Chapter 6
Particle Swarm Optimization: The Foundation

Dadabada Pradeep Kumar

Abstract Particle swarm optimization (PSO) is a very much popular swarm intelligence algorithm. Since its inception in the year 1995, it is being applied to solve optimization problems in many domains, including portfolio optimization. This chapter lays the basic PSO foundation and introduces existing PSO variants for researchers who want to solve the portfolio optimization problem. It starts with the introduction of PSO, describing the advantages, disadvantages, and applied areas of PSO. Later, the basic PSO procedure and its parameter selection mechanisms are presented. The chapter also presents three popular applications of PSO in finance, including portfolio optimization. Finally, the chapter ends by introducing the existing PSO variants to solve the portfolio optimization problem.

Keywords Portfolio optimization · PSO algorithm · Applications · Fitness · Position update · Velocity update · Swarm intelligence

6.1 Introduction

Optimization aims at obtaining optimal solutions to a problem from a set of feasible solutions based on one or several criteria. Optimization techniques cover large application areas in business, finance, service, industry, engineering, and computer science. For example, *portfolio optimization* is an optimization problem to select the best portfolio (asset distribution) with the objectives of maximizing factors such as expected return and minimizing costs like financial risk. Constraints, if any, can help in reducing the search space of feasible solutions. The *global optimal solution*, if possibly found, can be the best solution to the problem. However, sometimes, suboptimal solutions can also be considered the possible optimal solutions to the problem (Parsopoulos & Vrahatis, 2010).

D. P. Kumar (✉)
Area of Information Systems and Analytics, Indian Institute of Management Shillong, Shillong, Meghalaya, India

B. A. Mercangöz (ed.), *Applying Particle Swarm Optimization*, International Series in Operations Research & Management Science 306,
https://doi.org/10.1007/978-3-030-70281-6_6

Swarm intelligence (SI) is a distributed, intelligent computing mechanism for solving optimization problems. SI took its inspiration from the flocking of birds, swarming, and herding phenomenon invertebrates. Sometimes SI is considered a part of evolutionary computing, as it shares many similarities with it. SI starts working with individuals, where each individual tries to find out the optimal solution. The solution is shared among individuals, and then each individual improves themselves based on the information gathered from others. The most crucial SI property is that all the individuals work in a coordinated way without a coordinator's presence.

PSO is a population-based SI algorithm developed by Eberhart and Kennedy in 1995, inspired by the social behavior of bird flocking or fish schooling (Eberhart & Kennedy, 1995). From its inception, it is attracting a lot of researchers to solve optimization problems in different domains. In the beginning, PSO can only handle real-valued problems. Later, it has been extended to cover both binary and discrete problems (Eberhart & Shi, 2004).

PSO is a meta-heuristic algorithm that deals with a population of random solutions (particles). Each particle in PSO flies through the search space with a dynamically adjusted velocity and positions according to its own and its companion's historical behaviors. The particles move to optimal positions based on objective functions.

PSO is the most popular algorithm in comparison with other evolutionary algorithms (AlRashidi & El-Hawary, 2008; Eberhart & Shi, 2004; Pradeepkumar & Ravi, 2014, 2017; Ravi, Pradeepkumar, & Deb, 2017) as it is:

1. Very intuitive and flexible.
2. Less sensitive to the nature of the objective function.
3. Able to handle objective functions with stochastic nature.
4. Derivative-free.
5. Easy to comprehend and implement.
6. With the requirement of fewer user-defined parameters to tweak.
7. Without the requirement of a good initial solution to start its iteration process. However, PSO also has disadvantages. These include:

   – It does not always guarantee the optimal solution to the problem than the dynamic programming approach; instead, it results in a near-optimal solution.
   – It is slow to convergence in the refined search stage (weak local searchability).

As it is advantageous to apply, PSO is used in various domains involving optimization problems such as antennas, biomedicine, communication networks, clustering and classification, combinatorial optimization, control, design, distribution networks, electronics and electromagnetics, engines and motors, entertainment, fault diagnosis and recovery, finance, fuzzy and neuro-fuzzy systems, graphics and visualization, image and video analysis, metallurgy, modelling, neural networks, prediction and forecasting, power systems and plants, robotics, scheduling, security and military, sensor networks, and signal processing (AlRashidi & El-Hawary, 2008; Poli, 2008; Poli, Kennedy, & Blackwell, 2007; Pradeepkumar & Ravi, 2018).

6.2 Background

The particle swarm concept originated with the effort of Reeves (1983), who came up with the idea of particles. These particles are considered as independent entities that work in harmony to achieve the objective. Reynolds (1987) then added a concept of communication between the particles' social behavior, with the help of a flocking algorithm, whereby each particle adheres to the flocking rules. Later, Nowak, Szamrej, and Latané (1990) also helped us understand the principles underlying how particles are affected by the social environment. In addition to this, Heppner and Grenander (1990) related a roost concept, i.e., the flock aims for some roosting area. In these systems, the particles are autonomous, but a class of rules regulates their movements. These observations on collective behaviors in these social animals led to implementing this model to solve different optimization problems.

6.3 The Basic PSO Algorithm

The PSO technique encompasses the following features. PSO is a metaheuristic because it makes almost nil or very few inferences about the optimization problem. It can search for vast space with distributed candidate solutions. PSO exhibits SI in its optimization process. It mainly follows five fundamental principles observed in SI-based algorithms. Mark Millonas (1993) has stated these principles are followed by the particles while communicating with other fellow particles in the swarm.

In the procedure of basic PSO and its variants, a population of particles in the n-dimensional search space gets initialized randomly. Each particle represents a possible solution. Let $X_i = X_{i,1}, X_{i,2}, \ldots, X_{i,d}, \ldots, X_{N\,p}$ be a vector denoting the position and $V_i = V_{i,1}, V_{i,2}, \ldots, V_{i,d}, \ldots, V_{N\,p}$ be a vector denoting velocity of particle i. A particle's position and velocity can be updated dynamically until optimal values are obtained. The basic PSO procedure is depicted by a flowchart (see Fig. 6.1) and described in Algorithm 6.1, and the notations used in the algorithm are presented in Table 6.1.

It is worth noting that the updated equations, Eqs (1) and (2), are stochastic. As the velocities are getting updated dynamically, they may become too high, leading particles to become uncontrolled. Therefore, the V_{max} (Eberhart, Shi, & Kennedy, 2001), as in Eq. (6.3), helps in restricting the uncontrolled movement of particles in search space.

A parameter, namely, inertia weight (G) as in Eq. (6.4) (Shi & Eberhart, 1998a, 1999), helps in adjusting the trade-off between explorative and exploitative capabilities of PSO. The lesser the inertia weight is, the more the PSO's exploration capability will be and vice versa. And also, Clerc and Kennedy (2002) introduced constriction factor y, as in Eq. (6.5), which ensured convergence and improved the convergence rate of PSO.

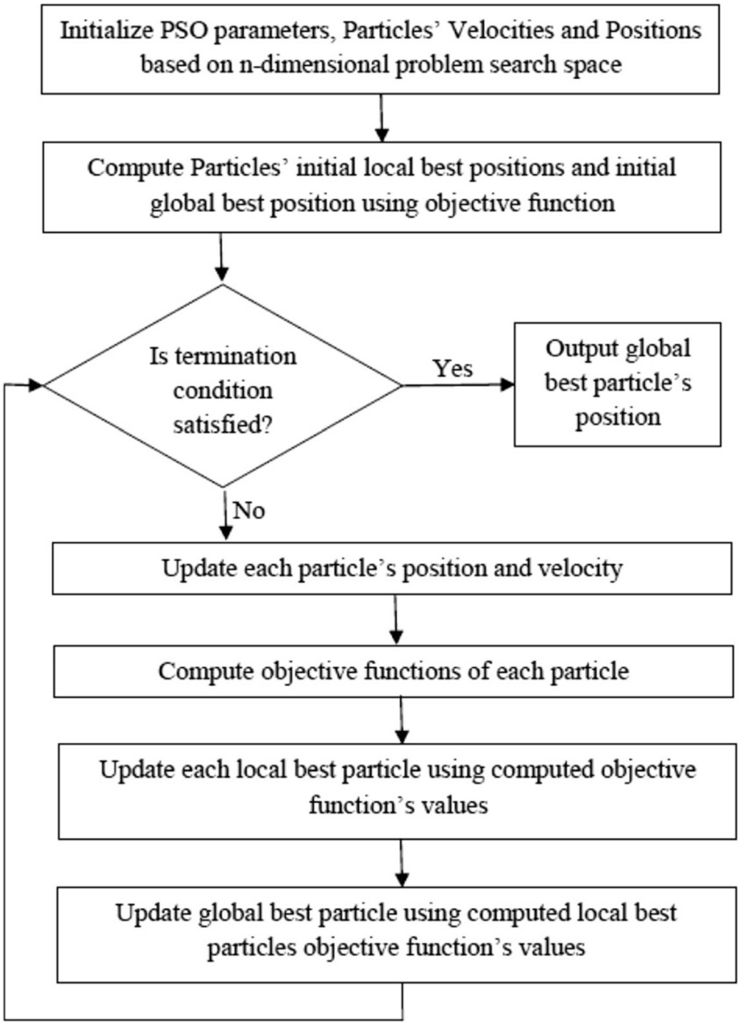

Fig. 6.1 Flowchart of particle swarm optimization's procedure

6.4 Parameter Selection

Shi and Eberhart (1998b), Rezaee Jordehi and Jasni (2013), and Wang, Tan, and Liu (2018) surveyed and presented various parameter selection methods found in the literature. Table 6.2 shows PSO parameters, the purpose of each parameter, and possible values or selection methods for each of these parameters. These parameter selections can help in achieving the best output from PSO. Furthermore, sensitivity analysis (Bartz-Beielstein, Parsopoulos, & Vrahatis, 2002), regression trees (Bartz-Beielstein, Parsopoulos, Vegt, & Vrahatis, 2004), and statistics (Bartz-Beielstein, Parsopoulos, & Vrahatis, 2004) can help in selecting the optimal parameters of the PSO algorithm so that PSO algorithm can solve practical problems better.

Algorithm 6.1: Particle Swarm Optimization

Input: $X[\,][\,]$; Position Matrix, $V[\,][\,]$; Velocity Matrix, $f(.)$; Objective function
Output: P_g; Global best particle
for *each i in* $\{1, 2, \ldots Np\}$ **do**
    for *each d in* $\{1, 2, \ldots n\}$ **do**
        //UB= Upper Boundary, LB=Lower Boundary of search space
        Initialize $X_{i,d}$ with a uniformly distributed random vector with (LB, UB)
        Initialize $V_{i,d}$ with a uniformly distributed random vector with $(-|UB - LB|, |UB - LB|)$
    $P_i = X_i$
$P_g = P_1$
for *each i in* $\{2, 3, \ldots Np\}$ **do**
    if $f(P_i)$ *is better than* $f(P_g)$ **then**
        $P_g = P_i$
        $gbest = f(P_g)$
for *each t in* $\{1, 2, \ldots MaxIterations\}$ **do**
    for *each i in* $\{1, 2, \ldots Np\}$ **do**
        for *each d in* $\{1, 2, \ldots n\}$ **do**
            /*C_1 and C_2 are two positive numbers, and r_{1d} and r_{2d} are two random numbers with uniform distribution in the interval [0,1].*/

$$V_{i,d}(t+1) = V_{i,d}(t) + C_1 r_{1d}(P_{i,d} - X_{i,d}) + C_2 r_{2d}(P_{gd} - X_{i,d}) \tag{6.1}$$

$$X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \tag{6.2}$$

        //Update the particle's best known position
        if $f(X_i)$ *is better than* $f(P_i)$ **then**
            $P_i = X_i$
            $lbest = f(P_i)$
        //Update the swarm's best known position
        if $f(P_i)$ *is better than* $f(P_g)$ **then**
            $P_g = P_i$
            $gbest = f(P_g)$

$$\text{If } |V_{i,d}| > V_{\max}, \text{then } V_{i,d} - \text{sign}(V_{i,d})V_{\max} \tag{6.3}$$

$$V_{i,d}(t+1) = \omega V_{i,d}(t) + C_1 r_{1d}(P_{i,d} - X_{i,d}) + C_2 r_{2d}\left(P_{gd} - X_{i,d}\right) \tag{6.4}$$

$$V_{i,d}(t+1) = \chi\left(V_{i,d}(t) + C_1 r_{1d}(P_{i,d} - X_{i,d}) + C_2 r_{2d}\left(P_{gd} - X_{i,d}\right)\right) \tag{6.5}$$

Table 6.1 Notations and their interpretation

Notation	Interpretation
X_i	Position vector of particle i
$X_{i,d}$	dth dimension of X_i
V_i	Velocity vector of particle i
V	dth dimension of V
$f(X_i)$	Objective function value of X_i
n	Dimension of problem in hand
t	Iteration number or time step
C_1	Cognitive acceleration coefficient
C_2	Social acceleration coefficient
P_i	The best position vector of particle i so far (local best position)
P_{best}	The best objective of particle i so far (local best fitness)
P_g	The best position vector of swarm particles so far (global best position)
g_{best}	The best objective of swarm particles (global best fitness)
$V_{\max}$	Maximum allowable velocity for particles
ω	Inertia weight
χ	Construction factor
N_p	Number of particles in swarm (swarm size)

6.5 PSO in Finance

PSO is applied to solve various optimization problems in finance. This section presents three such popular applications in finance:

6.5.1 *Financial Market Prediction*

The goal of financial market prediction problems such as FOREX rate prediction, stock market prediction, and commodity price prediction is to obtain accurate predictions to make the right decisions. One of the hybrid approaches using PSO is proposed by Pradeepkumar and Ravi (2014). In this approach, the artificial neural network (ANN) is used to obtain predictions. Later, the PSO-based regression model of errors is used to fine-tune the predictions obtained by ANN. The PSO minimizing mean squared error (MSE) is used to obtain optimal coefficients of the regression function of errors. The authors concluded that the proposed hybrid outperformed the standalone approaches.

Ravi et al. (2017) extended the approach aforementioned using multi-objective PSO (MOPSO) in place of PSO. The two objectives of MOPSO are the minimization of MSE and maximization of Dstat (directional change statistic). The authors concluded that MOPSO could yield optimal coefficients of regression in comparison with PSO.

Table 6.2 Parameter selection for PSO

Parameter	Purpose	Possible values/Selection methods
Swarm size (N_p)	Affects performance of PSO	20–50 (Sörensen & Glover, 2013; Wang et al., 2018)
Acceleration coefficients (C_1 & C_2)	Pull particles towards P_{best} and g_{best}	(1) $C_1 = C_2 = 2$ (Ozcan & Mohan, 1999)
		(2) Time-varying acceleration coefficients (Achayuthakan & Ongsakul, 2009; Bao & Mao, 2009)
		(3) Adaptive acceleration coefficients (Guo & Chen, 2009; Yun & Xue, 2009; Zhan, Xiao, Zhang, & Chen, 2007; Zhengija & Jianzhong, 2009; Ziyu & Dingxue, 2009)
		(4) $C_1 = C_2 = 1.49445$ (Clerc & Kennedy, 2002)
		(5) $C_1 = 2.8$, $C_2 = 1.3$ (Carlisle & Dozier, 2001; Schutte & Groenwold, 2005)
		(6) Genetic algorithm (Yu, Zhang, Chen, Song, & Hu, 2005)
		(7) Adaptive fuzzy algorithm (Juang, Tung, & Chiu, 2011)
		(8) Differential evolutionary algorithm (Parsopoulos & Vrahatis, 2002)
Inertia weight (ω)	Adjusts the trade-off between exploration and exploitation of capabilities of PSO	(1) Fixed inertia weight (Shi & Eberhart, 1999)
		(2) Fuzzy adaptive (Bajpai & Singh, 2007; Liu, Ouyang, Zhu, & Tang, 2010)
		(3) Linearly decreasing (Shi & Eberhart, 1998b, 1999)
		(4) Multi-stage linearly decreasing (Xin, Chen, & Hai, 2009)
		(5) Linearly increasing (Zheng, Ma, Jhang, & Qian, 2003)
		(6) Non-linear (Li, Xue, Niu, Chai, & Wu, 2009)
		(7) Random (Lin & Hong, 2007; Zhang, Tang, Hua, & Guan, 2015)
		(8) Chaotic (Feng, Teng, Wang, & Yao, 2007)
		(9) Exponential (Jianxin, Xin, Weiguo, & Rui, 2009)
		(10) Gaussian (Pant, Radha, & Singh, 2007)
		(11) Parallel (Liu, Su, Gao, & Xu, 2009)

(continued)

Table 6.2 (continued)

Parameter	Purpose	Possible values/Selection methods
		(12) Simulated annealing inertia weight (Hassan, Fayek, & Shaheen, 2006)
		(13) $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$ (Han, Yang, Ren, & Sun, 2010)
		(14) $\omega = [0.9, 1.2]$ (Shi & Eberhart, 1999)
		(15) $\omega = [0.5 + (\text{rnd}/2.0)]$ (Eberhart et al., 2001)
Maximum velocity ($V_{\max}$)	Constrains the speed of the particles	(1) Set to a fixed value (Wang et al., 2018)
		(2) Linearly decreased value with time (Fan, 2002)
		(3) Dynamically reduced based on success of search history (Fourie & Groenwold, 2002)
		(4) $V_{\max} = \frac{X_{\max} - X_{\min}}{N_I}$
		Where N_I is the number of intervals in the dth dimension selected by user. $X_{\max}$ and $X_{\min}$ are the maximum and minimum values that particles have achieved so far, respectively (Abido, 2001, 2002)
Maximum position ($X_{\max}$)	Constrains positions of the particles	(1) Absorbing wall, reflecting wall and invisible wall (Robinson & Rahmat-Samii, 2004)
		(2) Absorbing wall + reflecting wall (Huang & Mohan, 2005)
		(3) Hard position limit+absorbing wall +reflecting wall (Mikki & Kishk, 2005)
Stopping Criteria	Terminates the particles convergence process	(1) Prespecified number of iterations.
		(2) Achievement of a specified quality in solution.
		(3) Lapsing a specified time.
		(4) Lack of change in a certain successive iteration
		(5) a combination of above (Rezaee Jordehi & Jasni, 2013)

6.5.2 *Volatility Forecasting*

The volatility forecasting problem's goal is to obtain accurate predictions to assist various financial stakeholders. Pradeepkumar and Ravi (2017) presented a PSO-trained quantile regression neural network (QRNN), namely, PSOQRNN, to forecast volatility of financial markets. In this approach, the weights of QRNN are obtained using PSO so that the PSOQRNN could yield accurate forecasts. The authors concluded that PSO helped QRNN obtain accurate volatility forecasts compared to standalone QRNN and other similar volatility forecasting approaches.

6.5.3 Portfolio Optimization

The goal of portfolio optimization is to build the best investment portfolio according to a defined set of assets. Let us assume that we have selected N financial assets we want to invest in. They can be (daily, monthly, etc.) stocks, funds, bonds, ETF, etc. Each of these has many historical returns that are the relative price difference from one period to another.

Kunwar Madan (https://github.com/KunwarMadan/Optimal-Financial-Portfolio-Selection), in this context, presented an example of portfolio selection using PSO and genetic algorithm (GA) in Python. The author solved a 470-dimensional problem in which 470 stocks were considered in the portfolio. In 470-dimensional search space, PSO and GA are applied in finding the optimal combination of weights representing all stocks' capital using the Sharpe ratio. The author concluded that GA results after 2000 iterations were not even close to PSO results after 250 iterations. Hence, the author proved that PSO is better than GA to solve the portfolio optimization problem. The same fact is also proved by Chen and Zhu (2010). And also, the PSO is applied in constructing optimal risky portfolio (Cura, 2009; Dashti, Farjami, Vedadi, & Anisseh, 2007; Kendall & Su, 2005; Mercangoz, 2019) and in solving constrained portfolio selection problem (Chen, Zhang, Cai, & Xu, 2006; Cui, Cheng, & Bai, 2014; Zhu, Wang, Wang, & Chen, 2011).

6.6 Variants of PSO for Portfolio Optimization

Table 6.3 presents various variants of PSO proposed for solving portfolio optimization problem. The authors concluded that the proposed PSO variants outperformed basic PSO aforementioned and other PSO variants.

Table 6.3 Variants of PSO for portfolio optimization

Year	Author(s)	PSO variant
2009	Niu, Xue, Li, and Chai (2009)	Symbiotic Multi-swarm PSO (SMPSO)
2009	Mario Villalobos-Arias (2009)	PSO with stripes (MOPSO-ST)
2012	Sharma, Thulasiram, and Thulasiraman (2012)	Normalized PSO (NPSO)
2014	Soleimanivareki, Fakharzadeh, and Poormoradi (2014)	Fuzzy Adaptive PSO
2015	Yin, Ni, and Zhai (2015)	Heterogeneous Multiple Population PSO (HMIPPSO)

6.7 Conclusion

Portfolio optimization aims at building the best investment portfolio according to a defined set of assets and constraints. PSO is good at obtaining the near-best global optimal solution from the search space of feasible solutions. Literature provided a base for the readers that PSO and its variants are the best fit for achieving the portfolio optimization problem's objective. This chapter provided descriptions of basic PSO, its parameter selection methods, and its variants. The readers can also be further directed to refer to Yarpiz (2020) for solving portfolio optimization problem using various classic and other SI algorithms such as imperialist competitive algorithm (ICA), non-dominated sorting genetic algorithm II (NSGA-II), and strength Pareto evolutionary algorithm 2 (SPEA2).

References

Abido, A. (2001). Particle swarm optimisation for multi-machine power system stabilizer design. In *Proceedings of power engineering society summer meeting*. Washington, DC: IEEE Computer Society.

Abido, A. (2002). Optimal power flow using particle swarm optimisation. *International Journal of Electrical Power & Energy Systems, 24*, 563–571.

Achayuthakan, C., & Ongsakul, W. (2009). TVAC-PSO based optimal reactive power dispatch for reactive power cost allocation under deregulated environment. In *Proceedings of the IEEE international meeting of power and energy society* (pp. 1–9). Washington, DC: IEEE Computer Society.

AlRashidi, M. R., & El-Hawary, M. E. (2008). A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation, 13*(4), 913–918.

Bajpai, P., & Singh, S. N. (2007). Fuzzy adaptive particle swarm optimisation for bidding strategy in uniform price spot market. *IEEE Transactions on Power Systems, 22*, 2152–2160.

Bao, G. Q., & Mao, K. F. (2009). Particle swarm optimisation algorithm with asymmetric time-varying acceleration coefficients. In *Proceedings of the IEEE international conference on robotics and biomimetics* (pp. 2134–2139). Washington DC: IEEE Computer Society.

Bartz-Beielstein, T., Parsopoulos, K. E., Vegt, M. D., & Vrahatis, M. N. (2004). Designing particle swarm optimization with regression trees. In *Technical Report CI 173/04, SFB 531. University of Dortmund*. Dortmund, Germany: Department of Computer Science.

Bartz-Beielstein, T., Parsopoulos, K. E., & Vrahatis, M. N. (2002). Tuning PSO parameters through sensitivity analysis. In *Technical Report CI 124/02, SFB 531. University of Dortmund*. Dortmund, Germany: Department of Computer Science.

Bartz-Beielstein, T., Parsopoulos, K. E., & Vrahatis, M. N. (2004). Analysis of particle swarm optimization using computational statistics. In *Proceedings of the international conference of numerical analysis and applied mathematics (ICNAAM 2004)* (pp. 34–37). Chalkis, Greece: ICNAAM.

Carlisle, A., & Dozier, G. (2001). An off-the-shelf PSO. In *Proceedings of the workshop on particle swarm optimization*. Indiana, USA: Indianapolis.

Chen, W., Zhang, R., Cai, Y., & Xu, F. (2006). Particle swarm optimization for constrained portfolio selection problems. In *2006 International Conference on Machine Learning and Cybernetics* (pp. 2425–2429). China: Dalian. https://doi.org/10.1109/ICMLC.2006.258773.

Chen, Y., & Zhu, H. (2010). PSO heuristics algorithm for portfolio optimization. In Y. Tan, Y. Shi, & K. C. Tan (Eds.), *Advances in Swarm Intelligence. ICSI 2010* (Lecture Notes in Computer Science) (Vol. 6145). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-13495-1_23.

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability and convergence in a multi dimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(2), 58–73.

Cui, T., Cheng, S., & Bai, R. (2014, July). A combinatorial algorithm for the cardinality constrained portfolio optimization problem. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 491–498). New York: IEEE.

Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications, 10*(4), 2396–2406.

Dashti, M. A., Farjami, Y., Vedadi, A., & Anisseh, M. (2007). Implementation of particle swarm optimization in construction of optimal risky portfolios. In *2007 IEEE international conference on industrial engineering and engineering management* (pp. 812–816). Singapore: IEEE. https://doi.org/10.1109/IEEM.2007.4419303.

Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948). New York: IEEE.

Eberhart, R., Shi, Y., & Kennedy, J. (2001). *Swarm intelligence*. San Mateo, CA: Morgan Kaufmann.

Eberhart, R. C., & Shi, Y. (2004). Guest editorial special issue on particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 201–203.

Fan, H. (2002). A modification to particle swarm optimization algorithm. *Engineering Computations, 19*(8), 970–989.

Feng, Y., Teng, G. F., Wang, A. X., & Yao, Y. M. (2007). Chaotic inertia weight in parti- cle swarm optimisation. In *Proceedings of the IEEE international conference on innovative computing, information and control* (pp. 475–479). Washington, DC: IEEE Computer Society.

Fourie, P. C., & Groenwold, A. A. (2002). The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization, 23*(4), 259–267.

Guo, L., & Chen, X. (2009). A novel particle swarm optimisation based on the self-adaptation strategy of acceleration coefficients. In *Proceedings of the IEEE international conference on computational intelligence and security* (pp. 277–281). Washington, DC: IEEE Computer Society.

Han, W., Yang, P., Ren, H., & Sun, J. (2010). Comparison study of several kinds of inertia weight for PSO. In *Proceedings of the IEEE international conference on progress in informatics and computing* (pp. 280–284). Washington, DC: IEEE Computer Society.

Hassan, W. A., Fayek, M. B., & Shaheen, S. I. (2006). PSOSA: An optimised particle swarm technique for solving the urban planning problem. In *Proceedings of the IEEE international conference on computer engineering and systems* (pp. 401–405). Washington, DC: IEEE Computer Society.

Heppner, F., & Grenander, U. (1990). A stochastic nonlinear model for coordinated bird flocks. In S. Krasner (Ed.), *The ubiquity of chaos*. Washington, DC: AAAS Publications.

Huang, T., & Mohan, A. S. (2005). A hybrid boundary condition for robust particle swarm optimization. *Antennas Wirel Propag Lett, 4*, 112–117.

Jianxin, W., Xin, H. X., Weiguo, Z., & Rui, W. (2009). Exponential inertia weight particle swarm algorithm for dynamic optimisation of electromechanical coupling system. In *Proceedings of the IEEE international conference on intelligent computing and intelligent systems* (pp. 479–483). Washington DC: IEEE Computer Society.

Juang, Y. T., Tung, S. L., & Chiu, H. C. (2011). Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. *Information Sciences, 181*, 4539–4549.

Kendall, G., & Su, Y. (2005, January). Particle swarm optimisation approach in the construction of optimal risky portfolios. In *Proceedings of the 23rd IASTED International Multi-Conference Artificial Intelligence and Applications*. Innsbruck: IASTED.

Li, L., Xue, B., Niu, B., Chai, Y., & Wu, J. (2009). The novel nonlinear strategy of inertia weight in particle swarm optimisation. In *Proceedings of the IEEE international conference on bio-inspired computation* (pp. 1–5). Washington, DC: IEEE Computer Society.

Lin, G. Y., & Hong, D. Y. (2007). A new particle swarm optimisation algorithm with random inertia weight and evolution strategy. In *Proceedings of the IEEE international conference on computational intelligence and security* (pp. 199–203). Washington, DC: IEEE Computer Society.

Liu, C., Ouyang, C., Zhu, P., & Tang, W. (2010). An adaptive fuzzy weight PSO algorithm. In *Proceedings of genetic and evolutionary computing* (pp. 8–10). Washington, DC: IEEE Computer Society.

Liu, H., Su, R., Gao, Y., & Xu, R. (2009). Improved particle swarm optimisation using two novel parallel inertia weights. In *Proceedings of the IEEE international conference on intelligent computation technology and automation engineering and systems* (pp. 185–188). Washington, DC: IEEE Computer Society.

Mario Villalobos-Arias, M. (2009). Portfolio optimization using particle swarms with stripes. *Revista de Matematica: Teorıa y Aplicaciones, 16*(2), 205–220, cimpaucr issn: 1409- 2433.

Mercangoz, B. A. (2019). Particle swarm algorithm: An application on portfolio optimization. In J. Ray, A. Mukherjee, S. K. Dey, & G. Klepac (Eds.), *Metaheuristic Approaches to Portfolio Optimization* (pp. 27–59). Hershey: IGI Global. https://doi.org/10.4018/978-1-5225-8103-1.ch002.

Mikki, S., & Kishk, A. (2005). Improved particle swarm optimization technique using hard boundary conditions. *Microwave and Optical Technology Letters, 46*(5), 422–426.

Millonas, M. M. (1993). Swarms, phase transitions, and collective intelligence. In C. G. Langton (Ed.), *Proceedings of ALIFE III*. Santa Fe Institute, USA: Addison-Wesley.

Niu, B., Xue, B., Li, L., & Chai, Y. (2009, September). Symbiotic multi-swarm PSO for portfolio optimization. In *International Conference on Intelligent Computing* (pp. 776–784). Berlin, Heidelberg: Springer.

Nowak, A., Szamrej, J., & Latané, B. (1990). From private attitude to public opinion: A dynamic theory of social impact. *Psychological Review, 97*, 362. https://doi.org/10.1037/0033295X.97.3.362.

Ozcan, E., & Mohan, C. (1999). Particle swarm optimisation: Surfing the waves. In *Proceedings of the IEEE international congress on evolutionary computation* (pp. 1939–1944). Washington, DC: IEEE Computer Society.

Pant, M., Radha, T., & Singh, V. P. (2007). Particle swarm optimisation using Gaussian inertia weight. In *Proceedings of the IEEE international conference on computational intelligence and multimedia* (pp. 97–102). Washington, DC: IEEE Computer Society.

Parsopoulos, K. E., & Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing, 1*, 235–306.

Parsopoulos, K. E., & Vrahatis, M. N. (2010). Particle swarm optimization and intelligence: Advances and applications. In *Information Science Reference (an imprint of IGI Global)*. Hershey: IGI Global.

Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications, 2008*, 685175.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization–an overview. *Swarm Intelligence, 1*(1), 33–57.

Pradeepkumar, D., & Ravi, V. (2014, October). FOREX rate prediction using chaos, neural network and particle swarm optimization. In *International Conference in Swarm Intelligence* (pp. 363–375). Cham: Springer.

Pradeepkumar, D., & Ravi, V. (2017). Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. *Applied Soft Computing, 58*, 35–52.

Pradeepkumar, D., & Ravi, V. (2018). Soft computing hybrids for FOREX rate prediction: A comprehensive review. *Computers & Operations Research, 99*, 262–284.

Ravi, V., Pradeepkumar, D., & Deb, K. (2017). Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms. *Swarm and Evolutionary Computation, 36*, 136–149.

Reeves, W. T. (1983). Particle systems—A technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics, 2*(2), 91–108.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics and Interactive Techniques, 21*(4), 25–34.

Rezaee Jordehi, A., & Jasni, J. (2013). Parameter selection in particle swarm optimisation: A survey. *Journal of Experimental & Theoretical Artificial Intelligence, 25*(4), 527–542.

Robinson, J., & Rahmat-Samii, Y. (2004). Particle swarm optimization in electromagnetics. *IEEE Transactions on Antennas and Propagation, 52*(2), 397–407.

Schutte, J. F., & Groenwold, A. A. (2005). A study of global optimization using particle swarms. *Journal of Global Optimization, 31*, 93–108.

Sharma, B., Thulasiram, R., & Thulasiraman, P. (2012). Portfolio Management Using Particle Swarm Optimization on GPU. In *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications (ISPA)* (pp. 103–110). Leganes: IEEE. https://doi.org/10.1109/ISPA.2012.22.

Shi, Y., & Eberhart, R. (1998a). A modified Particle swarm optimiser. In *Proceedings of the IEEE international conference on computational intelligence* (pp. 69–73). Washington, DC: IEEE Computer Society.

Shi, Y., & Eberhart, R. (1998b). Parameter selection in particle swarm optimisation. In *Proceedings of the IEEE international conference on evolutionary programming* (pp. 591–600). Washington, DC: IEEE Computer Society.

Shi, Y., & Eberhart, R. (1999). Empirical study of particle swarm optimisation. In *Proceedings of the IEEE international conference on computational intelligence* (pp. 1945–1950). Washington, DC: IEEE Computer Society.

Soleimanivareki, M. A., Fakharzadeh, J., & Poormoradi, M. (2014). Fuzzy adaptive Pso approach for portfolio optimization problem. *The Journal of Mathematics and Computer Science, 12*(3), 235–242.

Sörensen, K., & Glover, F. W. (2013). Metaheuristics. In *Encyclopedia of operations research and management science* (pp. 960–970). New York: Springer US.

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Computing, 22*(2), 387–408.

Xin, J., Chen, G., & Hai, Y. (2009). A particle swarm optimiser with multi-stage linearly decreasing inertia weight. In *Proceedings of the IEEE international conference on computational sciences and optimisation* (pp. 505–508). Washington, DC: IEEE Computer Society.

Yarpiz. (2020). Portfolio optimization using classic and intelligent algorithms. MATLAB Central File Exchange. Retrieved Nov 17, 2020, https://www.mathworks.com/matlabcentral/fileexchange/53143-portfolio-optimization-using-classic-and-intelligent-algorithms.

Yin, X., Ni, Q., & Zhai, Y. (2015). A novel PSO for portfolio optimization based on heterogeneous multiple population strategy. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1196–1203). Sendai: CEC. https://doi.org/10.1109/CEC.2015.7257025.

Yu, H., Zhang, L., Chen, D., Song, X., & Hu, S. (2005). Estimation of model parameters using composite particle swarm optimization. *Journal of Chemical Engineering of Chinese Universities, 19*(5), 675–680.

Yun, W. G., & Xue, H. D. (2009). Particle swarm optimisation based on self-adaptive acceleration factors. In *Proceedings of the IEEE international conference on genetic and evolutionary computing* (pp. 637–640). Washington, DC: IEEE Computer Society.

Zhan, Z. H., Xiao, J., Zhang, J., & Chen, W. N. (2007). Adaptive control of acceleration coefficients for particle swarm optimisation based on clustering analysis. In *Proceedings of the IEEE international congress on evolutionary computation* (pp. 3276–3282). Washington, DC: IEEE Computer Society.

Zhang, L., Tang, Y., Hua, C., & Guan, X. (2015). A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. *Applied Soft Computing, 28*, 138–149.

Zheng, Y. L., Ma, L. H., Jhang, L. Y., & Qian, J. X. (2003). Empirical study of particle swarm optimiser with an increasing inertia weight. In *Proceedings of the IEEE international congress on evolutionary computation* (pp. 221–226). Washington, DC: IEEE Computer Society.

Zhengija, W., & Jianzhong, Z. (2009). A self-adaptive particle swarm optimisation algorithm with individual coefficients adjustment. In *Proceedings of the IEEE international symposium on intelligent information technology application* (pp. 396–399). Washington, DC: IEEE Computer Society.

Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle swarm optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications, 38*(8), 10161–10169.

Ziyu, T., & Dingxue, Z. (2009). A modified particle swarm optimisation with adaptive acceleration coefficients. In *Proceedings of the IEEE international conference on information processing* (pp. 330–332). Washington, DC: IEEE Computer Society.