Proceedings of the 23rd IASTED International Multi-Conference
ARTIFICIAL INTELLIGENCE AND APPLICATIONS
February 14-16, 2005, Innsbruck, Austria

A PARTICLE SWARM OPTIMISATION APPROACH IN THE CONSTRUCTION OF OPTIMAL RISKY PORTFOLIOS

Graham Kendall and Yan Su

Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science and IT
The University of Nottingham
United Kingdom
{gxk, yxs}@cs.nott.ac.uk

ABSTRACT

In this paper, we apply particle swarm optimisation to the construction of optimal risky portfolios for financial investments. Constructing an optimal risky portfolio is a high-dimensional constrained optimisation problem where financial investors look for an optimal combination of their investments among different financial assets with the aim of achieving a maximum reward-to-variability ratio. A particle swarm solver is developed and tested on various restricted and unrestricted risky investment portfolios. The particle swarm solver demonstrates high computational efficiency in constructing optimal risky portfolios of less than fifteen assets. The effectiveness of a weighting function in the particle swarm optimisation algorithm is also studied.

KEY WORDS
Swarm Intelligence, Particle Swarm Optimisation, Stock Market, Portfolio Management, Optimal Risky Portfolio.

1. Introduction

Swarm intelligence originates from the study of natural creatures that behave as a swarm in which individuals of the swarm follow simple rules, whereas the swarm exhibits complex, intelligent behaviour. Swarm intelligence research argues against the view that individuals are isolated information-processing entities and stresses the fact that intelligence arises among the interaction of intelligent entities [1].

The study of swarm intelligence has introduced a number of new optimisation techniques into the field of artificial intelligence. Dorigo et al. [2] developed ant colony optimisation (ACO) techniques that mimic ants' finding the shortest path to a food source by depositing pheromone on trails. Eberhart and Kennedy [3] introduced particle swarm optimisation (PSO), which is based on the analogy of birds flocking and fish schooling. PSO has been shown to be powerful, easy to implement, and computationally efficient [1]. In this paper, we apply particle swarm optimisation to a high-dimensional constrained optimisation problem – construction of optimal risky portfolios for financial investments (the ORP problem). A PSO solver is developed and tested on various restricted and unrestricted portfolios. Through experiments, the PSO solver demonstrates high computational efficiency in constructing optimal risky portfolios of less than fifteen assets.

2. Background

The fundamental concept behind particle swarm optimisation algorithms is that individuals in a swarm exchange previous experiences whilst the randomness of moving in the searching space is maintained. To some extent, particle swarm optimisation algorithms are similar to other evolutionary algorithms, such as genetic algorithms, in that all these optimisation algorithms maintain a population of potential solutions. Genetic algorithms, which evolve potential solutions through selection and reproduction, differ to PSO where potential solutions, called *particles*, are flown through the problem hyperspace. The flying of particles in the problem space is controlled by *velocities*. At each iteration, each particle's velocity is stochastically accelerated towards its previous best position and towards a global best position [1].

Initially designed for continuous optimisation problems, PSO was first applied to evolving artificial neural networks (ANNs), and achieved remarkable performance in terms of computational efficiency [4]. Based on the study of PSO and ANNs, PSO has been applied to a wide range of optimisation problems, such as human tremor analysis [5] and end milling of metal removal operation in manufacturing environments [6]. The PSO technique has been shown to be fast and accurate. There are different variations of PSO that aim to widen its applicability. Kennedy and Eberhart [7] describe a discrete binary version of the PSO algorithm. Yoshida et al. [8] describe a modified version of the continuous PSO algorithm, which is able to handle both discrete and continuous variables, for reactive power and voltage control problems. Eberhart and Shi [9] present a review on the developments and applications of particle swarm optimisation technique.

3. Optimal Risky Portfolios

A fundamental principle of financial investments is *diversification* where investors diversify their investments into different types of assets. Portfolio diversification minimises investors' exposure to risks, and maximises returns on portfolios. The Markowitz Mean-and-Variance model [10] for security selection of risky portfolio construction is described as below. Using a portfolio (E_p) with two risky assets, E_1 and E_2, as an example, assume the expected returns of the two risky assets are $E_1(r)$ and $E_2(r)$. The standard deviations of the two risky assets are σ_1 and σ_2. The covariance between E_1 and E_2 is *Cov(r₁, r₂)*. The expected return of the risky portfolio E_p, $E(r_p)$, is calculated using (1) below:

$$E(r_p) = w_1 E_1(r) + w_2 E_2(r) \tag{1}$$

Where w_1 is the weight of E_1 in the risky portfolio, w_2 is the weight of E_2 in the risky portfolio. The variance (σ_p^2) of the risky portfolio is calculated as shown in (2):

$$\sigma_p^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2Cov(r_{1,}r_2) \tag{2}$$

By varying the weights of E_1 and $E_{2,}$ i.e., w_1 and w_2, we will have a series values of $E(r_p)$ (risky portfolio' expected return) and σ_p (risky portfolio' standard deviation). A reward-to-variability ratio (S) will be calculated for each possible combination of w_1 and w_2 by using (3):

$$S = \frac{E_{r_p} - r_F}{\sigma_p} \tag{3}$$

where r_F is the expected return of the risk-free asset if there is any risk-free assets in the complete portfolio.

The simple two-asset risky portfolio described above can be extended to risky portfolios with any number of risky assets. The calculation of expected return ($E(r_p)$) of a multiple-asset risky portfolio is similar to (1). The calculation of standard deviation (σ_p) of a multiple-asset risky portfolio uses a border-multiplied covariance matrix (please refer to [11] for more details). There are two types of risky portfolios. *Unrestricted risky portfolios* do not have constraints on the short selling of stocks, i.e., investors can choose to sell a stock that the investor does not own, on the condition that the investor must buy it back after a time of period, hopefully at a lower price. In other words, for unrestricted risky portfolios, assets could have negative weights. *Restricted risky portfolios* place constraints on the short selling of portfolios' underlying equities, and require that all underlying assets must have positive weights. Both unrestricted optimal risky portfolios and restricted optimal risky portfolios must also satisfy another constraint, i.e., the total weights of all assets must sum to 1. To construct an optimal risky portfolio is to find the optimal combination of all assets in order to achieve the maximum reward-to-variability ratio. Mathematically, the restricted ORP problem for a risky portfolio with *N* assets is defined as (4):

$$\begin{aligned} &\underset{w_i}{Max}\ \frac{E(r_P) - r_F}{\sigma_P} \\ &s.t.\ \sum_{i=1}^{N} w_i = 1\ \&\ w_i \geq 0; i \in (0, N] \end{aligned} \tag{4}$$

The unrestricted ORP problem is defined as (5):

$$\begin{aligned} &\underset{w_i}{Max}\ \frac{E(r_P) - r_F}{\sigma_P} \\ &s.t.\ \sum_{i=1}^{N} w_i = 1 \end{aligned} \tag{5}$$

As the number of assets in the risky portfolio increases, construction of an optimal risky portfolio becomes an increasingly high-dimensional optimisation problem with a variety of constraints.

4. The Particle Swarm Solver

The major feature of PSO algorithms is their simplicity in implementation and high computational efficiency in solving optimisation problems. We implement a PSO solver for constructing optimal risky portfolios using the basic form of the PSO algorithm as described in [1]. Initially, a population of particles is generated satisfying all the constraints. A particle here essentially represents a possible portfolio combination. At each iteration, a particle moves to a new position in the problem space as shown below:

$$v_{i,j}^{k+1} = wv_{i,j}^{k} + c_1 rand_1(pbest - s_{i,j}^{k}) + c_2 rand_2(gbest - s_{i,j}^{k}) \tag{6}$$

where $v_{i,j}^{k+1}$ is particle *i*'s velocity on the *j*th dimension at iteration *k*+1. $v_{i,j}^{k}$ is particle *i*'s velocity on the *j*th dimension at iteration *k*. *w* is a weighting function. C_1 and C_2 are weighting factors of values of 2.0 [12]. $s_{i,j}^{k}$ is particle *i*'s position on the *j*th dimension at iteration *k*. *pbest* is the historical individual best position of particle *i*. *gbest* is the historical global best position of the swarm. Weighting function (*w*) is calculated using (7):

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{7}$$

where $w_{\max}$ is an initial weight of value 0.9, and $w_{\min}$ is the final weight of value 0.4 [12]. $iter_{\max}$ is the maximum number of iterations. *iter* is the current iteration number. Finally, the new position of particle *i*, s_i^{k+1}, is calculated as shown in (8):

$$s_i^{k+1} = s_i^{k} + v_i^{k+1} \tag{8}$$

Fig. 1 Restricted Optimal Risky Portfolio (Stock Indexes of Seven Countries) – PSO Solver vs. Excel Solver

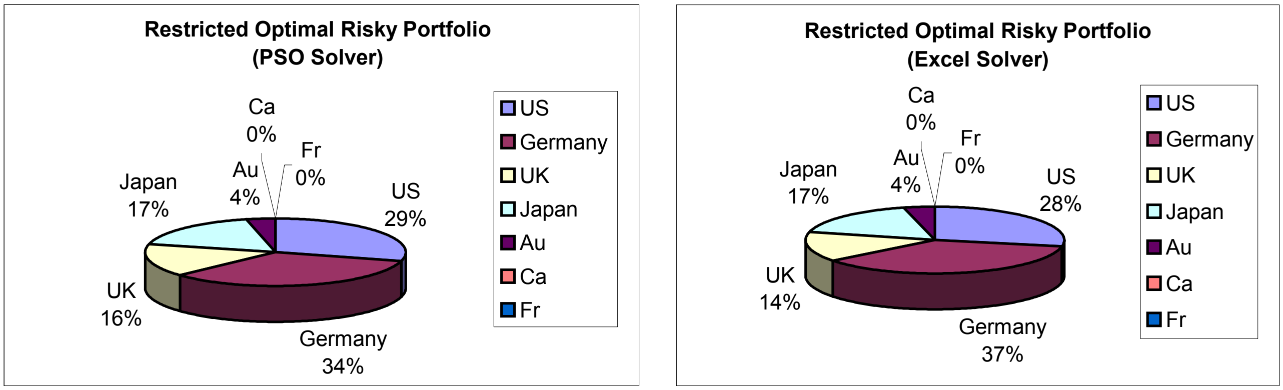


Table I. Restricted Optimal Risky Portfolio of Stock Indexes of Seven Countries (PSO Solver)

Assets	US	Germany	UK	Japan	Australia	Canada	France
E(r) (%)	15.7	21.7	18.3	17.3	14.8	10.5	17.2
SD (%)	21.1	25.0	23.5	26.6	27.6	23.4	26.6
Weights (Optimal)	0.2916	0.3391	0.1595	0.1726	0.037	0.0002	0.0000
PSO Solver	E(r) (%)	SD (%)	Ratio	Excel Solver	E(r) (%)	SD (%)	Ratio
	18.4	17.69	1.04014		18.5	17.79	1.03991

Fig. 2 Unrestricted Optimal Risky Portfolio (5 stocks) – PSO Solver vs. Excel Solver

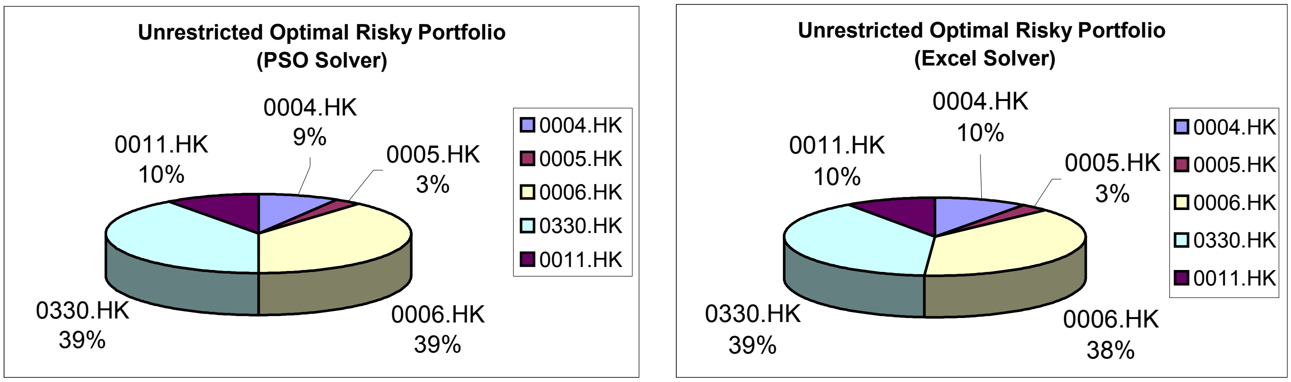


Table II. Unrestricted Optimal Risky Portfolio of 5 stocks (PSO Solver)

Assets	0004.HK	0005.HK	0006.HK	0330.HK	0011.HK
E(r) (%)	1.86	0.43	0.41	4.42	1.13
SD (%)	14.47	11.78	4.41	11.63	7.27
Weights (Optimal)	-0.1250	0.0377	0.4769	0.4900	0.1204

PSO Solver	E(r) (%)	SD (%)	Ratio	Excel Solver	E(r) (%)	SD (%)	Ratio
	2.3	5.38	0.4089		2.28	5.37	0.406

For restricted risky portfolios, the *pbest* and *gbest* are evaluated using (4). For unrestricted risky portfolios, the *pbest* and *gbest* are evaluated using (5). Whenever a particle flies to a new position in the problem space, all the constraints on the portfolio are satisfied to ensure a valid move.

5. Experiments and Discussion

The PSO solver is tested on one restricted risky portfolio with no risk-free assets, and three unrestricted risky portfolios with one risk-free asset. Table I shows the restricted risky portfolio of seven countries' stock indexes, together with each individual index's yearly expected returns (E(r)) and standard deviations (SD). Restricted portfolios requires no short sellings on the portfolio's underlying assets. In other words, the weights of individual assets in the restricted portfolio must be in the range of [0, 1], which conform to the optimal weights items in table I that shows the composition of the optimal risky portfolio of seven countries' stock indexes developed by the PSO solver. Figure 1 visually compares the optimal risky portfolio evolved by the PSO solver, and the optimal risky portfolio solved by using the traditional excel solver [13]. We used 50 particles with 500 iterations for the seven countries' stock indexes restricted portfolio. The program used 2.091 minutes. Clearly, the PSO solver is efficient in finding the restricted optimal risky portfolio and record a better reward-to-variability ratio as shown in table I.

Fig. 3 Unrestricted Optimal Risky Portfolio (12 stocks) – PSO Solver vs. Excel Solver

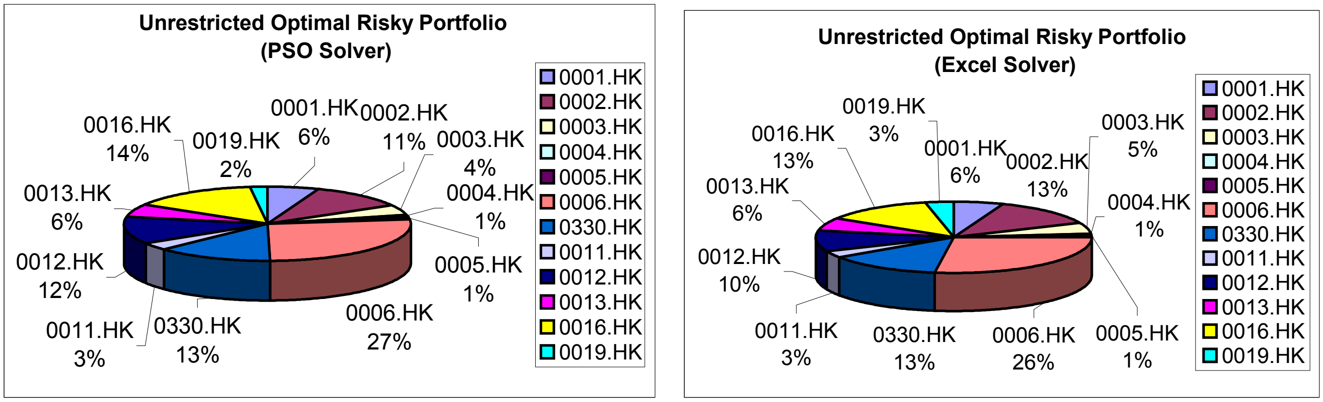


Table III. Unrestricted Optimal Risky Portfolio of 12 stocks (PSO Solver)

Assets	0001.HK	0002.HK	0003.HK	0004.HK	0005.HK	0006.HK		E(r)(%)	2.71
E(r)(%)	0.93	-0.05	0.22	1.86	0.43	0.41	PSO Solver	SD(%)	5.3
SD(%)	10.42	5.06	6.09	14.47	11.78	4.41			
Weights (Optimal)	-0.2063	-0.3775	-0.1518	-0.0376	0.0369	0.9333		Ratio	0.49245
Assets	0330.HK	0011.HK	0012.HK	0013.HK	0016.HK	0019.HK		E(r)(%)	2.87
E(r)(%)	4.42	1.13	0.66	0.75	1.17	0.92	Excel Solver	SD(%)	5.63
SD(%)	11.63	7.27	12.89	9.55	11.50	9.36			
Weights (Optimal)	0.4586	0.1201	-0.4103	0.2041	0.5003	-0.0698		Ratio	0.492

Fig. 4 Unrestricted Optimal Risky Portfolio (20 stocks) – PSO Solver vs. Excel Solver

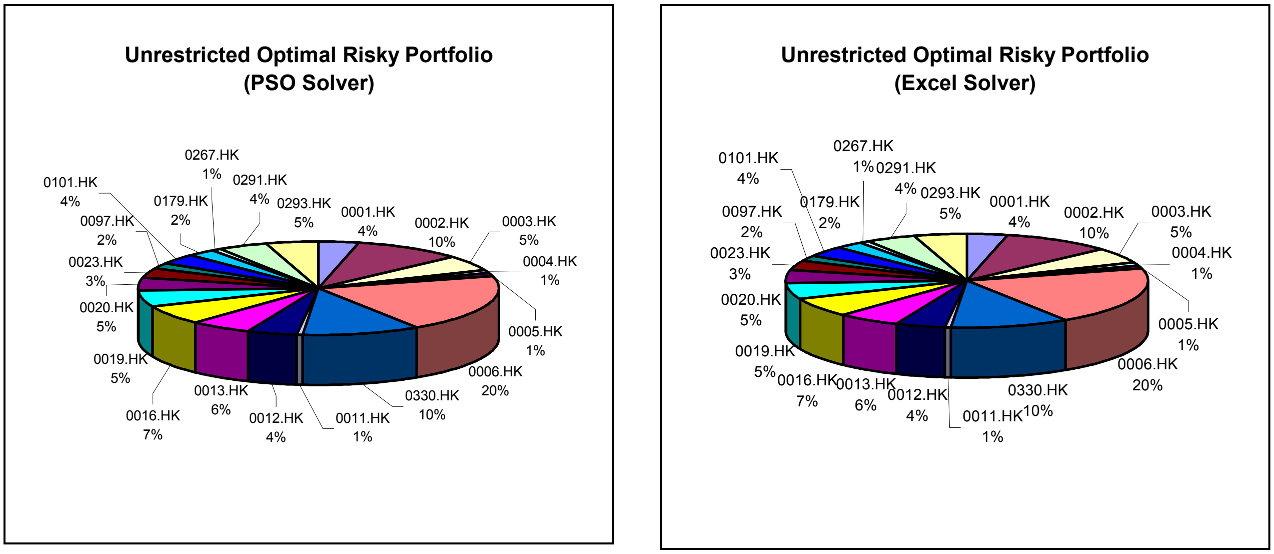


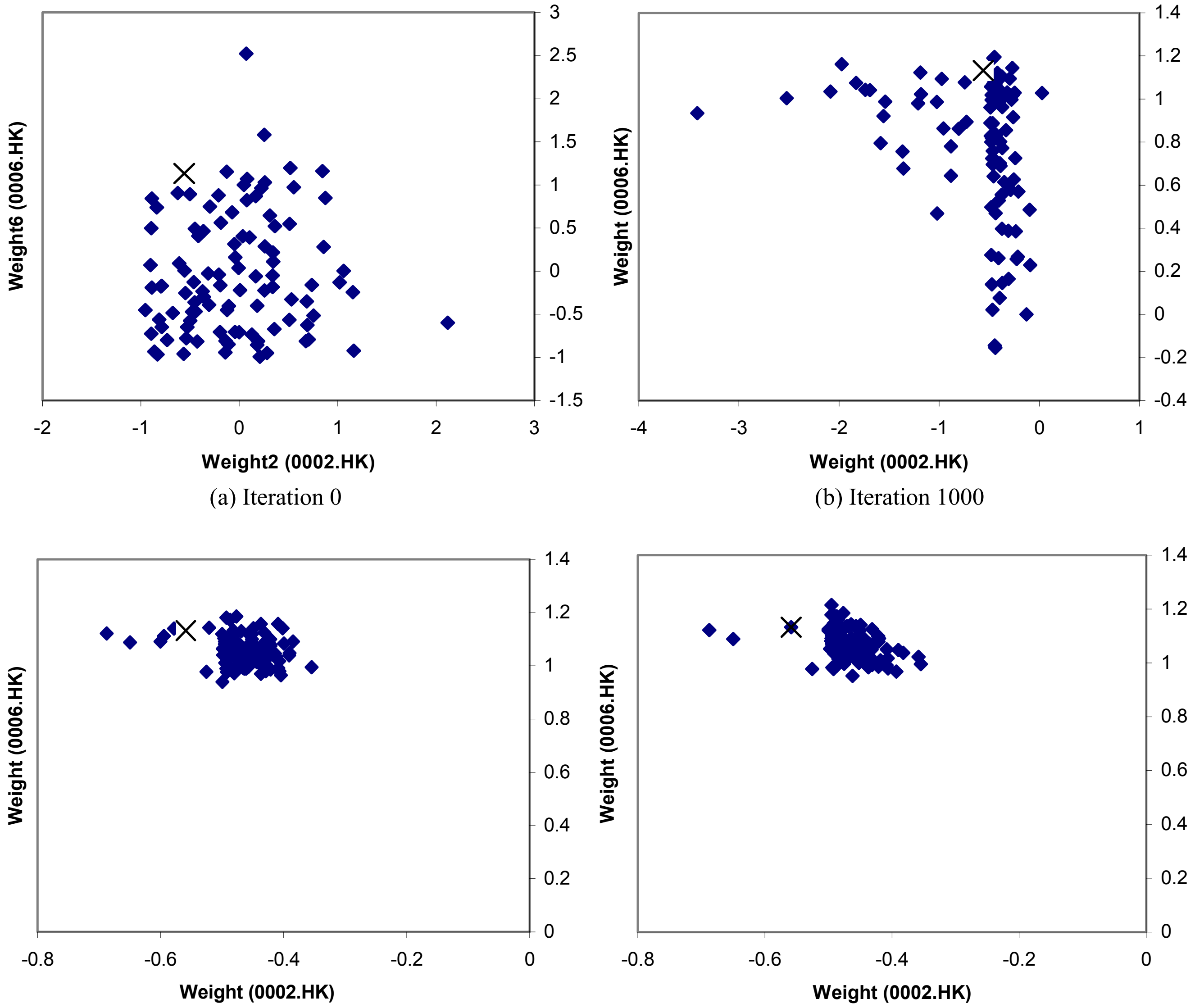
Table II, III, and IV demonstrate three unrestricted portfolios of 5 stocks, 12 stocks, and 20 stocks respectively. All stocks are randomly selected from the Hang Seng Index in the Hong Kong Stock Exchange. Individual stock's expected returns and standard deviations given in table II, III, and IV are based on each individual stock's historical monthly returns from 5 October 1998 to 2 October 2003. Unrestricted portfolios do not have constraints on short selling. In other words, the proportion of an asset in the portfolio could be negative, or greater than 1. As an example, in table IV, asset 0002.HK has a negative expected return. The optimal risk portfolio developed by the PSO solver chooses to short sell the stock 0002.HK with a weighting of –55.89% in the risky portfolio. The capital gained by short selling on unpromising stocks is used to invest more in promising stocks, such as asset 0006.HK in table IV with a weighting of 113.26% in the risky portfolio. Note, that even though the asset 0006.HK does not have a very strong expected return; but it has the smallest standard deviation among the 20 stocks, in other words, the safest asset in the risky portfolio. We also include a risk-free asset in the complete portfolio, i.e., a complete portfolio with both risk-free and risky assets. We are only investigating the construction of optimal risky portfolios problem. The study of constructing optimal complete portfolios is the subject of our future work. The risk-free asset used in unrestricted portfolios is the U.S. Treasury

Table IV. Unrestricted Optimal Risky Portfolio (20 stocks) – PSO Solver vs. Excel Solver

Assets	0001.HK	0002.HK	0003.HK	0004.HK	0005.HK	0006.HK	0330.HK	0011.HK	0012.HK	0013.HK
E(r)(%)	0.93	-0.05	0.22	1.86	0.43	0.41	4.42	1.13	0.66	0.75
SD(%)	10.42	5.06	6.09	14.47	11.78	4.41	11.63	7.27	12.89	9.55
Weights (Optimal)	-0.1987	-0.5589	-0.2865	0.0742	0.0641	1.1326	0.5865	-0.0307	-0.2496	0.3153
Assets	0016.HK	0019.HK	0020.HK	0023.HK	0097.HK	0101.HK	0179.HK	0267.HK	0291.HK	0293.HK
E(r)(%)	1.17	0.92	1.42	1.65	1.75	1.58	1.11	0.81	0.72	1.66
SD(%)	11.50	9.36	13.22	10.86	13.90	10.42	15.89	12.31	12.72	10.86
Weights (Optimal)	0.3832	-0.3069	-0.2639	0.1735	0.1231	0.2025	-0.1375	-0.049	-0.2444	0.2711

	E(r) (%)	SD (%)	Ratio		E(r) (%)	SD (%)	Ratio
PSO Solver	3.77	6.20	0.5919	Excel Solver	3.53	5.8	0.5914

Fig. 5 Particle Swarm Optimisation of 20-stock Risky Portfolio (2-dimension)



(a) Iteration 0

(b) Iteration 1000

(c) Iteration 3000

(d) Iteration 4830

Bill with a monthly yield to maturity of 0.1%.

Optimal weights items in table II, III, IV give the compositions of the optimal risky portfolios developed by our PSO solver for the 5-stock portfolio, 12-stock portfolio, and 20-stock portfolio respectively. Figures 2, 3, and 4 compare the optimal risky portfolios evolved by the PSO solver, and the optimal risky portfolios solved by the excel solver. For the 5-stock portfolio, we used 30 particles with 300 iterations. PSO solver used 0.201 minutes to find the optimal risky portfolio. For the 12-stock portfolio, we used 300 particles with 800 iterations of evolution. The program took 4.561 minutes. For the 20-stock portfolio, we used 800 particles and 5000 iterations. It took the PSO solver 1.66 hours to find the optimal risky portfolio. Through the experiments on unrestricted portfolios, the PSO solver demonstrated clear efficiency in finding optimal risky portfolios for portfolios with less than fifteen assets, but the search time increases rapidly for larger portfolios.

In summary, the traditional excel solver is unable to solve multiple-asset portfolios with large negative correlations, and is limited in the number of assets it can handle [11, 13]. Our PSO solver does not have constraints on the inputs to the evolutionary system and has no limitations on the number of assets included in the target portfolio. Our experiments on various restricted and unrestricted portfolios clearly demonstrate the efficiency of particle swarm optimisation technique in solving high-dimensional constrained optimisation problems. The downside, with large portfolios, for the PSO solver is the time needed for the evolutionary process. This is a common problem with most of the evolutionary approaches for optimisation problems.

In order to examine the particle swarm optimisation algorithm more closely, we choose the experiments on 20-asset portfolio as an example. We choose two dimensions from the 20-dimension problem space: weighting on asset 0002.HK and weighting on asset 0006.HK, and plot the behaviour of a swarm of 100 particles (particle 200 to particle 299) in Figure 5(a) to 5(d). Figure 5(a) shows the initial status of the swarm where particles are randomly scattered in the search space. The small cross (X) in Figure 5(a) to 5(d) shows the final optimal weights of asset 0002.HK and 0006.HK found by the PSO solver. After 1000 iterations, the swarm of particles presents a certain pattern of movement, as shown in Figure 5(b), which, interestingly, resembles bird flocking in the nature. Figure 5(c) shows the status of the swarm at iteration 3000 where we see most of the particles have successfully moved close to the target. This mimics the landing of birds on the ground when food is found. At iteration 4830, the first particle landed on the target as shown in Figure 5(d).

By closely examining Figure 5(c) and 5(d), we see the shape of the swarm has changed showing a tendency to move onto the target. However, there seems to be a threshold around –0.5 (x-axis) that most of the particles hardly overcame. We argue that the reason behind this is the weighting function used in the PSO algorithm. As shown in (7) in Sect. 4, the value of w decreases rapidly while the current iteration number increases. In other words, at the latter stages of evolution, the weighting function has little impact in (6). As discussed in [1], the weighting function is responsible for the randomness of particles' movement, while the other two items of the right-hand side of (6) are responsible for the convergence to *pbest* and *gbest*. For a small search space, with less dimensions for optimisation, a decreasing weighting function may be just what the evolution needs, i.e., converge to the optimal solution. However, for a large search space with a large number of optimisation dimensions, a rapid decreasing weighting function may result in premature convergence due to the lack of randomness in the latter stages of the evolutionary process. This is one possible aspect where the efficiency of the PSO solver could be improved for large portfolios with multiple assets.

For our future work, we intend to investigate the relationship between the weighting function and premature convergence of particle swarm optimisation algorithms, and methods for improving the efficiency of the PSO solver for large portfolios. The problem of constructing optimal risky portfolios also serves as a good platform for the study of the efficiency of different evolutionary algorithms, such as genetic algorithms and memetic algorithms. We also intend to use the PSO solver as a basis, and employ other evolutionary techniques, for the study of intelligent portfolio management for financial investments.

References:

[1] J. Kennedy, R. C. Eberhart, with Y. Shi, *Swarm intelligence* (San Francisco: Morgan Kaufmann Publishers, 2001).

[2] M. Dorigo, A. Coloni, and V. Maniezzo, Distributed optimization by ant colonies. *Proc. of First European Conference on Artificial Life*, 1991, 134-142.

[3] R. Eberhart and J. Kennedy, Particle swarm optimisation. *Proc. of IEEE International Conference on Neural Networks (ICNN'95)*, 1995, 1942-1948.

[4] R. C. Eberhart and Y. Shi, Evolving artificial neural networks. *Proc. 1998 Int'l Conf. On Neural Networks and Brain*, Beijing, P.R.C., 1998, PL5-PL13.

[5] R. C. Eberhart and X. Hu, Human tremor analysis using particle swarm optimization. *Proc. of the 1999 Congress on Evolutionary Computation*, Washington, DC, NJ: IEEE Service Center, 1999, 1927-1930.

[6] V. Tandon, Closing the gap between CAD/CAM and optimzed end milling. Master's thesis, Purdue School of Engineering and Technology, University Indianapolis.

[7] J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm optimization algorithm. *Proc. of the 1997 conference on Systems, Man, and Cybernetics (SMC'97)*, 1997, 4104-4109.

[8] H. Yoshida, K. Kawata, and Y. Fukuyama, A particle swarm optimisation for reactive power and voltage control considering voltage security assessment, *IEEE Transaction on Power Systems, 15*(4), 2001, 1232-1239.

[9] R. C. Eberhart and Y. Shi, Particle swarm optimisation: developments, applications, resources. *Proc. of Congress on Evolutionary Computation*, 2001, 81-86.

[10] H. Markowitz, Portfolio Selection, *Journal of Finance*, 7(1), 1952, 77-91.

[11] Z. Bodie, A. Kane, and A. J. Marcus, *Investments* (NY: McGraw-Hill/Irwin, 2002).

[12] Y. Shi and R. C. Eberhart, Parameter selection in particle swarm optimization. *Proc. of the 7th Annual Conference on Evolutionary Programming*, 1998, 591-600.

[13] C. W. Holden, *Excel modelling in investments* (NJ: Pearson Prentice Hall, 2004).