**Noname manuscript No.**
(will be inserted by the editor)

# Hybrid Algorithm for Constrained Portfolio Selection Problem

Received: date / Accepted: date

**Abstract** Since Markowitz's seminal work on the mean-variance model in modern portfolio theory, many studies have been conducted on computational techniques and recently meta-heuristics for portfolio selection problems. In this work, we propose and investigate a new hybrid algorithm integrating the population based incremental learning and differential evolution algorithms for the portfolio selection problem. We consider the extended mean-variance model with practical trading constraints including the cardinality, floor and ceiling constraints. The proposed hybrid algorithm adopts a partially guided mutation and an elitist strategy to promote the quality of solution. The performance of the proposed hybrid algorithm has been evaluated on the extended benchmark datasets in the OR Library. The computational results demonstrate that the proposed hybrid algorithm is not only effective but also efficient in solving the mean-variance model with real world constraints.

**Keywords** Mean-Variance Portfolio Optimization · Constrained Portfolio Selection Problem · Cardinality Constrained Portfolio Selection · Differential Evolution · Population Based Incremental Learning

## 1 Introduction

The portfolio selection problem (PSP) is concerned with the allocation of limited capital to a number of potential assets (investments) for a profitable investment strategy. The pioneering work to the PSP is the concept of efficient set developed by Nobel Laureate Harry Markowitz [32][52]. In his seminal work [32] which sets the foundations of modern portfolio theory (MPT), Markowitz viewed portfolio selection as a mean-variance optimization problem with regard to two criteria: to maximize the reward of a portfolio (measured by the mean of expected return), and to minimize the risk of the portfolio (measured by the variance of return). More formally, a desirable portfolio is defined to be a trade-off between risk and expected return.

With the continuous efforts of many researchers, Markowitz's seminal work has been widely extended. Markowitz et al [34], Pang [37] and Best and Hlouskova [8] adopted the mean-variance approach to compute the efficient frontier (see Section-2.1) of the PSP without taking into consideration of practical constraints. A number of exact approaches had also been proposed to solve the basic mean-variance PSP [33][35].

Although the Markowitz mean-variance model is the fundamental theory of MPT, direct application of this model is not of much practical uses mainly due to the fact that it is simplified with some unrealistic assumptions. It assumes a perfect market without taxes or transaction costs where short sales are not allowed, and securities are infinitely divisible, i.e. they can be traded in any (non-negative) fraction. From the practical point of view, real-world investors commonly face restrictions such as cardinality and bounding constraints. The cardinality constraint imposes a limit on the number of assets in the portfolio either to simplify the management of the portfolio or to reduce transaction costs. The bounding constraint restricts the proportion of each asset in the portfolio to lie between the lower and uppper bounds in order to avoid very small (or large) and unrealistic holdings. The more the model is extended to include relevant practical constraints the more it becomes difficult to solve.

Many researchers have investigated a variety of techniques to solve the constrained PSP. Some research has been conducted to solve the PSP with cardinality constraints by using different exact techniques [7][9][29][43][54]. However, these exact techniques may fail to find an optimal solution in a reasonable time and are computationally ineffective when applied to large-scale problems.

Extending the PSP with cardinality constraint already transforms the model from quadratic optimization model to quadratic mixed-integer problem (QMIP) which is proved to be NP-hard [36]. Since QMIPs are hard to solve optimally, many researchers have applied different heuristic optimization methods to the constrained PSP. Some research uses heuristic to solve the constrained PSP in the mean-variance framework. Fernandez et al [19] applied a Hopfield neural network heuristic to the PSP with cardinality and bounding constraints. Jobst et al [26] proposed two heuristics (integer restart and reoptimisation) using FortMP solver [17] for the constrained PSP. Perold [38] proposed a piecewise linear convex approximation of the cost function to locate efficient frontier for large-scale PSP considering a number of constraints.

Recent research on the constrained PSP with the mean-variance framework has also investigated local search based algorithms. Shearf [39] applied a hill climbing algorithm for the constrained PSP. Arriaga and Valenzuela-Rend [2] presented a Steepest Ascent Hill Climbing algorithm. The results indicated that it is competitive against genetic algorithms in terms of performance and execution time. Carama and Schyns [12][40] adopted simulated annealing (SA) to solve the Markowitz model with real-world constraints. It was claimed that the proposed algorithm can approximate the efficient frontier for medium size problems in a reasonable runtime. Busetti [10] also investigated tabu search (TS) to solve the PSP with cardinality, bounding and transaction cost constraints. Shearf [39], Chang et al [11] and Woodside-Oriakhi et al [56] presented TS and SA to solve PSP with cardinality and bounding constraints. Computational results on the OR-libraray datasets [5][6] were presented. Gaspero et al [22] proposed a hybrid technique that combined local search with a quadratic programming procedure to solve the constrained PSP.

In recent years, a majority of work in the literature had been focused on the population based metaheuristic algorithms for the PSP in mean-variance framework. Several works had applied genetic algorithms (GAs) to solve PSP with various constraints [11][19][45][56]. Experimental results showed that GA outperformed SA and TS. Some works had also applied the particle swarm optimization algorithms (PSOs) to compute the constrained efficient frontier of PSP [24][57][13].

Moral-Escudero et al [36] proposed a hybrid strategy to solve the PSP with cardinality constraints. The proposed hybrid method used GA to select the optimal subset of the available assets and quadratic programming to determine the proportion of capital to be invested in each asset. The results outperformed TS in several benchmark problems. Xu et al [58] also presented a hybrid algorithm to solve the PSP with cardinality and bounding constraints. The proposed algorithm hybridizes a population based incremental learning algorithm and a continuous population based incremental learning algorithm to optimize the selection and the proportion of assets respectively. The experimental results showed that the proposed algorithm was competitive to GA and PSO and achieved good results in searching efficient portfolios with high expected returns.

Several works had also been carried out for the PSP using multiobjective approaches. Ehrgott et al [16] applied a GA to optimize the PSP with objectives which are aggregated via user-specified utility functions. Skolpadungket et al [44] and Streichert et al [48] proposed various types of multiobjective GAs to solve the constrained PSP. Krink et al [27][28] proposed a differential evolution (DE) algorithm for multiobjective portfolio optimization. The proposed algorithm was compared with quadratic programming and NSGA-II. A comprehensive review of metaheuristics in portfolio selection problem could be found in [15] and [30].

In this work, we propose a hybrid algorithm to compute efficient frontier for the mean-variance model with the cardinality and bounding constraints. The rest of the paper is organized as follows. Section 2 presents the basic Markowitz mean-variance model and extends the model with cardinality, floor and ceiling constraints. Section 3 provides a detailed description on the components of the hybrid algorithm based on population based incremental learning and differential evolution algorithms. Section 4 presents experiments performed and the computational results. Conclusions are given in Section 5.

## 2 Problem Statement

### 2.1 The Markowitz Mean-Variance Model

The Markowitz mean-variance model (MV model) is formulated as an optimization problem over real-valued variables with a quadratic objective function and linear constraints as follows.

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

$$\textit{subject to} \quad \sum_{i=1}^{N} w_i \mu_i = R^* \tag{2}$$

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

$$0 \leq w_i \leq 1, i = 1, ..., N \tag{4}$$

where $N$ is the number of available assets, $\mu_i$ is the expected return of asset $i$ ($i = 1...N$), $\sigma_{ij}$ is the covariance between assets $i$ and $j$ ($i = 1...N$; $j = 1...N$), $R^*$ is the desired expected return, and $w_i$ ($0 \leq w_i \leq 1$) is the decision variable which represents the proportion held of asset $i$. Equation(1) minimizes the total variance (risk) associated with the portfolio whilst Equation(2), the *return constraint*, ensures that the portfolio has a predetermined expected return of $R^*$. Equation(3) defines the *budget constraint* (all the money available should be invested) for a feasible portfolio while Equation(4) requires that all investment should be *positive*, i.e., no short sales are allowed. We could trace out the set of efficient portfolios by solving the model (Eq(1)-(4)) repeatedly with different value of $R^*$ at each time.

By introducing a risk aversion parameter $\lambda \in [0, 1]$, the sensitivity of the investor to the risk can be defined in the model as follows.

$$\textit{minimize} \ \lambda \left[ \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \right] + (1-\lambda) \left[ -\sum_{i=1}^{N} w_i \mu_i \right] \tag{5}$$

$$\textit{subject to} \quad \sum_{i=1}^{N} w_i = 1 \tag{6}$$

$$0 \leq w_i \leq 1, i = 1, ..., N \tag{7}$$

In Equation(5), when $\lambda$ is zero, the model maximizes the mean expected return of the portfolio regardless of the variance (risk). On the other hand, when $\lambda$ equals one, the model minimizes the risk of the portfolio regardless of the mean expected return. For the portfolio selection problem in Eq(5)-(7), a portfolio is considered to be *efficient* based on the concept of the Pareto optimality [21]. In other words, for a given level of risk, compared to an efficient portfolio, there should be no portfolio with a higher expected return, or conversely for a given expected return there should be no portfolio with a lower risk. The complete set of these efficient portfolios forms the *efficient frontier* that represents the best trade-offs between the mean return and the variance (risk).[1] Figure-1 shows the unconstrained effcient frontier derived for the Hang Seng dataset (see Section-4.1) from the OR-library [5][6].

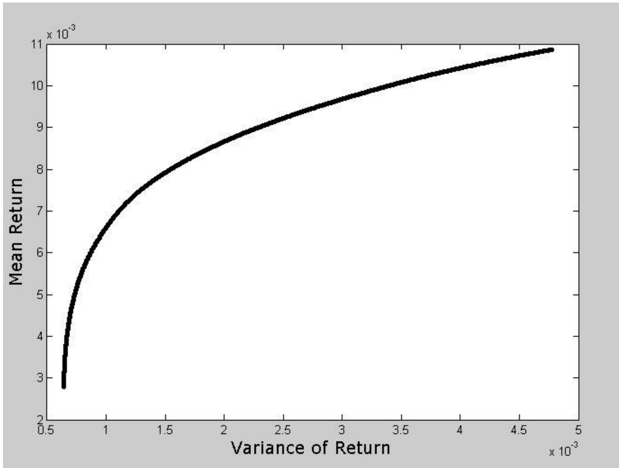


**Fig. 1** The unconstrained Efficient Frontier for the Hang Seng dataset.

### 2.2 The Mean Variance Model with Cardinality and Bounding Constraints (CCMV)

The basic mean variance model has several limitations which prohibit its use in practice. As a result several extensions and modifications have been developed in the literature to address real world constraints. In this work, we consider two common trading constraints, namely the cardinality and bounding constraints. Cardinality constraint specifies the maximum number of assets that a portfolio can hold to simplify the management of the portfolio and to reduce transaction costs. Bounding constraints[2] specify the lowest and highest limits on the proportion of each asset that can be held in a single portfolio. With these two constraints, the model can be described as follows.

[1] For an analytic derivation of the efficient frontier, see [35]

[2] In some literature, it is also known as quantity constraints or buy-in threshold constraints

$$minimize\ \lambda \left[ \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} \right] + (1-\lambda) \left[ - \sum_{i=1}^{N} w_i \mu_i \right] \quad (8)$$

$$subject\ to \qquad \sum_{i=1}^{N} w_i = 1, \quad (9)$$

$$\sum_{i=1}^{N} s_i = K, \quad (10)$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, i = 1, ..., N, \quad (11)$$

$$s_i \in \{0, 1\}, i = 1, ..., N \quad (12)$$

where $K$ is the desired number of invested assets in the portfolio, $s_i$ denotes whether asset $i$ is invested or not. If $s_i$ equals to one, asset $i$ is chosen to be invested and the proportion of capital $w_i$ lies in $[\epsilon_i,\ \delta_i]$, where $0 \leq \epsilon_i \leq \delta_i \leq 1$. Otherwise, asset $i$ is not invested and $w_i$ equals to zero. The above stated CCMV model is a mixed integer quadratic programming problem for which there exists no efficient algorithm. It may be seen as two subproblems: the selection of assets and the determination of the proportions of the selected assets. A new hybrid algorithm is presented in Section-3 to address these problems. When the basic model is extended to include the cardinality and bounding constraints the resultant efficient frontier might be discontinuous [11][26].

## 3 A Hybrid Algorithm (PBILDE) for the Portfolio Selection Problem

In this work, we propose a new hybrid approach based on evolutionary algorithms which iteratively evolve a population of candidate solutions towards better solutions. Inspired by the works in the literature [4][50][51][53][58], our hybrid approach, PBILDE, combines population based incremental learning (PBIL) and differential evolution (DE) to solve the CCMV model.

Population Based Incremental Learning (PBIL), proposed by Baluja [3][4], is one of the simplest yet effective Estimation of Distribution Algorithms (EDAs). It is based on the idea of evolving the individuals of the population based on statistical information gathered during evolution. Assuming there is no dependence between the variables, PBIL uses a probability vector to represent the distribution of all individuals. The probability vector is learnt towards the values that represent the best solution. The population of random samples is then generated based on the probabilites specified in the probability vector. For more comprehensive overviews of PBIL, see [3][4][25][41].

Differential Evolution (DE), proposed by Stron and Price [46][47], is one of the most successful evolutionary algorithms (EAs) for continuous optimization problems. Like a typical EA, DE has a random initial population that is then improved using mutation, crossover and recombination operations. DE mutates a (parent) vector in the population with a scaled difference of other randomly selected individual vectors. The resultant vector is then crossed over with the parent vector to generate a trial or offspring vector. The offspring vector replaces the parent vector if it has a better fitness value. Otherwise, the parent vector survive and is passed on to the next generation. There are several variants of DE in the literature [46][47]. They are varied by using different types of solution, different number of solutions to calculate the mutation values and different types of recombination operators. In this work, we use the scheme which can be classified as DE/rand/1/bin, where "rand" indicates that individuals are selected randomly to compute the mutation values, "1" denotes the number of pairs of solutions chosen and finally "bin" means that binomial recombination is used. For more comprehensive overviews of DEs, see [14][18][31][46][47].

### 3.1 Overview of the Hybrid Algorithm

We propose a hybrid algorithm, PBILDE, to efficiently address the CCMV model described in Section-2.2. PBILDE maintains a population of chromosomes, each representing a potential solution to the portfolio selection problem with cardinality and bounding constraints. It also maintains a real-valued probability vector to denote the probability of each asset being selected in high quality portfolios. As mentioned in the previous section, the portfolio selection problem can be seen as two sub-problems, the determination of the selection of assets and the allocation of capital to each asset. In each iteration of PBILDE, the probability vector is used to generate a population of solutions determining which assets are included in each solution. The DE offspring generation scheme (see Section-3.2.7) is used to allocate the proportions of assets.

In each iteration, PBILDE maintains an archive of the best solutions found during the evolution (see Section-3.2.4). A partially guided mutation (see Section-3.2.6) is also adopted to guide further search towards selecting favorable set of assets. The evolution process continues until a stopping criterion is met (i.e, the current best objective function value is better than a given value or it reaches to a certain number of generations). The detailed description of the algorithm and pseudocode (see Figure-2) are described in Section-3.2.

## 3.2 The Hybrid Algorithm

Let

$N$ = number of available assets
$NP$ = number of individuals in the population
$K$ = number of selected asset(s) in a portfolio, i.e. the cardinality
$\epsilon_i$ = minimum limit on the proportion of the $i^{th}$ asset
$\delta_i$ = maximum limit on the proportion of the $i^{th}$ asset

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

$w_i$ = proportion invested in the $i^{th}$ asset
$v_i$ = probability of the $i^{th}$ asset being selected
$M$ = number of portfolio(s) in the archive
$GBest$ = the archive maintaining the $M$ best portfolio(s) found so far
$best$ = the best portfolio in the archive $GBest$
$cbest$ = the best individual of the current population
$cworst$ = the worst individual of the current population
$s_i^{cbest}$ = $s_i$ of the best portfolio of the current population
$s_i^{cworst}$ = $s_i$ of the worst portfolio of the current population
$LR$ = learning rate
$NEG\_LR$ = negative learning rate
$MP$ = mutation probability
$MR$ = mutation rate
$CR$ = crossover rate
$F$ = scaling factor
$P^g$ = population of generation g
$Rand[x, y]$=uniform random integer in the range $[x, y]$
$rand[x, y]$=uniform random real-value in the range $[x, y]$

---

**Pseudocode:** PBILDE Algorithm

---

```
BEGIN
  INITIALIZATION:
     for i := 1 to N
        v_i = 0.5
     end for
     for each portfolio p_j, j := 1 to NP do
        randomly generate an individual
        if constraints are violated
           repair by Constraint Handling Techniques
           (see Section-3.2.8)
     end for
  Repeat until certain number of generations
    EVALUATE:
        for j := 1 to NP do
           evaluate f(p_j) \\see Eq-(13)
        end for
     CREATE ARCHIVE:
        GBest ← Maintain the M best portfolio(s)
                found so far
        cbest ← best portfolio of the current population
        if (f(cbest) > f(best))
           Replace the M worst individuals of the
           current population by the M best
           individuals from the archive, GBest
        end if
     UPDATE:
        update v_i by learning from the best and worst
        individuals of the current population
        (see Section-3.2.5)
     MUTATE:
        Perform Partially Guided Mutation
        (see Section-3.2.6)
     GENERATE OFFSPRING: (see Section-3.2.7)
        Generate a trial population by DE offspring
        generation scheme.
        Determine individuals of the next population
        using greedy selection.
END
```

---

**Fig. 2** Pseudocode of the Proposed PBILDE Algorithm.

### 3.2.1 Solution representation and encoding

In our solution representation, one probability vector of size $N$ is used to determine which assets are included in the portfolio. The probability vector $v$ is updated throughout the evolution by learning from the best solutions obtained from the population. Two vectors of size $N$ are used to define a portfolio $p$: a binary vector $s_i$, $i = 1, ..., N$ denoting whether asset $i$ is included in the portfolio, and a real-value vector $w_i$, $i = 1, ..., N$ representing the proportions of the capital invested in the assets. Some existing research adopts the same encoding method to define a portfolio [1][49].

| | | | | | |
|---|---|---|---|---|---|
| $v$ | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $s$ | 1 | 0 | 0 | 1 | 0 | $p_1$ |
| $w$ | 0.69 | 0.0 | 0.0 | 0.57 | 0.0 | |

$\vdots$

| | | | | | | |
|---|---|---|---|---|---|---|
| $s$ | 0 | 1 | 0 | 1 | 0 | $p_{NP}$ |
| $w$ | 0.0 | 0.29 | 0.0 | 0.87 | 0.0 | |

$\Downarrow$ $Normalize$

| | | | | | | |
|---|---|---|---|---|---|---|
| $s$ | 1 | 0 | 0 | 1 | 0 | $p_1$ |
| $w'$ | 0.55 | 0.0 | 0.0 | 0.45 | 0.0 | |

$\vdots$

| | | | | | | |
|---|---|---|---|---|---|---|
| $s$ | 0 | 1 | 0 | 1 | 0 | $p_{NP}$ |
| $w'$ | 0.0 | 0.25 | 0.0 | 0.75 | 0.0 | |

**Fig. 3** Example of an initial population and probability vector $v$.

*3.2.2 Initialization*

In PBILDE, the evolution is carried out on a population of a predefined number of individuals $p$ which are represented by $s_i$ and $w_i$. The probability vector $v_i$, which is maintained, is used to determine if asset $i$ is selected in a portfolio, i.e. $s_i = 1$ or $s_i = 0$. An initial population of the predetermined number of portfolios from the $N$ available assets is randomly generated. Initially, the probability vector $v_i$ is set to 0.5 to give equal chances to each asset being selected. The proportions of the selected assets in each solution are then randomly generated from the given lower and upper bounds by adopting Gaussian distribution. The randomly constructed portfolio could violate the constraints in the model and the constraint handling scheme described in Section-3.2.8 is applied to adjust and normalize the weights. (See Figure- 3)

*3.2.3 Evaluation*

To differentiate good and bad portfolios, the fitness of a portfolio $p$ is evaluated as follows:

$$f(p) = \lambda^* \left[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}\right] + \left(1-\lambda^*\right)\left[-\sum_{i=1}^{N} w_i \mu_i\right] \quad (13)$$

where $f(p)$ denotes the fitness of individual $p$, $\lambda^*$ denotes the value of $\lambda$, $w_i$ and $w_j$ denotes the $i^{th}$ and $j^{th}$ dimension of the proportion vector in individual $p$, respectively. The smaller the fitness value the better is the portfolio.

*3.2.4 Maintain of the Archive*

During the evolution, an archive ($GBest$) reserves the $M$ best portfolios. At each iteration during the evolution, the archive is updated to maintain the best individuals found so far. If the best individual in the new sampled population ($cbest$) at the current generation is worse than the global best individual found so far ($best$), then the $M$ worst individual(s) of the current population are replaced by the $M$ global best individual(s) from the archive. The strategy promotes the convergence of the algorithm. The idea to maintain the archive is to ensure that the global best solutions found by the algorithm are not lost and to exploit the global best solution(s) found during the search to help find better soultions.

*3.2.5 Update of the probability vector*

In PBILDE, the probability vector $v$ is used to store statistic information collected during the evoluation to guide the generation of the following populations. At each generation, the learning rate ($LR$) and negative learning rate ($NEG\_LR$) are used to update the probability vector ($v$). They control not only the speed at which the probability vector is shifted to resemble the best solution vector but also the portion of the search space that will be explored [20][42]. The probability vector is updated by learning from the best solution of the current population $s_i^{cbest}$ at a learning rate $LR$ as follows:

$$v_i = v_i \times (1 - LR) + s_i^{cbest} \times LR$$

In addition, after the probability vector is updated at the learning rate $LR$, if the $i^{th}$ asset is selected in the best solution but it is not selected in the worst solution or vice versa (i.e, $s_i^{cbest} \neq s_i^{cworst}$) then the $i^{th}$ asset has a higher probability of being selected or not selected. Hence, the probability vector is updated by a negative learning rate $NEG\_LR$ in order to move away from bad solutions i.e. learn from the bad individuals. When $s_i^{cbest} \neq s_i^{cworst}$, it is updated in the same way as PBIL in [58] as follows:

$$v_i = v_i \times (1 - NEG\_LR) + s_i^{cbest} \times NEG\_LR$$

*3.2.6 Mutation of the probability vector*

One of the factors to consider in designing the model in the population-based approach is to find an effective way to generate offspring. The approximate optimality principle [23] assumes that good solutions tend to have similar structure. This assumption is reasonable for many real-world problems. Based on this assumption, an ideal offspring generator aims to produce a solution which is close to the best solutions found so far in the hope that the resultant solution will be not far from the best solution and fall into a promising area of the search space [59].

At each itearation of the evolution, each dimension of the probability vector ($\upsilon$) is updated according to a certain mutation probability ($MP$). By taking into account of the balance between the exploitation and exploration of the search space, we adopt a new partially guided mutation. It gives an equal chance to mutate the probability vector ($\upsilon$) either randomly or based on the global best solution at a mutation rate $MR$ (i.e., guided mutation). The aim is to strike a balance between exploiting good structures in the best solutions and exploring other area of the search space. The pseudocode of the guided mutation is decribed in Figure-4.

In PBILDE, the probability vector ($\upsilon$) in the main evaluation is maintained by the update and mutation based on the best and worst individuals in the population. It is then utilized to influence the selection of assets in the next generation of portfolios. The proportion of the asset is generated by DE offspring generation scheme, as explained next.

**Pseudocode: Partially Guided Mutation**

```
for i := 1 to N do
    if rand(0,1] < MP \\MP: mutation probability
        if rand(0,1] < 0.5
            r = Rand[0,1]
            v_i = v_i × (1 − MR) + r × MR
        else
            v_i = s_i^cbest
        end if
    end if
end for
```

**Fig. 4** Pseudocode of Partially Guided Mutation

### *3.2.7 DE Offspring Generation*

The offspring generation scheme in PBILDE works with a population of solutions evolved during evolutions. The population of the next generation, $P^{g+1}$, is created based on the current population of the generation $P^g$ with $NP$ individuals (portfolios). It first generates a trial population $\overline{P}^{g+1}$. Each individual trial portfolio $\overline{p}_j^{g+1}$ contains two vectors:

$$\overline{w}_{j,i}^{g+1},\ \ j \in \{1,...,NP\};\ i \in \{1,...,N\}$$
$$\overline{s}_{j,i}^{g+1},\ \ j \in \{1,...,NP\};\ i \in \{1,...,N\}$$

where $\overline{w}_{j,i}$ denotes the proportion of the $i^{th}$ asset in the $j^{th}$ portfolio and $\overline{s}_{j,i}$ denotes whether the $i^{th}$ asset in the $j^{th}$ portfolio is selected or not.

A trial population is generated as described in Figure-5. For each trial portfolio, if the $i^{th}$ asset is selected then the weights of $i^{th}$ asset is generated by the mutation and crossover operations. Firstly, three mutually different indexes, $r_1$, $r_2$ and $r_3$, which are also different from the index $j$ of the current trial portfolio $\overline{p}_j^{g+1}$, are randomly selected from the parent population. The indexes $r_1$, $r_2$ and $r_3$ are randomly selected for each trial vector in the trial population.

In the mutation operation, the difference between two of the randomly selected vectors ($r_1$ and $r_2$) from the current population is multiplied by an amplification factor, $F$, and it is added to the third randomly selected vector ($r_3$) from the current population.

The binary crossover is performed to yield the trial vector. The crossover probability $CR$ represents the probablility of mutating the value of the trial vector. The condition $i == r'$ is to ensure that at least one element of the trial vector is different compared to the elements of the parent vector from the current generation. Similar to the initialization process, if the trial solution generated violate the constraints in the model, the constraint handling scheme (see Section-3.2.8) is applied.

**Pseudocode: Generate Trial Population**

```
for j := 1 to NP do
    r' := Rand[1,N]
    for i := 1 to N do
        Randomly select r1, r2, r3 ∈ {1,...,NP},
        r1 ≠ r2 ≠ r3 ≠ j
        if rand(0,1] < v_i
            s̄_{j,i}^{g+1} = 1
            if rand(0,1] < CR OR i == r'
                w̄_{j,i}^{g+1} = w_{r3,i}^g + F × (w_{r1,i}^g − w_{r2,i}^g)
            else
                w̄_{j,i}^{g+1} = w_{j,i}^g
            end if
        end if
    end for
end for
```

**Fig. 5** Pseudocode of Generating Trial Population

The population of the next generation $P^{g+1}$ is selected from the current population $P^g$ and the trial population $\overline{P}^{g+1}$. Each individual of the trial population is compared with the corresponding individual of the current populaion. PBILDE adopts the greedy selection in DE [47]. Under the greedy criterion, the better individual with the lower fitness value becomes a member individual of the next generation's population:

$$p_j^{g+1} = \begin{cases} \overline{p}_j^{g+1} & \textbf{if } f(\overline{p}_j^{g+1}) < f(p_j^g) \\ p_j^g & \textbf{otherwise} \end{cases}$$

*3.2.8 Constraint Handling*

During the population sampling, each constructed individual must be repaired if the representative portfolio does not satisfy the constraints of the problem. If the number of the selected assets is smaller or larger than $K$, then a repair operator selects or deletes an asset by using a heuristic which prioritizes the assets [13]. The idea is that the asset with a higher expected return and a lower covariance with other assets is believed to have higher chances to be in the best portfolio. The repair process continues untill the number of assets in a portfolio equals $K$, i.e. it satisfies the cardinality constraints (Eq-(10)).

The budget constraint in (Eq-(9)) is satisfied by firstly normalizing the weights: $w_i = w_i / \sum_{j=1}^{N} w_j$ over those assets selected based on the probability vector $\upsilon$. Moreover, the bounding constraint in Eq-(11) requires the proportion of asset $i$ to be in the range $[\epsilon_i,\ \delta_i]$. If the proportion of asset after the normalization violates the upper or lower bound constraints, then it is adjusted as follows:

$$w_i = \begin{cases} w_i + \psi \times (\theta_i/\delta^*) & \textbf{if } \delta_i > w_i \\ \delta_i & \textbf{if } \delta_i < w_i \\ w_i - \phi \times (\varphi_i/\epsilon^*) & \textbf{if } w_i > \epsilon_i \\ \epsilon_i & \textbf{if } w_i < \epsilon_i \end{cases}$$

**where**,

$$\theta_i = \delta_i - w_i$$

$$\varphi_i = w_i - \epsilon_i$$

$$\delta^* = \sum_{i=1}^{N} \theta_i \textbf{ where } \theta_i > 0,$$

$$\psi = \sum_{i=1}^{N} |\theta_i| \textbf{ where } \theta_i < 0,$$

$$\epsilon^* = \sum_{i=1}^{N} \varphi_i \textbf{ where } \varphi_i > 0$$

$$\phi = \sum_{i=1}^{N} |\varphi_i| \textbf{ where } \varphi_i < 0.$$

The same repair strategies have been used in the literature [11][13][58] to adjust the number of assets and the weight of assets in the portfolio. We adopt these strategies for a fair comparison in the experiments.

## 4 Computational Results

In this section, we describe the experiments performed and present computational results on both unconstrained and constrained PSP. The proposed PBILDE hybrid algorithm described in Section 3 has been firstly compared to two other approaches, DE and PBIL.

The DE approach differs from PBILDE in such a way that it performs selection of assets randomly before determining the proportions of assets in the weight vector. In other words, instead of using the probability vector, it makes no effort to learn from the population in order to decide which assets are favorable to be included.

The PBIL approach adopted in our experiment is originally proposed by Xu et al [58]. Xu et al [58] proposed a hybrid algorithm called PBIL_CCPS by integrating a PBIL and a continuous PBIL for the constrained PSP. It first builds a probabilistic model about the distribution of good individuals in the search space and then samples a new generation of population using the probablistic model. It maintains three vectors, a probability vector, a mean vector and a standard deviation vector, to learn from the previous generation. Like PBIL in [58], our adapted PBIL uses the same three vectors, probability vector, the mean and standard deviation vectors, and allocates a random proportion for the selected asset by Gaussian distribution. Unlike Xu et al [58], our PBIL approach with the archive of the best individuals (the elite) replaces the $M$ worst solutions of the current populaion with the $M$ global best solutions. Moreover, we introduce a partially guided mutation to exploit the information obtained during the evolution about the search space.

All three algorithms (PBILDE, PBIL and DE) in our study are applied with the elitism and partially guided mutation to demonstrate the effectiveness and efficiency of the hybrid PBILDE against the PBIL and DE with the same settings.

The proposed PBILDE has also been compared to a number of state-of-the-art approaches in the literature using the same evaluation methods to demonstrate the effectiveness of the hybrid algorithm for both the constrained and unconstrained PSP. All of our experiments are coded in C# and run on a core2duo with a 3.16GHz processor and 2GB RAM. The experimental results obtained for each algorithm are the average of 20 runs.

### 4.1 DataSets

A test data for the portfolio optimization problems from the OR-library [5][6] is used to evaluate the performance of the algorithms described above. These datasets contain the estimated returns and the covariance matrix of five different stock market indices: Hang Seng in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan. For each set of the test data, the number of assets $N$ is 31, 85, 89, 98 and 225, respectively. In the current literature of portfolio selection problem, this set of dataset has been widely adapted and tested, and is recognized as the benchmark to evaluate computational algorithms. All information of the dataset itself and their best known solutions can be accessed online [6].

### 4.2 Parameter Settings

In the parameter settings, the value of $\lambda$ in the objective function Eq-(5) is set as $\lambda_i = (i-1)/49$ where $i = 1, 2, ..., 50$. For each value of $\lambda$, each algorithm carried out in total $1000N$ fitness evaluations excluding the initializations.

*Unconstrained PSP*:
We set $K = N$, $\epsilon_i = 0$ $(i = 1, ..., N)$, $\delta_i = 1$ $(i = i, ..., N)$ and $NP = 20$ for the unconstrained problem.

For PBIL, the values of the learning rate ($LR$) and the negative learning rate ($NEG\_LR$) are 0.1 and 0.075, respectively. The mutation probability ($MP$) and mutation rate ($MR$) in the partially guided mutation, see Figure-4, are 0.05 and 0.05, respectively. The number of the best $M$ portfolios is set as $NP/4$. The probability of the learning rate of the mean vector and standard deviation vector $PLR$, is linearly increased from the range [0.05,0.4]. The above parameter values are set by referring those in [58] and testing.

In DE, the two parameters $CR$ and $F$ are set as 0.8 and 0.9, respectively, as proposed in [55]. The number of the best $M$ portfolios is set as $NP/4$.

In PBILDE, the values of the learning rate ($LR$) and the negative learning rate ($NEG\_LR$) are the same as those in PBIL. $CR$, $F$ and $M$ values are the same as those in DE. The mutation probability ($MP$) and mutation rate ($MR$) are $1/N$ and 0.05 respectively.

*Constrained PSP*:
In all three algorithms, we set $K = 10$, $\epsilon_i = 0.01$ $(i = 1, ..., N)$ and $\delta_i = 1$ $(i = i, ..., N)$ for constrained test. For PBIL and DE, the population size (NP) is set as 20. For PBILDE, we set mutation probability $MP = 1/N$ and number of population $NP = N/4$.

### 4.3 Performance Evaluation

To compare the efficiency of the algorithms, we compared the efficient frontier obtained by each algorithm with the optimal solutions provided by OR-library [5][6]. We adopt the same approach as previously used by Chang et al [11] to calculate the percentage deviation of each portfolio. It is evaluated by measuring the distance of the obtained efficient portfolio from the optimal efficient frontier. As mentioned in Section-4.2, 50 weighting parameter ($\lambda$) values are used to calculate the efficient frontier of the portfolio selection problem (see Eq-(8)). We maintain a set $V$ which consists of the best solution found for each $\lambda$. Each portfolio in set $V$ is used to evaluate the percentage deviation from the optimal efficient frontier for the unconstrained PSP.

For the constrained PSP, the efficient frontier becomes discontinuous when the cardinality constraint is included in the problem (see Section-2.2). It is therefore considered to be inappropriate to use only set $V$ to evaluate the performance of the algorithms. Another set $H$ is thus defined to store all efficient portfolios during the evolution. For each value of $\lambda$, let $p(\lambda)$ be the current best portfolio found by the algorithm. During the course of iteration, a newly found portfolio is added to $H$ if it is better than $p(\lambda)$. Those portfolios which are dominated by other portfolio in the set are then removed from the set $H$. The resulting set $H$ and set $V$ are used to calculate percentage deviation errors for the constrained PSP.

The same evaluation method of percentage deviation errors in Chang et al [11] has been adopted. Each obtained portfolio in the set $H$ and set $V$ is evaluated by measuring its distance (i.e, horizontally and vertically) from the optimal efficient frontier. The horizontal distance (x) from the efficient frontier is measured by considering the portfolio's expected return as fixed. Similarly, the vertical distance (y) from the efficient frontier is measured by considering the portfolio's risk as fixed. The final percentage deviation error is then measured by taking the minimum of these two values.

## 4.4 Experimental Results

### *4.4.1 Unconstrained PSP*

It has been observed that the population size of the algorithms does not lead to significantly different result for the unconstrained PSP. We therefore set the population size as 20. Table-1 provides the comparison on the results of set $V$ of three algorithms, namely PBILDE, DE and PBIL. PBILDE performed the best and obtained better results on 4 out of 5 datatsets. We can conclude from the results that PBILDE is an efficient algorithm. DE is the second best in three algorithms. By allocating the same number of evaluations to all three algorithms, similar CPU time is required.

| Instance | | | PBILDE | DE | PBIL |
|---|---|---|---|---|---|
| Index | N | | V | V | V |
| Hang Seng | 31 | MPE( %) | **0.0002** | 0.0280 | 0.2385 |
| | | MedPE( %) | **2.63E-06** | 2.81E-06 | 0.0257 |
| | | Time(s) | 109 | **105** | 134 |
| DAX 100 | 85 | MPE( %) | **0.0052** | 0.0089 | 1.1849 |
| | | MedPE( %) | **2.11E-05** | 2.15E-05 | 0.4292 |
| | | Time(s) | **1445** | 1522 | 2103 |
| FTSE 100 | 89 | MPE( %) | 0.0059 | **0.0049** | 0.9813 |
| | | MedPE( %) | 2.11E-06 | **1.98E-06** | 0.0799 |
| | | Time(s) | **1643** | 1898 | 2145 |
| S&P 100 | 98 | MPE( %) | **0.0078** | 0.0094 | 1.2361 |
| | | MedPE( %) | **3.54E-06** | 3.72E-06 | 0.1443 |
| | | Time(s) | **2094** | 2479 | 2700 |
| Nikkei | 225 | MPE( %) | 0.2733 | **0.2503** | 3.7411 |
| | | MedPE( %) | **2.25E-05** | 2.61E-05 | 2.0514 |
| | | Time(s) | **24823** | 28795 | 31903 |

**Table 1** Comparison Results of PBILDE with DE and PBIL for the Unconstrained PSP.

We also compare PBILDE with the results from Chang et al [11] and Xu et al [58] in Table-2, where *MedPE* and *MPE* denote the average values of the obtained median percentage error (MedPE) and mean percentage error (MPE) of set $V$ in 20 runs. The comparison results show that PBILDE can achieve better solution in most instances.

| Instance | | | PBILDE | Chang-GA | Chang-TS | Chang-SA | Xu-GA | Xu-PSO | Xu-PBIL |
|---|---|---|---|---|---|---|---|---|---|
| Index | N | | V | V | V | V | V | V | V |
| Hang Seng | 31 | MPE( %) | **0.0002** | 0.0202 | 0.8973 | 0.1129 | 0.0191 | 0.1422 | 0.0003 |
| | | MedPE( %) | **2.63E-06** | 0.0165 | 1.0718 | 0.016 | 0.0166 | 1.07E-05 | 1.24E-05 |
| DAX 100 | 85 | MPE( %) | 0.0052 | 0.0136 | 3.5645 | 0.0394 | 0.035 | 1.1044 | **0.0023** |
| | | MedPE( %) | **2.11E-05** | 0.0123 | 2.7816 | 0.0033 | 0.0124 | 4.77E-5 | 3.51E-05 |
| FTSE 100 | 89 | MPE( %) | **0.0059** | 0.0063 | 3.2731 | 0.2012 | 0.0109 | 1.143 | 0.0186 |
| | | MedPE( %) | **2.11E-06** | 0.0029 | 3.0238 | 0.0426 | 0.002 | 0.0084 | 2.45E-05 |
| S&P 100 | 98 | MPE( %) | **0.0078** | 0.0084 | 4.428 | 0.2158 | 0.043 | 2.0249 | 0.0137 |
| | | MedPE( %) | **3.54E-06** | 0.0085 | 4.278 | 0.0142 | 0.0085 | 0.5133 | 2.85E-05 |
| Nikkei | 225 | MPE( %) | 0.2733 | **0.0085** | 15.9163 | 1.7681 | 0.3715 | 8.1781 | 0.0606 |
| | | MedPE( %) | **2.25E-05** | 0.0084 | 14.2668 | 0.8107 | 0.0068 | 4.7023 | 2.69E-05 |

**Table 2** Comparison results of PBILDE with Chang et al [11] and Xu et al [58] for the Unconstrained PSP.

### *4.4.2 Constrained PSP*

In this section, before we compare the proposed PBILDE to other heuristic approaches, we outline a number of tests performed to decide the value of population size assignment and to adopt the new partially guided mutation and elitist scheme in PBILDE. Different population sizes are tested for the constrained PSP and the results are shown in Table-3. Unlike for the unconstrained PSP where the setting of population size does not lead to different perfomance, results show that for constrained PSP, setting population size ($NP$) as $N/4$ is better than both 20 and $2N$. It obtains more efficient points in set $H$ at a much higher computation time.

We tested the role of partially guided mutation (PGM) in PBILDE. The results shown in Table-4 are the average results of 20 runs as mentioned above. It is clear from the Table-4 that adopting the partially guided mutation in PBILDE contributes to better solution quality. We also tested the contribution of elitist strategy in PBILDE. Given the result shown in Table-5, we would conclude that it is an advantage to maintain the archive scheme in PBILDE.

Table-6 provides the comparison results of PBILDE, PBIL and DE with population size $NP = N/4$. PBILDE outperforms DE and PBIL in all instances. Results show that PBILDE uses up less CPU time on larger problems compared against PBIL and DE. Furthermore, the lack of consideration on an efficient selection of assets in DE penalizes the algorithm performance. Both PBIL and PBILDE use a probability vector in determing the selection of assets in a portfolio. Experimental results of PBIL compared with PBILDE show that the use of the probablistic model with the mean and standard deviation vectors in determining the proportions of the assets is not as effective as employing the DE within PBILDE. Figure-6 shows the comparison of the effi-

| Instance | | | NP = 20 | | NP = 2N | | NP=N/4 | |
|---|---|---|---|---|---|---|---|---|
| Index | N | | V | H | V | H | V | H |
| Hang Seng | 31 | avg MPE( %) | 1.1235 | 0.8865 | **1.1101** | 0.8925 | 1.1431 | **0.6196** |
| | | avg MedPE( %) | 1.2283 | 1.1050 | **1.2230** | 1.1060 | 1.2390 | **0.4712** |
| | | Number of EF points | 2923 | | 2165 | | **6367** | |
| | | Time(s) | **60** | | 99 | | 113 | |
| DAX 100 | 85 | avg MPE( %) | 2.4481 | 1.7449 | **2.4101** | 1.6597 | 2.4251 | **1.5433** |
| | | avg MedPE( %) | 2.5922 | 1.4291 | **2.5866** | 1.3945 | **2.5866** | **1.0986** |
| | | Number of EF points | 3347 | | 2021 | | **3378** | |
| | | Time(s) | **526** | | 818 | | 1358 | |
| FTSE 100 | 89 | avg MPE( %) | 1.0322 | 1.0177 | **0.9460** | **0.7204** | 0.9706 | 0.8234 |
| | | avg MedPE( %) | 1.0841 | 0.5443 | **1.0840** | 0.5203 | **1.0840** | **0.5134** |
| | | Number of EF points | 2919 | | 1574 | | **2957** | |
| | | Time(s) | **590** | | 962 | | 1496 | |
| S&P 100 | 98 | avg MPE( %) | 1.9144 | 1.7338 | **1.5688** | **1.2380** | 1.6386 | 1.3902 |
| | | avg MedPE( %) | 1.1617 | 0.8556 | **1.1594** | 0.9085 | 1.1692 | **0.7303** |
| | | Number of EF points | 4546 | | 2608 | | **4570** | |
| | | Time(s) | **762** | | 1014 | | 1901 | |
| Nikkei | 225 | avg MPE( %) | 0.6314 | 0.5198 | 0.5995 | 0.4604 | **0.5972** | **0.3996** |
| | | avg MedPE( %) | 0.6017 | 0.5233 | 0.5903 | 0.5262 | **0.5896** | **0.4619** |
| | | Number of EF points | 3967 | | 2560 | | **4000** | |
| | | Time(s) | **4955** | | 8070 | | 14918 | |
| Average of all instances | | avg MPE( %) | | 1.1805 | **1.3269** | 0.9942 | 1.3549 | **0.9552** |
| | | avg MedPE( %) | 1.3336 | 0.8914 | **1.3287** | 0.8911 | 1.3337 | **0.6551** |

**Table 3** Comparison results of PBILDE with different population size (NP) for the constrained PSP.

| Instance | | | PBILDE-with PGM | | PBILDE-without PGM | |
|---|---|---|---|---|---|---|
| Index | N | | V | H | V | H |
| Hang Seng | 31 | MPE( %) | **1.1431** | **0.6196** | 1.1444 | 0.7609 |
| | | MedPE( %) | **1.2390** | **0.4712** | 1.2402 | 0.7284 |
| | | Number of EF points | **6367** | | 6215 | |
| | | Time(s) | 113 | | **111** | |
| DAX 100 | 85 | MPE( %) | **2.4251** | **1.5433** | 2.4701 | 1.7668 |
| | | MedPE( %) | **2.5866** | **1.0986** | 2.6003 | 1.4315 |
| | | Number of EF points | **3378** | | 3321 | |
| | | Time(s) | 1358 | | **1332** | |
| FTSE 100 | 89 | MPE( %) | **0.9706** | **0.8234** | 1.0431 | 1.0258 |
| | | MedPE( %) | **1.0840** | **0.5134** | 1.0841 | 0.5213 |
| | | Number of EF points | **2957** | | 2937 | |
| | | Time(s) | 1496 | | **1453** | |
| S&P 100 | 98 | MPE( %) | **1.6386** | **1.3902** | 1.8451 | 1.7740 |
| | | MedPE( %) | **1.1692** | **0.7303** | 1.1595 | 0.8161 |
| | | Number of EF points | **4570** | | 4240 | |
| | | Time(s) | 1901 | | **1822** | |
| Nikkei | 225 | MPE( %) | **0.5972** | **0.3996** | 0.6142 | 0.4476 |
| | | MedPE( %) | **0.5896** | **0.4619** | 0.5965 | 0.4959 |
| | | Number of EF points | **4000** | | 3832 | |
| | | Time(s) | 14918 | | **14327** | |

**Table 4** Comparison results of PBILDE with and without partially guided mutation.

cient frontiers of PBILDE, PBIL and DE for the constrained PSP. We also evaluated the performance of the algorithms by the average fitness of the efficient portfolios obtained throughout the evolution. The fitness of the algorithm in a certain generation is measured

| Instance | | | PBILDE-with elitism | | PBILDE-without elitism | |
|---|---|---|---|---|---|---|
| Index | N | | V | H | V | H |
| Hang Seng | 31 | MPE( %) | **1.1431** | **0.6196** | 1.1241 | 0.7521 |
| | | MedPE( %) | **1.2390** | **0.4712** | 1.2410 | 0.7612 |
| | | Number of EF points | **6367** | | 6215 | |
| | | Time(s) | 113 | | **102** | |
| DAX 100 | 85 | MPE( %) | **2.4251** | **1.5433** | 2.4989 | 1.7300 |
| | | MedPE( %) | **2.5866** | **1.0986** | 2.6026 | 1.2384 |
| | | Number of EF points | **3378** | | 2817 | |
| | | Time(s) | 1358 | | **1232** | |
| FTSE 100 | 89 | MPE( %) | **0.9706** | **0.8234** | 1.0515 | 1.1300 |
| | | MedPE( %) | **1.0840** | **0.5134** | 1.0841 | 0.5500 |
| | | Number of EF points | **2957** | | 2790 | |
| | | Time(s) | 1496 | | **1333** | |
| S&P 100 | 98 | MPE( %) | **1.6386** | **1.3902** | 1.7889 | 1.7387 |
| | | MedPE( %) | **1.1692** | **0.7303** | 1.1609 | 0.8343 |
| | | Number of EF points | **4570** | | 4177 | |
| | | Time(s) | 1901 | | **1702** | |
| Nikkei | 225 | MPE( %) | **0.5972** | **0.3996** | 0.6125 | 0.4480 |
| | | MedPE( %) | **0.5896** | **0.4619** | 0.5961 | 0.4930 |
| | | Number of EF points | **4000** | | 3927 | |
| | | Time(s) | 14918 | | **11735** | |

**Table 5** Comparison results of PBILDE with and without elitism.

by the average mean percentage error deviation of the obtained efficient portfolios from the unconstrained efficient frontier (UCEF). The performance of the algorithms is provided in Figure-7. In all figures, the graphs represent the average of the mean percentage error in 20 runs. The results clearly demonstrate that our proposed algorithm PBILDE significantly outperforms DE and PBIL on all problems tested.

| Instance | | | PBILDE | | DE | | PBIL | |
|---|---|---|---|---|---|---|---|---|
| Index | N | | V | H | V | H | V | H |
| Hang Seng | 31 | MPE( %) | **1.1431** | **0.6196** | 1.2150 | 1.1932 | 1.3894 | 1.3737 |
| | | MedPE( %) | 1.2390 | **0.4712** | **1.2331** | 1.2807 | 1.5780 | 1.5267 |
| | | Time(s) | 113 | | **79** | | 95 | |
| DAX 100 | 85 | MPE( %) | **2.4251** | **1.5433** | 3.3077 | 2.9670 | 2.5129 | 2.9245 |
| | | MedPE( %) | 2.5866 | **1.0986** | 2.7410 | 2.5293 | **2.5850** | 2.6648 |
| | | Time(s) | 1358 | | **1274** | | 1478 | |
| FTSE 100 | 89 | MPE( %) | **0.9706** | **0.8234** | 1.3651 | 1.6203 | 1.3190 | 2.0282 |
| | | MedPE( %) | **1.0840** | **0.5134** | 1.0975 | 0.9832 | 1.1204 | 1.2599 |
| | | Time(s) | **1496** | | 1542 | | 1589 | |
| S&P 100 | 98 | MPE( %) | **1.6386** | **1.3902** | 3.2008 | 3.2170 | 2.4722 | 3.1763 |
| | | MedPE( %) | **1.1692** | **0.7303** | 1.5970 | 1.4973 | 1.2096 | 1.3810 |
| | | Time(s) | **1901** | | 1943 | | 1992 | |
| Nikkei | 225 | MPE( %) | **0.5972** | **0.3996** | 1.8934 | 2.2053 | 0.7554 | 0.8086 |
| | | MedPE( %) | **0.5896** | **0.4619** | 1.6428 | 1.7624 | 0.6592 | 0.6864 |
| | | Time(s) | **14918** | | 18327 | | 24806 | |
| Average of all instances | | avg MPE( %) | **1.3549** | **0.9552** | 2.1964 | 2.2406 | 1.6898 | 2.0623 |
| | | avg MedPE( %) | **1.3337** | **0.6551** | 1.6623 | 1.6106 | 1.4304 | 1.5038 |

**Table 6** Comparison results of PBILDE with population size(NP) = N/4 against DE and PBIL for the constrained PSP.

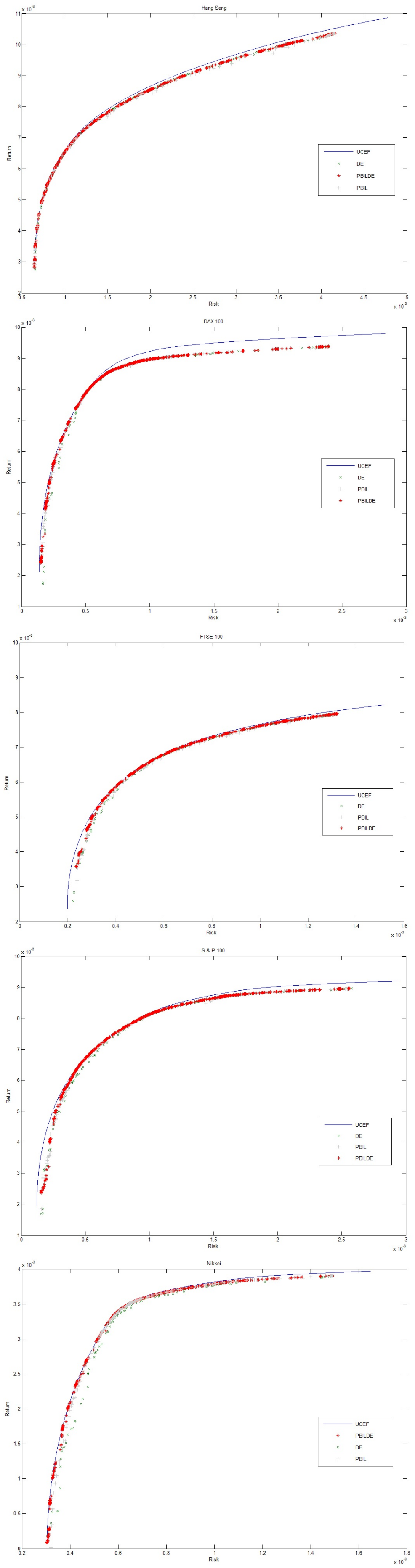


**Fig. 6** Comparison of heurisitc efficient frontiers for constrained PSP.

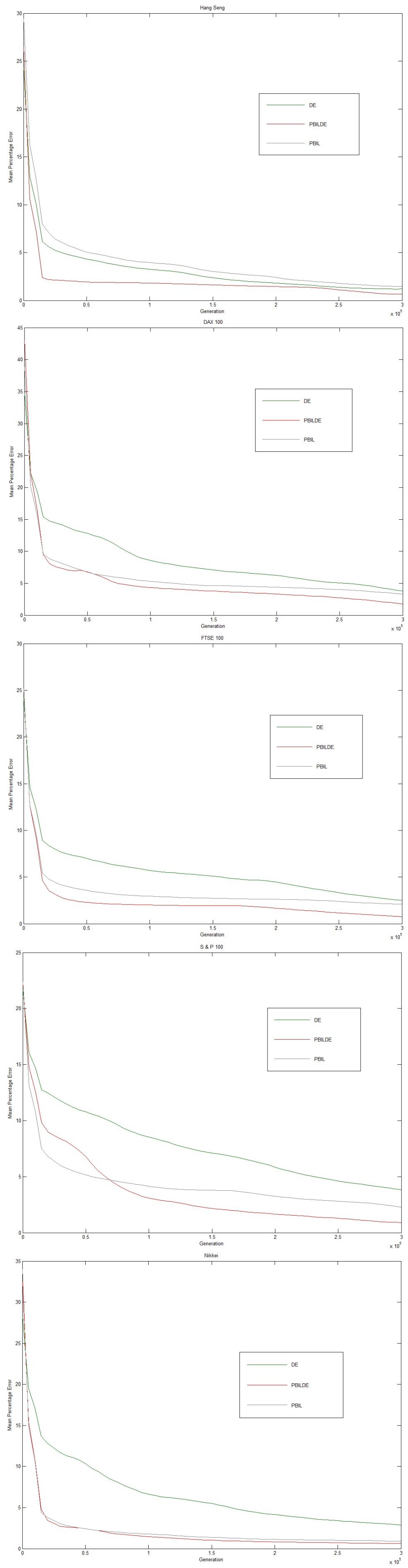


**Fig. 7** Mean performance of the algorithms for constrained PSP.

Chang et al [11] present three heurisitc algorithms based on GA, SA and TS for the constrained PSP and report GA performs better than SA and TS. Xu et al [58] also present a hybrid algorithm (PBIL_CCPS) and report it performs better than GA and PSO. We therefore compare PBILDE with the GA proposed by Chang et al [11] and PBIL_CCPS from Xu et al [58] for the constrained PSP. Both Chang et al [11] and Xu et al [58] adopted the CCMV model described in Section-2.2. The comparison results in Table-7 show that PBILDE outperforms GA and PBIL_CCPS in most instances.

| Instance | | | PBILDE | | Chang-GA [11] | | Xu-PBIL_CCPS [58] | |
|---|---|---|---|---|---|---|---|---|
| Index | N | | V | H | V | H | V | H |
| Hang Seng | 31 | MPE( %) | 1.1431 | **0.6196** | **1.0974** | 0.9457 | 1.1026 | 0.8472 |
| | | MedPE( %) | 1.2390 | **0.4712** | **1.2181** | 1.1819 | 1.2190 | 1.1013 |
| | | Number of EF points | **6367** | | 1317 | | 1540 | |
| DAX 100 | 85 | MPE( %) | **2.4251** | **1.5433** | 2.5424 | 1.9515 | 2.5163 | 2.0781 |
| | | MedPE( %) | 2.5866 | **1.0986** | **2.5466** | 2.1262 | 2.5739 | 2.2783 |
| | | Number of EF points | **3378** | | 1270 | | 1933 | |
| FTSE 100 | 89 | MPE( %) | **0.9706** | 0.8234 | 1.1076 | 0.8784 | 0.9960 | **0.7658** |
| | | MedPE( %) | **1.0840** | 0.5134 | 1.0841 | 0.5938 | 1.0841 | **0.4132** |
| | | Number of EF points | **2957** | | 1482 | | 1638 | |
| S&P 100 | 98 | MPE( %) | **1.6386** | **1.3902** | 1.9328 | 1.7157 | 2.2320 | 1.6340 |
| | | MedPE( %) | 1.1692 | **0.7303** | 1.2244 | 1.1447 | **1.1536** | 0.8453 |
| | | Number of EF points | **4570** | | 1560 | | 2177 | |
| Nikkei | 225 | MPE( %) | **0.5972** | **0.3996** | 0.7961 | 0.6431 | 1.0017 | 0.6451 |
| | | MedPE( %) | 0.5896 | **0.4619** | 0.6133 | 0.6062 | **0.5854** | 0.5596 |
| | | Number of EF points | **4000** | | 1823 | | 1468 | |
| Average of all instances | | avg MPE( %) | **1.3549** | **0.9552** | 1.4953 | 1.2269 | 1.5697 | 1.1940 |
| | | avg MedPE( %) | 1.3337 | **0.6551** | 1.3373 | 1.1306 | **1.3232** | 1.0395 |

**Table 7** Comparison results of PBILDE against other existing algorithms for the constrained PSP.

Various models have been proposed in the literature to solve the constrained PSP, where different variable definitions, objective functions, heuristic techniques, benchmarks and evaluation criteria have been employed. Therefore, it is very difficult, if not impossible, to conduct a fair comparison on different modelling approaches. For the completeness, we next provide the comparisons of our PBILDE against those of different approaches in Gaspero et al [22] and Woodside-Oriakhi et al [56] who use the OR-library instances with the same set of constraints.

Gaspero et al [22] present a hybrid technique (SD+QP) which combines local search metaheuristics and the quadratic programming (QP) procedure. In their work, they also reimplement the hybrid method based on a Hopfield neural network, originally proposed by Fernandez et al [19], and calculate the mean percentage deviation in set $H$. We compare PBILDE with this SD+QP approach [22] and the results are shown in Table-8. The comparison results show that PBILDE outperforms the SD+QP approach by Gaspero et al [22]. As reported in Table-8, the neural network approach by Ferandez et al [19] performs better than PBILDE in 3 out of 5 instances. However, PBILDE is better with regard to the overall average percentage error of all instances.

| Instance | | | PBILDE | Gaspero-SD+QP[22] | Fernandez-NN [19] |
|---|---|---|---|---|---|
| Index | N | | H | H | H |
| Hang Seng | 31 | MPE( %) | 0.6196 | 0.7000 | **0.3800** |
| DAX 100 | 85 | MPE( %) | 1.5433 | 2.9300 | **1.1300** |
| FTSE 100 | 89 | MPE( %) | **0.8234** | 1.9700 | 1.2500 |
| S&P 100 | 98 | MPE( %) | **1.3902** | 4.1000 | 2.8000 |
| Nikkei | 225 | MPE( %) | 0.3996 | 0.3000 | **0.3600** |
| Average all instances | | MPE( %) | **0.9552** | 2.000 | 1.1840 |

**Table 8** Comparison results of PBILDE against Gaspero et al [22] and Fernandez et al [19] for the constrained PSP.

Recently, Woodside-Oriakhi et al [56] propose a GA with subset optimization for the constrained PSP. The constrained portfolio selection problem was reformulated by relaxing constraint Eq(2), where the expected return may vary within 10% of the desired return range. The search of the algorithm is thus more flexible to explore a wider area of the search space of the relaxed problem. The same mechanism has been applied to develop a SA and TS. The weighted sum approach as described in Eq(5) approximates the constrained EF by accumulating the set of points which are not evenly distributed along the return axis whereas the Woodside-Oriakhi et al approach approximates the constrained EF by accumulating the set of efficient points which are evenly distributed among 50 values of the expected return in the prespecifed range.

The comparison results are shown in Table-9. The GA by Woodside-Oriakhi et al outperforms in all instances except the Hang Seng dataset. PBILDE outperforms the SA by Woodside-Oriakhi et al [56] in most instances. PBILDE is competitive to the TS by Woodside-Oriakhi et al [56]. However, the maximum and minimum precentage error results show that PBILDE results are stable compared to those of the three algorithms presented by Woodside-Oriakhi et al [56].

## 5 Conclusions

In this work, we have proposed an efficient and effective hybrid algorithm (PBILDE) to solve the portfolio

| Instance | | | PBILDE | Woodside-Oriakhi-GA | Woodside-Oriakhi-TS | Woodside-Oriakhi-SA |
|---|---|---|---|---|---|---|
| Index | N | | H | H | H | H |
| Hang Seng | 31 | MPE( %) | **0.6196** | 0.8501 | 0.8234 | 1.0589 |
| | | MedPE( %) | 0.4712 | 0.5873 | **0.3949** | 0.5355 |
| | | Minimum | 0.2816 | **0.0036** | 0.0068 | 0.0349 |
| | | Maximum | **0.6768** | 2.9034 | 4.6096 | 4.6397 |
| DAX 100 | 85 | MPE( %) | 1.5433 | 0.7740 | **0.7190** | 1.0267 |
| | | MedPE( %) | 1.0986 | **0.2400** | 0.4298 | 0.8682 |
| | | Minimum | 0.7537 | **0.0000** | 0.0149 | 0.0278 |
| | | Maximum | **1.6804** | 4.6811 | 2.7770 | 4.4123 |
| FTSE 100 | 89 | MPE( %) | 0.8234 | **0.1620** | 0.3930 | 0.8952 |
| | | MedPE( %) | 0.5134 | **0.0820** | 0.2061 | 0.3944 |
| | | Minimum | 0.4359 | **0.0000** | 0.0019 | 0.0230 |
| | | Maximum | 0.8695 | **0.7210** | 3.4570 | 10.2029 |
| S&P 100 | 98 | MPE( %) | 1.3902 | **0.2922** | 1.0358 | 3.0952 |
| | | MedPE( %) | 0.7303 | **0.1809** | 1.0248 | 2.1064 |
| | | Minimum | 0.4816 | **0.0007** | 0.0407 | 0.8658 |
| | | Maximum | **1.5726** | 1.6295 | 3.0061 | 8.6652 |
| Nikkei | 225 | MPE( %) | 0.3996 | **0.3353** | 0.7838 | 1.1193 |
| | | MedPE( %) | 0.4619 | **0.3040** | 0.6526 | 0.6877 |
| | | Minimum | 0.3739 | 0.0180 | **0.0085** | 0.0113 |
| | | Maximum | **0.4965** | 1.0557 | 2.6082 | 3.9678 |
| Average all instances | - | MPE( %) | 0.9552 | **0.4827** | 0.7510 | 1.4391 |
| | | MedPE( %) | 0.6550 | **0.2788** | 0.5416 | 0.9184 |
| | | Minimum | 0.4653 | **0.0045** | 0.0146 | 0.1926 |
| | | Maximum | **1.0591** | 2.1981 | 3.2916 | 6.3776 |

**Table 9** Comparison results of Hybrid Algorithm(PBILDE) against Woodside-Oriakhi et al [56] for the constrained PSP.

selection problem with cardinality, floor and ceiling constraints. The proposed PBILDE algorithm hybridizes a PBIL and a DE to explore and exploit the complex and constrained search space of the problem concerned. It also adopts a partially guided mutation and an elitist strategy to enhance the evolution over the search space. For the unconstrained problem, PBILDE outperforms in almost all instances compared against DE and PBIL with similiar or higher computational expenses. It also outperforms other existing approaches in the literature for the constrained problem. Results justify the effectiveness of the elitism and partially guided mutation in PBILDE. The comparison results against the PBIL, DE, as well as several algorithms in the literature again show that the proposed hybrid algorithm is highly competitive in most cases. The proposed PBILDE algorithm may be further extended to solve different PSP models with various constraints in our future work.

## References


1. Anagnostopoulos K, Mamanis G (2011) The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. Expert Systems with Applications 38(11):14,208–14,217
2. Arriaga J, Valenzuela-Rendón M (2012) Steepest ascent hill climbing for portfolio selection. In: Applications of Evolutionary Computation, Lecture Notes in Computer Science, vol 7248, Springer Berlin / Heidelberg, pp 145–154, URL `http://dx.doi.org/10.1007/978-3-642-29178-4_15`, 10.1007/978-3-642-29178-4_15
3. Baluja S (1994) Population-based incremental learning. Tech. Rep. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pa.
4. Baluja S, Caruana R (1995) Removing the genetics from the standard genetic algorithm. In: Machine learning: proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, July 9-12, 1995, Morgan Kaufmann Publishers, Inc., pp 38–46
5. Beasley J (1990) Or-library: Distributing test problems by electronic mail. Journal of the Operational Research Society 41(11):1069–1072
6. Beasley JE (1999) Or library dataset. Available from: `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html`, [Online; accessed 02-Oct-2011]
7. Bertsimas D, Shioda R (2009) Algorithm for cardinality-constrained quadratic optimization. Computational Optimization and Applications 43(1):1–22
8. Best M, Hlouskova J (2000) The efficient frontier for bounded assets. Mathematical Methods of Operations Research 52(2):195–212
9. Bienstock D (1996) Computational study of a family of mixed-integer quadratic programming problems. Mathematical Programming 74(2):121–140
10. Busetti F (2005) Metaheuristic approaches to realistic portfolio optimization. Master's thesis, University of South Aferica, URL `http://arxiv.org/ftp/cond-mat/papers/0501/0501057.pdf`
11. Chang T, Meade N, Beasley J, Sharaiha Y (2000) Heuristics for cardinality constrained portfolio optimisation. Computers and Operations Research 27(13):1271–1302
12. Crama Y, Schyns M (2003) Simulated annealing for complex portfolio selection problems. European Journal of operational research 150(3):546–571
13. Cura T (2009) Particle swarm optimization approach to portfolio optimization. Nonlinear Analysis: Real World Applications 10(4):2396–2406
14. Das S, Suganthan P (2011) Differential evolution: A survey of the state-of-the-art. Evolutionary Computation, IEEE Transactions on 15(1):4 –31, DOI 10.1109/TEVC.2010.2059031
15. Di Tollo G, Roli A (2006) Metaheuristics for the portfolio selection problem. Technical Report R-

2006-005, Dipartimento di Scienze, Universita ŞG DŠAnnunzioŤ Chieti–Pescara 200
16. Ehrgott M, Klamroth K, Schwehm C (2004) An mcdm approach to portfolio optimization. European Journal of Operational Research 155(3):752–770
17. Ellison HMLRMI EFD, Mitra G (1999) A Fortran Based Mathematical Programming System, FortMP. FortMP, Brunel University, UK and NAG Ltd., Oxford, UK
18. Feoktistov V (2006) Differential evolution: in search of solutions, vol 5. Springer-Verlag New York Inc
19. Fernández A, Gómez S (2007) Portfolio selection using neural networks. Computers and Operations Research 34(4):1177–1191
20. Folly K, Venayagamoorthy G (2009) Effects of learning rate on the performance of the population based incremental learning algorithm. In: Neural Networks, 2009. International Joint Conference on, IEEE, pp 861–868
21. Fonseca C, Fleming P (1995) An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation 3(1):1–16
22. Gaspero L, Tollo G, Roli A, Schaerf A (2011) Hybrid metaheuristics for constrained portfolio selection problems. Quantitative Finance 11(10):1473–1487
23. Glover F, Laguna M (1998) Tabu search, vol 1. Springer
24. Golmakani H, Fazel M (2011) Constrained portfolio selection using particle swarm optimization. Expert Systems with Applications 38(7):8327–8335
25. Gosling JN T, Tsang E (2004) Population-based incremental learning versus genetic algorithms: iterated prisoners dilemma. Tech. Rep. CSM-401, University of Essex, England, URL `http://dces.essex.ac.uk/research/CSP/finance/papers/GoJiTs-Pbil_vs_GA-csm401_2004.pdf`
26. Jobst N, Horniman M, Lucas C, Mitra G (2001) Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. Quantitative Finance 1:1–13
27. Krink T, Paterlini S (2008) Differential evolution for multiobjective portfolio optimization. Center for Economic Research (RECent) 21, URL `http://ideas.repec.org/p/mod/wcefin/08012.html`
28. Krink T, Paterlini S (2011) Multiobjective optimization using differential evolution for real-world portfolio optimization. Computational Management Science 8(1-2):157–179
29. Li D, Sun X, Wang J (2006) Optimal lot solution to cardinality constrained mean–variance formulation for portfolio selection. Mathematical Finance 16(1):83–101
30. Maringer D (2005) Portfolio management with heuristic optimization, vol 8. Springer Verlag
31. Maringer D (2008) Risk preferences and loss aversion in portfolio optimization. Computational Methods in Financial Engineering pp 27–45
32. Markowitz H (1952) Portfolio selection. Journal of Finance 7(1):77–91
33. Markowitz H (1956) The optimization of a quadratic function subject to linear constraints. Naval Research Logistics Quarterly 3(1-2):111–133
34. Markowitz H, Todd G, Sharpe W (2000) Mean-variance analysis in portfolio choice and capital markets, vol 66. Wiley
35. Merton R (1972) An analytic derivation of the efficient portfolio frontier. Journal of financial and quantitative analysis 7(4):1851–1872
36. Moral-Escudero R, Ruiz-Torrubiano R, Suarez A (2006) Selection of optimal investment portfolios with cardinality constraints. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, IEEE, pp 2382–2388
37. Pang J (1980) A new and efficient algorithm for a class of portfolio selection problems. Operations Research 28(3):754–767
38. Perold A (1984) Large-scale portfolio optimization. Management Science 30(10):1143–1160
39. Schaerf A (2002) Local search techniques for constrained portfolio selection problems. Computational Economics 20(3):177–190
40. Schyns M, Crama P (2001) Modelling financial data and portfolio optimization problems. PhD thesis, Doctoral Thesis, University of Liège, URL `http://orbi.ulg.ac.be/bitstream/2268/11831/1/MSthese.pdf`
41. Sebag M, Ducoulombier A (1998) Extending population-based incremental learning to continuous search spaces. Parallel Problem Solving from Nature-ŮPPSN V Berlin, Germany pp 418–427
42. Shapiro J (2003) The sensitivity of pbil to its learning rate and how detailed balance can remove it. Foundations of Genetic Algorithms 7:115–132
43. Shaw D, Liu S, Kopman L (2008) Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. Optimisation Methods & Software 23(3):411–420
44. Skolpadungket P, Dahal K, Harnpornchai N (2007) Portfolio optimization using multi-objective genetic algorithms. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, IEEE, pp 516–523
45. Soleimani H, Golmakani H, Salimi M (2009) Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regard-

ing sector capitalization using genetic algorithm. Expert Systems with Applications 36(3):5058–5063
46. Storn R, Price K (1995) Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. Rep. TR-95-012, Berkeley, CA, URL http://www.icsi.berkeley.edu/ftp/global/global/pub/techreports/1995/tr-95-012.pdf
47. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11(4):341–359
48. Streichert F, Ulmer H, Zell A (2003) Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In: Operations Research Proceedings 2003, Selected Papers of the International Conference on Operations Research (OR 2003), Springer, pp 3–5
49. Streichert F, Ulmer H, Zell A (2004) Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In: Evolutionary Computation, 2004. CEC2004. Congress on, IEEE, vol 1, pp 932–939
50. Sun J, Zhang Q, Tsang E (2005) De/eda: A new evolutionary algorithm for global optimization. Information Sciences 169(3):249–262
51. Vafashoar R, Meybodi M, MomeniÂăAzandaryani A (2012) Cla-de: a hybrid model based on cellular learning automata for numerical optimization. Applied Intelligence 36:735–748, DOI 10.1007/s10489-011-0292-1, URL http://dx.doi.org/10.1007/s10489-011-0292-1
52. Varian H (1993) A portfolio of nobel laureates: Markowitz, miller and sharpe. The Journal of Economic Perspectives 7(1):159–169
53. Vesterstrom J, Thomsen R (2004) A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Evolutionary Computation, 2004. CEC2004. Congress on, IEEE, vol 2, pp 1980–1987
54. Vielma J, Ahmed S, Nemhauser G (2007) A lifted linear programming branch-and-bound algorithm for mixed integer conic quadratic programs. Manuscript, Georgia Institute of Technology
55. Winker P, Lyra M, Sharpe C (2011) Least median of squares estimation by optimization heuristics with an application to the capm and a multifactor model. Computational Management Science 8(1):103–123
56. Woodside-Oriakhi M, Lucas C, Beasley J (2011) Heuristic algorithms for the cardinality constrained efficient frontier. European Journal of Operational Research 213(3):538–550
57. Xu F, Chen W, Yang L (2007) Improved particle swarm optimization for realistic portfolio selection. In: Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on, IEEE, vol 1, pp 185–190
58. Xu R, Zhang J, Liu O, Huang R (2010) An estimation of distribution algorithm based portfolio selection approach. In: 2010 International Conference on Technologies and Applications of Artificial Intelligence, IEEE, pp 305–313
59. Zhang Q, Sun J, Tsang E (2005) An evolutionary algorithm with guided mutation for the maximum clique problem. Evolutionary Computation, IEEE Transactions on 9(2):192–200