Interpolation Search for Swarm Intelligence to Optimization Problem

Takuto Tanaka and Koji Okuhara
Dept. of Information and Physical Sciences,
Graduate School of Information Science and Technology,
2-1 Yamadaoka, Suita, Osaka, 565-0871 Japan, Osaka University

E-mail: takuto-tanaka@ist.osaka-u.ac.jp

Abstract

Swarm Intelligence is the optimization technique based on the action of groups, such as a bird and a fish, and a colony of the ant. PSO which is one of the technique is developed and applied in various studies. However, convergence of PSO is groundless. From the point of view, in this paper, we have proposed a new hybrid dynamical system of Swarm Intelligence and Neural Network dynamics for finding better optimal solution. As the main results of this paper, we first show how to combine PSO and neural network mechanism by the theoretical analysis, and confirmed that proposed system can realize interpolation search based on global information of objective environment.

Keywords: Particle Swarm Optimization, Neural Network Dynamics

1 Introduction

Particle swarm optimization (PSO) is the technique of finding the optimal solution in consideration of the past search history from the best information (p-best) which a solid (particle) has, and the optimal value (g-best) of the group (swarm) formed from the solid. It is a parallel evolutionary computation technique developed by Kennedy [3] based on the social behavior. A standard study on PSO, treating both the social and computational method, is [4]. The PSO algorithm is initialized with a population of random candidate solutions, conceptualized as particles. Each particle is assigned a randomized velocity and is iteratively moved through the problem space. It is attracted towards the location of the best fitness achieved so far by the particle itself and by the location of the best fitness achieved so far across the whole population.

PSO is calculated updating the position and velocity of an each particle (see Fig. 1). The discrete time PSO algorithm is as follows:

$$x_d^{k+1} = x_d^k + v_d^{k+1}$$
$$v_d^{k+1} = wv_d^k + c_1 r_1(\hat{x}_{db}^k - x_d^k) + c_2 r_2(\hat{x}_{gb}^k - x_d^k)$$

where x_d^k is the position and v_d^k is the velocity of an individual d of the k-th iteration, r_1 and r_2 are [0,1] random number, c_1 and c_2 are parameters. In addition, w is a parameter to be called momentum, x_{db}^k x_{gb}^k shows the p-best and g-best respectively. Some stability analysis for the discrete time PSO have been discussed [5-7]. Recently, the development of the computer science of hardware and software is remarkable. In particular, the evolution method is performed flourishingly and developed and applied from the easiness of the application with the simulation base [2]. For example, hybrid method of PSO and Simplex method is suggested [1]. And also, method of PSO and GA (Genetic Algorithm) is suggested [2]. By the way, the importance of the optimization engineering for the large-scale problem rises more and more. Searching problems of log and pass also become large-scale by rise of Social Network Service (SNS). It takes time to solve these problems with the latest computer. Therefore, we provide the new hybrid dynamical system that it is found the most optimal solution with the number of few steps. From Continuous PSO algorithm, we obtain the new dynamical system Σ_n and we can formulate equivalent dynamics to a neural network from the continuous time PSO algorithm. Then, we consider application of reinforcement learning mechanism to realize interpolation search of global optimum.

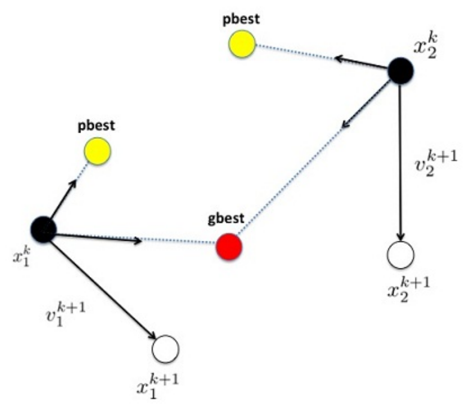


Fig. 1: Move of search point.

This paper is organized as follows. In section 2, we briefly describe the Continuous PSO algorithm. Next, we present the new dynamical system and derive it with steepest descent method In section 3. Then, we provide numerical experiments to show the effectiveness of the presented dynamics In section 4. Finally, Section 5 concludes the paper.

2 Continuous PSO algorithm

Now we consider the feasible region of problem solution and represent the continuous time PSO dynamics by matrices as follows Σ_m:

$$\begin{aligned} \dot{X} &= V \\ \dot{V} &= -\alpha V + \beta(X_{db} - X) + \gamma(X_{gb}T - X) \end{aligned}$$

However, it should be noted that X and V are not the only system states, X_{db} is also a state since it has a 'memory '.

$$\begin{aligned} \dot{X_{db}} &= a(X - X_{db})[I_n + \mathrm{diag}[\mathrm{sgn}(F(X_{gb} - F(X))]] \\ \dot{X_{gb}} &= X_{db}Q_j \quad \text{where } j = \arg \inf_{0<i\leq n} (f(x_{db_i})) \end{aligned}$$

where d denotes the problem dimension, $\Omega \subseteq R^d$ represents the feasible region, n denotes the number of birds, $f : \Omega \to R$ is the function to be minimized. In addition, for the sake of compactness, the following vectors and matrices are defined as follows:

- $X \triangleq [x_1 \cdots x_n] \in (\Omega \times \Re^n)$ the position matrix
- $V \triangleq [v_1 \cdots v_n] \in (\Re^d \times \Re^n)$ the velocity matrix
- $X_{db} \triangleq [x_{db_1} \cdots x_{db_n}] \in (\Omega \times \Re^n)$ the local best position matrix
- $X_{gb} \in \Re^d$ the global best position matrix
- $F \triangleq [f(x_1) \cdots f(x_n)] : (\Omega \times \Re^n) \to \Re^n$ is the stacked objective function raw vector
- T is a row vector composed of ones
- $Q_i \in \Re^n$ is a column vector having all elements equal to zero except the i^{th} element that equals one
- $I_n(\Re^n \in \Re^n)$ is the identity matrix of size n

Let diag$[y]$ be a diagonal matrix with the diagonal elements given by the elements of the vector y and sgn(y) denotes the sigma function for y, defined as: $\mathrm{sgn}(y) = 1$ if $y > 0$, and $\mathrm{sgn}(y) = -1$ if $y < 0$. Hence, assuming to be a positive constant, it is proposed to approximate the evolution of X for a minimization. Also stability analysis for the continuous time PSO has been discussed [8]. The notation used above is not the standard state space notation since the state variable X, V and X_{db} are not vectors but rather matrices of the adequate dimension previously defined. This description is motivated by the simplicity and compactness it provides without loss of clarity.

[Algorithm of CPSO]

Step 1 Set initial value of X, V and parameters α, β, γ and a.

Step 2 Derive initial value of X_{db}, X_{gb}.

Step 3 Calculate $\dot{V}$ and update V.

Step 4 Update X and evaluate X_{db}, X_{gb}.

Step 5 If it assume to be converge then finish otherwise return **Step 3**.

3 Proposal of hybrid PSO algorithm

We define the following Z,

$$Z = \beta(X_{db} - X) + \gamma(X_{gb}T - X)$$

and consider the following condition.

$$X = X_0 + \int_0^t V(s)ds$$

where X_0 is the initial position matrix. Here we introduce hybrid dynamical system Σ_n (see Fig. 2):

$$\begin{aligned} \dot{X} &= V \\ \dot{V} &= -\alpha V + Z \\ \dot{Z} &= \beta(\dot{X_{db}} - \dot{X}) + \gamma(\dot{X_{gb}}T - \dot{X}) + \delta(\dot{X_\mu}) \end{aligned}$$

where the dimensions of X, V, X_{db}, X_{gb}, and T are as defined above; and the time notation is dropped for brevity.

$$\begin{aligned} \dot{X_{db}} &= a(X - X_{db})[I_n + \mathrm{diag}[\mathrm{sgn}(F(X_{gb} - F(X))]] \\ \dot{X_{gb}} &= X_{db}Q_j \quad \textit{where } j = \arg \inf_{0<i\leq n} (f(x_{db_i})) \end{aligned}$$

where α, β, γ and δ are positive non zero reals, and satisfy $\beta + \gamma + \delta = 1$. These reals are parameters which you put weight to adjust PSO and neural network. X_μ is the new matrix which derived from neural network dynamics. X_μ is defined the following dynamics [7] (see Fig. 3) :

$$\begin{aligned} \dot{x}_{\mu i} &= -C \sum_{i=1}^{n} \frac{\partial f(y_i(t))}{\partial y_i} \frac{\partial \varphi(x(t))}{\partial x_i} (= \dot{z_i}(t)) \\ \dot{x_i} &= -a x_i(t) + z_i(t) \\ y_i(t) &= \varphi(x_i(t)) \end{aligned}$$

Here we introduce the discrete formula, thus z_i^k is vector of Z and ${x_\mu}_i^k$ is vector of X_μ about individual i of the k-th iteration respectively. Values a and C are parameters and φ is sigmoid function. Then, we adapt difference method.

From a viewpoint of theoretical analysis, $\dot{V}$ of PSO and $\dot{x_i}$of neural network regard it as an equivalent thing. $\beta(\dot{X_{db}} - \dot{X}) + \gamma(\dot{X_{gb}}T - \dot{X})$ controls the velocity of PSO

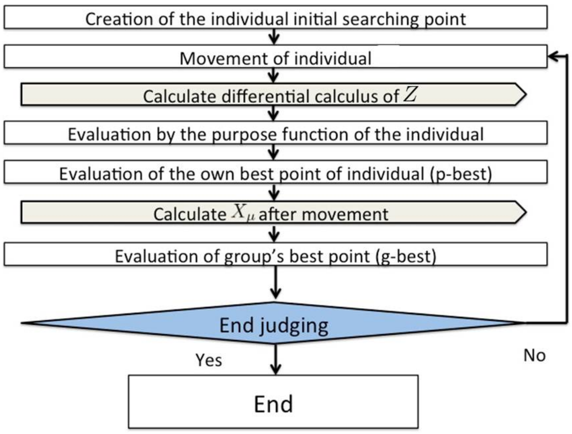


Fig. 2: Algorithm of proposed system Σ_n.

and $\delta(\dot{X_\mu})$ controls a neural network velocity. Therefore, $\dot{Z} = \beta(\dot{X_{db}} - \dot{X}) + \gamma(\dot{X_{gb}T} - \dot{X}) + \delta(\dot{X_\mu})$ of the hybrid proposed theoretically will have the global search which PSO has, and the local steepest descent method which a neural network has. Although the theoretical algorithm for combination of PSO and a neural network is considered by the continuous time model, a numerical simulation is done by the dispersion model. A setup of sampling time will change with the values of the coefficient of β, γ, δ.

Thus, we calculate and obtain the X_μ.

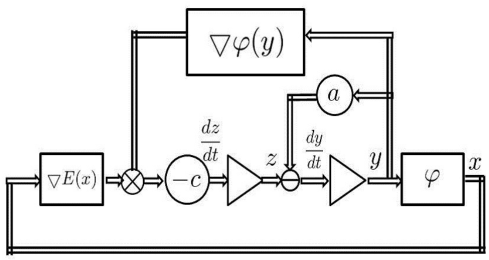


Fig. 3: Dynamics of X_μ

[Algorithm of proposed hybrid PSO]

Step 1 Set initial value of X, V, X_μ and parameters α, β, γ, δ and a.

Step 2 Derive initial value of X_{db}, X_{gb}, and Z.

Step 3 Calculate $\dot{Z}$ and update Z.

Step 4 Calculate $\dot{V}$ and update V.

Step 5 Update X and evaluate X_{db}, X_{gb}, and Z.

Step 6 Update X_μ.

Step 7 If it assume to be converge then finish otherwise return **Step 3**.

4 Numerical Experiments

In this section, we provide numerical experiments to compare the behavior of Σ_m and Σ_n and we discuss the relationship between the quality of the solution and these measures. To this end, we use the three kinds of test functions to be minimized which are used in [10]. The test functions in n-dimensional space are as follows. (see Fig. 4) :

$$f_{Griewank} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} \cos\left(\frac{\sqrt{i}}{x_i}\right) + 1$$

$$f_{Rosenbrock} = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

$$f_{Rastrigin} = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$$

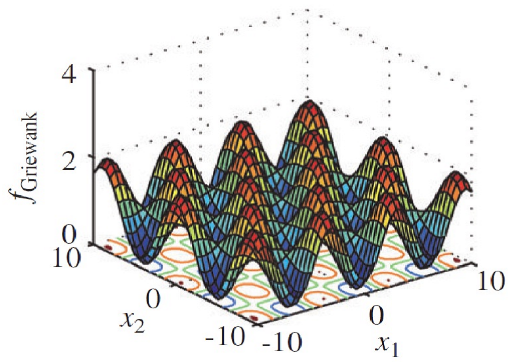


Fig. 4: Shapes of Griewank functions.

Figs. 4, 5, and 6 shows the shapes of the test functions in 2-dimensional space. The optimal values of the three functions are all 0 achieved at $x_i = 1$ for $f_{Rosenbrock}$ and at the origin for other functions.

Table 1: Performance of two different algorithms. a: PSO, b: Proposal of hybrid PSO

a: PSO, b: Proposal of hybrid PSO, Average of 150 times						
Function	Iterations = 1000		Iterations = 1500		Iterations = 2000	
Griewank	a: dim10	b: dim10	a: dim20	b: dim20	a: dim30	b: dim30
N = 20	0.094988	0.104158	0.035971	0.029494	0.019112	0.014708
N = 40	0.084636	0.090979	0.025511	0.026347	0.010046	0.013243
N = 80	0.074046	0.077824	0.032574	0.036061	0.014315	0.0127
Function	Iterations = 1000		Iterations = 1500		Iterations = 2000	
Rastrigin	a: dim10	b: dim10	a: dim20	b: dim20	a: dim30	b: dim30
N = 20	5.860594	5.83382	24.185181	26.653132	52.57797	42.33567
N = 40	4.285989	3.790377	19.33343	18.562738	41.644389	31.9873
N = 80	2.75856	2.708036	14.700459	14.922603	32.466173	31.2516
Function	Iterations = 1000		Iterations = 1500		Iterations = 2000	
Rosenbrock	a: dim10	b: dim10	a: dim20	b: dim20	a: dim30	b: dim30
N = 20	133.14007	41.74886	229.08707	146.95593	637.79283	319.25095
N = 40	70.82829	37.15916	219.00407	134.52409	442.88521	221.69113
N = 80	48.60264	14.74576	143.47258	86.38419	207.87556	99.74067

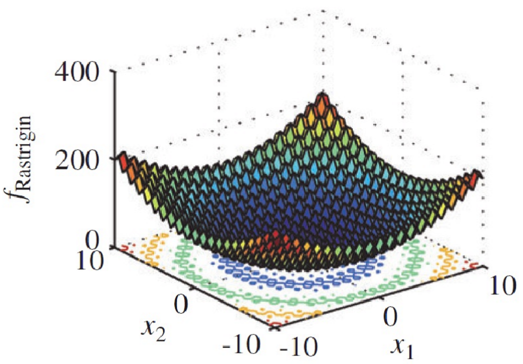


Fig. 5: Shapes of Rastrigin functions.

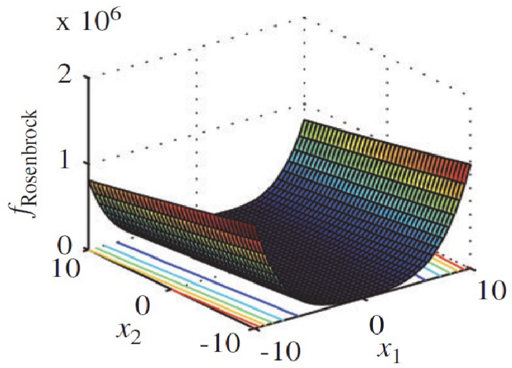


Fig. 6: Shapes of Rosenbrock functions.

For a minimization problem of each function, we consider investigate convergence time, sample convergence matrix of particle positions, and the best function values of the PSO algorithm with each parameter set in the stability region.

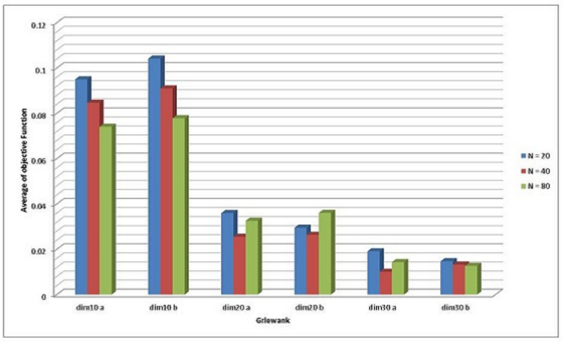
Each function of initial particle positions is randomly given from ($f_{Griewank}$: $\Re[300, 600]$), ($f_{Rastrigin}$: $\Re[2.56, 5.12]$), ($f_{Rosenbrock}$: $\Re[15, 30]$). Under these settings, we performed each algorithm 150 times for each test function and each parameter set $\alpha = 0.9, \beta = 0.33, \gamma = 0.33, \delta = 0.33$ in subsection 4.1-4.2.

4.1 Comparison of Convergence

We deal with 10, 20, and 30-dimensional problems as basic cases in experiment 1 and discuss the numerical results in detail. Let the number of particles be (20, 40, 80), and the number of iterations be (1000, 1500, 2000) in (10-dim, 20-dim, 30-dim) respectively.

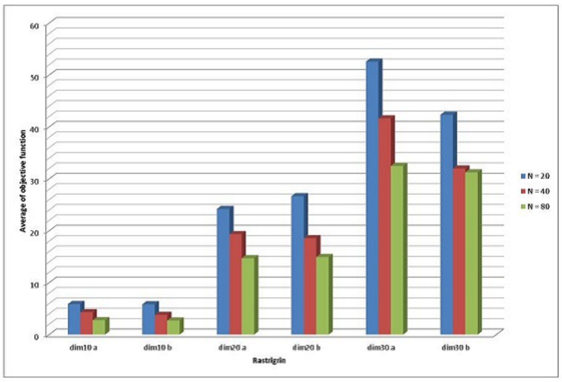
We define average of objective values of PSO and the hybrid method as the convergence performance. Figs. 7, 8, and 9 shows the best objective values of two approaches of each test functions. Overall features of figures mean that convergence performance of this work algorithm is better than PSO algorithm. As much as dimension is high, this feature is particularly remarkable. Also, for the test functions $f_{Rosenbrock}$ and $f_{Rastrigin}$, a difference of two approaches performance is the biggest . In contrast, for the case of the multimodel functions $f_{Griewank}$, both performances are too not different.

To sum up, the results suggest that the presented measures are still effective for the high-dimensional cases, and $f_{Rosenbrock}$ and $f_{Rastrigin}$ functions are easy to show an effect than $f_{Griewank}$ functions (see Table 1).



(a) (b) (a) (b) (a) (b)
10-dim. 20-dim. 30-dim.

Fig. 7: Individual convergence performance for two different algorithms. (Griewank) a: PSO, b: Proposal of hybrid PSO



(a) (b) (a) (b) (a) (b)
10-dim. 20-dim. 30-dim.

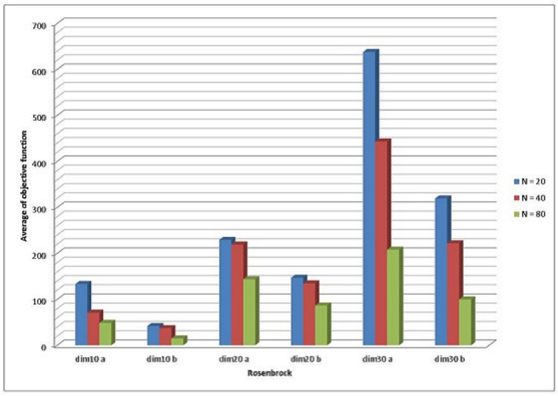
Fig. 8: Individual convergence performance for two different algorithms. (Rastrigin) a: PSO, b: Proposal of hybrid PSO

4.2 Comparison of Iteration

Next, we examine how much number of iterations each function converge at to evaluate two algorithm performance. We deal with 30-dimensional problems of three functions and let the number of particles be 80 as the case that an effect is remarkable, and the number of iterations be in ranges from 1 to 1600.

Figs. 8, 9, and 10 suggest that convergence rate of the hybrid method is better with few iteration than rate of PSO. It is particularly remarkable when the numbers of iteration be in ranges from 100 to 200 and using $f_{Rosenbrock}$ and $f_{Rastrigin}$. This result can lead observation that the hybrid algorithm is near the most suitable value even little number of iterations.

As the main results of this paper, we first show how to combine PSO and neural network mechanism by the theoretical analysis, and confirmed that proposed system can realize interpolation search based on global information of objective environment.



(a) (b) (a) (b) (a) (b)
10-dim. 20-dim. 30-dim.

Fig. 9: Individual convergence performance for two different algorithms. (Rosenbrock) a: PSO, b: Proposal of hybrid PSO

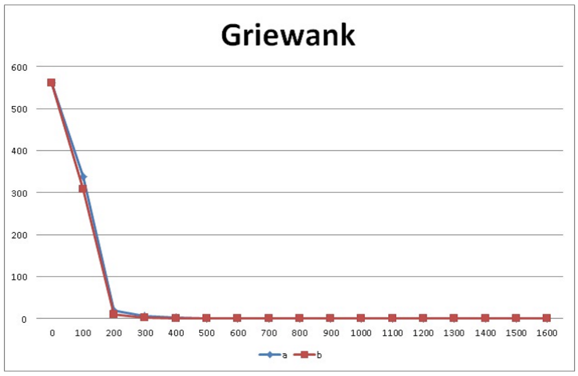


Fig. 10: Best fitness of function Griewank versus iteration for two approaches. a: PSO, b: Proposal of hybrid PSO

5 Conclusions

In this paper, we have proposed the hybrid dynamics with neural network and we have formulated equivalent dynamics to a neural network from the continuous time PSO algorithm. The proposed hybrid method and PSO algorithm were then tested some difficult continuous functions from the literature. Both algorithms were successful to converge with most steps over all runs.

Considering the computational expense, the proposed algorithm showed very competitive performance with high-dimensional cases and few steps in the literature while PSO algorithm also performs comparably. And also, $f_{Rosenbrock}$ and $f_{Rastrigin}$ functions are easy to show an effect. These observations lead us to conclude that the proposed hybrid dynamics Σ_n is effective at locating best-practice optimum solutions for variable function, in particular uni-modal function, for high-dimensional cases or number of few steps.

In the future, an action for performance enhancements is demanded through the consideration in wider benchmark problem and comparison with the simplex

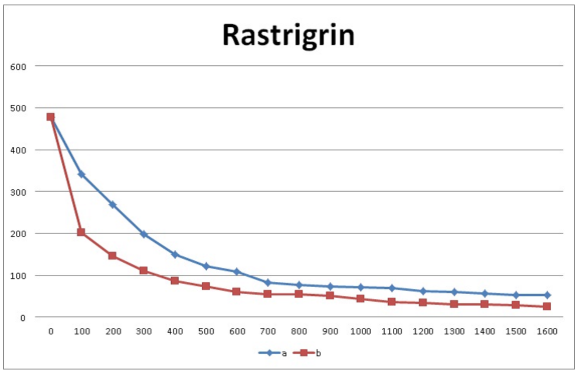


Fig. 11: Best fitness of function Rastrigin versus iteration for two approaches. a: PSO, b: Proposal of hybrid PSO

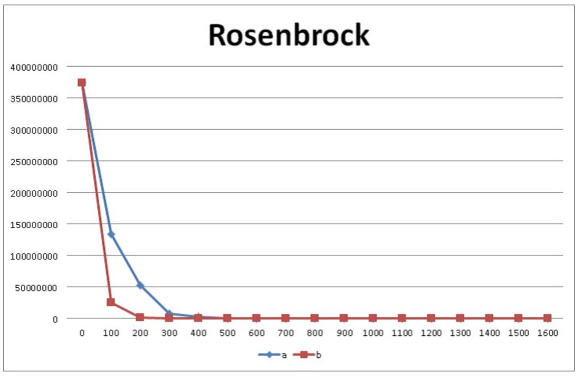


Fig. 12: Best fitness of function Rosenbrock versus iteration for two approaches. a: PSO, b: Proposal of hybrid PSO

method or DE (differential evolution). And more, since the behavior of the proposed algorithm depends on the function to be minimized, we cannot determine the best parameter set (β, γ, δ) uniquely in practical applications by using the proposed analysis methods. However, the proposed method may estimate a complexity global function by the estimate of the incline of neural network. And if the complexity of the function can be estimated in this way, a tuning method of the parameter sets may be developed by exploiting the proposed analysis method, which is one of the future research directions.

References

[1] Y.Shimizu: Proposal of Evolutionary Simplex Method for Global Optimization Problem, *Information Systems Society of Japan.*, vol. 24, No5, pp. 119-126 (2011).

[2] Shu-Kai S. Fan, Erwie Zahara: A hybrid simplex search and particle swarm optimization for unconstrained optimization, *European Journal of Operational Research.*, pp. 527-548 (2007).

[3] J. Kennedy, R.C. Eberhart: Particle swarm optimization, *IEEE Conf. On Neural Networks, IV, Piscataway, NJ.*, pp. 1942-1948 (1995).

[4] J. Kennedy, R.C. Eberhart, Y. Shi: Swarm intelligence, *Morgan Kaufimann Publishers, San Francisco, CA.*, pp. 1942-1948 (2001).

[5] V. Kadirkamanathan, K. Selvarajah, P. J. Fleming: Stability analysis of the particle dynamics in particle swarm optimizer, *IEEE Trans. Evolutionary Computation.*, pp. 245-255 (2006).

[6] M. Clerc, J. Kennedy: The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans. Evolutionary Computation*, pp. 58-73 (2002).

[7] M. Jiang, Y. P. Luo, S. Y. Yang: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, *Information Processing Letters*, vol. 102, No. 1, pp. 8-16 (2007).

[8] H. M. Emara and H.A. Abdel Fattah: Continuous swarm optimization technique with stability analysis, *Proceeding of the 2004 American Control Conference*, pp. 2811-2817 (2004).

[9] M. Kazuaki: A Study on Constrained Optimization with Nonlinear Dynamical Systems on Manifolds, *Keio University Science and Technology*, pp. 1-191 (2005).

[10] Trelea, I. C.: "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, pp. 317-325 (2003).