Expert Systems with Applications 39 (2012) 4558–4566

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization

Guang-Feng Deng [a,*], Woo-Tsong Lin [a], Chih-Chung Lo [b]

[a] Department of Management Information Systems, National Chengchi University, 64, Sec. 2, Chihnan Rd., Wenshan Dist, Taipei 116, Taiwan, ROC
[b] Department of Applied Informatics, Fo Guang University, 160, Linwei Rd., Jiaosi, Yilan County 26247, Taiwan, ROC

ARTICLE INFO

Keywords:
Particle swarm optimization
Cardinality constrained portfolio optimization problem
Markowitz mean–variance model
Nonlinear mixed quadratic programming problem

ABSTRACT

This work presents particle swarm optimization (PSO), a collaborative population-based meta-heuristic algorithm for solving the Cardinality Constraints Markowitz Portfolio Optimization problem (CCMPO problem). To our knowledge, an efficient algorithmic solution for this nonlinear mixed quadratic programming problem has not been proposed until now. Using heuristic algorithms in this case is imperative. To solve the CCMPO problem, the proposed improved PSO increases exploration in the initial search steps and improves convergence speed in the final search steps. Numerical solutions are obtained for five analyses of weekly price data for the following indices for the period March, 1992 to September, 1997: Hang Seng 31 in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan. The test results indicate that the proposed PSO is much more robust and effective than existing PSO algorithms, especially for low-risk investment portfolios. In most cases, the PSO outperformed genetic algorithm (GA), simulated annealing (SA), and tabu search (TS).

1. Introduction

Portfolio optimization, which is the allocation of wealth among several assets, is an essential problem in modern risk management. Expected returns and risks are the most important parameters in portfolio optimization problems. Investors generally prefer to maximize returns and minimize risk. However, high returns generally involve increased risk.

The Markowitz mean–variance model, which is among the best models for solving the portfolio selection problem, can be described in terms of the mean return of the assets and the variance of return (risk) between these assets (Markowitz, 1952). The basic model obtains the "efficient frontier", which is the portfolio of assets that achieves a predetermined level of expected return at minimal risk. For every level of desired mean return, this efficiency frontier indicates the best investment strategy.

From a practical perspective, portfolio selection problem consider many constraints of real-world investments, including trading limitations, portfolio size, etc. However, the basic Markowitz mean–variance model does not include cardinality constraints to ensure the investment in a given number of different assets, nor does it include bounding constraints to limit the funds invested in each asset. Although portfolio optimization using the standard Markowitz model is NP-hard, the solution to this problem with a sufficiently small number of variables can be solved by using quadratic programming. The problem becomes much more difficult if the number of variables is increased or if additional constraints such as cardinality constraints are introduced (Maringer & Kellerer, 2003). Such constraints formed nonlinear mixed integer programming problems, which are considerably more difficult to solve than the original problem. Exact solution methods are inadequate. Therefore, proposed heuristic solutions for the portfolio selection problem include evolutionary algorithms, tabu search (TS) simulated annealing (SA) and neural networks (Chang, Meade, Beasley, & Sharaiha, 2000; Chang, Yang, & Chang, 2009; Crama & Schyns, 2003; Fernandez & Gomez, 2007; Maringer & Kellerer, 2003; Soleimani, Golmakani, & Salimi, 2009).

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart in 1995, is based on a psychosocial model of social influence and social learning and has proven effective in many empirical studies (Ali & Kaelo, 2008; Jiang, Hu, Huang, & Wu, 2007; Kennedy, Eberhart, & Shi, 2001; Tsai et al., 2010; Yang, Yuan, Yuan, & Mao, 2007). This study proposes a novel application of PSO, a collaborative population-based meta-heuristic algorithm for the Markowitz mean–variance model, which includes cardinality and bounding constraints, to solve Cardinality Constrained Markowitz Portfolio Optimization problems (CCMPO problems). Due to the many variations of the original PSO, this study first investigated the performance of PSO variants in solving CCMPO problems. The results showed that the constraints cause PSO to stagnate to the local optimum. Therefore, remedies are proposed to avoid stagnation in CCMPO problems. The reflection strategy is to keep desired

* Corresponding author. Tel.: +886 919 300038.
E-mail addresses: deng@nccu.edu.tw (G.-F. Deng), lin@mis.nccu.edu.tw (W.-T. Lin), locc@mail.fgu.edu.tw (C.-C. Lo).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.129

assets in portfolio in the search process. The replacement minimum hold strategy randomly adds assets with the minimum hold weight when assets needed to obtain a new solution are insufficient. The mutation strategy increases the search space.

The performance of the proposed PSO was compared with other variants of PSO and with heuristic algorithms, including genetic algorithm (GA), simulated annealing (SA), and tabu search (TS). Performance was compared using five problems, involving 31-255 dimensions corresponding to weekly data for March, 1992–September, 1997. The test data were obtained from the following indices: Hang Seng 31 in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan. Results show that the proposed PSO is much more robust and effective than existing PSO algorithms in terms of tracing out the efficient frontier accurately, especially in risk-aversion CCMPO problems. Compared to other heuristic algorithms, the proposed PSO obtained better solutions in most test problems.

Following this introduction, Section 2 presents the model formulation for the cardinality constrained portfolio optimization problems, and Section 3 describes the application of PSO for solving this problem. The computational experiment in Section 4 evaluates the PSO model and experimental results. Section 5 presents conclusions and proposes future works.

2. Portfolio optimization problems and particle swarm optimization

This section presents the standard Markowitz portfolio model and demonstrates an efficient frontier calculation. The cardinality constraints are then given for the Markowitz mean–variance model to be solved.

2.1. Portfolio optimization problems

The notation used in this analysis is based on Markowitz mean–variance model for solving the portfolio selection problem. Let N be the number of different assets, u_i be the expected return of asset i $(i = 1, \ldots, N)$, $\sigma_{i,j}$ be the covariance between assets i and j $(i = 1, \ldots, N; j = 1, \ldots, N)$, The decision variables xi represent the proportion $(0 \leqq x_i \leqq 1)$ of the portfolio invested in asset i $(i = 1, \ldots, N)$ and a weighting parameter λ. Using this notation, the standard Markowitz mean–variance model for the portfolio selection problem can be presented as

$$\text{Minimize } \lambda\left[\sum_{i=1}^{N}\sum_{j=1}^{N} x_i \cdot x_j \cdot \sigma_{ij}\right] - (1-\lambda)\left[\sum_{i=1}^{N} x_i \cdot u_i\right] \tag{1}$$

$$\text{subject to } \sum_{i=1}^{N} x_i = 1 \tag{2}$$

$$0 \leqslant x_i \leqslant 1, \quad i = 1, \ldots, N \tag{3}$$

where $\lambda \in [0,1]$ is the risk aversion parameter. The case $\lambda = 0$ represents the maximum expected return for the portfolio (disregarding risk), and the optimal portfolio is the single asset with the highest return. The case $\lambda = 1$ represents the minimal total risk for the selected portfolio (disregarding return), and the optimal portfolio includes numerous assets. The two extremes $\lambda = 0$ and $\lambda = 1$, λ represent the tradeoff between risk and return. Eq. (2) ensures that the sum of the proportions is 1. The equation $\sum_{i=}^{N}\sum_{j=1}^{N} x_i \cdot x_j \cdot \sigma_{ij}$ obtains total variance(risk) , which should be minimized and the equation $\sum_{i=1}^{N} x_i \cdot u_i$ obtains the total portfolio return, which should be maximized.

The portfolio selection problem is a multi-objective optimization problem, and all non-dominated solutions can be used to produce the efficient frontier. Fig. 1 shows the efficient frontier as plotted by varying value λ corresponding to the benchmark Hang Seng 31. To calculate cardinality constraints for the Markowitz Optimal Model, this study used the model formulation presented in (Chang et al., 2000; Fernandez & Gomez, 2007). In addition to the previously defined variables, let K be the desired number of assets in the portfolio, let ε_i be the minimum proportion of the portfolio allocated to asset i $(i = 1,\ldots,N)$ if any of asset i is held, and let δ_i be the maximum proportion allocated to asset i $(i = 1,\ldots,N)$ if any of asset i is held , where $0 \leqq \varepsilon_i \leqq \delta_i \leqq 1$ $(i = 1, \ldots, N)$. In practice ε_i represents a "min-buy" or "minimum transaction level" for asset i, and δ_i limits portfolio exposure to asset i. Zero-one decision variables are as follows:

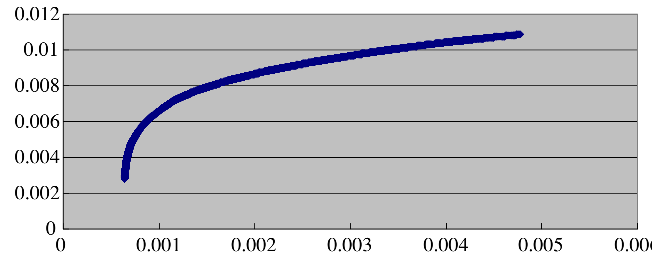

Fig. 1. Standard efficient frontier for Hang Seng 31 dataset.

$$z_i \begin{cases} 1 & \text{if any of asset } i (i = 1 \ldots N) \text{ is held,} \\ 0 & \text{otherwise} \end{cases}$$

The cardinality constrained portfolio optimization problem is

$$\textit{Minimize } \lambda\left[\sum_{i=1}^{N}\sum_{j=1}^{N} x_i \cdot x_j \cdot \sigma_{ij}\right] - (1-\lambda)\left[\sum_{i=1}^{N} x_i \cdot u_i\right] \tag{4}$$

$$\text{subject to } \sum_{i=1}^{N} x_i = 1 \tag{5}$$

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

$$\varepsilon_i z_i \leqslant x_i \leqslant \delta_i z_i, \quad i = 1, \ldots, N \tag{7}$$

$$z_i \in [0, 1], \quad i = 1, \ldots, N \tag{8}$$

Eq. (5) ensures that the sum of the proportions is 1, and Eq. (6) ensures that exactly K assets are held. Eq. (7) ensures that if any of asset i is held $(z_i = 1)$ its proportion w_i must lie between ε_i and δ_i whilst if no asset i is held $(z_i = 0)$, its proportion w_i is zero. Eq. (8) is the integrality constraint.

2.2. Particle swarm optimization

The PSO is a social population-based search algorithm of social influence that learns from its neighborhood. A PSO swarm resembles a population, and a particle resembles an individual. The PSO is initialized with a particle swarm, and each particle position represents a possible solution. The particles fly through the multidimensional search space by dynamically adjusting velocities according to its own experience and that of its neighbors (Clerc, 2006; Kennedy et al., 2001).

At each iteration t, the position $x_{i,j}^{t}$ of the ith particle is updated by a velocity $v_{i,j}^{t+1}$. The position is updated for the next iteration using

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \tag{9}$$

where $x_{i,j}^{t}$ denotes the position of particle i in the dimension j search space at time step t. The position of the particle is changed by adding a velocity $v_{i,j}^{t+1}$ to the current position. The velocity update rule is calculated as

$$v_{i,j}^{t+1} = v_{i,j}^{t} + c_1 \cdot r_1 \cdot (p_{i,j} - x_{i,j}^{t}) + c_2 \cdot r_2 \cdot (p_{g,j} - x_{i,j}^{t}) \tag{10}$$

The rule depends on three components: its current velocity $v_{i,j}^{t}$, the weighted difference vectors $(p_{i,j} - x_{i,j}^{t})$, and $(p_{g,j} - x_{i,j}^{t})$, where $v_{i,j}^{t}$ is the velocity of particle i in dimension $j = 1, \ldots, n$ at time step t, the personal best position, $p_{i,j}$ associated with particle i in dimension j is the best position the particle has visited since the first time step. The global best position $p_{g,j}$ at time step t is the best position discovered by all particles found since the first time step.The r_1 and r_2 are random values in the range [0, 1], sampled from a uniform distribution. These random values introduce a stochastic element to the algorithm. The c_1 and c_2 are positive acceleration coefficients used to scale the contribution of the cognitive and social components, respectively, and are also referred to as trust parameters, where c_1 expresses the confidence of a particle in itself, and c_2 expresses the confidence of a particle in its neighbors. Particles gain strength by cooperation and are most effective when c_1 and c_2 are well-balanced. Low values result in smooth particle trajectories, which allow particles to roam among different regions. High values cause particles to move abruptly from one region to another region.To improve PSO convergence, Shi and Eberhart (1998) proposed a strategy for incorporating inertial weight w as a mechanism for controlling swarm exploitation and exploration by weighting the contribution of the previous velocity. The w control how much the memory of the previous flight direction influences the new velocity. For $w \geqq 1$, velocity increase over time, accelerates to maximum velocity, and the swarm diverges. Particle fail to change direction to move back towards promising areas. For $w \leqq 1$, particles decelerate until their velocity is zero. The velocity update with inertia is given as

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^{t} + c_1 \cdot r_1 \cdot (p_{i,j} - x_{i,j}^{t}) + c_2 \cdot r_2 \cdot (p_{g,j} - x_{i,j}^{t}) \tag{11}$$

The authors in (11) also proposed maximum velocity $v_{\max}$. The $v_{\max}$ was calculated as a fraction δ ($0 < \delta < 1$) of the distance between the bounds of the search space as follows

$$v_{\max,j} = \delta(x_{\max,j} - x_{\min,j}) \tag{12}$$

The PSO with constant inertia w and maximum velocity limitation $v_{\max}$ is referred to here as *Basic-PSO*.

Variations of the original PSO have obtained improved performance. Several PSO variants for comparison purposes were introduced. Fourie and Groenwold (2002) suggested a dynamic inertia weight and maximum velocity reduction as in (13). The inertia weight and maximum velocity were then reduced by fractions α and β respectively, if no improvement in p_g occurred after a specified number of iterations h, i.e.,

$$\textit{if} \quad f(p_g^t) = f(p_g^{t-h}) \quad \text{then} \quad \omega^{t+1} = \alpha \cdot w^t \quad \text{and} \quad v_{\max,j}^{t} = \beta \cdot v_{\max,j}^{t} \tag{13}$$

where α and β are such that $0 < \alpha, \beta < 1$. This version is referred to as *PSO with dynamic inertia and maximum velocity* or *PSO-DIV*.

Clerc modified the basic PSO by using constriction coefficient χ to balance the exploration-exploitation trade-off (Clerc & Kennedy, 2002). The velocity update equation changes to

$$v_{i,j}^{t+1} = \chi[v_{i,j}^{t} + \phi_1(p_{i,j} - x_{i,j}^{t}) + \phi_2(p_{g,j} - x_{i,j}^{t})] \tag{14}$$

where

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi(\phi - 4)}|}$$

with

$$\phi = \phi_1 + \phi_2, \quad \phi_1 = c_1 r_1, \quad \phi_2 = c_2 r_2 \tag{15}$$

Conditions $\varphi \geqq 4$ and $\kappa \in [0, 1]$ ensure that the swarm converges. Parameter κ controls swarm exploration and exploitation abilities. For $\kappa \approx 0$, fast convergence is obtained by local exploitation. The swarm then exhibits hill-climbing behavior. On the other hand, $\kappa \approx 1$ results in slow convergence with a high degree of exploration. Usually, κ is set to a coefficient value. However, a high initial degree of exploration with local exploitation in the later search phases can be achieved used an initial value close to 1 and decreasing it to zero. This version is referred to as *PSO with constriction*, or *PSO-C*.

3. PSO for CCMPO problems

This section describes the proposed PSO approach to CCMPO problems. Here, a particle position is formulated as a portfolio, and the position of particle i in dimension j is the proportion of capital to be invested in the jth asset.

This study found that PSO quickly stagnates to the local optimum in CCMPO problems in order to satisfy the constraint on the desired number of assets K in the portfolio especially when CCMPO problems consider high values for risk aversion parameter λ. This study therefore considered possible remedies for constraint satisfaction in the PSO to avoid stagnation in early phase. Fig. 2 gives the basic structure of the proposed PSO. The different steps are described below in detail.

3.1. Initialization

In PSO initialization phase, a size N swarm is randomly generated. The individuals in the swarm are randomly valued for each dimension between the bounds. Similarly, the velocity is initialized to zero in each dimension.

```
    S.n_s denote the total number of particles in the swarm S S.x_i denote the
position of particle i in the swarm S
    S.y_i denote the personal best position of particle i has visited since the first
time step in the swarm S.
    S.ŷ denote the best position discovered by all particles so far in the swarm S.
    for e=1 to E do
      λ =(e-1)/(E-1)
      Create and initialize an n_x-dimensional swarm, S;
      For iteration= 1: MaxInerations
      For each particle i=1,2,…,S.n_s
        Constraint handling method to satisfy the constraints equation (19);
        Evaluate portfolio using equation ;
        If f(S.x_i) < f(S.y_i)
             S.y_i = S.x_i ;            //set the personal best position
        End
        If f(S.y_i) < f(S.ŷ )
              S.ŷ = S.y_i ;             //set the global best position
        End
      End
      For each particle i=1,2,…, S.n_s
          Update the velocity using equation (9);
          Update the position using equation (11);
           Boundary handling method using equation (16,17,18);
           Mutation
      End
      Until stopping condition criterion is true;
    End for
    Error calculation using final return and risk compare to standard efficient
frontier
    End for
```

Fig. 2. PSO for CCMPO problems.

3.2. Constraints satisfaction

New positions may leave the search space when updating particle positions in search process. In this case, the intuitive solution is setting the value of the new position to the boundary value for the asset of the portfolio. This causes rapid stagnation of the PSO to the local optimum. To attain better diversity to the search space, the reflection strategy suggested in Paterlini and Krink (2006) is applied during the initial search phase. That is if the value of the new position leaves the domain of the search space, it is reflect back into the domain by

$$x_{i,j}^t = x_{i,j}^t + 2(x_j^l - x_{i,j}^t) \quad \text{if } x_{i,j}^t < x_j^l \tag{16}$$

$$x_{i,j}^t = x_{i,j}^t - 2(x_{i,j}^t - x_j^l) \quad \text{if } x_{i,j}^t > x_j^u \tag{17}$$

where x_j^u and x_j^l are the upper and lower bounds of each jth component, respectively. This method allows the particle to explore a larger search area and to escape from local minima, which improves solution quality. The reflection strategy terminates when no improvement is obtained after numerous iterations. Then the boundary value was set by

$$x_{i,j}^t = x_j^l, \quad \text{if } x_{i,j}^t < x_j^l. \qquad x_{i,j}^t = x_j^u \quad \text{if } x_{i,j}^t > x_j^u \tag{18}$$

For handling the cardinality constraints, K is the desired number of assets in the portfolio. Given a set Q of K assets, Let K^{new} represent the number of assets after updating positions in portfolio (the numbers of the proportion w_i greater than 0). If $K^{new} < K$, then some assets must be added to Q; if $K^{new} > K$, then some assets must be removed from Q until $K^{new} = K$.

Considering the removal of assets in the case $K^{new} > K$. This study deletes the smallest assets. If $K^{new} < K$ assets, assets remaining to be added must be identified. This study randomly adds an asset $i \notin Q$ and assigns the minimum proportional value ε_i to the new asset.

According to Eq. (7), the value of x_i must also satisfy $0 \leqq \varepsilon_i \leqq x_i \leqq \delta_i \leqq 1$ for $i \in Q$. Let s_i represent the proportion of the new position belonging to Q. If $s_i < \varepsilon_i$, the minimum proportional value of ε_i replaces asset s_i. If $s_i > \varepsilon_i$, the proportional share of the free portfolio is calculated as follows

$$x_i = \varepsilon_i + \frac{s_i}{\sum_{j \in Q,\, s_i > \varepsilon_i} s_i} \left(1 - \sum_{j \in Q}^{\varepsilon_i}\right) \tag{19}$$

This minimizes the proportional value of ε_i for the useless assets $i \in Q$ so that particles converge faster in the search process, especially in CCMPO problems involving low values for risk aversion parameter λ.

3.3. Inertia weight (w)

The inertia weight w controls how previous velocity affects present velocity. High w values emphasize exploration for the global search the optimal solution while low values emphasize the local search around the current search area. All population-based search techniques rely on global exploration and local exploitation to achieve good performance. Generally, exploration should be most intensive in initial stages when the algorithm has very little knowledge about the search space, whereas later stages require additional exploitation requiring the algorithm to exploit information it has gained so far.

Since CCMPO problems involve complex search space, the parameter w becomes vital in PSO algorithms. Therefore, the proposed PSO uses the time variant w for CCMPO problems introduced by Shi and Eberhart (Tripathi, Bandyopadhyay, & Pal, 2007). The w is linearly reduced during the search process. Therefore, inertia values are initially large and decrease over time. Particles tend to explore in the initial search steps and tend to exploit as time increasingly. The w at time step t update is obtained by

$$w(t) = (w(0) - w(n_t))\frac{(n_t - t)}{n_t} + w(n_t) \tag{20}$$

where n_t is the maximum number of time steps required to execute the algorithm, $w(0)$ is the initial inertia weight, $w(n_t)$ is the final inertia weight, and $w(t)$ is the inertia at time step t. Usually $w(0) = 0.9$ and $w(n_t) = 0.4$.

3.4. Acceleration coefficients (c_1 and c_2)

If c_1 is larger than c_2, each particle has a stronger attraction to its own best position, and excessive wandering occurs. On the other hand, if c_2 exceeds c_1, particles are most attracted to the global best position, which causes them to rush towards the optima prematurely. The ratio between c_1 and c_2 coefficients is problem-dependent. Most applications use $c_1 = c_2$. Since c_1 and c_2 are usually static, their optimized values are found empirically.

To ensure a more global search during initial stages and a more local search during the final stages of CCMPO problems, the proposed PSO adopts time variants c_1 and c_2 as demonstrated by Ratnaweera, Halgamuge, and Watson (2004). Over time c_1 decreases linearly, and c_2 increases linearly. This strategy focuses on exploration in the early stages of optimization process by trusting itself, and encourages convergence to a good optimum near the end of the optimization process by trusting the best particle. The c_1 and c_2 at time step t update is given as

$$\begin{aligned} c_1(t) &= (c_{1,min} - c_{1,max})\frac{t}{n_t} + c_{1,max} \\ c_2(t) &= (c_{2,max} - c_{2,min})\frac{t}{n_t} + c_{2,min} \end{aligned} \tag{21}$$

where n_t is the maximum number of time steps, usually $c_{1,max} = c_{2,max} = 2.5$ and $c_{1,min} = c_{2,min} = 0.5$.

3.5. Mutation

Similarly, to allow our proposed PSO algorithm to maximize diversity, mutation operator was used based on (Tripathi et al., 2007). A similar mutation operator was used in the PSO for multi-modal function optimization. Given a particle, a randomly chosen variable, say g_k, is mutated as given below:

$$g'_k \begin{cases} g_k^{+}\Delta(t, \mathrm{UB} - g_k) & \text{if } \mathit{flip} = 0, \\ g_k + \Delta(t, g_k - \mathrm{LB}) & \text{if } \mathit{flip} = 1. \end{cases} \tag{22}$$

where *flip* denotes the random event of returning 0 or 1. The UB denotes the upper limit of the variable g_k while LB denotes the lower limit. The function Δ is defined as

$$\Delta(t, x) = x \cdot \left(1 - r^{\left(1 - \frac{t}{\max\_t}\right)^b}\right) \tag{23}$$

where r is a random number generated in the range [0,1], max_t is the maximum number of iterations, and t is the iteration number. The parameter b determines the dependence of the mutation on the iteration number.

3.6. Termination

The algorithm terminates when no improvement occurs over repeated iterations.

4. Computational experiments

To test the performance of the proposed PSO for CCMPO problems, two computational experiments were performed. The first experiment compared the performance of the proposed PSO with classic variants of PSO to CCMPO problems. The second experiment compared the solutions obtained by the proposed PSO with those obtained by heuristic algorithms.

4.1. Definition of experiments

The proposed PSO searches for efficient frontiers by testing 50 different values for the risk aversion parameter λ in the cardinality-constrained Markowitz portfolio model. The experiment employed the five benchmark datasets used earlier in (Chang et al., 2000; Fernandez & Gomez, 2007) These data correspond to weekly price data from March, 1992 to September, 1997 for the following indices: Hang Seng 31 in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan. The number N of different assets considered for each index was 31, 85, 89, 98 and 225, respectively. The sets of mean return of each asset, covariance between these assets and efficient frontier 2000 points are publicly available at http://people.brunel.ac.uk/mastjjb/jeb/orlib/portinfo.html. The cardinality constraints used the values $K = 10$, $\varepsilon_i = 0.01$ and $\delta_i = 1$ for problem formulation.

The three criteria used to quantify the performance of PSO for CCMPO problem solving were accuracy, robustness, diversity. Accuracy refers to the quality of the solution obtained. This analysis used the standard deviation (risk) and return of the best solution for each λ to compare standard efficient frontiers and to measure percentage error respectively, and the lower value of standard deviation error and mean returns error was used as the percentage error associated with a portfolio.

For example, let the pair (s_i, r_i) represent the standard deviation(risk) and mean return of a point obtained by PSO. Additionally, let s_i^* be the standard deviation corresponding to r_i according to a linear interpolation in the standard efficient frontier. The standard deviation of return error e_i for any point (s_i, r_i) is defined as the value $100\,(s_i^* - s_i)/s_i^*$. Similarly, by using the return r_i^* corresponding to s_i according to a linear interpolation in the standard efficient frontier, mean return error η_i can be defined as the quantity $100\;(r_i - r_i^*)/r_i^*$. The error measure defined in (Chang et al., 2000) was calculated by averaging the minimums between the mean return errors e_i and the standard deviation of return errors η_i.

The smaller the variance of a performance criterion over a number of trials, the more robust the algorithm. Variance was calculated as an expression of robustness for all algorithms.

Diversity is essential in population-based optimization algorithms. High diversity directly implies that a large area of the search space can be explored. Incorrect parameter settings may cause divergent or cyclic behavior and prevent convergence to a stable point. In simple terms, diversity can be defined as dispersal of particles. Formally, diversity can be calculated as

$$diversity(S) = \frac{1}{n_s}\sum_{i=1}^{n_s}\sqrt{\sum_{j=1}^{n_x}(x_{i,j} - \bar{x}_j)^2} \tag{24}$$

where $\bar{x}_j$ is the average of the jth dimension over all particles, i.e., $\bar{x}_j = \frac{\sum_{i=1}^{n_s} x_{i,j}}{n_s}$.

4.2. Experiment 1: comparison of PSO variants

This section compares the proposed PSO to classic variant PSO including basic PSO, PSO-DIV and PSO- Constriction to CCMPO problems. Numerous empirical studies show that the PSO is sensitive to control parameter choices such as inertia weight, acceleration coefficients and velocity clamping (van den Bergh & Engelbrecht, 2006). This study applied the following parameters suggested in the literature:

For *basic PSO*, the value $(w, c_1, c_2, v_{\max})$ set to (0.7298, 1.49618, 1.49618, 1) as suggested in van den Bergh and Engelbrecht (2006). For *PSO-DIV*, We set $\alpha = \beta = 0.99$, $h = 10$ and initial $v_{\max} = 1$ suggested in Fourie and Groenwold (2002). For *PSO-Constriction*, we set $c_1 = 2.8$ and $c_2 = 1.3$ as suggest in Schutte and Groenwold (2005). In the proposed PSO, the value $c_{1,max} = c_{2,max}$ set 2.5, $c_{1,min} = c_{2,min} = 0.5$, the $w_{initial} = 0.9$ linearly decreased $w_{final} = 0.4$, and $b = 5$ as suggested in Engelbrecht (2005), Ratnaweera et al. (2004), and Tripathi et al. (2007). The reflection strategy terminates when no improvement exceeds 20 iterations. Swarm size is set to 100 for all PSOs, and the algorithms terminate when no

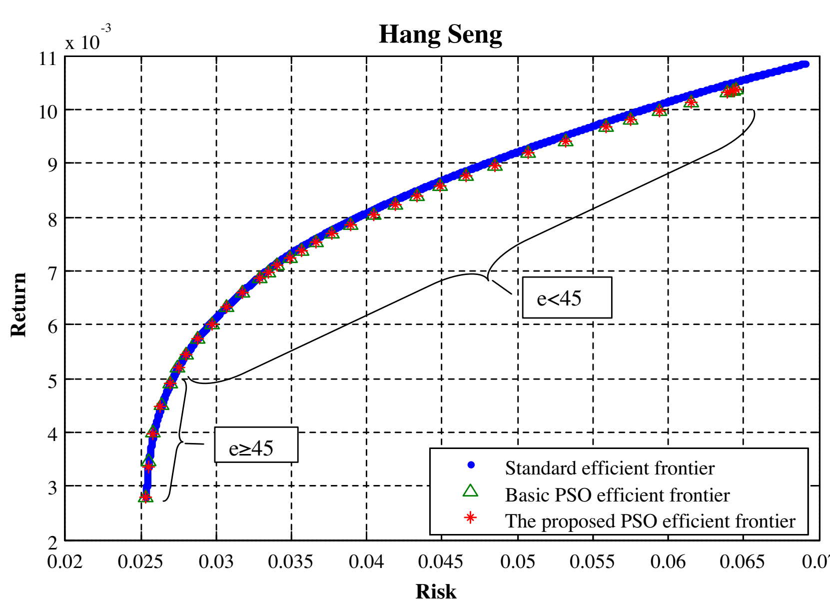

Fig. 3. Efficient frontiers for Hang Seng.

improvement occurs over 100 iterations. The average value of twenty-five trials for each test was recorded. All PSO algorithms presented in this paper were coded in MATLAB language and tested on a Pentium M processor with 1.6 GHz CPU speed and 1 GB RAM machine, running Windows.

Efficient frontiers formed by points were identified in each set of Pareto optimal portfolios obtained for each PSO and then compared with standard efficient frontiers. Figs. 3–7 show these comparisons. For clarity, each figure only shows three efficient frontiers. Since basic PSO efficient frontiers and other variant PSO have similar characteristics, basic PSO efficient frontiers were displayed in figure. Table 1 shows the minimum mean percentage error of portfolio for the proposed and variant PSOs. The best minimum mean percentage error for each problem is in boldface. Clearly, the proposed PSO generally obtained a lower minimum mean percentage error than variant PSOs did. Table 1 also exhibits the average CPU times and the number of iterations of convergence for each method. The comparison results indicate that the proposed PSO required no more iterations than variant PSOs did, and the CPU time required to solve the CCMPO problems was reasonable and practical. When solving complex CCMPO problems (number of assets >100) such as the last problem, the time required to run a trial was only slightly longer than simple CCMOP problems (number of assets <100).

For risk aversion parameter λ, when value λ is low, the portfolio emphasizes to maximize return regardless of risk. The objective function of CCMPO problems according to Eq. (1) became almost linear. The portfolio in this case typically included significant investments less than the $K = 10$ assets (here, the term "significant investment" refers to any investment exceeding $1/K$). When value λ

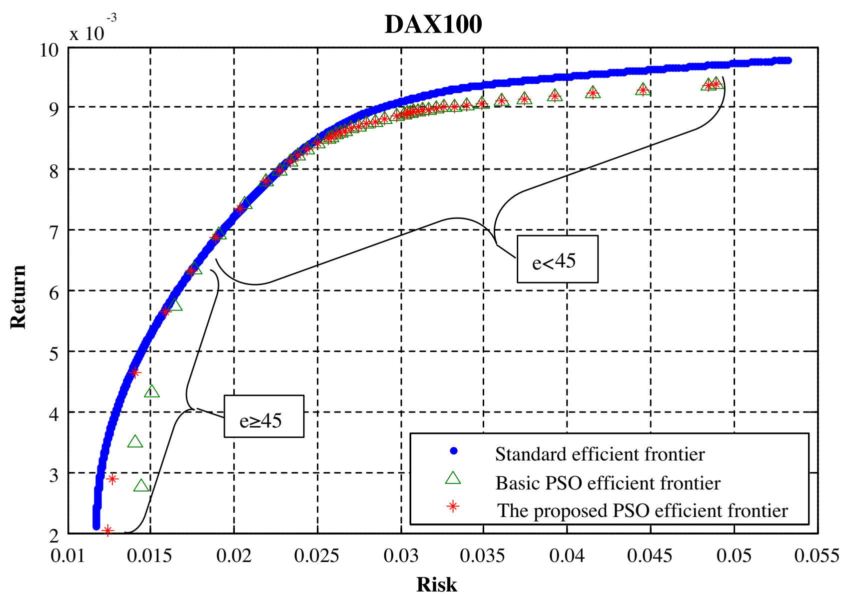

Fig. 4. Efficient frontiers for DAX100.

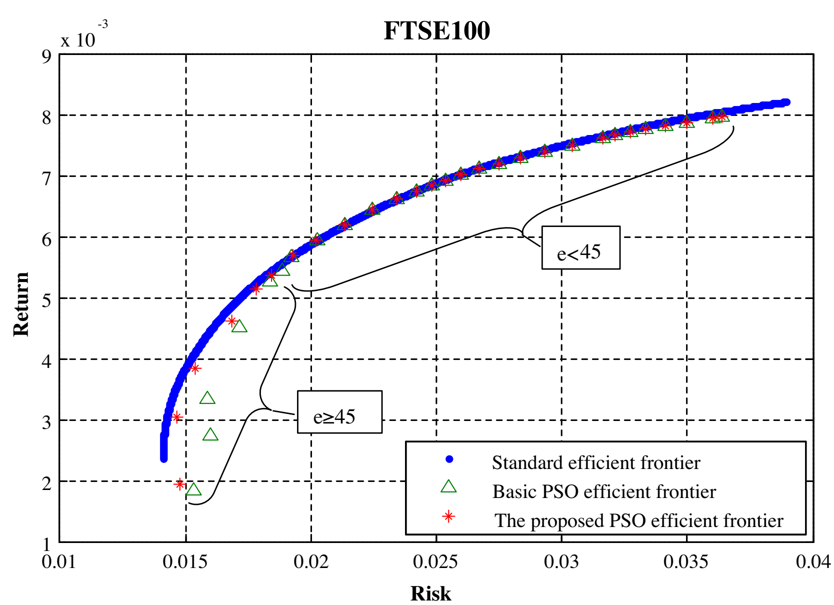

Fig. 5. Efficient frontiers for FTSE100.

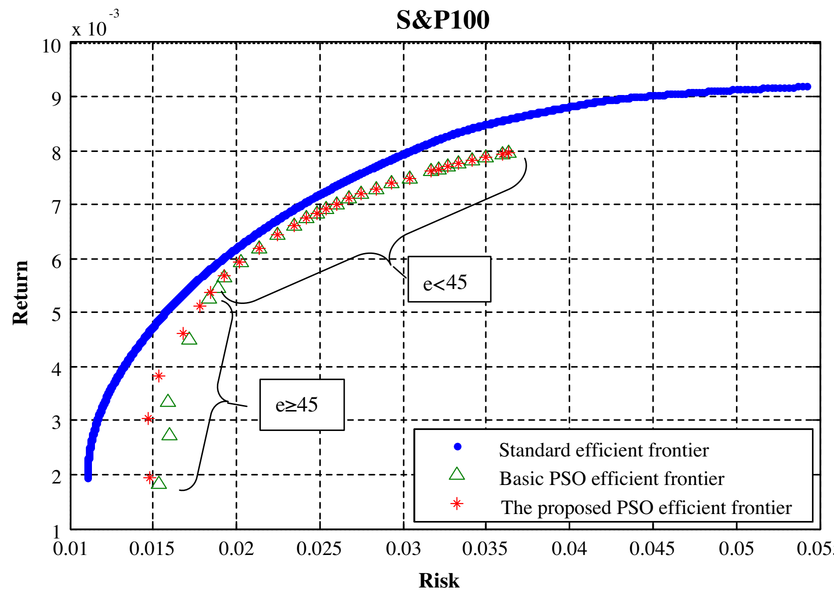

Fig. 6. Efficient frontiers for S&P100.

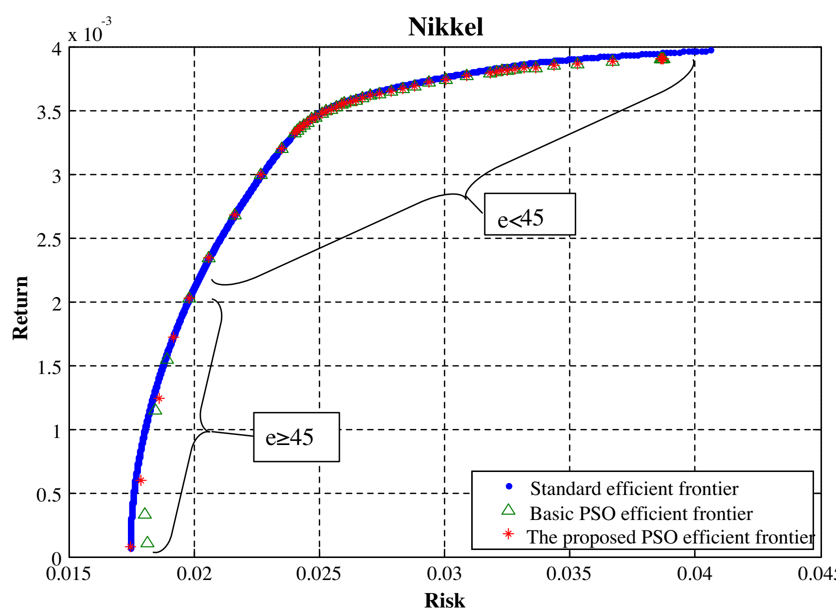

Fig. 7. Efficient frontiers for Nikkel.

is high, the portfolio emphasizes to minimize risk regardless of return, the objective function becomes nonlinear. In the portfolio obtained in this case, the diversity of significant investments were close to K assets.

The experiment further tested low value $\lambda(e < 45)$ and high values $\lambda(e \geqslant 45)$ for variant PSO and the proposed PSO, respectively. Table 2 shows the comparison results. When high value of λ were considered (emphasize risk). The proposed PSO outperformed all other variant PSOs except in the smallest benchmark problem (the first problem). Additionally, all PSOs performed comparably, when low value of λ was considered (regardless of the risk).

Table 2 also shows the robustness comparisons for all PSOs. When CCMPO problems emphasize risk aversion (high values $\lambda(e \geqslant 45)$), the proposed PSO obtained the lowest variance of all PSOs. When CCMPO problems place emphasis on return (high values $\lambda(e < 45)$), all PSOs obtained similar variance. Overall, the proposed PSO was more robust than other PSOs.

4.3. Experiment 2: comparative performance of the proposed PSO with other heuristics

To compare the proposed PSO with other heuristics, the same data sets were considered in the constrained portfolio problem. Table 3 shows that comparable results were obtained for minimum mean percentage error. The heuristics used to compare the proposed PSO were genetic algorithm (GA), simulated annealing(SA), and tabu search(TS).The results on GA, SA, and TS are from (Chang et al., 2000). The proposed PSO was run for 1000 iterations using 100 particles. The parameter settings were approximately the same number of solution searched by the heuristic with which

Table 1
Experimental result for CCMOP problems.

Problem name/assets(N)		Basic PSO	PSO-DIV	PSO-Constriction	Proposed PSO
Hang Seng 31	Mean percentage error	1.104717681	1.095675909	1.095702307	**1.095302002**
	Best cost	1.095717385	1.095575769	1.095502701	1.095299022
	CPU time(s)/iterations of Stopping	4.5/393	4.9/432	4.9/446	4.8/427
DAX100 85	Mean percentage error	2.92051537	3.495334379	2.590166003	**2.54171247**
	Best cost	2.862105063	3.425427691	2.548363	2.450878
	CPU time(s)/iterations of Stopping	24.0/492	23.5/455	27.0/523	26.8/511
FTSE100 89	Mean percentage error	1.427812792	1.617620472	1.297192971	**1.06283486**
	Best cost	1.399256536	1.585268063	1.271249	1.012178
	CPU time(s)/iterations of Stopping	30.4/487	31.1/512	32.0/538	31.4/515
S&P100 98	Mean percentage error	2.555478598	2.56155639	2.208278669	**1.68903754**
	Best cost	2.399256536	2.385268063	2.071249	1.622178
	CPU time(s)/iterations of Stopping	35.6/555	34.5/532	36.0/572	36.6/581
Nikkei 225	Mean percentage error	0.964592	0.897954	0.844298	**0.687015592**
	Best cost	0.94530016	0.84019492	0.80612	0.631875
	CPU time(s)/iterations of Stopping	78.2/565	79.0/576	79.5/580	75.8/545

Table 2
Results of robustness analysis for CCMOP problems.

Problem name/assets (N)	e	Mean percentage error/variance	Basic PSO	PSO-DIV	PSO-constriction	Proposed PSO
Hang Seng 31	$e < 45$	Mean	1.23979088	1.23979085	1.23979084	1.23979085
		Variance	3.672E-04	4.173E-04	2.172E-04	2.512E-04
	$e \geqslant 45$	Mean	0.114180888	0.038833008	0.039053065	0.035717117
		Variance	0.0231	0.0341	0.0268	0.0212
DAX100 85	$e < 45$	Mean	2.42053749	2.4200721	2.41892175	2.41895154
		Variance	5.091E-05	7.516E-05	4.565E-05	7.663E-05
	$e \geqslant 45$	Mean	6.587019823	11.38059109	3.845957192	3.44195929
		Variance	4.213	5.621	2.225	0.5327
FTSE100 89	$e < 45$	Mean	0.77771612	0.77130061	0.81686727	0.78080596
		Variance	5.752E-03	7.112E-03	8.766E-03	7.391E-03
	$e \geqslant 45$	Mean	6.195188387	7.823966127	4.819581445	3.131046793
		Variance	2.154	2.857	1.843	0.4417
S&P100 98	$e < 45$	Mean	1.32944344	1.31971421	1.13107077	1.07017639
		Variance	8.916E-03	6.842E-03	4.706E-03	5.845E-03
	$e \geqslant 45$	Mean	11.54640309	11.66839904	10.10780326	6.22735264
		Variance	3.762	5.291	2.114	0.782
Nikkei 225	$e < 45$	Mean	0.65249758	0.66909846	0.65112283	0.64049833
		Variance	9.253E-03	9.162E-03	7.765E-03	7.895E-03
	$e \geqslant 45$	Mean	3.253288667	2.576231333	2.260922	1.0281446
		Variance	1.762	1.291	0.814	0.282

Table 3
Comparison of the proposed PSO with other heuristics for CCMOP problems.

Problem name	Assets (N)	GA	SA	TS	Proposed PSO
Hang Seng	31	1.0974	1.0957	1.1217	**1.0953**
DAX100	85	2.5424	2.9297	3.3049	**2.5417**
FTSE100	89	1.1076	1.4623	1.6080	**1.0628**
S&P100	98	1.9328	3.0696	3.3092	**1.6890**
Nikkei	225	0.7961	**0.6732**	0.8975	0.6870
Average		1.4953	1.8461	2.0483	**1.4152**

we compare our results. The results for the proposed PSO were averaged over 25 trials. The minimum mean percentage error for each problem is given in boldface. The proposed PSO almost always obtained the best performance in most cases.

5. Conclusion

This work developed an improved PSO for identifying the efficient frontier in portfolio optimization problems. The standard Markowitz mean–variance model was generalized to include cardinality and bounding constraints. Such constraints convert the portfolio selection problem into a mixed quadratic and integer programming problem, for which computationally efficient algorithms have not been developed. The proposed PSO was tested on CCMPO problem set. The test results confirmed that incorporating the improved constraint handling increased PSO exploration efficiency in the initial search steps and increased convergence speed in the final search steps. Comparisons with other PSO variants also showed that the proposed PSO is much more robust and effective, especially for low-risk investments. Solution comparisons showed that the proposed PSO outperformed genetic algorithm (GA), simulated annealing (SA), and tabu search (TS). Further research using Ant colony optimization to solve the CCMPO problem is currently underway.

References

Ali, M. M., & Kaelo, P. (2008). Improved particle swarm algorithms for global optimization. *Applied Mathematics and Computation, 196*(2), 578–593.

Chang, T. J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research, 27*(13), 1271–1302.

Chang, T. J., Yang, S. C., & Chang, K. J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications, 36*(7), 10529–10537.

Clerc, M. (2006). *Particle swarm optimization*. London: ISTE.
Clerc, M., & Kennedy, J. (2002). The particle swarm – Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73.
Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research, 150*(3), 546–571.
Engelbrecht, A. P. (2005). *Fundamentals of computational swarm intelligence*. Chichester, West Sussex, England; Hoboken, NJ: Wiley.
Fernandez, A., & Gomez, S. (2007). Portfolio selection using neural networks. *Computers & Operations Research, 34*(4), 1177–1191.
Fourie, P., & Groenwold, A. (2002). The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization, 23*(4), 259–267.
Jiang, Y., Hu, T., Huang, C., & Wu, X. (2007). An improved particle swarm optimization algorithm. *Applied Mathematics and Computation, 193*(1), 231–239.
Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. San Francisco, CA: Kaufmann.
Maringer, D., & Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *Or Spectrum, 25*(4), 481–495.
Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, 77–91.
Paterlini, S., & Krink, T. (2006). Differential evolution and particle swarm optimisation in partitional clustering. *Computational Statistics and Data Analysis, 50*(5), 1220–1247.
Ratnaweera, A., Halgamuge, S., & Watson, H. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*(3), 240–255.
Schutte, J., & Groenwold, A. (2005). A study of global optimization using particle swarms. *Journal of Global Optimization, 31*(1), 93–108.
Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. Paper presented at the evolutionary computation proceedings, the 1998 IEEE International Conference on 1998. IEEE world congress on computational intelligence.
Soleimani, H., Golmakani, H. R., & Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications, 36*(3, Part 1), 5058–5063.
Tripathi, P. K., Bandyopadhyay, S., & Pal, S. K. (2007). Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. *Information Sciences, 177*(22), 5033–5049.
Tsai, S.-J., Sun, T.-Y., Liu, C.-C., Hsieh, S.-T., Wu, W.-C., & Chiu, S.-Y. (2010). An improved multi-objective particle swarm optimizer for multi-objective problems. *Expert Systems with Applications, 37*(8), 5872–5886.
van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences, 176*(8), 937–971.
Yang, X., Yuan, J., Yuan, J., & Mao, H. (2007). A modified particle swarm optimizer with dynamic adaptation. *Applied Mathematics and Computation, 189*(2), 1205–1213.