

Convergence Analysis of Swarm Algorithm

Hongbo Liu^{1,2,4}, Ajith Abraham^{1,4}, Václav Snášel^{3,4}

¹School of Information Science and Technology, Dalian Maritime University, Dalian, 026, Chi na

²Department of Computer Science, Dalian University of Technology, Dalian, 116023, China

³Department of Computer Science, VŠB-Technical University of Ostrava,

708 33 Ostrava-Poruba, Czech Republic

⁴Machine Intelligence Research Labs - MIR Labs, USA, <http://www.mirlabs.org>

lhb@dlut.edu.cn, ajith.abraham@ieee.org, vaclav.snasel@vsb.cz

Abstract

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm whereby the collective behaviors of unsophisticated individuals interacting locally with their environment cause coherent functional global patterns to emerge. Although the swarm algorithms have exhibited good performance across a wide range of application problems, it is difficult to analyze the convergence. We discuss the swarm intelligent model namely the particle swarm based on its iterated function system. The dynamic trajectory of the particle is described based single individual. We also attempt to theoretically prove that the swarm algorithm converges with a probability of 1 towards the global optimal.

1. Introduction

Swarm Intelligence (SI) is mainly inspired by social behaviour patterns of organisms that live and interact within large groups of unsophisticated autonomous individuals. In particular, it incorporates swarming behaviours observed in flocks of birds, schools of fish, or swarms of bees, colonies of ants, and even human social behavior, from which the intelligence is emerged [1, 2, 3, 4, 5]. SI provides a framework to explore distributed problem solving without centralized control or the provision of a global model. The particle swarm model helps to find optimal regions of complex search spaces through interaction of individuals in a population of particles [6]. It has exhibited good performance across a wide range of applications [7, 8, 9, 10, 11].

However, the executing efficiency and effectiveness of the algorithms are ignored in many important works, since it is difficult to evaluate the performance and convergence of considered algorithms. Trelea discussed the swarm algo-

rithm using standard results from the dynamic system theory. And the exploration – exploitation tradeoff is discussed and illustrated [12]. Bergh and Engelbrecht illustrated particle trajectories for general swarms to include the influence of the inertia term. And they also provides a formal proof that each particle converges to a stable point [13]. Jiang *et al.* investigated the swarm algorithm using stochastic process theory. They analyzed the stochastic convergent condition of the particle swarm system and corresponding parameter selection guidelines [14]. It has been shown that the trajectories of the particles oscillate as different sinusoidal waves and converge quickly. Various methods have been used to identify some other particle to influence the individual. In this paper, we analysis the convergence of swarm algorithm through iterative function system and probabilistic theory.

2 Swarm Algorithm

A particle swarm model consists of a swarm of particles moving in a d -dimensional search space where the fitness f can be calculated as a certain quality measure. Each particle has a position represented by a position-vector \vec{x}_i (i is the index of the particle), and a velocity represented by a velocity-vector \vec{v}_i . Each particle remembers its own best position so far in a vector \vec{p}_i , and its j -th dimensional value is $p_{i,j}$. The best position from the swarm thus far is then stored in a vector \vec{p}_g , and its j -th dimensional value is $p_{g,j}$. During the iteration time t , the update of the velocity from the previous velocity is determined by Eq.(1). Subsequently, the new position is determined by the sum of the previous position and the new velocity by Eq.(2).

$$v_{i,j}(t) = wv_{i,j}(t-1) + c_1r_1(p_{i,j}(t-1) - x_{i,j}(t-1)) + c_2r_2(p_{g,j}(t-1) - x_{i,j}(t-1)) \quad (1)$$

$$x_{i,j}(t) = x_{i,j}(t-1) + v_{i,j}(t) \quad (2)$$

where r_1 and r_2 are the random numbers, uniformly distributed within the interval $[0,1]$ for the j -th dimension of i -th particle. c_1 is a positive constant termed as the coefficient of the self-recognition component; c_2 is a positive constant termed as the coefficient of the social component. The variable w is the inertia factor, for which value is typically setup to vary linearly from 1 to 0 during the iterated processing. From Eq.(1), a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm. In the particle swarm model, the particle searches the solutions in the problem space within a range $[-s, s]$ (If the range is not symmetrical, it can be translated to the corresponding symmetrical range.) In order to guide the particles effectively in the search space, the maximum moving distance during one iteration is clamped in between the maximum velocity $[-v_{max}, v_{max}]$ given in Eq.(3), and similarly for its moving range given in (4):

$$v_{i,j} = \text{sign}(v_{i,j}) \min(|v_{i,j}|, v_{max}) \quad (3)$$

$$x_{i,j} = \text{sign}(x_{i,j}) \min(|x_{i,j}|, x_{max}) \quad (4)$$

The value of v_{max} is $\rho \times s$, with $0.1 \leq \rho \leq 1.0$ and is usually chosen to be s , i.e. $\rho = 1$. The pseudo-code for particle-search is illustrated in Algorithm 1.

Algorithm 1 Particle Swarm Algorithm

```

01. Initialize the size of the particle swarm  $n$ , and other
01. parameters; Initialize the positions and the velocities
01. for all the particles randomly.
02. While (the end criterion is not met) do
03.  $t = t + 1$ ;
04. Calculate the fitness value of each particle;
05.  $\vec{p}^* = \text{argmin}_{i=1}^n (f(\vec{p}^*(t-1)), f(\vec{p}_1(t)),$ 
05.  $f(\vec{p}_2(t)), \dots, f(\vec{p}_i(t)), \dots, f(\vec{p}_n(t)))$ ;
06. For  $i=1$  to  $n$ 
07.  $\vec{p}_i^\#(t) = \text{argmin}_{i=1}^n (f(\vec{p}_i^\#(t-1)), f(\vec{p}_i(t)))$ ;
08. For  $j=1$  to  $d$ 
09. Update the  $j$ -th dimension value of  $\vec{p}_i$  and  $\vec{v}_i$ 
09. according to Eqs.(1), (3), (2), (4);
10. Next  $j$ 
11. Next  $i$ 
12. End While.

```

The particle swarm algorithm can be described generally as a population of vectors whose trajectories oscillate around a region which is defined by each individual's previous best success and the success of some other particle. Eberhart and Kennedy called the two basic methods as "gbest model" and "lbest model" [2]. Some previous studies has been shown that gbest model converges quickly on

problem solutions but has a weakness for becoming trapped in local optima, while lbest model converges slowly on problem solutions but is able to "flow around" local optima, as the individuals explore different regions [15, 16]. Some related research and development during the recent years are also reported in [9, 17].

3. Individual Trajectory

Some previous studies have discussed the trajectory of particles and the convergence of the algorithm [6, 12, 13]. It has been shown that the trajectories of the particles oscillate as different sinusoidal waves and converge quickly. We represent particle swarm as an iterated function system. The dynamic trajectory of the particle is described based single individual.

Clerc and Kennedy have stripped the particle swarm model down to a most simple form [6]. If the self-recognition component c_1 and the coefficient of the social-recognition component c_2 in the particle swarm model are combined into a single term c , i.e. $c = c_1 + c_2$, the best position \vec{p}_i can be redefined as follows:

$$\vec{p}_i \leftarrow \frac{(c_1 \vec{p}_i + c_2 \vec{p}_g)}{(c_1 + c_2)} \quad (5)$$

Then, the update of the particle's velocity is defined by:

$$\vec{v}_i(t) = \vec{v}_i(t-1) + c(\vec{p}_i - \vec{x}_i(t-1)) \quad (6)$$

The system can be simplified even further by using $\vec{y}_i(t-1)$ instead of $\vec{p}_i - \vec{x}_i(t-1)$. Thus, the reduced system is then:

$$\begin{cases} \vec{v}(t) = \vec{v}(t-1) + c\vec{y}(t-1) \\ \vec{y}(t) = -\vec{v}(t-1) + (1-c)\vec{y}(t-1) \end{cases}$$

This recurrence relation can be written as a matrix-vector product, so that

$$\begin{bmatrix} \vec{v}(t) \\ \vec{y}(t) \end{bmatrix} = \begin{bmatrix} 1 & c \\ -1 & 1-c \end{bmatrix} \cdot \begin{bmatrix} \vec{v}(t-1) \\ \vec{y}(t-1) \end{bmatrix}$$

Let

$$\vec{P}_t = \begin{bmatrix} \vec{v}_t \\ \vec{y}_t \end{bmatrix}$$

and

$$A = \begin{bmatrix} 1 & c \\ -1 & 1-c \end{bmatrix}$$

we have an iterated function system for the particle swarm model:

$$\vec{P}_t = A \cdot \vec{P}_{t-1} \quad (7)$$

For the iterated system determined by Eq.(7), the eigenvalues of A are λ_1 and λ_2 . We are looking for pair of values (c, k) so that

$$A^k = I \quad (8)$$

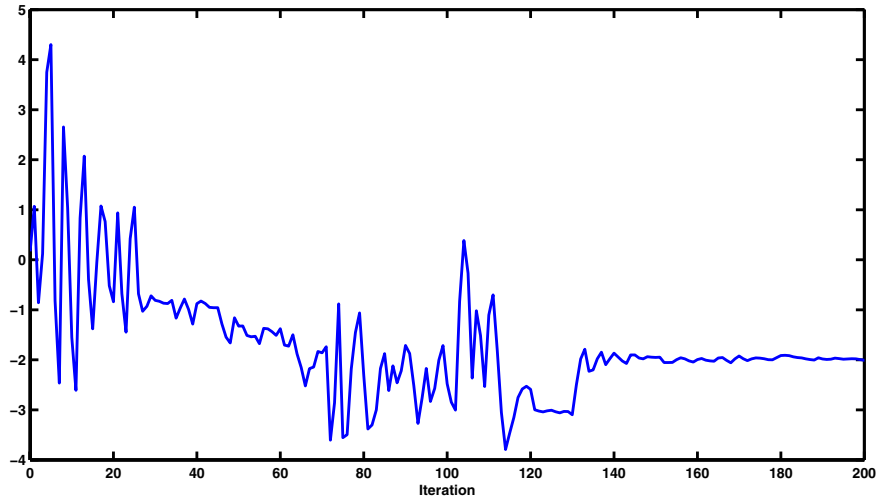


Figure 1. Trajectory of a single particle.

where I is the identity matrix. We have $\det(A) > 0$ (equal to 1, in fact), so it exists P so that

$$P^{-1}AP = \Lambda \quad (9)$$

where

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Eq.(8) can then be rewritten

$$(P\Lambda P^{-1})^k = \Lambda^k = I \quad (10)$$

It means we must have

$$\lambda_1^k = \lambda_2^k = 1 \quad (11)$$

But we have

$$\begin{cases} \lambda_1 = 1 - \frac{c}{2} + \sqrt{\Delta} \\ \lambda_2 = 1 - \frac{c}{2} - \sqrt{\Delta} \end{cases} \quad (12)$$

with c is strictly positive, and

$$\Delta = \left(1 - \frac{c}{2}\right)^2 - 1$$

So it is possible for Eq.(11) only if the eigenvalues are true complex numbers, i.e. if Δ is strictly negative. It implies that c must be smaller than 4.

It is easy to see that λ_1 and λ_2 are conjugate complex number. It means $\lambda_1 = \bar{\lambda}_2$.

So finally we have k solution for $\lambda_1^k = 1$ these solution can be computed by De Moivre Theorem [18]:

$$\cos\left(\frac{s2\pi}{k}\right) + i\sin\left(\frac{s2\pi}{k}\right)$$

where $s = 0, \dots, k-1$.

There are an infinity of such cycles but in [1, 4] the only possible ones are for cycle sizes $k \in \{3, 4, 5, 6\}$.

For $k = 3$ we have the following root of unity:

$$1, -\frac{1}{2} + \frac{i\sqrt{3}}{2}, -\frac{1}{2} - \frac{i\sqrt{3}}{2}$$

For $k = 4$:

$$1, -1, i, -i$$

For $k = 5$:

$$1, \frac{\sqrt{5}-1}{4} + i\frac{\sqrt{10+2\sqrt{5}}}{4}, \frac{\sqrt{5}-1}{4} - i\frac{\sqrt{10+2\sqrt{5}}}{4}, \\ \frac{-\sqrt{5}-1}{4} + i\frac{\sqrt{10-2\sqrt{5}}}{4}, \frac{-\sqrt{5}-1}{4} - i\frac{\sqrt{10-2\sqrt{5}}}{4}$$

For $k = 6$:

$$1, -1, -\frac{1}{2} + \frac{i\sqrt{3}}{2}, -\frac{1}{2} - \frac{i\sqrt{3}}{2}, \\ \frac{1}{2} + \frac{i\sqrt{3}}{2}, \frac{1}{2} - \frac{i\sqrt{3}}{2}$$

The solution must hold Eq.(12) and $0 < c < 4$. It is easy see that we get only the following solutions:

$$(k, c) = (3, 3), (4, 2), (5, \frac{3-\sqrt{5}}{2}), (5, \frac{3+\sqrt{5}}{2}), (6, 1)$$

The trajectory of a single particle is illustrated in Figure 1.

4. Probabilistic Analysis

For analyzing the convergence of the swarm algorithm, we first introduce the definitions and lemmas [19, 20, 21, 22, 23, 24, 25].

Consider the problem (P) as

$$(P) = \begin{cases} \min f(\vec{x}) \\ \vec{x} \in \Omega = [-s, s]^n \end{cases} \quad (13)$$

where $\vec{x} = (x_1, x_2, \dots, x_n)^T$. \vec{x}^* is the global optimal solution to the problem (P), let $f^* = f(\vec{x}^*)$. Let

$$\begin{aligned} D_0 &= \{\vec{x} \in \Omega | f(\vec{x}) - f^* < \varepsilon\} \\ D_1 &= \Omega \setminus D_0 \end{aligned} \quad (14)$$

for every $\varepsilon > 0$.

Since i -th dimensional value of the particle's velocity is generated by Eq.(1), the variable velocity belongs to the normal distribution. When $v_{max} = s$, $v \sim [-s, s]$. During the iterated procedure from the time t to $t + 1$, let q_{ij} denote that $\vec{x}(t) \in D_i$ and $\vec{x}(t + 1) \in D_j$. Accordingly the particles' positions in the swarm could be classified into four states: q_{00} , q_{01} , q_{10} and q_{11} . Obviously $q_{00} + q_{01} = 1$, $q_{10} + q_{11} = 1$.

Definition 1 (Convergence in terms of probability) Let ξ_n a sequence of random variables, and ξ a random variable, and all of them are defined on the same probability space. The sequence ξ_n converges with a probability of ξ if

$$\lim_{n \rightarrow \infty} P(|\xi_n - \xi| < \varepsilon) = 1 \quad (15)$$

for every $\varepsilon > 0$. The sequence ξ_n converges almost surely or almost everywhere or with probability of 1 or strongly towards ξ if

$$P\left(\lim_{n \rightarrow \infty} \xi_n = \xi\right) = 1; \quad (16)$$

or

$$P\left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} [|\xi_k - \xi| \geq \varepsilon]\right) = 0 \quad (17)$$

for every $\varepsilon > 0$.

Lemma 1 (Borel-Cantelli Lemma) Let $\{A_k\}_{k=1}^{\infty}$ be a sequence of events occurring with a certain probability distribution, and let A be the event consisting of the occurrences of a finite number of events A_k for $k = 1, 2, \dots$. Then

$$P\left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\right) = 0 \quad (18)$$

if

$$\sum_{n=1}^{\infty} P(A_n) < \infty; \quad (19)$$

$$P\left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\right) = 1 \quad (20)$$

if the events are totally independent and

$$\sum_{n=1}^{\infty} P(A_n) = \infty. \quad (21)$$

Lemma 2 (Particle State Transference) $q_{01} = 0$; $q_{00} = 1$; $q_{11} \leq c \in (0, 1)$ and $q_{10} \geq 1 - c \in (0, 1)$.

Proof 1 In the swarm algorithm, the best solution is updated and saved during the whole iterated procedure. So $q_{01} = 0$ and $q_{00} = 1$.

Let \vec{x} is the position with the best fitness among the swarm so far as the time t , i.e. $\vec{x} = \vec{p}^*$. As the definition in Eq.(14), $\exists r > 0$, when $\|\vec{x} - \vec{x}\|_{\infty} \leq r$, we have $|f(\vec{x}) - f^*| < \varepsilon$. Denote $Q_{\vec{x}, r} = \{x \in \Omega | \|\vec{x} - \vec{x}\|_{\infty} \leq r\}$. Accordingly

$$Q_{\vec{x}, r} \subset D_0 \quad (22)$$

Then,

$$\begin{aligned} &P\{(\vec{x} + \Delta\vec{x}) \in Q_{\vec{x}, r}\} \\ &= \prod_{i=1}^n P\{|x_i + \Delta x_i - \hat{x}_i| \leq r\} \\ &= \prod_{i=1}^n P\{\hat{x}_i - x_i - r \leq \Delta x_i \leq \hat{x}_i - x_i + r\} \end{aligned} \quad (23)$$

where x_i , Δx_i and \hat{x}_i are the i -th dimensional values of \vec{x} , $\Delta\vec{x}$ and \vec{x} , respectively. Moreover, $\hat{v} \sim [-s, s]$, so that

$$P((\vec{x} + \Delta\vec{x}) \in Q_{\vec{x}, r}) = \prod_{i=1}^n \int_{\hat{x}_i - x_i - r}^{\hat{x}_i - x_i + r} \frac{\eta}{2\sqrt{2\pi}s} e^{-\frac{y^2}{2s^2}} dy \quad (24)$$

Denote $P_1(\vec{x}) = P\{(\vec{x} + \Delta\vec{x}) \in Q_{\vec{x}, r}\}$ and \mathcal{C} is the convex closure of level set for the initial particle swarm. According to Eq.(24), $0 < P_1(\vec{x}) < 1$ ($\vec{x} \in \mathcal{C}$). Again, since \mathcal{C} is a bounded closed set, so $\exists \vec{y} \in \mathcal{C}$,

$$P_1(\vec{y}) = \min_{\vec{x} \in \mathcal{C}} P_1(\vec{x}), \quad 0 < P_1(\vec{y}) < 1. \quad (25)$$

Considering synthetically Eqs.(22) and (25), so that

$$q_{10} \geq P_1(\vec{x}) \geq P_1(\vec{y}) \quad (26)$$

Let $c = 1 - P_1(\vec{y})$, thus,

$$q_{11} = 1 - q_{10} \leq 1 - P_1(\vec{y}) = c \quad (0 < c < 1) \quad (27)$$

and

$$q_{10} \geq 1 - c \in (0, 1) \quad (28)$$

Theorem 1 Assume that the swarm algorithm provides position series $\vec{p}_i(t) (i = 1, 2, \dots, n)$ at time t by the iterated procedure. \vec{p}^* is the best position among the swarm explored so far, i.e.

$$\vec{p}^*(t) = \arg \min_{1 \leq i \leq n} (f(\vec{p}^*(t-1)), f(\vec{p}_i(t))) \quad (29)$$

Then,

$$P\left(\lim_{t \rightarrow \infty} f(\vec{p}^*(t)) = f^*\right) = 1 \quad (30)$$

Proof 2 For $\forall \varepsilon > 0$, let $p_k = P\{|f(\vec{p}^*(k)) - f^*| \geq \varepsilon\}$, then

$$p_k = \begin{cases} 0 & \text{if } \exists T \in \{1, 2, \dots, k\}, \vec{p}^*(T) \in D_0 \\ \bar{p}_k & \text{if } \vec{p}^*(t) \notin D_0, t = 1, 2, \dots, k \end{cases} \quad (31)$$

According to Lemma 2,

$$\bar{p}_k = P\{\vec{p}^*(t) \notin D_0, t = 1, 2, \dots, k\} = q_{11}^k \leq c^k. \quad (32)$$

Hence,

$$\sum_{k=1}^{\infty} p_k \leq \sum_{k=1}^{\infty} c^k = \frac{c}{1-c} < \infty. \quad (33)$$

According to Lemma 1,

$$P\left(\bigcap_{t=1}^{\infty} \bigcup_{k \geq t} |f(\vec{p}^*(k)) - f^*| \geq \varepsilon\right) = 0 \quad (34)$$

As defined in Definition 1, the sequence $f(\vec{p}^*(t))$ converges almost surely or almost everywhere or with probability 1 or strongly towards f^* . The theorem is proven.

5. Curve Illustration

In this section, we provide two examples to illustrate the convergence through the algorithm performance curves.

In our first experiment, the algorithm was tested using the numerical function. The optimization objective is Quadric function, but it has dimensional effect ($i * x_i$) with noise. We tested the algorithms exploring 200 dimensions. For the function, the goal was to find the global minima. The parameters c_1 and c_2 were set to 1.49. Inertia weight w was decreased linearly from 0.9 to 0.1. The experiment was repeated 10 times with different random seeds. Each trial had a fixed number of 8,000 iterations. The objective functions were evaluated 360,000 times during each trial, because the swarm size was set to 20. The average fitness values of the best solutions throughout the optimization run were recorded. Figure 2 illustrates the mean best function values for the objective function using the swarm algorithm. Quartic function with noise:

$$f_3 = \sum_{i=1}^n (ix_i^4) + \text{random}[0, 1).$$

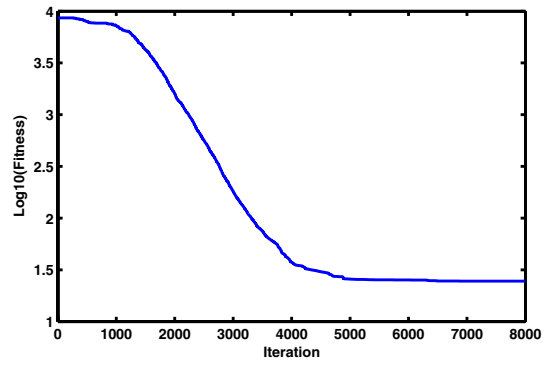


Figure 2. Performance for 200-D Quartic with Noise function.

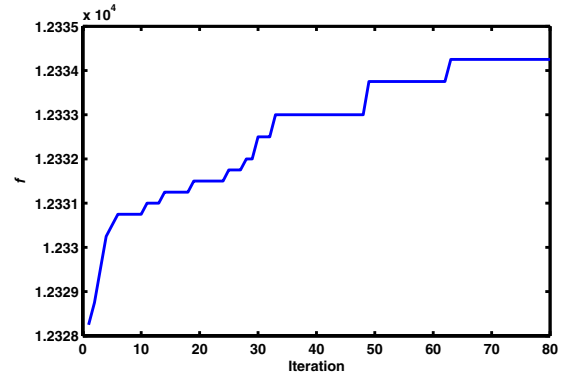


Figure 3. Performance for the NS (35, 1400, 17)

$$\vec{x} \in [-1.28, 1.28]^n, \\ \min(f_3(\vec{x}^*)) = f_3(\vec{0}) = 0.$$

The second experiment is using the swarm algorithm to Neighbor Selection (NS) in P2P Networks. In order to maximize the disjointness of content, the system has to maximize the number of content pieces each peer can contribute to its neighbors by determining the connections. We discussed to apply the particle swarm algorithm for the problem (35,1400,17) [26]. The algorithm was repeated 4 times with different random seeds. Each trial had a fixed number of 80 iterations. Other specific parameter settings of the algorithms are described in [26]. The average fitness values of the best solutions throughout the optimization run were recorded. Figure 3 illustrates the algorithm performance during the search processes for the NS problem. As evident, the algorithm provided good convergence performance for the NS problems.

6. Conclusions

In this paper, we focused on the convergence of swarm algorithm based on iterative function system and probabilistic theory. Particle swarm was investigated as a case and the swarm model was represented by the iterated function system. The dynamic trajectory of the particle was derived. We provided theoretically prove that the swarm algorithm converges with a probability of 1 towards the global optimal. Algorithm performance curves were illustrated through optimization experiments. The works would be helpful for developing new swarm intelligent swarm models and selecting appropriate parameters [27].

7. Acknowledgements

The authors would like to thank Maurice Clerc, Bo Li and Ran He for their scientific collaboration in this research work. This work was supported by NSFC (60873054).

References

- [1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.
- [2] J. Kennedy, and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, CA, 2001.
- [3] A.P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley, NY, 2005.
- [4] M. Clerc. *Particle Swarm Optimization*. ISTE Publishing Company, London, 2006.
- [5] M. Su, S. Su, and Y. Zhao. “A swarm-inspired projection algorithm”. *Pattern Recognition*, 2009, 42(11), pp. 2764–2786.
- [6] M. Clerc and J. Kennedy. “The Particle Swarm-explosion, Stability, and Convergence in A Multidimensional Complex Space”. *IEEE Transactions on Evolutionary Computation*, 2002, 6, pp. 58–73.
- [7] K.E. Parsopoulos, and M.N. Vrahatis. “Recent Approaches to Global Optimization Problems through Particle Swarm Optimization”. *Natural Computing*, 2002, 1, pp. 235–306. 235–306.
- [8] J.F. Schute, and A.A. Groenwold. “A Study of Global Optimization Using Particle Swarms”. *Journal of Global Optimization*, 2005, 3, pp. 193–108.
- [9] H. Liu, and A. Abraham. “An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems”. *Journal of Universal Computer Science*, 2007, 13(7), pp. 1032–1054.
- [10] B. Yue, W. Yao, A. Abraham and H. Liu. “A New Rough Set Reduct Algorithm Based on Particle Swarm Optimization”. *Lecture Notes in Computer Science*, 2007, 4527, pp. 397–406.
- [11] H. Liu, A. Abraham, and A.E. Hassanien. “Scheduling Jobs on Computational Grids Using Fuzzy Particle Swarm Optimization Algorithm”. *Future Generation Computer Systems*, <http://dx.doi.org/10.1016/j.future.2009.05.022>, 2009.
- [12] T.I. Cristian “The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection”. *Information Processing Letters*, 2003, 85(6), pp. 317–325.
- [13] F. van den Bergh, and A.P. Engelbrecht. “A Study of Particle Swarm Optimization Particle Trajectories”. *Information Sciences*, 2006, 176, pp. 937–971.
- [14] M. Jiang, Y.P. Luo, and S.Y. Yang. “Stochastic Convergence Analysis and Parameter Selection of the Standard Particle Swarm Optimization Algorithm”. *Information Processing Letters*, 2007, 102, pp. 8–16.
- [15] H. Liu, B. Li, Y. Ji, and T. Sun. “Particle Swarm Optimization from lbest to gbest”. ; In : , *Applied Soft Computing Technologies: The Challenge of Complexity*, Series: Advances in Soft Computing, Abraham, A., Baets, B.D., Köppen, M., Nickolay, B. (Eds.), Springer, 2006, pp. 537–545.
- [16] S. Ghosh, D. Kundu, K. Suresh, S. Das, A. Abraham, B.K. Panigrahi, and V. Snášel. “On Some Properties of the lbest Topology in Particle Swarm Optimization”. *Proceedings of Ninth International Conference on Hybrid Intelligent Systems*, China, IEEE Computer Society Press, 2009, pp. 370–376.
- [17] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar. “Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions”. *IEEE Transactions on Evolutionary Computation*, 2006, 10(3), pp. 281–295.
- [18] T. Andreescu, and D. Andrica. *Complex Numbers from A to ... Z*. Birkhäuser, Boston, 2005.
- [19] C. Guo, and H. Tang. “Global Convergence Properties of Evolution Strategies”. *IEEE Mathematica Numerica Sinica*, 2001, 23(1), pp. 105–110.
- [20] R. He, Y. Wang, Q. Wang, J. Zhou, and C. Hu. “An Improved Particle Swarm Optimization Based on Self-adaptive Escape Velocity”. *Chinese Journal of Software*, 2005, 16(12), pp. 2036–2044.
- [21] P. Halmos. *Measure theory*. Van Nostrand and Co., New York, 2005.
- [22] Z. Xu, G. Cheng, and Y. Liang. “Search Capability for An Algebraic Crossover”. *Journal of Xi’an Jiaotong University*, 1999, 33(10), pp. 88–99.
- [23] L.D. Whitley. “Fundamental Principles of Deception in Genetic Search”. *Foundation of Genetic Algorithms*, California: Morgan Kaufmann Publishers, 1991, pp. 221–241.
- [24] J.H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.
- [25] D.E. Goldberg. “Genetic Algorithms in search, optimization, and machine learning”. Addison-Wesley Publishing Corporation, Inc., 1989.
- [26] S. Sun, A. Abraham, G. Zhang, and H. Liu. “A Particle Swarm Optimization Algorithm for Neighbor Selection in Peer-to-Peer Networks”. *Proceedings of the Sixth Computer Information Systems and Industrial Management Applications (CISIM’07)*, IEEE Computer Society Press, 2007, pp. 166–172.
- [27] H. Liu, A. Abraham, and M. Clerc. “Chaotic Dynamic Characteristics in Swarm Intelligence”. *Applied Soft Computing Journal*, 2007, 7, pp. 1019–1026.