International Journal of Pure and Applied Mathematics
Volume 120 No. 6 2018, 11731-11747
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue
http://www.acadpubl.eu/hub/

# Different variants of Particle Swarm Optimization, its limitations and future directions

Vrushali K.Bongirwar[1], Dr.Archana V.Potnurwar[2],
Dr Shailendra Aote[3],
[1,2,3]Assistant Prof.
[1,3]Ramdeobaba College of Engineering and Management, Nagpur
[2]P.I.E.T,Nagpur [2]archanapotnurwar@gmail.com

July 30, 2018

**Abstract**

The optimization problems attract the attention of many researchers for studying swarm intelligence techniques. Swarm intelligence (SI) is a nature inspired computing technique mainly used for solving optimization problems. Ant colony optimization, particle swarm optimization (PSO), Biogeography based optimization, cuckoo search are some of the techniques in SI. Out of all these techniques, PSO has become more popular and stronger algorithm in the field of optimization. Here we discussed about working of PSO and different variants of it. All these variants are used to solve different types of optimization problems. We have also discussed limitations in existing approaches, which can be helpful for the researchers to carry out their research.

***Key Words*:**Particle swarm Optimization, Variants, future directions

# 1 Introduction

Problem solving is one of the most complicated intellectual activities of the human brain. The process of problem solving deals with finding the solution in the presence of the constraints. An exact solution to some problems might simply be infeasible, especially if it has larger dimensionality. In those problems, solution near to the exact value might be deemed very good and sufficient. Knapsack problem, linear programming problem is an example of an optimization problem. Another example of an optimization problem is to arrange the transistors on a computer chip, so that it will occupy the smallest area and the number of components it will used are as few as possible [1]. Optimization is generally used in many problems like Scheduling Problems, Resource Allocation, Decision Making, Industrial Planning. Furthermore, these optimization techniques cover large application areas in business, industry, engineering and computer science. Swarm intelligence (SI) is the distributed intelligent computing mechanism for solving optimization problems. SI took its inspiration from flocking of birds, swarming and herding phenomenon in vertebrates. Sometimes SI is considered as a part of evolutionary computing, as it shares many similarities with it. Ant colony optimization and particle swarm optimization are generally used techniques in SI. Now a days, Biogeography based optimization also attracts the attention of many researchers. SI starts its working with a group of individuals, where each individual try to find out the optimal solution. The solution is shared among individuals, and then each individual improves themselves based on the information, which is gathered from others. In general, the properties of swarm intelligence are as follows,
•It consist of a group of individuals
•All the individuals have similar characteristics
•Each individual interacts with each other based on simple behavioral rules. This interaction involves exchanging the local information among them.
• The overall output of the system depends on the interaction of individuals with each other and with their group behavior.
The most important property of SI is that all the individuals work in a coordinated way without the presence of a coordinator. PSO is the global optimization technique, which is inspired from the

flocking of bird, school of fish, a swarm of bees and human social behavior. It is proposed initially by James Kennedy and Russell Eberhart in 1995 [3]. Due to, involvement of less parameters and fast convergence, PSO attracts attention of many researchers and becomes the promising algorithm to solve all kinds of optimization problems. The simulation by PSO is explained with the following scenario: Let the group of birds be randomly searching the food. All the birds are unaware about the exact location of the food. But they know how the food is from the current position. This information is updated in each iteration. There is no central coordinator to lead the swarm, instead, all the birds cooperate with each other and update its position. PSO incorporates swarming behavior of birds, where each individual is evolved from cooperation and competition among themselves. Each individual is referred as a particle. Each particle adjusts its velocity according to its own and neighbors velocity. The working of PSO is as follows. Each particle is considered as a point in a D dimensional space. The position and velocity of i$^{th}$ particle is represented as,
$X_i = (X_{i1}, X_{i2}, , X_{iD}) and V_i = (V_{i1}, V_{i2}, ., V_{iD})$
respectively. The best previous position is considered as the personal best (pbest). The pbest position of any particle is represented as,
$P_i = (P_{i1}, P_{i2}, ..P_{iD})$ The symbol g is used to represent the index of the best particle among all the particles in the population and is called as global best position. The velocity of each particle i is represented as

$$V_{iD} = V_{iD} + c1 \times rand\,() \times (P_{iD} - X_{iD}) + c2 \times rand\,() \times (P_{gD} - X_{iD}) \qquad (1.1)$$

The Particle is manipulated according to the following equation

$$X_{iD} = X_{iD} + V_{iD} \qquad (1.2)$$

Where, the random number generator, rand () generates the value in the range [0, 1]. The velocity update equation consists of three parts, where first part is its own previous velocity. The second part is the cognition part, which represents the personal experience of the particle itself. It considers its personal best performance for future travel. The third part is the social part, which considers the behavior of the group best particle. The equation (1.1) is used to calculate the particles new velocity based on its previous velocity, the distances of its current position from its own best position and

3

the groups best position. Then the particle occupies a new position according to the equation (1.2). The fitness function is used to measure the performance of each particle. Equation (1.1) and (1.2) are the basic equations of PSO behavior, which were updated in later time. Inertia weight (w) is added in the velocity update equation to balance the global and local search [4]. The new velocity update equation is,

$$V_{iD} = w \times V_{iD} + c1 \times rand() \times (P_{iD} - X_{iD}) + c2 \times rand() \times (P_{gD} - X_{iD}) \qquad (1.3)$$

where,
$V_{iD}$ = Velocity of the i$^{th}$ particle in D$^{th}$ dimension
$X_{iD}$ = Position of the i$^{th}$ particle in D$^{th}$ dimension
$P_{iD}$ = Personal best position of i$^{th}$ particle
$P_{gD}$ = Global best position of i$^{th}$ particle
w = inertia weight
c1 = Cognitive acceleration coefficient
c2 = Social acceleration coefficient

Velocity update equation consists of three parts, where each part has its own influence [4]. Consider the absence of the first part, and then each particle movement depends on its personal and group experience. The velocity of each particle is memoryless. Over the period of time, every particle will move towards the same position and the search space contracts through each generation. The importance of the second and third part is to make the use of, whether particle is updated in the proper direction or not. The working of PSO is as follows: Initialize the velocity and position of particles in the search space. This search space is bound between lower and upper bound. Initially, a current position of each particle is considered as its personal best. Evaluate the position of the particle by some fitness function. The particle having minimum fitness value is considered as a global best particle. Calculate the particles new velocity and position. If this new position is better than previous position, then making this position as its personal best, otherwise its personal best position remains unchanged. During the search, if the particle moves outside the search space, it is necessary to reinitialize its velocity and position in the search space. Repeat this process for certain number of iterations. This paper presents the different techniques of PSO. Section 2 discusses the different variants of PSO. Limitations and future directions are given in section

3. Finally section 4 discusses the conclusion.

# 2 Literature Survey

PSO is initially proposed by James Kennedy and Russel Eberhart in 1995. This basic version was not handling difficult problems. To increase the performance and to achieve the efficiency in solving the optimization problems, different variants of PSO are proposed in later years.

## 2.1 History

The first modification is proposed by its founder, where they proposed new optimizer using PSO. This version is considered as lbest version [19], which becomes more popular to solve various problems. To calculate the gbest, it uses neighbors of current particles and chooses global best from its neighbors only. Yuhui Shi and Russell Eberhart [4] introduced w as an inertia weight to increase the performance of PSO. Inertia weight is used to control the exploration and exploitations. To solve large scale dimensional problems, efforts were put towards improving the performance by using parameter adjustment, update in velocity equation, methods to prevent and control premature stagnation.

## 2.2 Methods and Types of PSO

This section briefs about different variants in each of the above mentioned categories. All these variants are proposed to address different kinds of optimization problems.

### 2.2.1 Methods with parameter adjustment

Efforts started in adjusting the parameter in 1998. Shi declared that [5, 20], inertia weight must be linearly decreasing from large value to small value during the PSO run. Initially the PSO has the global search ability and later it prefers local search. Linearly decreasing inertia weight is used, which decreases the value from 0.9 to 0.4. Yong-Ling Zheng [21] has proposed a linearly increasing inertia weight, which increases the value from 0.4 to 0.9. Clerk in 1999 came up with the new factor in the PSO equation. This factor is known as constriction factor. The work is extended by R. C. Eberhart, Y. Shi [7], where they compare the use of inertia weight and constriction factor in the equation. From their study it is concluded that the use of constriction factor in place of w, c1 and c2 gives better results. It is also stated that this constriction factor must be used to limit the maximum velocity, when the dynamic range of the variables are used on each dimension. Asanga Ratnaweera, Saman K. Halgamuge [8] proposes a time varying acceleration coefficients which decrease the value of c1 from 2.5 to 0.5 and increases the value of c2 from 0.5 to 2.5. They have also given a new formula to calculate the value of inertia weight. A linearly decreasing Vmax is introduced in addition to the time-varying inertia weight, in [24]. A nonlinear change in inertia weight by designing fuzzy methods is also introduced [25]. To guarantee the convergence and to improve the convergence velocity, a PSO variant with a constriction factor was introduced by Clerc and Kennedy [26]. Adaptive Constriction Factor for Location-related Particle Swarm [27] is proposed in 2007, which adapt a technique to solve complex problems with PSO. Constriction factor associated with each particle varies, based on the relative location of the better particles. It actually determines the direction in which constriction factor needs to be updated. The methodology of dynamically varying constriction factor is termed as a PSO-cf. Sabine Helwig and Rolf Wanka in [28] studied the particle trajectories in the initial iterations. It is proved that many particles leave the search space at the beginning of the process. To get those particles in next iterations, a bound handling strategy performs well. To decrease the number of particles those are leaving the search space, zero and half-diff velocity initialization method is proposed.

#### 2.2.2 PSO Variants

As discussed earlier, the first version of PSO comes in 1995 called as gbest version followed by lbest version. In 1998, Yuhui Shi added the inertia weight in the equation. The values of w, c1 and c2 are replaced by Constriction coefficient in 1999. A new locally convergent Particle Swarm Optimizer is proposed by F. Van Den Berg in 2002 [33]. The main reason behind stagnation is discussed that if the swarm reaches to local well, then because of a decrease in inertia weight and decreasing velocity, it does not find any momentum to leave the space. To overcome this issue, Guaranteed Convergence PSO (GCPSO) is proposed. The gbest particle uses a different velocity update equation. This algorithm performs well for a small number of particles. The problem of convergence still exists because all the particles move into local well and the solution is present in some other region, called global minimum. Multi-start PSO (MPSO) is developed by Van den Bergh [33], which automatically gets restarted when the stagnation is detected. Various criteria that cause the stagnation are studied. These include objective function slope, maximum swarm radius and cluster analysis. The first criteria monitors, whether the improvement has been seen recently in the function value being optimized. The latter two criteria check the proximity of the particles to one another. Restarting in MPSO means to start the search with a new sequence of random numbers. Each search is independent of its previous search, therefore, while restarting; particles lose their memories of the previous search. The global best of each restarts must be stored, so that after a fixed number of restarts, the best of all global bests is proposed as the most desirable solution. A quantum swarm PSO (QPSO) sometimes called as a discrete PSO is proposed in 2004 for discrete optimization problems [35]. It is inspired from physics and biology. The authors define a particle Q(t) based on the quantum bit (qubit). The sigmoid function is randomly replaced, and the best chromosome's guidance is also used to move close to the optimum. This new kind of discrete PSO is called as QPSO. Fully Inspired PSO (FIPS) is proposed by Rui Mendes in 2004 [36]. It comes with the idea that instead of using global best particle, consider all other particles to calculate new velocity. Five types of variants are proposed here. The fully informed particle swarm (FIPS) with returning a constant is proposed here, which considers the particle having same contributions. A fully informed swarm (wFIP), where the contribution of each neighbor was weighted by the goodness of its previous best is proposed. Another fully informed (wdFIPS) particle swarm is proposed, where the each particle considers its neighbor and finds the average distance from the target particle. A fully informed model (Self) is proposed, where the particles considers its previous best, but the half of the weight is assigned to each particle. The last fully informed model (wSelf) is given, where each particle considers its neighbor with the complete weight and its previous best is considered with half weight. The computational cost is more as it considers all other particles instead of only some neighbors. Hierarchical Particle Swarm Optimizer (HPSO), uses neighborhood structure, which is based on the arrangement of particles in the tree [37]. The particle moves up and down in the tree, based on their previous solution. This gives good particles that move up in the hierarchy, a larger influence on the swarm. The variant of H-PSO is introduced, where the structure of the hierarchy is dynamically adapted during the course of run. Another variant is that, based on the position of the particle with respect to their level in the hierarchy, different behavior is assigned to each individual particle. Another variant of H-PSO (AH-PSO) is proposed, where it dynamically changes the branching degree of the tree topology, which helps in improving the performance of H-PSO. Another extension of H-PSO is to use different values for the inertia weight of the particles according to their level in the hierarchy. Comprehensive learning particle swarm optimizer (CLPSO). The particles velocity is updated based on its historical information. The diversity of the particle is preserved, which avoid the premature convergence of the swarm. A new velocity update equation is used here, which depends on the learning probability (p). A random number is generated. If this random number is greater than p then the particle is generated based on its personal best only along its direction. Otherwise, it uses the tournament selection method to select another particle, based on which, it will calculate the new velocity. This method performs well for global optimization problems. The main aim of RegPSO is to automatically trigger the swarm in case of stagnation. This mechanism takes the swarm from suboptimal solutions and continued its process to find the true global optimum. The maximum deviation from the gbest position is determined and particles are regrouped within a range of the maximum deviation. This is a computationally simple and effective PSO algorithm. Heterogeneous PSO says that at least one pair of particles must differ in velocity update rule, parameters in velocity update rules, neighborhood topology etc. This is considered as a particles configuration. Adaptive particle swarm is proposed, which is built on dynamic heterogeneity. Here because of the behavior of the swarm, some trigger is generated. To give the response to the trigger, changes in the configuration take place. Fractional PSO is proposed to address multidimensional problems. The process is used in, where optimum dimension is unknown. Here swarm particle uses both positional and dimensional optima. It avoids the used of dimension definition initially. The process in two dimensions is as follows. Select the point, which is better in each dimension. Compare the selected point with the point, which is calculated from the general position update formula. If this is better, then select this point as a new position of the particle. Multi-swarm particle swarm optimization (MPSO) is proposed to overcome the premature convergence problem . Multiple sub-swarms are used to maintain the swarm diversity. To share the information among sub-swarms, a cooperative mechanism is introduced. MPSO follows an adaptive reinitializing strategy. To achieve the global optimum, swarm diversity is used to guide the re-initialization strategy. This swarm diversity is also used to maintain the local and global exploration. Pradipta Ghosh introduces a new variant of PSO referred to as Hierarchical Dynamic Local Neighborhood based Particle Swarm Optimization The proposed technique follows the dynamic hierarchy of the particles in their arrangement. The particles search for the better solution within each hierarchy, using dynamically varying sub-swarms. It means that these sub-swarms exchange their information and are regrouped frequently. Results are extremely well on CEC-2005 test suit as compared to other promising algorithms.

#### 2.2.3 Cooperative coevolution for Large Scale Dimensional Problems

The most used model for large scale optimization (LSO) problems is cooperative coevolution (CC) model. Potter and Jong suggested this model in 1994 for solving optimization problems by using genetic algorithm (CCGA-1) This model talks about dividing the large dimensional space into sub components randomly. Frans van den Bergh extends these efforts for solving the optimization problems by using PSO By using this method (CPSO), solution quality and robustness of the traditional PSO method are increased. One hypothesis is that the increased diversity of the cooperative swarms is responsible for the improved robustness for multimodal problems.

The increase in performance is due to the exponential increase in the volume of the search space, while the number of particles must be kept small and fixed, so that the efficiency of the algorithm is maintained. Larger the swarm size, larger the unwanted particles are. These unwanted particles do not contribute to the solution, especially during later iterations, so it would be impractical to increase the number of particles to match the increase in volume. The main idea behind the CPSO algorithm is to decompose the large search space into several smaller spaces, which in turn increases the rate of convergence as compared to the standard PSO. Zhenyu Yang proposed a large scale evolutionary optimization using cooperative coevoluti, which are capable to solve nonseparable problems. A novel two sub-swarms, based on multi-phases (TSEM-PSO) is proposed to deal with the problem of stagnation . The complete swarm is divided into two identical sub-swarms. The first sub-swarm adopts the standard PSO formulation and the second sub-swarm adopting the method as given: When the two sub-swarms evolve independently, the exchanged numbers of particles is different in different searching phase and its amount decreases over the period of time. It helps to increase the information exchange between the particles, improve the diversity of the population and increase the speed and chances of finding the convergence of the algorithm.

#### 2.2.4 Hybridized PSO and other techniques

A novel particle swarm optimization model with the centroid of the population is proposed to deal with the local optimum solution. This also improves the global optimum efficiency and the accuracy of the particle swarm optimization [80]. New techniques in parameter selection are proposed in the case of convergence of the proposed model. Regression based fitness approximation is studied to find optimum more quickly. Aging leader (ALC-PSO) PSO is proposed which defines an aging mechanism of leader particles . Distance based PSO model is proposed to solve multimodal optimization problems, where any particle can switch to different optimal solutions, when needed. The necessity is to maintain the optimal system performance. As the initialization of the swarm is one of the major components in getting good performance of the PSO, Borhan Kazimipour proposed Initialization Methods for Large Scale Global

Optimization in 2013 The Second best particle is also added into the velocity update equation. The behavior analysis of leader particle is studied in This study is done to understand the stagnation cause and avoiding it by doing some adjustment in leader particle.

## 3 Limitations in existing approaches

A lot of variants are suggested to improve the performance of PSO. Most of them have some limitations in terms of convergence speed, dimensional space, memory requirement, premature convergence, etc. Some of the limitations are gathered here, which are helpful to carry out the research.
1) lbest version of an algorithm is not suitable for more number of particles.
2) Most of the algorithm does not perform well on multimodal problems.
3) Tradeoff between optimal solution and the convergence speed.
4) Algorithms like FIPS perform well, but memory requirement is more as it stores the information of all the particles.
5) Premature convergence is always the problem with any evolutionary algorithms.
6) The difficulty in achieving the solution increases with respect to the number of dimensions. Cooperative strategies are used to deal with this problem.
7) Non separable problems, where every variable is dependent on some other variables are still facing difficulty to get the optimal solution.
8) The main difficulty in designing an effective strategy is to group the swarm along the dimensions.
9) Some algorithms perform well on some benchmark functions, but its performance degrades on other functions.
Based on the literature review and limitations in existing approaches, different gaps in the working of PSO are found. Work in bridging the above mentioned gaps leads to the future research.

# 4 Conclusion

This paper presents the literature in the PSO field. The discussion starts with the need to study optimization problems and swarm intelligence solution to it followed by the brief working of PSO. The history of PSO development, which was extended in later years, is also discussed. Different PSO variants are proposed with reference to the methods of parameter adjustment, different versions of PSO, techniques to deal with large scale optimization, hybridized PSO. PSO is mostly hybridized with DE. Limitations of existing approaches are given, which gives the direction for further research.

# References

[1] Mahamed G. H. Omran, Particle Swarm Optimization Methods for Pattern Recognition and Image Processing, Ph.D. Thesis at University of Pretoria, Pretoria, November 2004.

[2] Mahamed G. H. Omran, Particle Swarm Optimization Methods for Pattern Recognition and Image Processing, Ph.D. Thesis at University of Pretoria, Pretoria, November 2004.

[3] Ajith Abraham, He Guo, and Hongbo Liu, Swarm Intelligence: Foundations, Perspectives and Applications, Studies in Computational Intelligence (SCI) 26, Springer-Verlag Berlin Heidelberg , 325 (2006).

[4] James Kennedy and Russell Eberhart , Particle Swarm Optimization, IEEE, International Conference on Neural Network, Perth, Australia1995, pp. 1942-1948.

[5] Yuhui Shi and Russell Eberhart, A Modified Particle Swarm Optimizer, proceedings of the IEEE congress on Evolutionary Computation(CEC 1998), Piscataway, NJ. 69-73, 1998.

[6] Y. Shi and R. C. Eberhart, ,Empirical study of particle swarm optimization, in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 3, 1999, pp. 101106.

[7] Clerc, M., The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, Proc. I999 ICEC, Washington, DC, pp 1951 - 1957.

[8] R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 1, 2000, pp. 8488.

[9] Ratnaweera A, Halgamuge SK, and Watson HC, Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients,IEEE Transactions on Evolutionary Computation, VOL. 8, NO. 3, JUNE 2004, pp. 240255.

[10] A. Immanuel Selvakumar, K. Thanushkodi, A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems, IEEE Transactions On Power Systems, Vol. 22, No. 1, February 2007, pp. 42-51.

[11] Pisut Pongchairerks, Particle swarm optimization algorithm applied to scheduling problems, ScienceAsia 35 (2009): pp. 8994.

[12] Yashar Mehdad, Bernardo Magnini, Optimizing Textual Entailment Recognition Using Particle Swarm Optimization, Proceedings of the 2009 Workshop on Applied Textual Inference, ACL-IJCNLP, Suntec, Singapore, 2009, pages 3643.

[13] Mohammad Syafrullah and Naomie Salim, Improving Term Extraction Using Particle Swarm Optimization Techniques, Journal Of Computing, Volume 2, Issue 2, February 2010, ISSN 2151-9617, pp. 116-120.

[14] Sarina Sulaiman, Siti Mariyam Shamsuddin, Fadni Forkan, Ajith Abraham, "Intelligent Web Caching Using Neurocomputing and Particle Swarm Optimization Algorithm", AMS, 2008, Asia International Conference on Modelling Simulation, Asia International Conference on Modelling Simulation 2008, pp. 642-647, doi:10.1109/AMS.2008.40

[15] Bilal M. Zahran and Ghassan Kanaan, Text Feature Selection using Particle Swarm Optimization Algorithm, World Applied Sciences Journal 7 (Special Issue of Computer IT):2009, pp. 69-74.

[16] K. Tang, X. Yao, P. Suganthan, C. MacNish, Y. Chen, C. Chen, and Z. Yang, Benchmark functions for the CEC2008 special session and competition on large scale global optimization, Nature Inspired Computat. Applicat. Lab., Univ. Sci. Technol. China, Hefei, China, Tech,rep. 2007 [Online]. Available: http://nical.ustc.edu.cn/cec08ss.php.

[17] K. Tang, X. Li, P. Suganthan, Z. Yang, and T. Weise, Benchmark functions for the CEC2010 special session and competition on large scale global optimization, Nature Inspired Computat. Applicat. Lab., Univ. Sci. Technol. China, Hefei, China, Tech. Rep., 2009 [Online]. Available: http://nical.ustc.edu.cn/cec10ss.php.

[18] Seyedali Mirjalili, Andrew Lewis, Ali Safa Sadiq, Autonomous Particles Groups for Particle Swarm Optimization, Arab J Sci Eng (2014) 39, pp.46834697.

[19] Joaqun Derrac, Salvador Garca, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (2011), pp. 318.

[20] Eberhart, R. C., and Kermedy, J., A New Optimizer Using Particles Swarm Theory, Proc. Sixth International Symposium on Micro Machine and Hmm Science (Nagoya, Japan), IEEE Service Center, Pkcataway, pp. 39-43.

[21] Y Shi, and R Eberhart, Parameter selection in particle swarm optimization, Proceedings of 7th Annual Conference on Evolution Computation,1998, pp. 591-601.

[22] Yong-Ling Zheng, Long-Hua Ma, Li-Yan Zhang, Ji-Xin Qian, On The Convergence Analysis And Parameter Selection In

Particle Swarm Optimization, Proceedings of the Second International Conference on Machine Learning and Cybernetics, Wan, 2-5 November 2003, pp. 1802-1806.

[23] Kevin J. Binkely, Masfumi hagiwara, balancing Exploration and Exploitation in particle Swarm Optimization: Velocity Based Reinitialization, Technical paper, Transaction of Japanese Society for Artificial Intelligence 23(1), 2008, pp. 27-35.

[24] Chukiat Worasucheep, A Particle Swarm optimization with Stagnation Detection and Dispersion, IEEE Congress on Evolutionary Computation, CEC-2008, pp. 424-429.

[25] H. Y. Fan and Y. Shi, Study on Vmax of particle swarm optimization, in Proc.Workshop Particle Swarm Optimization, Indianapolis, IN, Apr. 2001.

[26] H. Y. Fan and Y. Shi , Particle swarm optimization with fuzzy adaptive inertia weight, in Proc.Workshop Particle Swarm Optimization, Indianapolis, IN, 2001, pp. 101106.

[27] M. Clerc and J. Kennedy, The particle swarm: Explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evolutionary Computation, vol. 6, no. 1, Feb. 2002, pp. 5873.

[28] Xiang-Han Chen, Wei-Ping Lee, Chen-Yi Liao, Jang-Ting Dai, Adaptive Constriction Factor for Location-related Particle Swarm, Proceedings of the 8th WSEAS International Conference on Evolutionary Computing, Vancouver, British Columbia, Canada, June 19-21, 2007, pp. 307-313.

[29] Sabine Helwig and Rolf Wanka, Theoretical Analysis of Initial Particle Swarm Behavior, Proc. of the 10th International Conference on Parallel Problem Solving from Nature, G. Rudolph et al. (Eds.), Dortmund, Germany, September 2008, LNCS 5199, pages 889898.

[30] Zhi-Hui Zhan, Jun Zhang, Yun Li and Henry Shu-Hung Chung, Adaptive Particle Swarm Optimization, IEEE Transactions on

Systems, Man, And CyberneticsPart B: Cybernetics, Vol. 39, NO. 6, December 2009, pp 1362-1381.

[31] Yuntao Dai, Liqiang Liu, Ying Li, An Intelligent Parameter Selection Method for Particle Swarm Optimization Algorithm, Fourth International Joint Conference on Computational Sciences and Optimization, 2011, pp.960-964.

[32] Wei Zhang, Di Ma, Jin-jun Wei, Hai-feng Liang, A parameter selection strategy for particle swarm optimization based on particle positions, Expert Systems with Applications 41, 2014, pp- 35763584.

[33] Limin Zhang, Yinggan Tang, Changchun Hua, Xinping Guan, A new particle swarm optimization algorithm with adaptive inertiaweight based on Bayesian techniques, Applied Soft Computing 28, 2015, pp. 138149.

[34] F. van den Bergh, An Analysis of Particle Swarm Optimizers, Ph.D. dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, 2002.

[35] Ran Cheng and Yaochu Jin, A Social Learning Particle Swarm optimization algorithm for scalable optimization, Information Sciences, 291, 2015, pp. 43-60.

[36] Shuyuan Yang, Min Wang, Licheng Jiao, A Quantum Particle Swarm Optimization, proceedings of the IEEE congress on Evolutionary Computation(CEC 2004), pp. 320 324.

[37] Rui Mendes, James Kennedy, and Jos Neves, The Fully Informed Particle Swarm: Simpler, Maybe Better, IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, June 2004, pp. 204-210.

[38] Stefan Janson and Martin Middendorf, A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant, IEEE Transactions On Systems, Man, And CyberneticsPart B: Cybernetics, Vol. 35, No. 6, December 2005, pp. 1272- 1282.