODE-GL $\leftrightarrow$ NSGA-II-GL	~	~
MODE-GL $\leftrightarrow$ SPEA2	+	+
MODE-GL $\leftrightarrow$ SPEA2-GL	~	~
NSGA-II $\leftrightarrow$ NSGA-II-GL	-	-
NSGA-II $\leftrightarrow$ SPEA2	~	~
NSGA-II $\leftrightarrow$ SPEA2-GL	-	-
NSGA-II-GL $\leftrightarrow$ SPEA2	+	+
NSGA-II-GL $\leftrightarrow$ SPEA2-GL	~	~
SPEA2 $\leftrightarrow$ SPEA2-GL	-	-

Table 3: Student's t-Test Results of Different Algorithms on S & P 500 dataset.

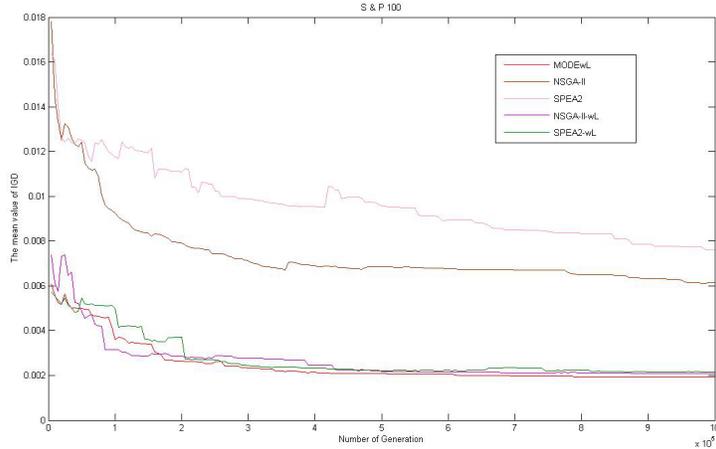


Figure 9: Comparison of convergence of algorithms for S & P 100.

## 6. Conclusions

In this work, we have investigated the portfolio optimization problem with six widely used practical constraints in real life trading scenarios. This work focuses on downside risk as an alternative risk measure in financial markets and adopts a realistic framework for portfolio optimization that moves away from most widely considered mean-variance approach. Value-at-Risk (VaR) is used as a risk measure and a historical simulation approach is adopted to calculate VaR. This technique is nonparametric and does not require any distributional assumptions.

The portfolio optimization in the VaR context involves additional complexities since VaR is non-linear, non-convex and non-differentiable, and typically exhibits multiple local extrema and discontinuities especially when real-world trading constraints are incorporated [32]. A new efficient learning-guided hybrid multi-objective evolutionary algorithm (MODE-GL) has been developed to solve mean-VaR portfolio optimization problems with practical trading constraints. The proposed MODE-GL approach extracts the important features of non-dominated solutions throughout the evolution. Incorporating a learning mechanism and prior problem-specific knowledge exploitation in the evolution process allows the approach to generate promising offspring solutions. The new MODE-GL approach introduced here thus aims to promote convergence by concentrating on the promising regions of the search space. Two extended variants of differential evolution mutation schemes promote exploration of the search in order to explore the solution space to maximize the probability of obtaining the

global optimum.

We have demonstrated that maintaining a secondary population of solution sets in combination with a learning-guided candidate solution generation scheme contributes to better performance over existing state-of-the-art multi-objective evolutionary algorithms, NSGAII and SPEA2. The experimental results using real datasets show that the quality of the generated Pareto set approximations significantly improved for MODE-GL while the efficiency is mainly because the proposed algorithm is computationally cheap as it only uses a single update at each generation. Moreover, the results also show that the performance of the NSGA-II and SPEA2 improves significantly when the learning-guided solution generation scheme is incorporated. In this work, we consider up to six practical trading constraints. However, it does not reflect the factors that may occur in some market trading scenarios. In financial markets, buying and selling assets may entail brokerage fees and taxes imposed on investors. Transaction cost is one of the additional factors that may be a concern for portfolio managers. Therefore, it is important to extend the portfolio optimization model with transaction cost constraint as a direction for future work. The MODE-GL approach shows great promise in tackling an important class of portfolio investment problems using realistic constraints in an efficient way and thus has significant potential for adoption in practice.

## 7. Acknowledgement

This research was supported by the School of Computer Science, The University of Nottingham.

## References

- [1] Alexander, G. J. and Baptista, A. M. (2002). Economic implications of using a mean-var model for portfolio selection: A comparison with mean-variance analysis. *Journal of Economic Dynamics and Control*, 26(7):1159–1193.
- [2] Alfaro-Cid, E., Baixauli-Soler, J. S., and Fernandez-Blanco, M. O. (2011). Minimising value-at-risk in a portfolio optimisation problem using a multi-objective genetic algorithm. *International Journal of Risk Assessment and Management*, 15(5):453–477.
- [3] Anagnostopoulos, K. and Mamanis, G. (2011). Multiobjective Evolutionary Algorithms for Complex Portfolio Optimization Problems. *Computational Management Science*, 8(3):259–279.
- [4] Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent measures of risk. *Mathematical finance*, 9(3):203–228.
- [5] Baixauli-Soler, J., Alfaro-Cid, E., and Fernandez-Blanco, M. (2011). Mean-var portfolio selection under real constraints. *Computational Economics*, 37(2):113–131.

- [6] Bakshi, G., Kapadia, N., and Madan, D. (2003). Stock return characteristics, skew laws, and the differential pricing of individual equity options. *The Review of Financial Studies*, 16(1):pp. 101–143.
- [7] Balbás, A., Balbás, B., and Balbás, R. (2010). Capm and apt-like models with risk measures. *Journal of Banking & Finance*, 34(6):1166–1174.
- [8] Basel Committee on Banking Supervision (1996). Amendment to the capital accord to incorporate market risks. <http://www.bis.org/publ/bcbs24a.pdf>.
- [9] Basel Committee on Banking Supervision (2004). Basel II: International Convergence of Capital Measurement and Capital Standards: a Revised Framework. <http://www.bis.org/publ/bcbs107.pdf>.
- [10] Basel Committee on Banking Supervision (2010). Basel III: A global regulatory framework for more resilient banks and banking systems. *Bank for International Settlements*.
- [11] Bawa, V. S. (1975). Optimal rules for ordering uncertain prospects. *Journal of Financial Economics*, 2(1):95–121.
- [12] Bawa, V. S. and Lindenberg, E. B. (1977). Capital market equilibrium in a mean-lower partial moment framework. *Journal of Financial Economics*, 5(2):189–200.
- [13] Benati, S. and Rizzi, R. (2007). A mixed integer linear programming formulation of the optimal mean/value-at-risk portfolio problem. *European Journal of Operational Research*, 176(1):423–434.
- [14] Biglova, A., Ortobelli, S., Rachev, S. T., and Stoyanov, S. (2004). Different approaches to risk estimation in portfolio theory. *The Journal of Portfolio Management*, 31(1):103–112.
- [15] Chang, T., Meade, N., Beasley, J., and Sharaiha, Y. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research*, 27(13):1271–1302.
- [16] Charpentier, A. and Oulidi, A. (2009). Estimating allocations for value-at-risk portfolio optimization. *Mathematical Methods of Operations Research*, 69(3):395–410.
- [17] Coello, C. (2006). Evolutionary multi-objective optimization and its use in finance. *Handbook of Research on Nature Inspired Computing for Economy and Management*. Idea Group Publishing.
- [18] Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *Evolutionary Computation, IEEE Transactions on*, 8(3):256–279.

- [19] Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191(11):1245–1287.
- [20] Cont, R. (2001). *Empirical properties of asset returns: stylized facts and statistical issues*. Taylor & Francis.
- [21] Cont, R., Deguest, R., and Scandolo, G. (2010). Robustness and sensitivity analysis of risk measurement procedures. *Quantitative Finance*, 10(6):593–606.
- [22] Dallagnol, V. A. F., van den Berg, J., and Mous, L. (2009). Portfolio management using value at risk: A comparison between genetic algorithms and particle swarm optimization. *International Journal of Intelligent Systems*, 24(7):766–792.
- [23] Daniélsson, J., Jorgensen, B. N., de Vries, C. G., and Yang, X. (2008). Optimal portfolio allocation under the probabilistic var constraint and incentives for financial innovation. *Annals of finance*, 4(3):345–367.
- [24] Das, S. and Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4–31.
- [25] Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.
- [26] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.
- [27] Duffie, D. and Pan, J. (1997). An overview of value at risk. *The Journal of derivatives*, 4(3):7–49.
- [28] Fama, E. F. (1965). The behavior of stock-market prices. *Journal of business*, pages 34–105.
- [29] Feng, M., Wächter, A., and Staum, J. (2015). Practical algorithms for value-at-risk portfolio optimization problems. *Quantitative Finance Letters*, 3(1):1–9.
- [30] Fishburn, P. C. (1977). Mean-risk analysis with risk associated with below-target returns. *The American Economic Review*, pages 116–126.
- [31] Fonseca, C. and Fleming, P. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 3(1):1–16.
- [32] Gaivoronski, A. A. and Pflug, G. (2005). Value-at-risk in portfolio optimization: properties and computational approach. *Journal of Risk*, 7(2):1–31.

- [33] Ghaoui, L. E., Oks, M., and Oustry, F. (2003). Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51(4):543–556.
- [34] Gilli, M. and Këllezi, E. (2001). A global optimization heuristic for portfolio choice with var and expected shortfall. *Computational Methods in Decision-making, Economics and Finance, Applied Optimization Series*. Kluwer Academic Publishers.
- [35] Gilli, M., Këllezi, E., and Hysi, H. (2006). A data-driven optimization heuristic for downside risk minimization. *Swiss Finance Institute Research Paper*, (06-2).
- [36] Goh, J. W., Lim, K. G., Sim, M., and Zhang, W. (2012). Portfolio value-at-risk optimization for asymmetrically distributed asset returns. *European Journal of Operational Research*, 221(2):397 – 406.
- [37] Harlow, W. V. (1991). Asset allocation in a downside-risk framework. *Financial Analysts Journal*, pages 28–40.
- [38] Jackson, P., Maude, D. J., and Perraudin, W. (1997). Bank capital and value at risk. *The Journal of Derivatives*, 4(3):73–89.
- [39] Jevne, H. K., Haddow, P. C., and Gaivoronski, A. A. (2012). Evolving constrained mean-var efficient frontiers. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE.
- [40] Jorion, P. (2006). *Value at risk: the new benchmark for managing financial risk*. McGraw-Hill New York, 3rd edition edition.
- [41] Knowles, J. and Corne, D. (2002). On metrics for comparing nondominated sets. In *Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on*, volume 1, pages 711–716. IEEE.
- [42] Knowles, J., Thiele, L., and Zitzler, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. revised version.
- [43] Knowles, J. D. and Corne, D. W. (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172.
- [44] Kolm, P. N., Tütüncü, R., and Fabozzi, F. J. (2014). 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234(2):356 – 371. 60 years following Harry Markowitz’s contribution to portfolio theory and operations research.
- [45] Kon, S. J. (1984). Models of stock returns a comparison. *The Journal of Finance*, 39(1):147–165.

- [46] Krokmal, P., Zabaranin, M., and Uryasev, S. (2011). Modeling and optimization of risk. *Surveys in Operations Research and Management Science*, 16(2):49 – 66.
- [47] Linsmeier, T. J. and Pearson, N. D. (2000). Value at risk. *Financial Analysts Journal*, pages 47–67.
- [48] Lwin, K. and Qu, R. (2013). A hybrid algorithm for constrained portfolio selection problems. *Applied intelligence*, 39(2):251–266.
- [49] Lwin, K., Qu, R., and Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24:757–772.
- [50] Maio, F. D., Baronchelli, S., and Zio, E. (2014). Hierarchical differential evolution for minimal cut sets identification: Application to nuclear safety systems. *European Journal of Operational Research*, 238(2):645 – 652.
- [51] Manganelli, S. and Engle, R. F. (2001). Value at risk models in finance. Technical report, European Central Bank Frankfurt am Main.
- [52] Mansini, R., Ogryczak, W., and Speranza, M. G. (2007). Conditional value at risk and related linear programming models for portfolio optimization. *Annals of operations research*, 152(1):227–256.
- [53] Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):pp. 77–91.
- [54] Markowitz, H. (1959). Portfolio selection: Efficient diversification of investments. *John Wiley and Sons, New York*.
- [55] Morgan, J. P. (1996). RiskMetrics (TM): Technical document, 4th ed. *New York: Morgan Guaranty Trust Company*.
- [56] Natarajan, K., Pachamanova, D., and Sim, M. (2008). Incorporating asymmetric distributional information in robust value-at-risk optimization. *Management Science*, 54(3):573–585.
- [57] Pérignon, C. and Smith, D. R. (2010). The level and quality of value-at-risk disclosure by commercial banks. *Journal of Banking & Finance*, 34(2):362–377.
- [58] Prakash, A. J., Chang, C.-H., and Pactwa, T. E. (2003). Selecting a portfolio with skewness: Recent evidence from us, european, and latin american equity markets. *Journal of Banking & Finance*, 27(7):1375–1390.
- [59] Pritsker, M. (1997). Evaluating value at risk methodologies: accuracy versus computational time. *Journal of Financial Services Research*, 12(2-3):201–242.

- [60] Robič, T. and Filipič, B. (2005). Demo: Differential evolution for multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 520–533. Springer.
- [61] Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42.
- [62] Rockafellar, R. T. and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471.
- [63] Rossello, D. (2015). Ranking of investment funds: Acceptability versus robustness. *European Journal of Operational Research*, 245(3):828–836.
- [64] Roy, A. D. (1952). Safety first and the holding of assets. *Econometrica: Journal of the Econometric Society*, pages 431–449.
- [65] Salcedo-Sanz, S. (2009). A survey of repair methods used as constraint handling techniques in evolutionary algorithms. *Computer science review*, 3(3):175–192.
- [66] Sarykalin, S., Serraino, G., and Uryasev, S. (2008). Value-at-risk vs. conditional value-at-risk in risk management and optimization. *Tutorials in Operations Research. INFORMS, Hanover, MD*, pages 270–294.
- [67] Sierra, M. R. and Coello, C. A. C. (2005). Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance. In *EMO’05*, pages 505–519.
- [68] Skolpadungket, P., Dahal, K., and Harnpornchai, N. (2007). Portfolio optimization using multi-objective genetic algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 516–523. IEEE.
- [69] Storn, R. and Price, K. (1995). Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA.
- [70] Streichert, F., Ulmer, H., and Zell, A. (2004). Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 932–939. IEEE.
- [71] Valle, C., Meade, N., and Beasley, J. (2014). Absolute return portfolios. *Omega*, 45(0):20 – 41.
- [72] Van Veldhuizen, D. A. and Lamont, G. B. (2000). On measuring multiobjective evolutionary algorithm performance. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 204–211. IEEE.
- [73] Walpole, R. E., Myers, R. H., Myers, S. L., and Ye, K. (1998). *Probability and statistics for engineers and scientists*, volume 8. Prentice Hall Upper Saddle River, NJ:.

- [74] Woodside-Oriakhi, M., Lucas, C., and Beasley, J. (2013). Portfolio rebalancing with an investment horizon and transaction costs. *Omega*, 41(2):406 – 420. Management science and environmental issues.
- [75] Wozabal, D., Hochreiter, R., and Pflug, G. C. (2010). A difference of convex formulation of value-at-risk constrained optimization. *Optimization*, 59(3):377–400.
- [76] Yahoo, W. (2014). Yahoo Finance. <https://finance.yahoo.com/>. Accessed:30-05-2014.
- [77] Zhang, J. and Sanderson, A. C. (2009). Jade: adaptive differential evolution with optional external archive. *Evolutionary Computation, IEEE Transactions on*, 13(5):945–958.
- [78] Zhao, Z., Wang, J., Zhao, J., and Su, Z. (2012). Using a grey model optimized by differential evolution algorithm to forecast the per capita annual net income of rural households in china. *Omega*, 40(5):525 – 532.
- [79] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.
- [80] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.
- [81] Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm.
- [82] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4):257–271.
- [83] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132.