

**A new IPSO-SA approach for cardinality constrained portfolio optimization****Marzieh Mozafari<sup>a\*</sup>, Sajedah Tafazzoli<sup>a</sup>, Fariborz Jolai<sup>b</sup>**<sup>a</sup>*Department of Industrial Engineering, Amirkabir University of Technology, 424 Hafez Ave., Tehran 15916-34311, Iran*<sup>b</sup>*Department of Industrial Engineering, College of Engineering, University of Tehran, P.O. Box 11155/4563, Tehran, Iran***ARTICLE INFO***Article history:*Received 1 October 2010  
Received in revised form  
7 January 2011  
Accepted 10 January 2011  
Available online  
14 January 2011**Keywords:**Portfolio optimization  
Cardinality constraint  
Hybrid solution approach  
Improved particle swarm  
optimization  
Simulated annealing**ABSTRACT**

The problem of portfolio optimization has always been a key concern for investors. This paper addresses a realistic portfolio optimization problem with floor, ceiling, and cardinality constraints. This problem is a mixed integer quadratic programming where traditional optimization methods fail to find the optimal solution, efficiently. The present paper develops a new hybrid approach based on an improved particle swarm optimization (PSO) and a modified simulated annealing (SA) to find the cardinality constrained efficient frontier. The proposed algorithm benefits from simple and easy characteristics of PSO with an adaptation of inertia weights and constriction factor. In addition, incorporating an SA procedure into IPSO helps escaping from local optima and improves the precision of convergence. Computational results on benchmark problems with up to 225 assets signify that our proposed algorithm exceeds not only the standard PSO but also the other heuristic algorithms previously presented to solve the cardinality constrained portfolio problem.

© 2011 Growing Science Ltd. All rights reserved

**1. Introduction**

Today, managers and investors confront a wide range of possible securities or assets where each has its own potential risk and rewards. The traditional portfolio optimization techniques attempt to select an appropriate portfolio of assets with a reasonable trade-off between the risk and the rate of return. The mean-variance approach introduced by Markowitz (1954) for portfolio optimization problems plays an important role in developing modern portfolio selection techniques. Markowitz model is a quadratic programming problem, which can be solved through exact methods such as active set methods, interior point techniques, etc.

Markowitz seminal work has been widely extended in different ways. Following his work, alternative models such as mean-absolute deviation (MAD) and absolute deviation were proposed for the same problem. Konno and Yamazaki (1991) proposed a linear programming model which is equivalent to Markowitz model whenever returns are multivariate and normally distributed. Ignoring the covariance matrix in the Markowitz model results in great estimation risk and could cause serious error (Simaan, 1997). A piecewise linear approximation, weighted goal programming (Lee &

\* Corresponding author. Tel: +982164545329  
E-mail: m.mozafari@aut.ac.ir (M. Mozafari)

Chesser, 1980) and mini-max model (Young, 1998) are the other examples of these models. Some authors have added some practical constraints such as transaction costs, liquidity, buy-in threshold, cardinality, etc to Markowitz model to make it more realistic. However, adding constraints to the portfolio optimization problem make it intractable even for small instances.

When conventional techniques cannot be used, heuristic algorithms can assist. Di Tollo and Roli (2008) provided an overview of the literature on the application of meta-heuristics to the portfolio selection problem. These methods consist of simulated annealing (SA) (Crama & Schyns, 2003; Armañanzas & Lozano, 2005), threshold accepting (TA) (Dueck & Winker, 1992; Gilli et al., 2006), tabu search (TS) (Glover et al. 1995; Rolland, 1997) genetic algorithm (GA) (Arnone, 1993; Yang, 2006) and ant colony optimization (ACO) (Armañanzas & Lozano, 2005; Maringer, 2001). Chang et al. (2000) proposed GA, SA and TS for cardinality constrained problem and the proposed model of this paper will be compared with the results of this paper.

A relatively new meta-heuristic approach, where its application in portfolio optimization is still limited, is particle swarm optimization (PSO). Mous et al. (2006) compared PSO and GA which was applied to portfolio selection, Xu et al. (2007) and Gao and Chu (2010) proposed improved PSO algorithms for portfolio problem with transaction costs and quality constraints. Mishra et al. (2009) studied portfolio optimization problem from multi-objective perspective and considered four well-known multi-objective evolutionary algorithms including multi-objective PSO (MOPSO) which outperformed its counterparts for some numerical experiments. Cura (2009) presented a PSO algorithm to solve cardinality constrained portfolio. Compared with other meta-heuristics, PSO is simple, high speed, large scope and easy to be implemented by programs. However, there are still many issues in particle swarm, such as slow convergence during the latter search, poor precision and converging to local optimum. To overcome the above problems, a lot of revised PSO have emerged.

Investors generally incline to restrict the number of assets in the portfolio and purchase just a subset of them. Therefore, cardinality is a practical constraint that has to be considered in decisions. In this paper, we address a portfolio optimization problem with floor, ceiling, and cardinality constraints. This problem is a mixed integer programming with quadratic objective function where traditional optimization methods fail to find the optimal solution, efficiently. We propose a new hybrid solution approach based on an improved PSO (IPSO) and a simulated annealing (SA) procedure to optimize the cardinality constrained portfolio problem. Our proposed algorithm takes advantage of inertia weights mechanism and constriction factor approach in updating the particle's velocity to improve the PSO searching capabilities. In addition, an SA procedure is embedded into IPSO to improve the capacity of fine-tuning solution in the latter period of the search. Incorporating SA procedure can enhance the ability of IPSO for jumping out of the local optimum. Computational experiments on standard benchmark problems are carried out to assess the effectiveness of the IPSO-SA algorithm to solve cardinality constrained portfolio problem.

The rest of this paper is organized as follows. Section 2 describes the formulation of the portfolio optimization problem with cardinality constraints. In Section 3, an IPSO-SA algorithm is proposed to optimize the cardinality constrained portfolio problem. Section 4 is devoted to computational experiments and involves parameter setting for the proposed algorithm, numerical results and statistical performance analysis on benchmark data sets. Finally, Section 5 discusses the concluding remarks.

## **2. Cardinality constrained portfolio optimization model**

The well-known mean-variance Markowitz (1954) model which played an important role in the development of modern portfolio problems can be expressed as follows,

$$\min \lambda \left[ \sum_{i=1}^N \sum_{j=1}^N x_i x_j \sigma_{ij} \right] - (1 - \lambda) \left[ \sum_{i=1}^N x_i \mu_i \right] \quad (1)$$

subject to

$$\sum_{i=1}^N x_i = 1, \quad (2)$$

$$0 \leq x_i \leq 1, \quad i = 1, \dots, N. \quad (3)$$

where  $N$  is the number of distinctive assets,  $x_i$  is the proportion of asset  $i$  in the portfolio,  $\sigma_{ij}$  is the covariance between the returns of assets  $i$  and  $j$ ,  $\mu_i$  denotes the mean return of asset  $i$  and  $\lambda$  is the risk aversion parameter. When  $\lambda$  is zero, the model maximizes the mean return of the portfolio, neglecting the risk. In contrast, when  $\lambda$  is 1, the model minimizes the risk of the portfolio regardless of the mean return.

Chang et al. (2000) have extended the Markowitz model by adding two constraints: floor-ceiling and cardinality constraints. The cardinality constrained portfolio optimization model can be expressed as follows,

$$\min \lambda \left[ \sum_{i=1}^N \sum_{j=1}^N x_i x_j \sigma_{ij} \right] - (1 - \lambda) \left[ \sum_{i=1}^N x_i \mu_i \right] \quad (4)$$

subject to

$$\sum_{i=1}^N x_i = 1, \quad (5)$$

$$\sum_{i=1}^N y_i = K, \quad (6)$$

$$\varepsilon_i y_i \leq x_i \leq \delta_i y_i \quad i = 1, \dots, N, \quad (7)$$

$$y_i \in \{0, 1\}, \quad i = 1, \dots, N \quad (8)$$

where  $K$  is a given value which restricts the number of assets in the portfolio. The decision variable  $y_i \in \{0, 1\}$  is 1, if asset  $i$  is assigned to portfolio, otherwise  $y_i = 0$ . Parameters  $\varepsilon_i$  and  $\delta_i$  are respectively the floor and ceiling bounds for asset's proportion.

The cardinality constrained mean-variance model is a mixed integer nonlinear program, which cannot be solved efficiently by the existing algorithms due to its NP-Hard nature (Fernandez & Gomez, 2007). In the following section, we propose a new hybrid optimization strategy to solve medium and large instances of the cardinality constrained portfolio optimization problem encountered in real-world applications.

### 3. The IPSO-SA algorithm for solving cardinality constrained portfolio optimization

Particle swarm optimization (PSO) which was originally developed by Kennedy and Eberhart (1995a, 1995b) is a population based stochastic optimization technique imitating the social behaviors of bird flocking. PSO has many similarities to evolutionary computation techniques like GA; however, the typical evolution operators such as selection, crossover and mutation are not involved. PSO has been widely applied in solving combinatorial optimization problems due to its exploration capability, implementation simplicity and performance reliability (Chen et al, 2006). In PSO, individual particles move thorough the problem search space seeking an optimal solution. The new position of each particle is adjusted according to its velocity and the distance between its current position and the local best position as well as the entire swarm best known position. As the algorithm is iterated, the swarm

focuses more and more on an area of the search space containing high-quality solutions (Blum & Li, 2008).

The combination of global exploration and local exploitation features during the optimization process is the key success of PSO algorithm. However, the main pitfall of the standard PSO is that it may be trapped into a sub-optimal solution region. Hence, many authors have extended the standard PSO and proposed efficient variants. For example, Shi and Eberhart (1998) proposed the inertia weights approach to balance the global and local search abilities in standard PSO. Inertia weights PSO investigates the space globally in early steps and searches around the optimal solution locally in final steps. Clerc and Kennedy (2002) developed a PSO variant by using a constriction factor in velocity update equation, and improved the convergence of the algorithm. Koshino et al. (2007) incorporated the inertia weights and constriction factor approaches and rose to higher-level performance for PSO algorithm. Pram et al. (2003) presented the fitness-distance-ratio based PSO, with near neighbor interactions. This algorithm considers another particle, *nbest*, in velocity update, which has a higher fitness value and is nearer to the updated particle. Some authors have hybridized PSO algorithms adding features of other meta-heuristic approaches and achieved more robust and effective methods for optimizing hard problems. Examples are Tasgetiren et al. (2004), Liu et al. (2007), Shen et al. (2008), Behnamian and Fatemi Ghomi (2010), and Tang et al. (2010). In this paper, a new hybrid PSO algorithm is proposed for the solution of mean-variance cardinality constrained problem. The proposed algorithm is inspired by recent improved variant of PSO, called IPSO, which integrates merits of inertia weights and constriction factor approaches (IWCFA). A modified simulated annealing approach with strong local-search power is incorporated to further improvement. The proposed IPSO-SA approach makes full utilization of the exploration capabilities in IPSO and SA algorithms and offsets the weaknesses of each algorithm.

### 3.1. Encoding scheme

A particle is represented by a  $2 \times N$  dimensions matrix in which the first row involves proportion variables  $x_{pi}$  and the second row includes binary decision variables  $y_{pi}$ .  $N$  reflects the number of assets in the portfolio. The initial position and the velocity populations are generated randomly containing  $P$  particles.

### 3.2. Constraint handling and fitness evaluation

To ensure the feasibility of a candidate solution, we employ a constraint handling approach inspired from Cura (2009). When the number of assets is fewer than  $K$ , we select the maximum c-valued asset and conclude this asset to portfolio with a probability  $\beta$ ; or an asset is chosen randomly with  $(1 - \beta)$  probability. On the contrary, when the number of assets is more than  $K$ , we remove the minimum c-value asset or a random one from the portfolio. C-value provides the proportion between mean return and mean risk for a given asset, with respect to parameter  $\lambda$ ; and can be computed as follows,

$$\theta_i = 1 + (1 - \lambda)\mu_i, \quad i = 1, \dots, N \quad (9)$$

$$\rho_i = 1 + \lambda \frac{\sum_{j=1}^N \sigma_{ij}}{N}, \quad i = 1, \dots, N \quad (10)$$

$$\Omega = -1 \times \min(0, \theta_1, \dots, \theta_N), \quad (11)$$

$$\Psi = -1 \times \min(0, \rho_1, \dots, \rho_N), \quad (12)$$

$$c_i = \frac{\theta_i + \Omega}{\rho_i + \Psi}, \quad i = 1, \dots, N \quad (13)$$

We used  $\beta = 0.5$  for asset reposition probability.

Besides, in order to meet constraint 5, we can easily reposition  $x_{pi}$  with  $x_{pi} / \sum_{i=1}^N x_{pi}$  and to guarantee the boundary constraints on  $x_i$ , we provide a substitute for  $x_i$  such that

$$x_{pi} = \begin{cases} x_{pi} + \frac{\delta_i}{\delta^*} \eta & \text{if } d_i > 0 \\ \delta_i & \text{if } d_i < 0 \\ x_{pi} + \frac{\varepsilon_i}{\varepsilon^*} \gamma & \text{if } e_i > 0 \\ \varepsilon_i & \text{otherwise} \end{cases}, \tag{14}$$

where  $d_i = \delta_i - x_{pi}$  and  $\delta^*$  are the sum of  $d_i$ , for  $i \in Q$  (i.e. where  $y_i$  is equal to 1) and  $d_i > 0$ ,  $e_i = x_{pi} - \varepsilon_i$  and  $\varepsilon^*$  are the sum of  $e_i$ , where  $i \in Q$  and  $e_i > 0$ ,  $\eta$  is the sum of  $(-1 \times d_i)$ , where  $i \in Q$  and  $d_i < 0$  and finally  $\gamma$  is the sum of  $(-1 \times e_i)$ , where  $i \in Q$  and  $e_i < 0$ .

To assess how much a solution fits to the optimization purpose, a fitness function is evaluated. The fitness value of each particle  $p$  is defined as follows,

$$f_p = \lambda [\sum_{i=1}^N \sum_{j=1}^N y_{pi} x_{pi} y_{pj} x_{pj} \sigma_{ij}] - (1 - \lambda) [\sum_{i=1}^N y_{pi} x_{pi} \mu_i]. \tag{15}$$

### 3.3. Simulated annealing procedure

Simulated annealing (SA) is a trajectory-based heuristic, which performs a stochastic neighborhood search of the solution space. SA's major advantage over classical local search methods is its ability to avoid getting trapped into local optimum while searching for a global optimum. SA exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system. Kirkpatrick et al. (1983) and Cerny (1985) pioneered the use of this algorithm for combinatorial problems.

Starting from a current solution  $p_1$ , SA generates another solution  $p_2$  by taking a stochastic step in some neighborhood of  $p_1$ . If the new solution improves the value of the objective function, it replaces  $p_1$  as the new current solution. Otherwise, the new solution is accepted with a given probability. The possibility of moving to solutions with a higher cost (i.e., performing degrading moves) characterizes SA and enables it to escape from local optimum. The probabilistic acceptance of the worst solution depends on the cost difference between the two solutions and it also decreases during the search. A Boltzmann function, Eq. (16), inspired from thermodynamics models often defines this probability,

$$\text{probability}(n, p, p_n) = \begin{cases} 1 & \Delta f_n < 0 \\ \exp(-\frac{\Delta f_n}{T_n}) & \Delta f_n \geq 0 \end{cases} \tag{16}$$

where  $\Delta f_n = f(p) - f(p_n)$  and  $T_n$  is the temperature at stage  $n$ . The temperature is kept unchanged during each stage, which consists of a constant number of iterations. After each stage, the temperature is decreased by a constant factor  $\bar{\alpha} \in (0,1)$ . Simulated annealing algorithms differ from each other with respect to the various factors such as neighborhood search, cooling (annealing) schedule and termination criterion. After evaluating the fitness value of each particle, we sort the population based on the objective function values in an ascending order to choose the global best solution of the swarm. We apply SA algorithm as a local search mechanism around this global best solution (*Gbest*). Thus, *Gbest* changes in each iteration. The other critical characteristic of SA, which determines the search quality of solution space, is the definition of neighborhood structure. Here we considered generating neighbors by modifying the weights of a subset of the assets of the current portfolio. Two neighborhood structure examined by modifying the weight of only one asset:

1. Choose randomly one asset, which is currently in portfolio. The quantity of a chosen asset is either increased or decreased by a factor  $q \in (0,1)$  and  $idR$  is also neighborhood structure (Schaerf, 2002).
2. Choose one asset randomly. If it is currently in portfolio, set its quantity to zero. Otherwise, add it to portfolio and set its quantity, randomly.

Applying both of the neighborhood structure to our portfolio optimization problem, the second one has better performance in reaching better solutions and escaping from local optima.

The other characteristic of SA algorithm is the feasibility of the final solution. There are two different approaches about the feasibility of the produced final solution. In the first approach, throughout all iterations of the SA algorithm, feasibility must be preserved and any solution violating the constraints will not be considered. Therefore, the neighborhood of a current solution must entirely consist of feasible solutions. On the contrary, to the first approach, the second one permits the consideration of infeasible solutions but adds a penalty term to the objective function for each violated constraint. Both approaches, “all-feasible” vs. “penalty” are not equally convenient in all situations. Since penalty approach does not seem to be appropriate for cardinality constrained portfolio optimization (Crama, Y. & Schyns, 2003), we use the so called all-feasible approach. In order to satisfy the constraints at any step of the search process, the constraint handling mechanism must be performed on the produced neighbor solution at every stage of SA. The initial temperature of the proposed SA is derived from the objective function value of the initial starting solution ( $|Gbest^t/1000|$ ).  $2N$  iterations are performed at each temperature. The proposed SA procedure pseudo code is displayed in Fig.1.

---

```

Begin SA
Temp = Gbest/1000;  $\bar{\alpha}$  ;
Gb=Gbest;
For iteration_count = 0 to 2N
  For a specified number of iteration at the same temperature do
    Generate a neighbor solution of Gb, called Gb'
    Make Gb' feasible;
    Compute fitness value for Gb';
    If  $f(Gb') < f(Gb)$ 
      Gb=Gb';
    Else if  $f(Gb') \geq f(Gb)$  &
       $\exp[-(f(Gb') - f(Gb))/temp] > rand$ 
      Gb=Gb';
    End if
  End for
  Temp =  $\bar{\alpha}$  * Temp;
End for
End

```

---

**Fig. 1.** SA procedure

### 3.4. Particles' velocity and position update

Particle swarm contains two primary operators: velocity and position update. In every iteration, each particle is updated to find a better position using its own memory as well as the swarm memory of the best-found solution. The first is the best solution it has achieved so far called  $Pbest$ . The latter is the best value obtained so far by any particles in the swarm called  $Gbest$ . Considering  $Pbest^t$  and  $Gbest^t$ , respectively as the particle's best and the swarm's best found solutions up to iteration  $t$ , the original PSO updates the velocity and position of each particle  $p$  through the following equations,

$$v_p^{t+1} = w^t v_p^t + c_1 r_1 (Pbest^t - x_p^t) + c_2 r_2 (Gbest^t - x_p^t), \quad (17)$$

$$x_p^{t+1} = x_p^t + v_p^{t+1}, \quad (18)$$

where  $r_1$  and  $r_2$  are two random numbers uniformly distributed in the interval  $[0,1]$ .  $c_1$  and  $c_2$  denotes the cognitive and social scaling parameters and  $w^t$  is the particle inertia weight. In this paper, each particle is moved by an improved way proposed by Koshino et al, (2007) which takes advantage of

both inertia weights and constriction factor approaches. According to inertia weights approach, the iteration number changes  $w^t$ , gradually. In this way, the problem space is investigated globally in the early steps, and locally near the optimal solution in the final steps. Constriction factor approach uses coefficients to the present velocity, the direction of the global best position, and the direction of the individual-best, in order to control the behavior of the swarm. Applying this improved approach, the velocity of each particle is updated as follows,

$$vy_{pi}^{t+1} = k(w^t \cdot vy_{pi}^t + c_1 r_1 (Pbest(y)_i^t - y_{pi}^t) + c_2 r_2 (Gbest(y)_i^t - y_{pi}^t)) \quad (19)$$

$$vx_{pi}^{t+1} = \begin{cases} k(w^t vx_{pi}^t + c_1 r_1 (Pbest(x)_i^t - x_{pi}^t) + c_2 r_2 (Gbest(x)_i^t - x_{pi}^t)) & \text{if } y_{pi}^{t+1} = 1, \\ vx_{pi}^t & \text{otherwise.} \end{cases} \quad (20)$$

where  $w^t$  and  $k$  can be calculated from the following equations.

$$w^t = w_{max} - \frac{w_{max} - w_{min}}{t_{max}} \cdot t, \quad (21)$$

$$k = \frac{2}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|}, \quad \phi = c_1 + c_2, \quad \phi > 4 \quad (22)$$

where  $w_{max}$  and  $w_{min}$  are the maximum and the minimum inertia weights, respectively.  $t_{max}$  denotes the maximum number of iterations.  $vx_{pi}^t$  reflects the velocity of particle  $p$  on dimension  $x_i$ , and  $vy_{pi}^t$  indicates the velocity of particle  $p$  on dimension  $y_i$  at iteration  $t$ .  $vx_{pi}^{t+1}$  would be updated when asset  $i$  is selected by particle (or portfolio)  $p$  at iteration  $t + 1$ , i.e.  $y_{pi}^{t+1} = 1$ .  $Pbest(x)_i^t$  and  $Pbest(y)_i^t$  are the best positions so far for particle  $p$  on dimensions  $x_i$  and  $y_i$ , respectively.  $Gbest(x)_i^t$  and  $Gbest(y)_i^t$  are the best position of the swarm on dimension  $x_i$  and  $y_i$ . Thus, the position of each particle on dimension  $x_i$  can be updated using the Eq. 23.

$$x_{pi}^{t+1} = \begin{cases} x_{pi}^t + vx_{pi}^{t+1} & \text{if } y_{pi}^{t+1} = 1, \\ x_{pi}^t & \text{otherwise.} \end{cases} \quad (23)$$

Since  $y_{pi}^t$  is a binary variable, a discrete variant of PSO will be utilized to update the position of the particles on dimension  $y_i$ , as follows.

$$y_{pi}^{t+1} = \text{round}\left(\frac{1}{1 + e^{-\tau}} - \varpi\right), \quad (24)$$

where  $\tau = y_{pi}^t + vy_{pi}^{t+1}$ , and parameter  $\varpi$  is a small number.

After moving each particle, the feasibility of the new solution must be ensured through the constraint handling process described before and the fitness of new population is evaluated. This procedure continues until the iteration counter reaches the maximum number of iterations  $t_{max}$ . Fig.2 depicts the entire pseudo code of the IPSO-SA algorithm proposed for the solution of cardinality constrained portfolio optimization problem.

#### 4. Computational experiments

In this section, computational results of the IPSO-SA algorithm to find cardinality constrained efficient frontier are discussed. The parameter setting to improve the performance of the proposed algorithm with statistical analysis is implemented. Then, the performance of fine-tuned algorithm is compared with previously proposed algorithms and its positive significant effect on results is proved through ANOVA and a multiple comparison test. To compare the results of the IPSO-SA with previously proposed algorithms, we used the benchmark instances from OR-Library (Beasley, 1996) available at <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files>. The data corresponds to weekly prices

between March 1992 and September 1997 from different well known indices of Hang Seng in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan. The numbers of different assets for each benchmark instance are 31, 85, 89, 98 and 225, respectively. All the results have been computed using the values  $K=10$ ,  $\varepsilon_i=0.01$  and  $\delta_i=1$  ( $i=1,\dots,N$ ) for the problem formulation, and  $\nabla\lambda=0.02$  for the implementation of the algorithms. With  $\nabla\lambda=0.02$ , the number of different  $\lambda$  values, denoted by  $\xi$ , is 51 and the number of particles is set to 50.

---

```

Begin IPSO-SA
 $\lambda=0$ ;
While ( $\lambda \leq 1$ )
  Generate the initial velocity vectors  $v_x$  and  $v_y$ ;
  Generate all particles in the swarm randomly;
  Make feasible particle  $p$  for  $p=1,2,\dots,P$ ;
  Compute fitness value  $f_p$  for  $p=1,2,\dots,P$ ;
  Find the best particle in the swarm ( $G_{best}$ );
  For iteration_count=1 to  $t_{max}$ 
    Calculate  $w^t$  and  $k$ ;
    For  $p=1$  to  $P$ 
      For  $i=1$  to  $N$ 
        Update particle  $p$  on dimension  $y_i$ ;
        If  $y_{pi}=1$ 
          Compute the velocity  $v_{x_{pi}}$ ;
          If  $v_{x_{pi}} + x_{pi} \geq 0$ 
            Update particle  $p$  on dimension  $x_i$ ;
          Else
             $y_{pi}=0$ ;
             $x_{pi}=0$ ;
          End if
        Else
           $x_{pi}=0$ ;
        End if
      End for
    End for
    Run SA for  $G_{best}$ ;
     $\lambda = \lambda + \nabla\lambda$ ;
  End while
End

```

---

**Fig. 2.** IPSO-SA algorithm for the solution of cardinality constrained portfolio optimization problem

In order to test the effectiveness of the proposed algorithm, we compute the heuristic efficient frontier and compare it with standard efficient frontiers. Three error measures are used for the evaluation of the heuristic efficient frontier relative to the standard efficient frontier: mean Euclidean distance, variance of return error and mean return error.

Error measures can be quantified as follows: Let  $(v_i^s, r_i^s)$  ( $i=1, \dots, 2000$ ) be the variance and the mean return of the point in the standard efficient frontier, and let  $(v_j^h, r_j^h)$  ( $j=1, \dots, \xi$ ) be the variance and the mean return of the point in the heuristic efficient frontier. Let  $(v_e^s, r_e^s)$  be the point in the standard efficient frontier that has minimum Euclidean distance from the heuristic point  $(v_j^h, r_j^h)$  where  $e$  is defined as follows,

$$e = \arg \min_{i=1, \dots, 2000} (\sqrt{(v_i^s - v_j^h)^2 + (r_i^s - r_j^h)^2}) \quad j=1, \dots, \xi \quad (25)$$

Therefore mean Euclidean distance, variance of return error and mean return error are defined respectively as follows,

$$\text{Mean Euclidean distance} = \left( \sum_{j=1}^{\xi} \sqrt{(v_e^s - v_j^h)^2 + (r_e^s - r_j^h)^2} \right) \times \frac{1}{\xi}, \quad (26)$$

$$\text{Variance of return error} = \left( \sum_{j=1}^{\xi} 100 \left| \frac{(v_e^s - v_j^h)}{v_j^h} \right| \right) \times \frac{1}{\xi}, \quad (27)$$

$$\text{Mean return error} = \left( \sum_{j=1}^{\xi} 100 \left| \frac{(r_e^s - r_j^h)}{r_j^h} \right| \right) \times \frac{1}{\xi}. \quad (28)$$



#### 4.1 Parameter setting for the proposed algorithm based on statistical analysis

Each meta-heuristic has parameters which affect its solution quality and computation time. In order to improve the efficiency of an algorithm, these parameters should be tuned and set to their best values. A well-known approach to study the effect of different factors and their interactions is design of experiments (DOE) and its associated statistical tool, ANOVA. Analysis of variance (ANOVA) is a method to evaluate whether the impact of different factors on performance is significant or not.

Since considering all parameters of the proposed algorithm and their combinations increases the dimension of the experiment in ANOVA, we choose to study four main parameters for tuning the algorithm. Table 1 shows these parameters and their study levels. Note that cognitive and social parameters of IPSO part ( $c_1$  and  $c_2$  respectively) are set to their default value 2 which perform best for our algorithm.

**Table 1**

Experimental design parameters

| Algorithm parameters |                                    | Factor   | Levels              |
|----------------------|------------------------------------|----------|---------------------|
| PSO parameters       | $w_{max}$                          | <i>a</i> | 1.5, 1.3, 1.1       |
|                      | $w_{min}$                          | <i>b</i> | 0.1, 0.2, 0.3       |
| SA parameters        | Cooldown factor ( $\bar{\alpha}$ ) | <i>c</i> | 0.9, 0.95, 0.99     |
|                      | Neighborhood search type           | <i>d</i> | 1-opt, 2-opt, 3-opt |

A general factorial experiment with four factors, each having three levels and two observations per combination is designed. The ANOVA and multiple comparison tests were performed with SAS software version 9.1 for Hang Seng benchmark data set. Similar results are expected for other benchmark instances. Table 2 shows the results of ANOVA test for parameter tuning based on mean Euclidean distance error values. These results indicate the significant effects of  $w_{max}$  and  $\bar{\alpha}$  factor on the performance of the proposed model. In order to determine the statistically similar and different means, the Student-Newman-Keuls (SNK) test is used to group them. The results of SNK test for significant parameters in Table 3 indicate that  $w_{max} = 1.5$  and  $\bar{\alpha} = 0.9$  or 0.95 present the lowest error measure and subsequently the highest performance. To examine the impact of parameters on computation time of the algorithm, another ANOVA experiment is conducted with parameters in the previous study. The results of factorial ANOVA followed by SNK test suggest 2-opt as the best level for neighborhood search type to improve the efficiency of the algorithm in computation time.

**Table 2**

$3^4$  factorial ANOVA for Hang Seng data set

| Source      | D.F | SS         | MS         | F            | Pr > F |
|-------------|-----|------------|------------|--------------|--------|
| <i>a</i> *  | 2   | 4.6845E-12 | 2.3423E-12 | <b>19.68</b> | <.0001 |
| <i>b</i>    | 2   | 4.9872E-14 | 2.4936E-14 | 0.21         | 0.8114 |
| <i>c</i> *  | 2   | 8.0470E-13 | 4.0235E-13 | <b>3.38</b>  | 0.0389 |
| <i>d</i>    | 2   | 2.8090E-13 | 1.4045E-13 | 1.18         | 0.3125 |
| <i>ab</i>   | 4   | 2.8711E-13 | 7.1777E-14 | 0.6          | 0.6615 |
| <i>ac</i>   | 4   | 8.8577E-13 | 2.2144E-13 | 1.86         | 0.1254 |
| <i>ad</i>   | 4   | 2.5377E-13 | 6.3444E-14 | 0.53         | 0.7118 |
| <i>bc</i>   | 4   | 4.5282E-13 | 1.1320E-13 | 0.95         | 0.439  |
| <i>bd</i>   | 4   | 2.6191E-13 | 6.5478E-14 | 0.55         | 0.6995 |
| <i>cd</i>   | 4   | 4.2956E-13 | 1.0739E-13 | 0.9          | 0.4667 |
| <i>abc</i>  | 8   | 1.2212E-12 | 1.5265E-13 | 1.28         | 0.2643 |
| <i>abd</i>  | 8   | 1.3061E-12 | 1.6326E-13 | 1.37         | 0.2215 |
| <i>acd</i>  | 8   | 3.1257E-13 | 3.9072E-14 | 0.33         | 0.9529 |
| <i>bcd</i>  | 8   | 8.2547E-13 | 1.0318E-13 | 0.87         | 0.5478 |
| <i>abcd</i> | 16  | 1.8770E-12 | 1.1731E-13 | 0.99         | 0.4802 |
| Error       | 81  | 9.6409E-12 | 1.1902E-13 |              |        |
| Total       | 161 | 2.3574E-11 |            |              |        |

(\*) asterisk indicates significance at 5 percent level.

**Table 3**

Results from SNK test for Hang Seng data set

| $w_{max}$ | mean       | SNK grouping | $\bar{\alpha}$ | mean       | SNK grouping |
|-----------|------------|--------------|----------------|------------|--------------|
| 1.5       | 0.0000779  | A            | 0.9            | 0.00007807 | A            |
| 1.1       | 0.00007827 | B            | 0.95           | 0.00007814 | A,B          |
| 1.3       | 0.00007827 | B            | 0.99           | 0.00007824 | B            |

An analysis of the effects of these parameters on the other two error measures, variance of return error and mean return error, with the same approach determined best value for  $w_{min}$  at 0.2 or 0.1 level.

#### 4.2. Numerical results

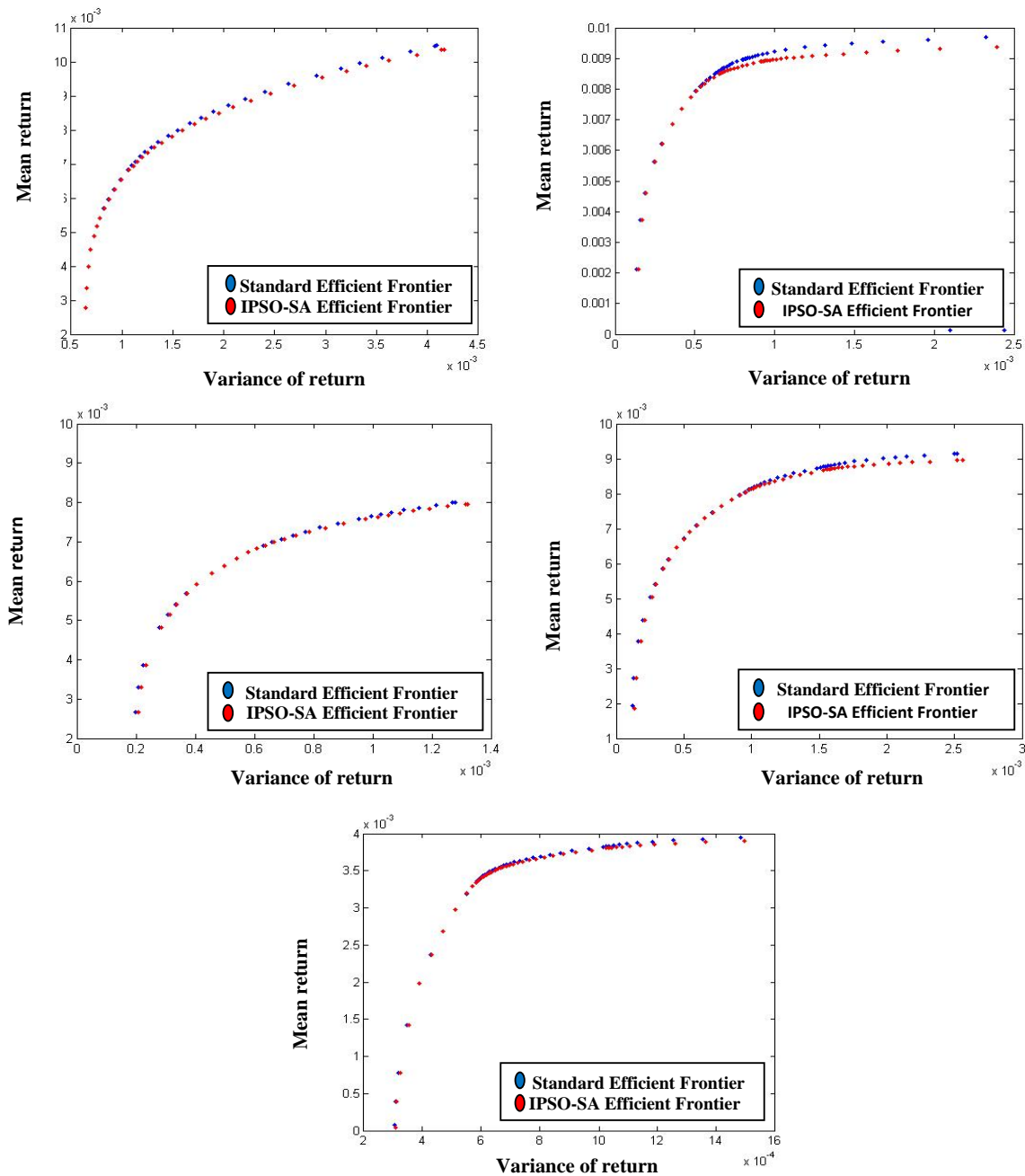
Computational results of IPSO-SA algorithm to find cardinality constrained efficient frontier are presented in this section. To examine the efficiency of the proposed algorithm, the performance of the IPSO-SA has been compared with GA, TS, SA and standard PSO using three defined error measures for five benchmark data sets.

Table 4 shows the comparative results, and Fig.3 illustrates the comparison of efficient frontiers. It is worth to mention that we implement IPSO-SA algorithm in MATLAB 7.6.0 using a core 2 Dou, 2.1 GHz computer with 2 GB of memory. The execution time of IPSO-SA was 183 seconds for Hang Seng, 563 seconds for DAX 100, 767 seconds for FTSE 100, 848 seconds for S&P 100 and about 72 minutes for Nikkei.

**Table 4**

Experimental results of five heuristics on portfolio selection

| Index     | Assets | Errors                      | Chang et al. (2000) |        |        | Cura (2009) | (present paper) |
|-----------|--------|-----------------------------|---------------------|--------|--------|-------------|-----------------|
|           |        |                             | GA                  | TS     | SA     | PSO         | IPSO-SA         |
| Hang Seng | 31     | Mean Euclidean distance     | 0.004               | 0.004  | 0.004  | 0.0049      | 0.0001          |
|           |        | Variance of return error(%) | 1.6441              | 1.6578 | 1.6628 | 2.2421      | 1.6388          |
|           |        | Mean return error(%)        | 0.6072              | 0.6107 | 0.6238 | 0.7427      | 0.6059          |
| DAX 100   | 85     | Mean Euclidean distance     | 0.0076              | 0.0082 | 0.0078 | 0.009       | 0.0001          |
|           |        | Variance of return error(%) | 7.218               | 9.0309 | 8.5485 | 6.8588      | 6.7806          |
|           |        | Mean return error(%)        | 1.2791              | 1.9078 | 1.2817 | 1.5885      | 1.2770          |
| FTSE 100  | 89     | Mean Euclidean distance     | 0.002               | 0.0021 | 0.0021 | 0.0022      | 0.0000          |
|           |        | Variance of return error(%) | 2.866               | 4.0123 | 3.8205 | 3.0596      | 2.4701          |
|           |        | Mean return error(%)        | 0.3277              | 0.3298 | 0.3304 | 0.364       | 0.3247          |
| S&P 100   | 98     | Mean Euclidean distance     | 0.0041              | 0.0041 | 0.0041 | 0.0052      | 0.0001          |
|           |        | Variance of return error(%) | 3.4802              | 5.7139 | 5.4247 | 3.9136      | 2.6281          |
|           |        | Mean return error(%)        | 1.2258              | 0.7125 | 0.8416 | 1.404       | 0.7846          |
| Nikkei    | 225    | Mean Euclidean distance     | 0.0093              | 0.001  | 0.001  | 0.0019      | 0.0000          |
|           |        | Variance of return error(%) | 1.2056              | 1.2431 | 1.2017 | 2.4274      | 0.9583          |
|           |        | Mean return error(%)        | 5.3266              | 0.4207 | 0.4126 | 0.7997      | 1.7090          |



**Fig. 3.** IPSO-SA efficient frontier vs. the standard efficient frontier

#### 4.3. Analysis of results

In this section, we statistically compare the performance of our proposed algorithm with GA, TS, SA and standard PSO using analysis of variance. To test the equality of the performance for different meta-heuristic algorithms, a randomized complete block experiment is designed. In this experiment, the effect of benchmark problems is blocked to reduce experimental error and to provide more reliable results (Coffin & Saltzman, 2000).

The ANOVA for randomized complete block design is performed with five meta-heuristic algorithms as treatments, five benchmark problems as blocks and one observation per cell. The results of ANOVA comparative study of the algorithms are presented in Table 5. In this analysis, the

performance of the algorithms are measured with mean Euclidean distance error. According to ANOVA test, there is a significant difference among the performance of different algorithms.

Table 5  
ANOVA for comparison of algorithms performance

| Source                     | D.F | SS         | MS         | F    | Pr > F |
|----------------------------|-----|------------|------------|------|--------|
| Meta-heuristic algorithms* | 4   | 0.00008481 | 0.00002120 | 5.78 | 0.0045 |
| Test problems*             | 4   | 0.00006644 | 0.00001661 | 4.53 | 0.0123 |
| Error                      | 16  | 0.00005873 | 0.00000367 |      |        |
| Total                      | 24  | 0.00020998 |            |      |        |

We used SNK test for multiple comparisons of mean responses to determine the most efficient algorithms with respect to error measures. SNK grouping revealed that our proposed IPSO-SA algorithm (group A) performs better than GA, TS, SA and standard PSO (group B).

### 5. Concluding remarks

This paper studied a realistic portfolio optimization problem with floor, ceiling, and cardinality constraints, which plays an important role in financial engineering. A new IPSO-SA algorithm based on an improved particle swarm optimization and a modified simulated annealing was proposed to find the cardinality constrained efficient frontier. The proposed algorithm benefits from simple and easy characteristics of PSO with an adaptation of inertia weights and constriction factor approaches. In addition, incorporating SA procedure into IPSO helps escape from local optimum and improve the precision of the convergence.

Computational experiments on five benchmark data sets for up to 225 assets were executed to assess the effectiveness of the IPSO-SA algorithm to solve the cardinality constrained portfolio problem. A parameter setting method with statistical analysis was employed to support the algorithm. Then, the performance of fine-tuned algorithm was compared with previously proposed algorithms and its positive significant effect on results was proved through ANOVA and a multiple comparison test. The results indicate that our proposed algorithm can find high quality solutions with near-zero errors in reasonable amount of time.

### References

- Armañanzas, R., & Lozano, J. A. (2005). A multiobjective approach to the portfolio optimization problem. *Proceedings of the IEEE Congress on Evolutionary Computation*, Edinburgh, UK, 2 1388-1395.
- Arnone, S., Loraschi, A., & Tettamanzi, A. (1993). A genetic approach to portfolio selection. *Neural Network World. International Journal on Neural and Mass-Parallel Computing and Information Systems*, 3(6), 597-604.
- Beasley, J.E. (1996). Obtaining test problems via Internet. *Journal of Global Optimization*, 8(4), 429-433.
- Behnamian, J., Fatemi Ghomi, S. M. T. (2010) Development of a PSO-SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. *Expert Systems and Applications*, 37(2), 974-984.
- Blum, C., & Li, X. (2008). *Swarm Intelligence in Optimization*, in Blum, C. & Merkle, D. (eds.), *Swarm Intelligence - Introduction and Applications*, Springer, 43 – 85.
- Cerny, V. (1985). Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1), 41-51.
- Chang, T. J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimization. *Computers and Operational Research*, 27, 1271-1302.

- Chen, A. L., Yang, G. K., & Wu, Z. M. (2006). Hybrid Discrete Particle Swarm Optimization Algorithm for Capacitated Vehicle Routing Problem. *Journal of Zhejiang University Science A*, 7, 607-614.
- Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6, 58-73.
- Coffin, M. & Saltzman, M. J. (2000). Statistical analysis of computational tests of algorithms and heuristics. *INFORMS Journal on Computing*, 12(1), 24-44.
- Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150, 546-571.
- Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications*, 10(4), 2396-2406.
- Di Tollo, G., & Roli, A. (2008). Metaheuristics for the Portfolio Selection Problem. *International Journal of Operations Research*, 5(1), 13-35.
- Dueck, G., & Winker, P. (1992). New concepts and algorithms for portfolio choice. *Applied Stochastic Models and Data Analysis*, 8, 159-178.
- Eberhart, R. C., & Kennedy, J. (1995b). A new optimizer using particle swarm theory. Proceeding of 6th International Symposium on Micromachine and Human Science. Nagoya, Japan, 39-43.
- Fernandez, A., & Gomez, S. (2007) Portfolio selection using neural networks. *Computers & Operations Research*, 34(4), 1177-1191.
- Gao, J., & Chu, Z. (2010). A new particle swarm optimisation based on MATLAB for portfolio selection problem. *International Journal of Modelling, Identification and Control*, 9(1-2), 206 - 211.
- Gilli, M., K ellezi, E., & Hysi, H. (2006). A data-driven optimization heuristic for downside risk minimization. *The Journal of Risk*, 8(3), 1-19.
- Glover, F., Mulvey, J. M., & Hoyland, K. (1995). Solving dynamic stochastic control problems in finance using tabu search with variable scaling. Proceedings of the Metaheuristics International Conference, Breckenridge, Colorado, 429-448.
- Kennedy, J., & Eberhart, R.C. (1995a). Particle swarm optimization. IEEE International Conference on Neural Networks, 4, 1942-1948.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, P. M. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.
- Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio in optimization model and its application to Tokyo stock market. *Management Science*, 37(5), 519-531.
- Koshino, M., Murata, H., & Kimura, H. (2007). Improved particle swarm optimization and application to portfolio selection. *Electronics and Communications in Japan (Part 3)*, 90(3), 13-25.
- Lee, S. M., & Chesser, D. L. (1980). Goal programming for portfolio selection. *Journal of Portfolio Management*, 6(3), 22-26.
- Liu, B., Wang, L. & Jin, Y.H. (2007). An effective PSO-based memetic algorithm for flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37, 18-27.
- Maringer, D. G. (2001). Optimizing portfolios with ant systems. Proceedings of the International ICSC Congress on Computational Intelligence: Methods and Applications, Bangor, Wales, United Kingdom, 288-294.
- Markowitz, H. M. (1952). Portfolio selection. *Journal of Finance*, 7(1), 77-91.
- Mishra, S. K., Panda, G., & Meher, S. (2009). Multi-objective particle swarm optimization approach to portfolio optimization, World Congress on Nature and Biologically Inspired Computing, Coimbatore, India, 1612 - 1615.

- Mous, L., Dallagnol, V. A. F., Cheung, W., & van den Berg, J. (2006). A comparison of particle swarm optimization and genetic algorithms applied to portfolio selection. Proceedings of the Workshop on Nature Inspired Cooperative Strategies for Optimization NICSO, Granada, Spain, 109-121.
- Peram, T., Veeramachaneni, K., & Mohan, C. K. (2003) Fitness-distance-ratio based particle swarm optimization. Swarm Intelligence Symposium, Indiana, USA, 174-181.
- Rolland, E. (1997). A tabu search method for constrained real number search: Applications to portfolio selection. Technical report, Department of Accounting and Management Information Systems, Ohio State University, Columbus, Ohio.
- Schaerf, A. (2002). Local search techniques for constrained portfolio selection problems. *Computational Economics*, 20, 177-190.
- Shen Q., Shi, W. M., & Kong, W. (2008). Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. *Computational Biology and Chemistry*, 32, 53-60.
- Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. *Proceedings IEEE Congress on Evolutionary Computation*, pp. 69-73.
- Simaan, Y. (1997). Estimation risk in portfolio selection: The mean variance model versus the mean absolute deviation model. *Management Science*, 43(10), 1437-1446.
- Tang, J., Zhang, G., Lin, B., & Zhang, B. (2010). Power mutation embedded modified PSO for global optimization problems. Tan Y., Shi Y. Tan, K.C. (eds.) *Advances in Swarm Intelligence, Lecture Notes in Computer Science*, 6145, 566-573.
- Tasgetiren M.F., Sevkli M., Liang Y.C., Gencyilmaz G. (2004). Particle swarm optimization algorithm for makespan and maximum lateness minimization in permutation flowshop sequencing problem. Proceedings of the fourth international symposium on intelligent manufacturing systems, Sakarya, Turkey, 431-41.
- Xu, F., Chen, W., & Yang, L. (2007). Improved particle swarm optimization for realistic portfolio selection. Proceedings of Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 1, 185-190.
- Yang, X. (2006). Improving portfolio efficiency: A genetic algorithm approach. *Computational Economics*, 28, 1-14.
- Young, M. R. (1998). A minimax portfolio selection rule with linear programming solution. *Management Science*, 44(5), 673-683.