Lee, K.Y. and M. A. El-Sharkawi (Editors), Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems, IEEE Power Engineering Society, Publication 02TP160, pp. 45-51, 2002. Modern Heuristic Optimization Techniques, Wiley, Chapter 4, pp. 71-88, 2008.

Chapter 5

Fundamentals of Particle Swarm Optimization Techniques

Abstract: This chapter presents fundamentals of particle swarm optimization (PSO) techniques. While a lot of evolutionary computation techniques have been developed for combinatorial optimization problems, PSO has been basically developed for continuous optimization problem, based on the backgrounds of artificial life and psychological research. PSO has several variations including integration with selection mechanism and hybridization for handling both discrete and continuous variables. Moreover, recently developed constriction factor approach is useful for obtaining high quality solutions.

Key words: Continuous optimization problem, Mixed-integer nonlinear optimization problem, Constriction factor

1. INTRODUCTION

Natural creatures sometimes behave as a swarm. One of the main streams of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer. Reynolds developed *boid* as a swarm model with simple rules and generated complicated swarm behavior by CG animation [1].

From the beginning of 90's, new optimization technique researches using analogy of swarm behavior of natural creatures have been started. Dorigo developed ant colony optimization (ACO) mainly based on the social insect, especially ant, metaphor [2]. Each individual exchanges information through pheromone implicitly in ACO. Eberhart and Kennedy developed particle swarm optimization (PSO) based on the analogy of swarm of bird and fish school [3]. Each individual exchanges previous experiences in PSO. These researches are called "Swarm Intelligence" [4][5]. This chapter describes mainly about PSO as one of *swarm intelligence* techniques.

Other evolutionary computation (EC) techniques such as genetic algorithm (GA) also utilize some searching points in the solution space. While GA can handle combinatorial optimization problems, PSO can handle continuous optimization problems originally. PSO has been expanded to handle combinatorial optimization problems, and both discrete and continuous variables as well. Efficient treatment of mixed-integer nonlinear optimization problems (MINLP) is one of the most difficult problems in optimization field. Moreover, unlike other EC techniques, PSO can be realized with only small program. Namely PSO can handle MINLP with only small program. The feature of PSO is one of the advantages compared with other optimization techniques.

This chapter is organized as follows: Chapter II explains basic PSO method and chapter III explains variation of PSO such as discrete PSO and hybrid PSO. Chapter IV describes parameter sensitivities and constriction factor approach. Chapter V shows some applications of PSO and Chapter VI concludes this chapter with some remarks.

2. BASIC PARTICLE SWARM OPTIMIZATION

2.1 Background of Particle Swarm Optimization

Natural creatures sometimes behave as a swarm. One of the main streams of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer. Swarm behavior can be modeled with a few simple rules. School of fishes and swarm of birds can be modeled with such simple models. Namely, even if the behavior rules of each individual (agent) are simple, the behavior of the swarm can be complicated. Reynolds called this kind of agent as *boid* and generated complicated swarm behavior by CG animation [1]. He utilized the following three vectors as simple rules.

(1) to step away from the nearest agent
(2) to go toward the destination
(3) to go to the center of the swarm

Namely, behavior of each agent inside the swarm can be modeled with simple vectors. This characteristic is one of the basic concepts of PSO.

Boyd and Richerson examine the decision process of human being and developed the concept of individual learning and cultural transmission [6]. According to their examination, people utilize two important kinds of information in decision process. The first one is their own experience; that is, they have tried the choices and know which state has been better so far, and they know how good it was. The second one is other people's experiences; that is, they have knowledge of how the other agents around them have performed. Namely, they know which choices their neighbors have found are most positive so far and how positive the best pattern of choices was. Namely each agent decides his decision using his own experiences and other peoples' experiences. This characteristic is another basic concept of PSO.

2.2 Basic method

According to the background of PSO and simulation of swarm of bird, Kennedy and Eberhart developed a PSO concept. Namely, PSO is basically developed through simulation of bird flocking in two-dimension space. The position of each agent is represented by XY axis position and also the velocity is expressed by vx (the velocity of X axis) and vy (the velocity of Y axis). Modification of the agent position is realized by the position and velocity information.

Bird flocking optimizes a certain objective function. Each agent knows its best value so far (pbest) and its XY position. This information is analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (gbest) among pbests. This information is analogy of knowledge of how the other agents around them have performed. Namely, Each agent tries to modify its position using the following information:

- the current positions (x, y),
- the current velocities (vx, vy),
- the distance between the current position and pbest
- the distance between the current position and gbest

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$v_i^{k+1} = wv_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k) \tag{1}$$

where, v_i^k : velocity of agent i at iteration k,
w : weighting function,
c_j : weighting factor,
rand : random number between 0 and 1,
s_i^k : current position of agent i at iteration k,
$pbest_i$: pbest of agent i,
gbest : gbest of the group.

The following weighting function is usually utilized in (1):

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{2}$$

where, w_{max} : initial weight,
w_{min} : final weight,
$iter_{max}$: maximum iteration number,
iter : current iteration number.

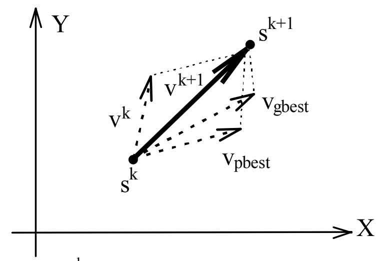
Using the above equation, a certain velocity, which gradually gets close to pbest and gbest can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{3}$$

Fig. 2 shows a concept of modification of a searching point by PSO and Fig. 3 shows a searching concept with agents in a solution space. Each agent changes its current position using the integration of vectors as shown in fig. 2.

The general flow chart of PSO can be described as follows:

Step. 1 Generation of initial condition of each agent
Initial searching points (s_i^0) and velocities (v_i^0) of each agent are usually generated randomly within the allowable range. The current searching point is set to pbest for each agent. The best-evaluated value of pbest is set to gbest and the agent number with the best



s^k : current searching point,
s^{k+1} : modified searching point,
V^k : current velocity,
V^{k+1} : modified velocity,
V_{pbest} : velocity based on pbest
V_{gbest} :velocity based on gbest

Fig.2 Concept of modification of a searching point by PSO.

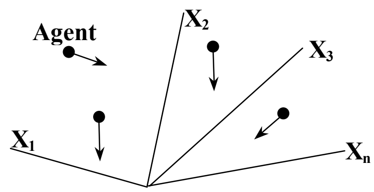


Fig. 3 Searching concept with agents in a solution space by PSO.

value is stored.

Step. 2 Evaluation of searching point of each agent
The objective function value is calculated for each agent. If the value is better than the current pbest of the agent, the pbest value is replaced by the current value. If the best value of pbest is better than the current gbest, gbest is replaced by the best value and the agent number with the best value is stored.

Step. 3 Modification of each searching point
The current searching point of each agent is changed using (1)(2)(3).

Step. 4 Checking the exit condition
The current iteration number reaches the predetermined maximum iteration number, then exit. Otherwise, go to step 2.

Fig. 4 shows the general flow chart of PSO. The features of the searching procedure of PSO can be summarized as follows:

(a) As shown in (1)(2)(3), PSO can essentially handle continuous optimization problem.
(b) PSO utilizes several searching points like genetic algorithm (GA) and the searching points gradually get close to the optimal point using their pbests and the gbest.
(c) The first term of right-hand side (RHS) of (1) is corresponding to diversification in the search procedure. The second and third terms of that are corresponding to intensification in the search procedure. Namely, the method has a well-balanced mechanism to utilize

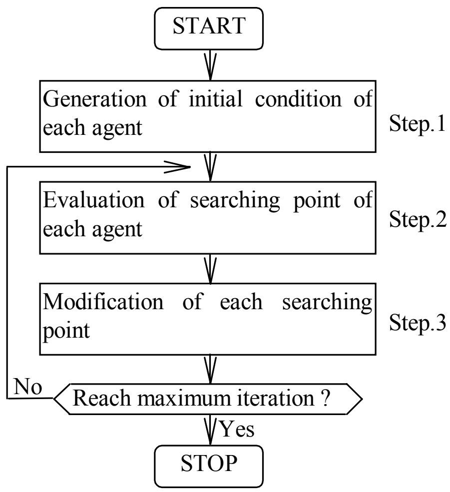


Fig.4 A general flow chart of PSO.

diversification and intensification in the search procedure efficiently.

(d) The above concept is explained using only XY-axis (two-dimension space). However, the method can be easily applied to n-dimension problem. Namely, PSO can handle continuous optimization problems with continuous state variables in a n-dimension solution space.

The above feature (c) can be explained as follows [7]. The RHS of (2) consists of three terms. The first term is the previous velocity of the agent. The second and third terms are utilized to change the velocity of the agent. Without the second and third terms, the agent will keep on "flying" in the same direction until it hits the boundary. Namely, it tries to explore new areas and, therefore, the first term is corresponding to diversification in the search procedure. On the other hand, without the first term, the velocity of the "flying" agent is only determined by using its current position and its best positions in history. Namely, the agents will try to converge to the their pbests and/or gbest and, therefore, the terms are corresponding to intensification in the search procedure. The basic PSO has been applied to a learning problem of neural networks and Schaffer f6, the famous benchmark function for GA, and efficiency of the method has been confirmed [3].

3. VARIATIONS OF PARTICLE SWARM OPTIMIZATION

3.1 Discrete PSO [8]

The original PSO described in II is basically developed for continuous optimization problems. However, lots of practical engineering problems are formulated as combinatorial optimization problems. Kennedy and Eberhart developed a discrete binary version of PSO for the problems [8]. They proposed a model wherein the probability of an agent's deciding yes or no, true or false, or making some other decision, is a function of personal and social factors as follows:

$$P(s_i^{k+1}=1) = f(s_i^k, v_i^k, pbest_i, gbest) \qquad (4)$$

The parameter v, an agent's predisposition to make one or the other choice, will determine a probability threshold. If v is higher, the agent is more likely to choose 1, and lower values favor the 0 choice. Such a threshold requires staying in the range [0, 1]. One of the functions accomplishing this feature is the sigmoid function, which usually utilized with neural networks.

$$sig(v_i^k) = \frac{1}{1+\exp(-v_i^k)} \qquad (5)$$

The agent's disposition should be adjusted for success of the agent and the group. In order to accomplish this, a formula for each v_i^k that will be some function of the difference between the agent's current position and the best positions found so far by itself and by the group. Namely, like the basic continuous version, the formula for binary version of PSO can be described as follows:

$$v_i^{k+1} = v_i^k + rand \times (pbest_i - s_i^k) + rand \times (gbest - s_i^k) \qquad (6)$$

$$\rho_i^{k+1} < sig(v_i^{k+1})\ then\ s_i^{k+1} = 1;\ else\ s_i^{k+1} = 0 \qquad (7)$$

where, rand: a positive random number drawn from a uniform distribution with a predefined upper limit.

ρ_i^{k+1} : a vector of random numbers of [0.0, 1.0].

In the binary version, the limit of rand is often set so that the two rand limits sum to 4.0. These formulas are iterated repeatedly over each dimension of each agent. The second and third term of RHS of (6) can be weighted like the basic continuous version of PSO. v_i^k can be limited so that sig(v_i^k) does not approach too closely to 0.0 or 1.0. This ensures that there is always some chance of a bit flipping. A constant parameter V_{max} can be set at the start of a trial. In practice, V_{max} is often set in [-4.0, +4.0]. The entire algorithm of the binary version of PSO is almost the same as that of the basic continuous version except the above decision equations.

3.2 PSO for MINLP[9]

Lots of engineering problems have to handle both discrete and continuous variables using nonlinear objective functions. Kennedy and Eberhart discussed about integration of binary and continuous version of PSO [5]. Fukuyama, et al., presented a PSO for MINLP by modifying the continuous version of PSO [9]. The method can be briefly described as follows.

Discrete variables can be handled in (1) and (3) with little modification. Discrete numbers instead of continuous numbers can be used to express the current position and velocity. Namely, discrete random number is used for *rand* in (1) and the whole calculation of RHS of (1) is discritized to the existing discrete number. Using this modification for discrete numbers, both continuous and discrete number can be handled in the algorithm with no inconsistency. In [9], the PSO for MINLP was successfully applied to a reactive power and voltage control problem with promising results.

3.3 Hybrid PSO (HPSO) [10]

HPSO utilizes the basic mechanism of PSO and the natural selection mechanism, which is usually utilized by EC methods such as GAs. Since search procedure by PSO deeply depends on pbest and gbest, the searching area may be limited by pbest and gbest. On the contrary, by introduction of the natural selection mechanism, effect of pbest and gbest is gradually vanished by the selection and broader area search can be realized. Agent positions with low evaluation values are replaced by those with high evaluation values using the selection. The exchange rate at the selection is added as a new optimization parameter of PSO. On the contrary, pbest information of each agent is maintained. Therefore, both intensive search in a current effective area and dependence on the past high evaluation position are realized at the same time. Fig. 5 shows a general flow chart of HPSO. Fig. 6 shows concept of step. 2, 3, and 4 of the general flow chart.

3.4 Lbest model

Eberhart and Kennedy called the above-mentioned basic method as “gbest model”. They also developed “lbest model” [5]. In the model, agents have information only of their own and their nearest array neighbor’ bests (lbests), rather than that of the entire group. Namely, in (1), gbest is replaced by lbests in the model.

4. PARAMETER SELECTIONS AND CONSTRICTION FACTOR APPROACH

4.1 Parameter Selection

PSO has several explicit parameters whose values can be adjusted to produce variations in the way the algorithm searches the solution space. The parameters in (1)(2) are as follows:

c_j : weighting factor,
w_{max} : initial weight of the weight function,
w_{min} : final weight of the weight function,

Shi and Eberhart tried to examine the parameter selection of the above parameters [11][12]. According to their examination, the following parameters are appropriate and the values do *not* depend on problems:

c_j=2.0, w_{max}=0.9, w_{min}=0.4.

The values are also appropriate for power system problems [9][13].

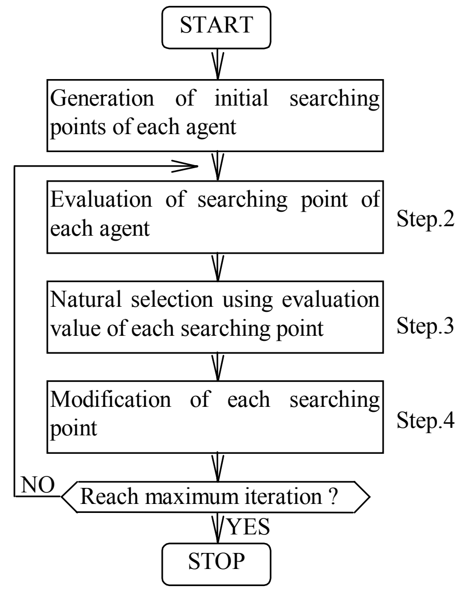


Fig.5 A general flow chart of HPSO.

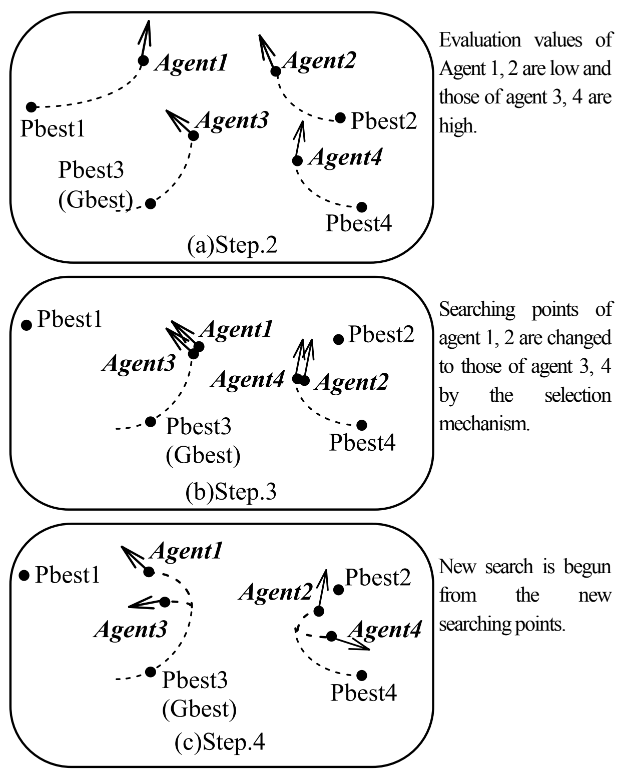


Fig. 6 Concept of searching process by HPSO.

4.2 Constriction factor

The basic system equation of PSO (equ. (1), (2), and (3)) can be considered as a kind of difference equations. Therefore, the system dynamics, namely, search procedure, can be analyzed by the eigen value analysis. The constriction factor approach utilizes the eigen value analysis and controls the system behavior so that the system behavior has the following features [14]:

(a) The system does not diverge in a real value region and finally can converge,

(b) The system can search different regions efficiently.

The velocity of the constriction factor approach (simplest constriction) can be expressed as follows instead of (1) and (2):

$$v_i^{k+1} = K[v_i^k + c_1 \times rand() \times (pbest_i - s_i^k) + c_2 \times rand() \times (gbest - s_i^k)] \quad (8)$$

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, where\, \varphi = c_1 + c_2, \varphi > 4 \quad (9)$$

For example, if φ=4.1, then χ=0.73. As φ increases above 4.0, χ gets smaller. For example, if φ=5.0, then χ=0.38, and the damping effect is even more pronounced. The constriction factor approach results in convergence of the agent over time. Unlike other EC methods, the constriction factor approach of PSO ensures the convergence of the search procedures based on the mathematical theory. Namely, the amplitude of the each agent's oscillation decreases as it focuses on a previous best point. The constriction factor approach can generate higher quality solutions than the conventional PSO approach [15].

However, the constriction factor only considers dynamic behavior of each agent and the effect of the interaction among agents; namely, the effect of pbest and gbest in the system dynamics is one of the future works [14].

5. RESEARCH AREAS AND APPLICATIONS

Ref. [16]-[66] shows other PSO related papers. Most of papers are related to the method itself, and its modification and comparison with other EC methods. PSO is a new EC technique and there are a few applications. Table 1 shows applications of PSO in general fields. The last four applications are in power system fields. Detailed description of [9][32][66] and [50] can be found in Chap.13. Application of PSO to various fields is at the early stage. More applications can be expected.

6. CONCLUSIONS

This chapter presents fundamentals of particle swarm optimization (PSO) techniques. While a lot of evolutionary computation techniques have been developed for combinatorial optimization problems, PSO has been basically developed for continuous optimization problem. PSO has several variations including integration with selection mechanism and hybridization for handling both discrete and continuous variables. Moreover, recently developed constriction factor approach is based on mathematical analysis and useful for obtaining high quality solutions. A few applications are already appeared using PSO. PSO can be an efficient optimization tool for nonlinear continuous optimization problems, combinatorial optimization problems, and mixed-integer nonlinear optimization problem (MINLP).

Table 1 PSO applications.

Application field	No.
Neural network learning algorithm	[14][56]
Human tremor analysis	[7]
Rule Extraction in Fuzzy Neural Network	[33]
Battery Pack State-of-Charge Estimation	[46]
Computer Numerically Controlled Milling Optimization	[65]
Reactive Power and Voltage Control	[9][32][66]
Distribution state estimation	[50]
Power System Stabilizer Design	[16]
Fault State Power Supply Reliability Enhancement	[51]

*) No. shows the paper No. shown in bibliographies section.

REFERENCES

[1] C. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Compter Graphics*, Vol.21, No.4, pp.25-34, 1987.

[2] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies", *Proc. of First European Conference on Artificial Life*, pp.134-142, Cambridge, MA: MIT Press 1991.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks (ICNN'95)*, Vol. IV, pp.1942-1948, Perth, Australia, 1995.

[4] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence : From Natural to Artificial Systems*, Oxford Press, 1999.

[5] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[6] R. Boyd and P. Recharson, *Culture and the Evolutionary Process*, University of Chicago Press, 1985.

[7] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'98)*, pp.69-73, Anchorage, May 1998.

[8] 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)*, pp.4104-4109, 1997.

[9] Y. Fukuyama, et al., "A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment", *IEEE Transaction on Power Systems*, Vol.15, No.4, November 2000.

[10] P. Angeline, "Using Selection to Improve Particle Swarm Optimization", *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'98)*, Anchorage, May 1998.

[11] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization", *Proc. of the 1998 Annual Conference on Evolutionary Programming*, San Diego, 1998.

[12] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'98)*, Anchorage, Alaska, May 4-9, 1998.

[13] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "Practical Distribution State Estimation Using Hybrid Particle Swarm Optimization", *Proc. of*

IEEE Power Engineering Society Winter Meeting, Columbus, Ohio, January 2001.

[14] M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization", *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'99)*, 1999.

[15] R. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization", *Proc. of the Congress on Evolutionary Computation (CEC2000)*, pp.84-88, 2000.

[16] M. A. Abido, "Particle Swarm Optimization for Multimachine Power System Stabilizer Design", *Proc. of IEEE Power Engineering Society Summer Meeting*, July 2001.

[17] P. Angeline, "Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Differences", *Proceeding of The Seventh Annual Conf. on Evolutionary Programming*, March 1998.

[18] P. Angeline, "Using Selection to Improve Particle Swarm Optimization", *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'98)*, Anchorage, Alaska, USA, May 4-9, 1998.

[19] A. Carlisle and G. Dozier, "Adapting Particle Swarm Optimization to Dynamic Environments", *Proceedings of International Conference on Artificial Intelligence*, Monte Carlo Resort, Las Vegas, Nevada, USA

[20] A. Carlisle, and G. Dozier, "An off-the-shelf particle Swarm Optimization", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001 (in press).

[21] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization", *Proc. of 1999 Congress on Evolutionary Computation (CEC'99)*, Washington, DC, pp 1951-1957, 1999.

[22] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp.39-43, Piscataway, NJ: IEEE Service Center, 1995.

[23] R. Eberhart and X. Hu, "Human tremor analysis using particle swarm optimization", *Proc. of Congress on Evolutionary Computation (CEC'99)*, Washington, DC, pp 1927–1930. Piscataway, NJ: IEEE Service Center, 1999.

[24] R. Eberhart and Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", *Proc. of the Seventh Annual Conference on Evolutionary Programming*, March 1998.

[25] R. Eberhart and Y. Shi, "Evolving artificial neural networks", *Proc. of International Conference on Neural Networks and Brain*, Beijing, P.R.C., PL5-PL13, 1998.

[26] R. Eberhart and Y. Shi, Y, "Comparison between genetic algorithms and particle swarm optimization", In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., *Evolutionary Programming* VII: Proc. 7th Annual Conferece on Evolutionary Programming, San Diego, CA. Berlin: Springer-Verlag, 1998.

[27] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *Proc. of Congress on Evolutionary Computation (CEC2000)*, San Diego, CA, pp 84-88, 2000.

[28] R. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms", *Proc. of Congress on Evolutionary Computation (CEC2001)*, Seoul, Korea, Piscataway, NJ: IEEE Service Center, 2001.

[29] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources", *Proc. of Congress on Evolutionary Computation (CEC2001)*, Seoul, Korea, Piscataway, NJ: IEEE Service Center, 2001.

[30] R. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*, Boston: Academic Press Professional, 1996.

[31] H.-Y. Fan and Y. Shi, "Study of Vmax of the particle swarm optimization algorithm", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[32] Y. Fukuyama and H. Yoshida, "A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems", *Proc. of Congress on Evolutionary Computation (CEC2001)*, Seoul, Korea. Piscataway, NJ: IEEE Service Center, 2001.

[33] Z. He, C. Wei, L. Yang, X. Gao, S. Yao, R. Eberhart, and Y. Shi, "Extracting Rules from Fuzzy Neural Network by Particle Swarm Optimization", *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'98)*, Anchorage, Alaska, USA, May 4-9, 1998.

[34] A. Ismail, A. P. Engelbrecht, "Training Product Units in Feedforward Neural Networks using Particle Swarm Optimization", *Proceedings of the International Conference on Artificial Intelligence*, Durban, South Africa, pp 36-40, 1999,

[35] J. Kennedy, "The particle swarm: social adaptation of knowledge", *Proc. of International Conference on Evolutionary Computation (CEC'97)*, Indianapolis, IN, 303-308. Piscataway, NJ: IEEE Service Center, 1997.

[36] J. Kennedy, "Minds and cultures: Particle swarm implications", *Socially Intelligent Agents: Papers from the 1997 AAAI Fall Symposium. Technical Report FS-97-02*, 67-72. Menlo Park, CA: AAAI Press, 1997.

[37] J. Kennedy, "Methods of agreement: inference among the eleMentals", *Proc. of International Symposium on Intelligent Control*, Piscataway, NJ: IEEE Service Center, 1998.

[38] J. Kennedy, "The behavior of particles", In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. *Evolutionary Programming VII: Proc. 7th Ann. Conf. on Evolutionary Programming Conf.*, San Diego, CA, pp.581–589. Berlin: Springer-Verlag, 1998.

[39] J. Kennedy, "Thinking is social: Experiments with the adaptive culture model", *Journal of Conflict Resolution*, Vol.42, pp.56-76, 1998.

[40] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance", *Proc. of Congress on Evolutionary Computation (CEC'99)*, 1931–1938, Piscataway, NJ: IEEE Service Center, 1999.

[41] J. Kennedy, "Stereotyping: improving particle swarm performance with cluster analysis", *Proc. of the 2000 Congress on Evolutionary Computation (CEC2000)*, San Diego, CA, Piscataway, NJ: IEEE Press, 2000.

[42] J. Kennedy, "Out of the computer, into the world: externalizing the particle swarm", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[43] J. Kennedy, and R. Eberhart, "Particle swarm optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks (ICNN)*, Perth, Australia, IEEE Service Center, Piscataway, NJ, IV: pp.1942-1948, 1995.

[44] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm", *Proceedings of the 1997 Conference on Systems, Man, and Cybernetics (SMC'97)*, pp.4104-4109. IEEE Service Center, Piscataway, NJ, 1997.

[45] J. Kennedy and R. Eberhart, "The particle swarm: social adaptation in information processing systems", In Corne, D., Dorigo, M., and Glover, F., Eds., *New Ideas in Optimization*, London: McGraw-Hill, 1999.

[46] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*, San Francisco: Morgan Kaufmann Publishers, 2001.

[47] J. Kennedy and W. Spears, "Matching Algorithms to Problems: An experimental Test of the Particle Swarm and some Genetic Algorithms on the Multimodal Problem Generator", *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'98)*, Anchorage, Alaska, USA, May 4-9, 1998.

[48] M. Løvbjerg, T. Rasmussen, and T. Krink, "Hybrid Particle Swarm Optimization with Breeding and Subpopulations", *Proceedings of the third Genetic and Evolutionary Computation Conference (GECCO-2001)*, 2001.

[49] C. Mohan, and B. Al-kazemi, "Discrete particle swarm optimization",

Proceedings of the Workshop on Particle Swarm Optimization, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[50] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "Practical Distribution State Estimation Using Hybrid Particle Swarm Optimization", *Proc. of IEEE Power Engineering Society Winter Meeting*, Columbus, Ohio, USA, 2001.

[51] K. Nara and Y. Mishima, "Particle Swarm Optimisation for Fault State Power Supply Reliability Enhancement", *Proc. of IEEE International Conference on Intelligent Systems Applications to Power Systems (ISAP2001)*, Budapest, June 2001.

[52] E. Ozcan and C. Mohan, "Analysis of a Simple Particle Swarm Optimization System", *Intelligent Engineering Systems Through Artificial Neural Networks*, Vol.8, pp. 253-258, 1998.

[53] E. Ozcan and C. Mohan, C. K, "Particle Swarm Optimization: Surfing the Waves", *Proc. of 1999 Congress on Evolutionary Computation (CEC'99), Washington*, DC, USA, July 6-9, 1999.

[54] K. Parsopoulos, V. Plagianakos, G. Magoulas, and M. Vrahatis, "Stretching technique for obtaining global minimizers through particle swarm optimization", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[55] T. Ray and K. M. Liew, "A Swarm with an Effective Information Sharing Mechanism for Unconstrained and Constrained Single Objective Optimization Problems", *Proc. of the 2001 Congress on Evolutionary Computation (CEC2001),* Seoul Korea, 2001.

[56] J. Salerno, " Using the Particle Swarm Optimization Technique to Train a Recurrent Neural Model", *Proc. of 9th International Conference on Tools with Artificial Intelligence (ICTAI'97)*, 1997.

[57] B. Secrest and G. Lamont, "Communication in particle swarm optimization illustrated by the traveling salesman problem", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[58] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization", In *Evolutionary Programming VII: Proc. EP98*, New York: Springer-Verlag, pp. 591-600, 1998.

[59] Y. Shi and R. Eberhart, "A modified particle swarm optimizer", *Proceedings of the IEEE International Conference on Evolutionary Computation (CEC'98)*, pp.69-73. Piscataway, NJ: IEEE Press, 1998.

[60] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization", *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pp.1945–1950, Piscataway, NJ: IEEE Service Center, 1999.

[61] Y. Shi and R. Eberhart, "Experimental study of particle swarm optimization", *Proc. of SCI2000 Conference*, Orlando, FL, 2000.

[62] Y. Shi and R. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization", *Proc. of Congress on Evolutionary Computation (CEC2001*), Seoul, Korea. Piscataway, NJ: IEEE Service Center, 2001.

[63] Y. Shi and R. Eberhart, "Particle Swarm Optimization with Fuzzy Adaptive Inertia Weight", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI, 2001.

[64] P. Suganthan, "Particle swarm optimiser with neighbourhood operator", *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pp.1958–1962. Piscataway, NJ: IEEE Service Center, 1999.

[65] V. Tandon, "Closing the gap between CAD/CAM and optimized CNC end milling", Master's thesis, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis, 2000.

[66] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage stability", In G. L. Torres and A. P. Alves da Silva, Eds., *Proc. of Intl. Conf. on Intelligent System Application to Power Systems (ISAP'99)*, Rio de Janeiro, Brazil, pp.117–121, 1999.