Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009

Network-Structured Particle Swarm Optimizer Considering Neighborhood Relationships

Haruna Matsushita and Yoshifumi Nishio

Abstract— **This study proposes a new Network-Structured Particle Swarm Optimizer considering neighborhood relationships (NS-PSO). All particles of NS-PSO are connected to adjacent particles by a neighborhood relation of the 2-dimensional network. The directly connected particles share the information of their own best position. Each particle is updated depending on the neighborhood distance on the network between it and a winner, whose function value is best among all particles. Simulation results show the searching efficiency of NS-PSO.**

I. Introduction

PARTICLE SWARM OPTIMIZATION (PSO) [1] is an evolutionary algorithm to simulate the movement of flocks of birds. Due to the simple concept, easy implementation, and quick convergence, PSO has attracted much attention and is used to wide applications in different fields in recent years. However, PSO greatly depends on its parameters and converges prematurely in case of solving complex problems which have local optima.

Furthermore, in the standard PSO algorithm, there are no special relationships between particles. Each particle position is updated according to its personal best position and the best particle position among the all particles. Then, which position is more important for each particle is determined at random in every generation. On the other side, in the real world, various personal relationships exist, such as the hierarchical relationship, the trust relationships, the parents-child relationship and so on.

Various topological neighborhoods have been considered by researches [2]–[5]. They have applied ring neighborhood, the von Neumann neighborhood, or some other topological neighborhoods. However, the parameters are increased by using fourth term for considering the neighboring best position when updating the velocity. Moreover, if there are no connections in respective particles, the computation costs are increased because we have to search each particle's neighborhood by using particle position or its cost with a neighborhood radius in every generation.

In this study, we propose a new PSO algorithm with topological neighborhoods; Network-Structured Particle Swarm Optimizer considering neighborhood relationships (NS-PSO). All particles of NS-PSO are connected to adjacent particles by a neighborhood relation, which dictates the topology of the 2-dimensional network. The connected particles, namely neighboring particles on the network, share the information of their own best position. In every generation, we find a winner particle, whose function value is the best among all particles, and each particle is updated depending on the neighborhood distance between it and the winner on the network.

Simulation results and comparisons with the standard PSO show that the proposed NS-PSO can effectively enhance the searching efficiency by measuring in terms of accuracy, robustness and parameter-dependence.

Haruna Matsushita and Yoshifumi Nishio are with the Department of Electrical and Electronic Engineering, Tokushima University, Tokushima, 770–8506, Japan (phone: +81–88–656–7470; fax: +81–88–656–7471; email: {haruna, nishio}@ee.tokushima-u.ac.jp).

II. Standard Particle Swarm Optimization (PSO)

PSO is an evolutionary algorithm to simulate the movement of flocks of birds. In the algorithm of PSO, multiple solutions called "particles" coexist. At each time step, the particle flies towards its own past best position and the best position among all particles. Each particle has two information; position and velocity. The position vector of each particle i and its velocity vector are represented by $\boldsymbol{X}_i = (x_{i1}, \cdots, x_{id}, \cdots, x_{iD})$ and $\boldsymbol{V}_i = (v_{i1}, \cdots, v_{id}, \cdots, v_{iD})$, respectively, where $(d = 1, 2, \cdots, D)$, $(i = 1, 2, \cdots, M)$ and $x_{id} \in [x_{\min}, x_{\max}]$.

(PSO1) (Initialization) Let a generation step $t = 0$. Randomly initialize the particle position $\boldsymbol{X}_i$ and its velocity $\boldsymbol{V}_i$ for each particle i, and initialize $\boldsymbol{P}_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ with a copy of $\boldsymbol{X}_i$. Evaluate the objective function $f(\boldsymbol{X}_i)$ for each particle i and find $\boldsymbol{P}_g$ with the best function value among the all particles.

(PSO2) Evaluate the fitness $f(\boldsymbol{X}_i)$. For each particle i, if $f(\boldsymbol{X}_i) < f(\boldsymbol{P}_i)$, the personal best position (called *pbest*) $\boldsymbol{P}_i = \boldsymbol{X}_i$. Let $\boldsymbol{P}_g$ represents the best position with the best fitness among all particles so far (called *gbest*). Update $\boldsymbol{P}_g$, if needed.

(PSO3) Update $\boldsymbol{V}_i$ and $\boldsymbol{X}_i$ of each particle i depending on its *pbest* and *gbest* according to

$$\begin{aligned} v_{id}(t+1) &= w v_{id}(t) + c_1 \mathrm{rand}(\cdot)\,(p_{id} - x_{id}(t)) \\ &\qquad + c_2 \mathrm{Rand}(\cdot)\,(p_g - x_{id}(t)), \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1), \end{aligned} \tag{1}$$

where w is the inertia weight determining how much of the previous velocity of the particle is preserved. c_1 and c_2 are two positive acceleration coefficients, generally $c_1 = c_2$. $\mathrm{rand}(\cdot)$ and $\mathrm{Rand}(\cdot)$ are two uniform random numbers samples from $U(0, 1)$.

(PSO4) Let $t = t + 1$ and go back to (PSO2).

978-1-4244-3553-1/09/$25.00 ©2009 IEEE

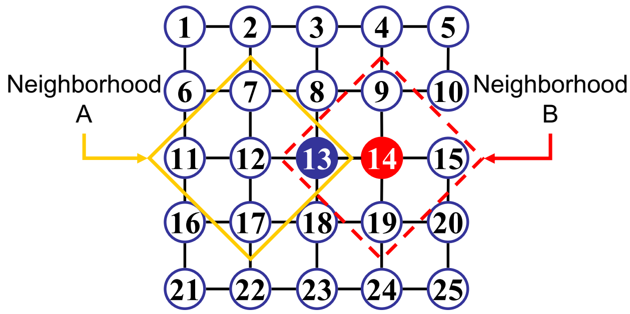


Fig. 1. Network-structure and neighborhood relationships. 1-neighborhood of the particle 12 is Neighborhood A containing the particle 7, 11, 12, 13 and 17. 1-neighborhood of the particle 14 is Neighborhood B containing the particle 9, 13, 14, 15 and 19. If $f(\boldsymbol{P}_{13})$ is the smallest value among Neighborhood A, then *lbest* of the particle 12 is $\boldsymbol{L}_{12} = \boldsymbol{P}_{13}$. If $f(\boldsymbol{P}_{14})$ is the smallest value among Neighborhood B, then *lbest* of the particle 14 is $\boldsymbol{L}_{14} = \boldsymbol{P}_{14}$.

III. Network-Structured Particle Swarm Optimizer Considering Neighborhood Relationships (NS-PSO)

The algorithm of NS-PSO is almost same as the standard PSO. The most important feature of NS-PSO is that all particles are organized on a rectangular 2-dimensional grid as Fig. 1. In other words, the particles are connected to adjacent particles by the neighborhood relation, which dictates the topology of the network. Furthermore, the particles share the local best position between the neighborhood particles directly connected. We should note the mean of *neighborhood*. In some research, the neighboring particles were defined as the particle with similar cost values or as the particles whose position are close to each other. However, in this study, the particles, which are directly connected on the network, are defined as *neighborhood* even if their positions or their costs are far.

(NS-PSO1) (Initialization) Let a generation step $t = 0$ and initialize particle informationton according to (PSO1). Define g as the winner c. Find $\boldsymbol{L}_i = (l_{i1}, l_{i2}, \cdots, l_{iD})$ with the best function value among the directly connected particles, namely own neighbors.

(NS-PSO2) Evaluate the fitness $f(\boldsymbol{X}_i)$ and find a winner particle c with the best fitness among the all particles at current time t;

$$c = \arg\min_i \{f(\boldsymbol{X}_i(t))\}. \tag{2}$$

For each particle i, if $f(\boldsymbol{X}_i) < f(\boldsymbol{P}_i)$, the personal best position *pbest* $\boldsymbol{P}_i = \boldsymbol{X}_i$.

If $f(\boldsymbol{X}_c) < f(\boldsymbol{P}_g)$, update *gbest* $\boldsymbol{P}_g = \boldsymbol{X}_c$, where $\boldsymbol{X}_c = (x_{c1}, x_{c2}, \cdots, x_{cD})$ is the position of the winner c.

(NS-PSO3) Find each local best position (called *lbest*) $\boldsymbol{L}_i$ among the particle i and its neighborhoods which are directly connected with i on the network as Fig. 1. For each particle i, if $f(\boldsymbol{P}_i) < f(\boldsymbol{L}_i)$, update *lbest* $\boldsymbol{L}_i = \boldsymbol{P}_i$.

(NS-PSO4) Update $\boldsymbol{V}_i$ and $\boldsymbol{X}_i$ of each particle i depending on its *lbest*, position of the winner $\boldsymbol{X}_c$ and the distance on the network between i and the winner c, according to

$$\begin{aligned} v_{id}(t+1) &= wv_{id}(t) + c_1 \text{rand}(\cdot)\left(l_{id} - x_{id}(t)\right) \\ &\quad + c_2 h_{c,i}\left(x_{cd} - x_{id}(t)\right), \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1), \end{aligned} \tag{3}$$

where $h_{c,i}$ is the fixed neighborhood function defined by

$$h_{c,i} = \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2}\right), \tag{4}$$

where $\|\boldsymbol{r}_i - \boldsymbol{r}_c\|$ is the distance between network nodes c and i on the network, the fixed parameter σ corresponds to the width of the neighborhood function. Therefore, the large σ strengthens particles' spreading force to the whole space, and the small σ strengthens their convergent force toward the winner.

(NS-PSO5) Let $t = t + 1$ and go back to (NS-PSO2).

IV. Experimental Results

In order to evaluate the performance of NS-PSO, we use eight benchmark optimization problems summarized in Table I. The optimum solution x^* of Rosenbrock's function f_2 is $[1, 1, \ldots, 1]$, and x^* of the other functions are all $[0, 0, \ldots, 0]$. The optimum function values $f(x^*)$ of all functions are 0. f_1, f_2, f_3 and f_4 are unimodal functions shown in Figs. 2(a)–(d), and f_5, f_6, f_7 and f_8 are multimodal functions shown in Figs. 2(e)–(h) with numerous local minima. Griewank's function with full definition range as Fig. 2(h) looks similar to Sphere function as Fig. 2(a) which is the unimodal. However, in the inner area as Fig. 2(i), there are a lot of small peaks and valleys. All the functions have D variables. In this study, D is set to 50 and 100 to investigate the performances in various dimensions.

The population size M is set to 36 in PSO, and the network size is 6×6 in NS-PSO. For PSO and NS-PSO, the parameters are set as $w = 0.7$ and $c_1 = c_2 = 1.6$. The neighborhood radius σ of NS-PSO for f_6 with $D = 100$ are 3, while they are all 1.5 for f_6 with $D = 50$ and the other functions with both dimensions. Because Ackley's function f_6 with $D = 100$ is difficult to optimize, the particles of NS-PSO need much information of the neighbors.

We carry out the simulations repeated 30 times for all the optimization functions with 3000 generations. The performances with the minimum and mean function values and the standard deviation over 30 independent runs on eight functions with 50 and 100 dimensions are listed in Tables II and III, respectively. We can see that NS-PSO can obtain the better mean values for all the test functions. In particular, NS-PSO greatly improves the performance from PSO on f_1, f_3, f_4 and f_8 with $D = 50$ and on f_1, f_3, f_4 and f_6 with $D = 100$. Furthermore, NS-PSO can achieve the better minimum values 7 times on both dimensions. From these results, we can say that NS-PSO algorithm can obtain the better results for the test functions with both $D = 50$ and 100.

Figure 3 shows the mean *gbest* value of every generation over 30 runs for eight test functions with 100 dimensions.

TABLE I
EIGHT TEST FUNCTIONS.

Function name	Test Function	Initialization Space
Sphere function;	$f_1(x) = \sum_{d=1}^{D-1} x_d^2,$	$x \in [-2.048, 2.047]^D$
Rosenbrock's function;	$f_2(x) = \sum_{d=1}^{D-1} \left(100\left(x_d^2 - x_{d+1}\right)^2 + (1-x_d)^2\right),$	$x \in [-2.048, 2.047]^D$
3rd De Jong's function;	$f_3(x) = \sum_{d=1}^{D} \lvert x_d \rvert,$	$x \in [-2.048, 2.047]^D$
4th De Jong's function;	$f_4(x) = \sum_{d=1}^{D} d x_d^4,$	$x \in [-1.28, 1.27]^D$
Rastrigin's function;	$f_5(x) = \sum_{d=1}^{D} \left(x_d^2 - 10\cos(2\pi x_d) + 10\right),$	$x \in [-5.12, 5.12]^D$
Ackley's function;	$f_6(x) = \sum_{d=1}^{D-1} \left(20 + e - 20e^{-0.2\sqrt{0.5(x_d^2+x_{d+1}^2)}} - e^{0.5(\cos(2\pi x_d)+\cos(2\pi x_{d+1}))}\right),$	$x \in [-30, 30]^D$
Stretched V sine wave function;	$f_7(x) = \sum_{d=1}^{D-1} (x_d^2 + x_{d+1}^2)^{0.25}\left(1 + \sin^2(50(x_d^2 + x_{d+1}^2)^{0.1})\right),$	$x \in [-10, 10]^D$
Griewank's function;	$f_8(x) = \sum_{d=1}^{D} \frac{x_d^2}{4000} - \prod_{d=1}^{D} \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1,$	$x \in [-600, 600]^D$

TABLE II
COMPARISON RESULTS OF PSO AND NS-PSO ON 8 TEST FUNCTIONS WITH $D = 50$.

f	Method	Mean	Minimum	Std
f_1	PSO	2.31e-18	1.25e-25	1.26e-17
	NS-PSO	**8.34e-26**	2.16e-30	2.74e-25
f_2	PSO	55.88	16.33	27.55
	NS-PSO	**41.60**	37.22	9.63
f_3	PSO	1.70e-06	1.15e-09	6.81e-06
	NS-PSO	**1.59e-07**	3.16e-12	7.46e-07
f_4	PSO	9.83e-36	1.34e-41	4.15e-35
	NS-PSO	**3.27e-42**	9.94e-48	1.13e-41
f_5	PSO	149.71	99.50	34.84
	NS-PSO	**87.69**	54.72	21.91
f_6	PSO	235.98	123.85	59.03
	NS-PSO	**180.79**	10.32	239.37
f_7	PSO	69.88	49.71	9.51
	NS-PSO	**37.08**	17.72	7.91
f_8	PSO	9.10e-02	0	0.2788
	NS-PSO	**2.61e-02**	0	4.11e-02

TABLE III
COMPARISON RESULTS OF PSO AND NS-PSO ON 8 TEST FUNCTIONS WITH $D = 100$.

f	Method	Mean	Minimum	Std
f_1	PSO	6.66e-04	9.02e-07	1.04e-03
	NS-PSO	**2.24e-05**	3.12e-08	7.95e-05
f_2	PSO	179.64	88.90	46.02
	NS-PSO	**103.01**	92.44	20.04
f_3	PSO	0.5293	3.91e-03	0.8531
	NS-PSO	**0.0610**	3.26e-03	0.4721
f_4	PSO	1.42e-07	7.26e-12	7.53e-07
	NS-PSO	**3.47e-10**	1.00e-14	1.65e-09
f_5	PSO	398.75	303.46	54.51
	NS-PSO	**211.30**	150.24	36.70
f_6	PSO	783.73	564.46	95.70
	NS-PSO	**360.09**	126.44	196.00
f_7	PSO	195.75	174.56	12.89
	NS-PSO	**141.39**	106.25	20.66
f_8	PSO	0.2660	2.56e-04	0.3684
	NS-PSO	**0.1516**	6.50e-06	0.2791

The convergence rate of NS-PSO is almost same or slower than the standard PSO. In the standard PSO, the particles move toward *gbest* or toward *pbest*, however, the direction, which more particles move toward, is decided at random on every generation. On the other hand, the neighborhood gaussian function is used in NS-PSO, then, the particles

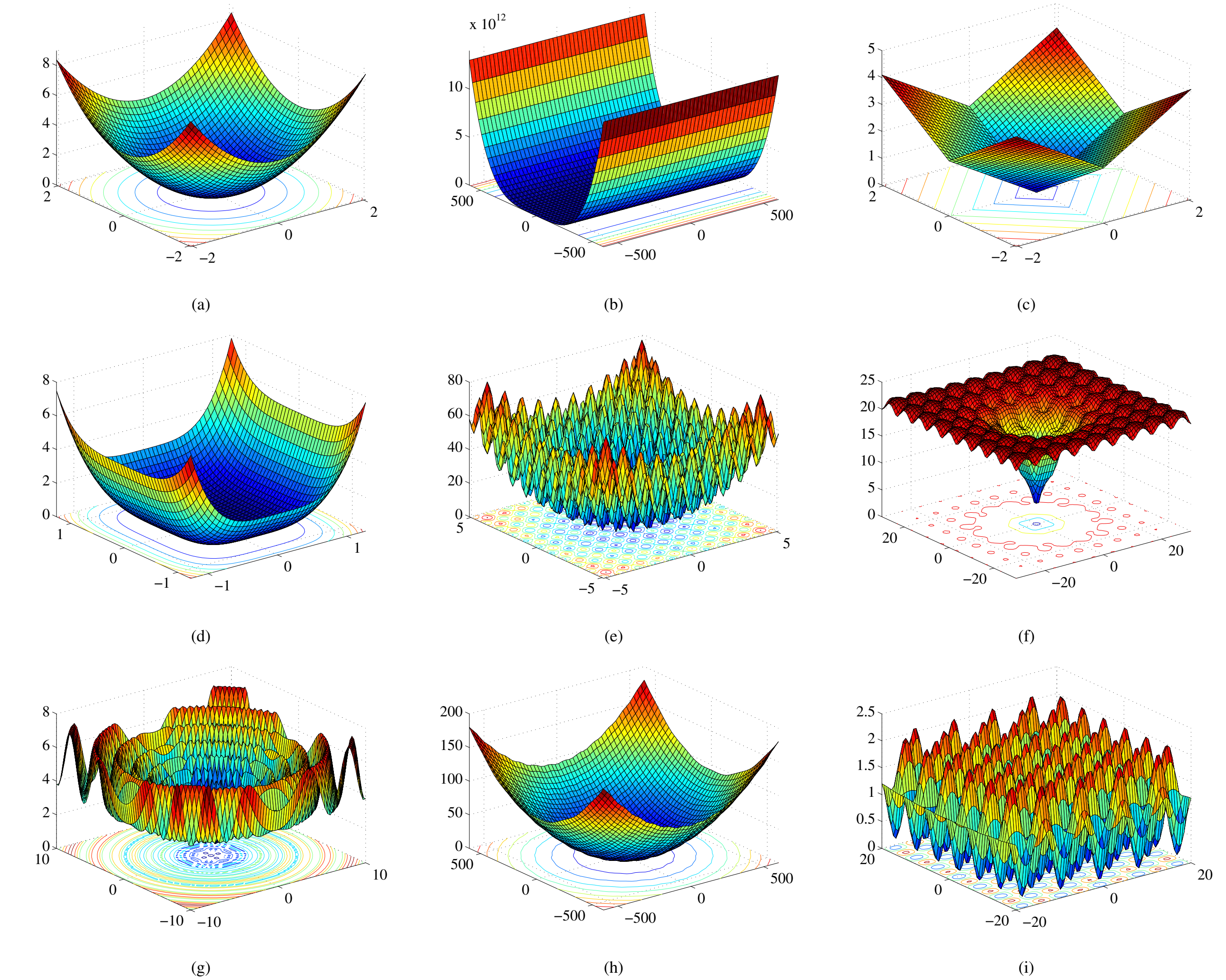


Fig. 2. Eight test functions with two variables. First and second variables are on the x-axis and y-axis, respectively, and z-axis shows its function value. (a) Sphere function. (b) Rosenbrock's function. (c) 3^{rd} De Jong's function. (d) 4^{th} De Jong's function. (e) Rastrigin's function. (f) Ackley's function. (g) Stretched V sine wave function. (h) Griewank's function with full definition range. (i) Inner area of Griewank's function from -20 to 20.

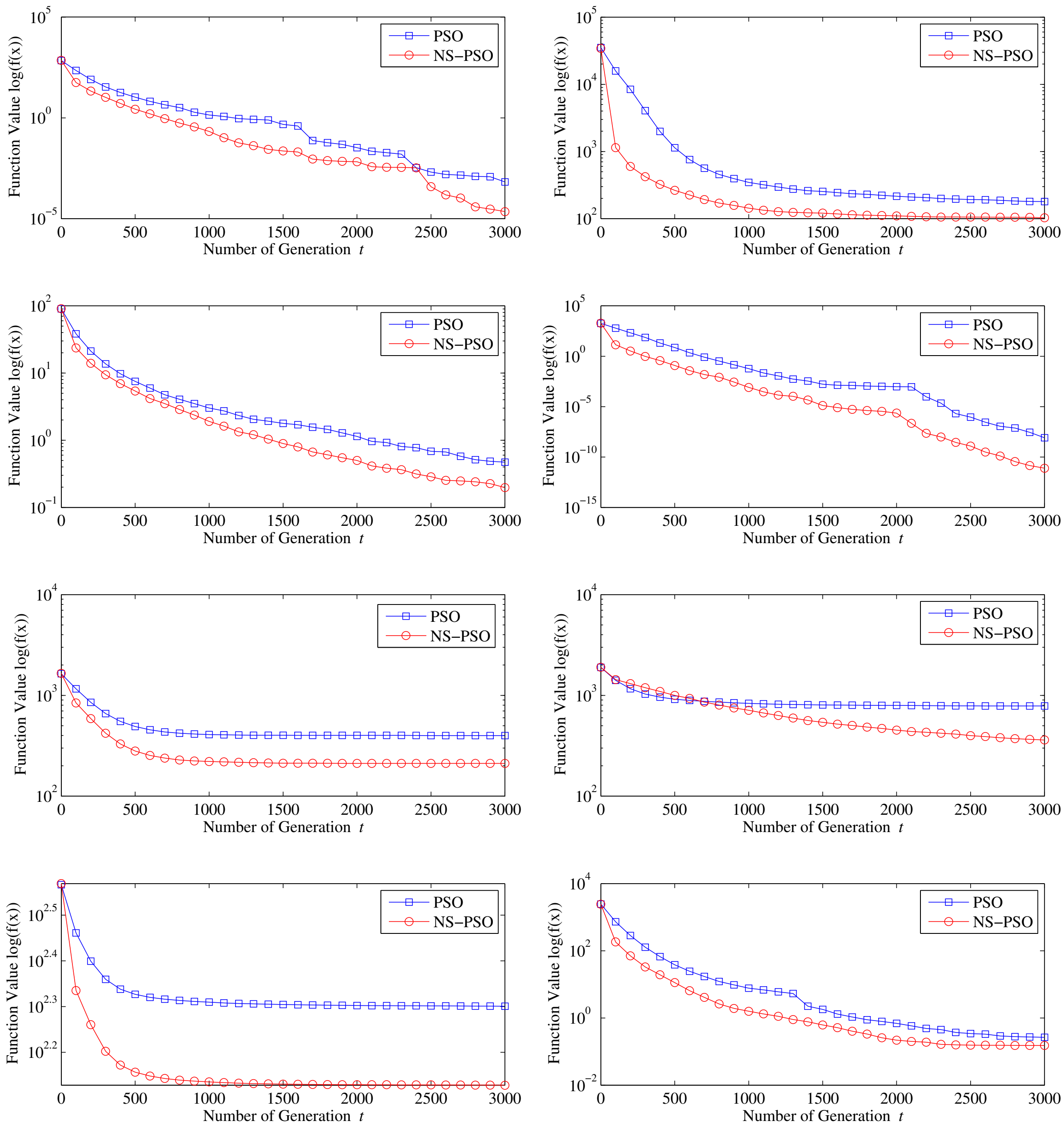
move according to the neighborhood distance between the winner and them. The winner's neighborhood particles move toward the winner, so that they spread to whole space. For the particles which are not 1-neighbors of the winner but are connected near the winner, the gravitation toward the winner is strong. The other particles fly toward their *lbest*. In other words, the roles of the NS-PSO particles are determined by the connection relationship, and they produce the diversity of the particles. These effects avert the premature convergence, and the particles of NS-PSO can easily escape from the local optima.

Next, we consider the robustness of both algorithms by measuring by the standard deviation of all simulations. From the standard deviations in Tables II and III, the standard PSO and NS-PSO obtain the smaller values 2 and 6 times on both dimensions, respectively. Therefore, NS-PSO appears to be more robust than the standard PSO. The particle's velocity and its position of NS-PSO is updated by according to the connection relationship although the standard PSO is updated at completely random. Therefore, NS-PSO can achieve the stable better performance unaffected by the initial states of the particles.

V. Parameter Dependence

Furthermore, we investigate the effect of the parameters; the inertia weight w and the acceleration coefficients c_1 and c_2, on performance quality and their sensitivity.

Figure 4 shows the mean fitness value of *gbest* with different w over 30 runs for all the test functions with 100 dimensions. The fixed parameters are same as above simulations. We can see that NS-PSO achieve better performances than the standard PSO even if the inertia weight w is varied. If we decrease or increase w by just 0.1, the performance of the standard PSO becomes drastically worse as especially



(g) (h)

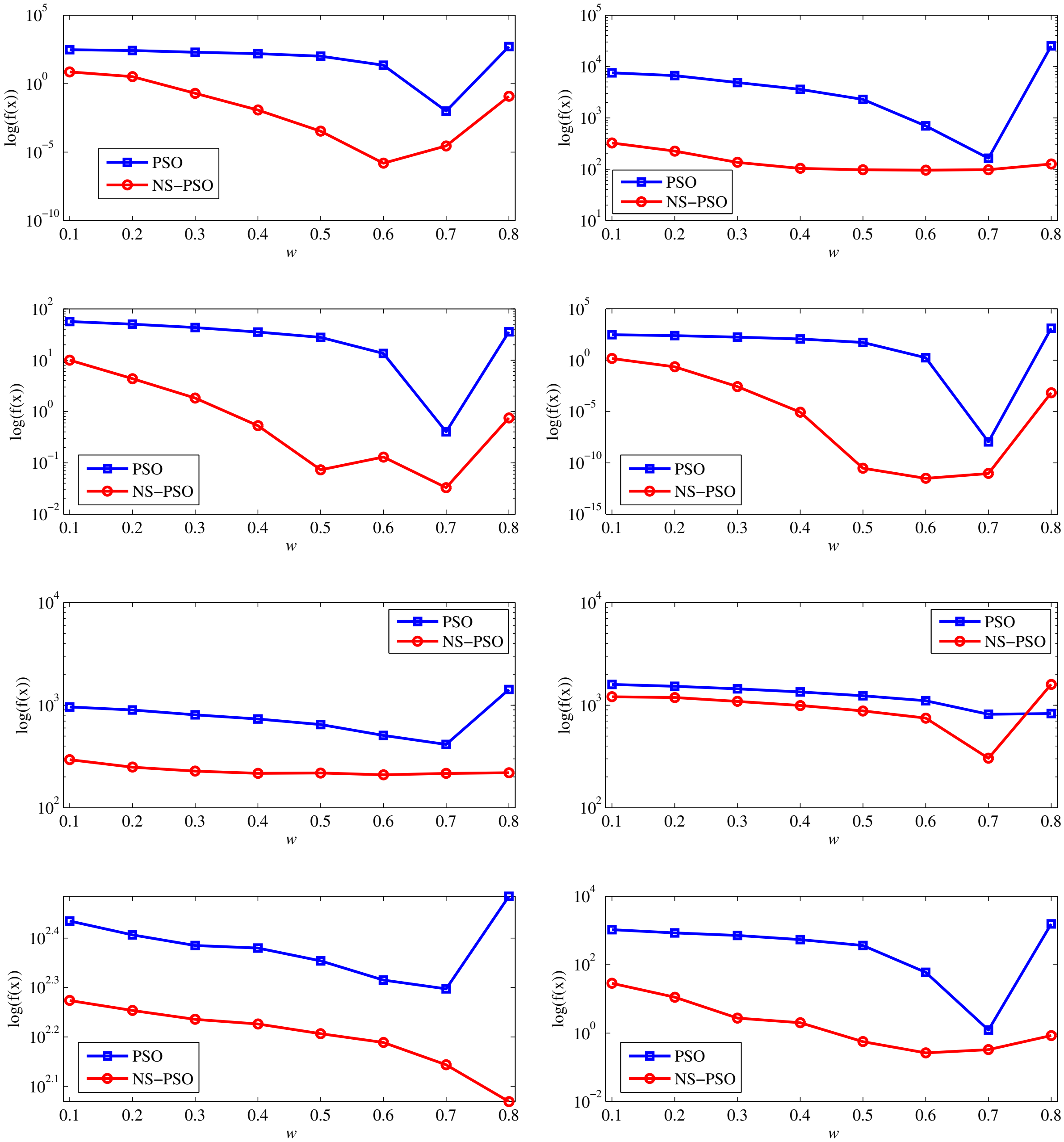
Fig. 3. Mean *gbest* value of every generation for 100-dimensional eight functions. (a) Sphere function. (b) Rosenbrock's function. (c) 3^{rd} De Jong's function. (d) 4^{th} De Jong's function. (e) Rastrigin's function. (f) Ackley's function. (g) Stretched V sine wave function. (h) Griewank's function.

Figs. 4(a)–(d), (g) and (h). Oh the other hand, NS-PSO is more effective and robust than the standard PSO if the inertia weight w is set as a small value ($w \in [0.1, 0.6]$) [6] as especially multimodal functions shown in Figs. 4(e)–(h).

Figure 5 shows the mean fitness value of *gbest* with different $c_1 (= c_2)$ over 30 runs for all the test functions with 100 dimensions. We should note that for f_3 and f_8, the minimum *gbest* value of the standard PSO among all the $c_1 (= c_2)$ is smaller than NS-PSO as Figs. 5(c) and (g). However, the standard PSO is extremely sensitive to small changes in $c_1 (= c_2)$ whereas the performances of NS-PSO are stable. For other functions, NS-PSO can obtain better results than PSO even if the acceleration coefficients $c_1 (= c_2)$ are varied. From these results, we can say that NS-PSO is more effective and its parametrical dependence is not stronger than the standard PSO.

VI. Conclusions

This study has proposed a new Particle Swarm Optimization (PSO) with topological neighborhoods; Network-Structured Particle Swarm Optimizer considering neighborhood relationships. All particles of NS-PSO are connected



(g) (h)

Fig. 4. Mean *gbest* value with different w for 100-dimensional eight functions. Fixed parameters; $c_1 = c_2 = 1.6$. $\sigma = 3$ for Ackley's function f_6 and 1.5 for other all functions. (a) Sphere function. (b) Rosenbrock's function. (c) $3^{\rm rd}$ De Jong's function. (d) $4^{\rm th}$ De Jong's function. (e) Rastrigin's function. (f) Ackley's function. (g) Stretched V sine wave function. (h) Griewank's function.

to adjacent particles by neighborhood relation of the 2-dimensional network. The directly connected particles share the information of their own best position. Each particle is updated depending on the neighborhood distance on the network between it and a winner, whose function value is the best among all particles. In the simulation results, we have confirmed the accuracy, robustness and parameter-dependence of PSO and NS-PSO, and it is clear that the overall performance of NS-PSO is much efficient and robust.

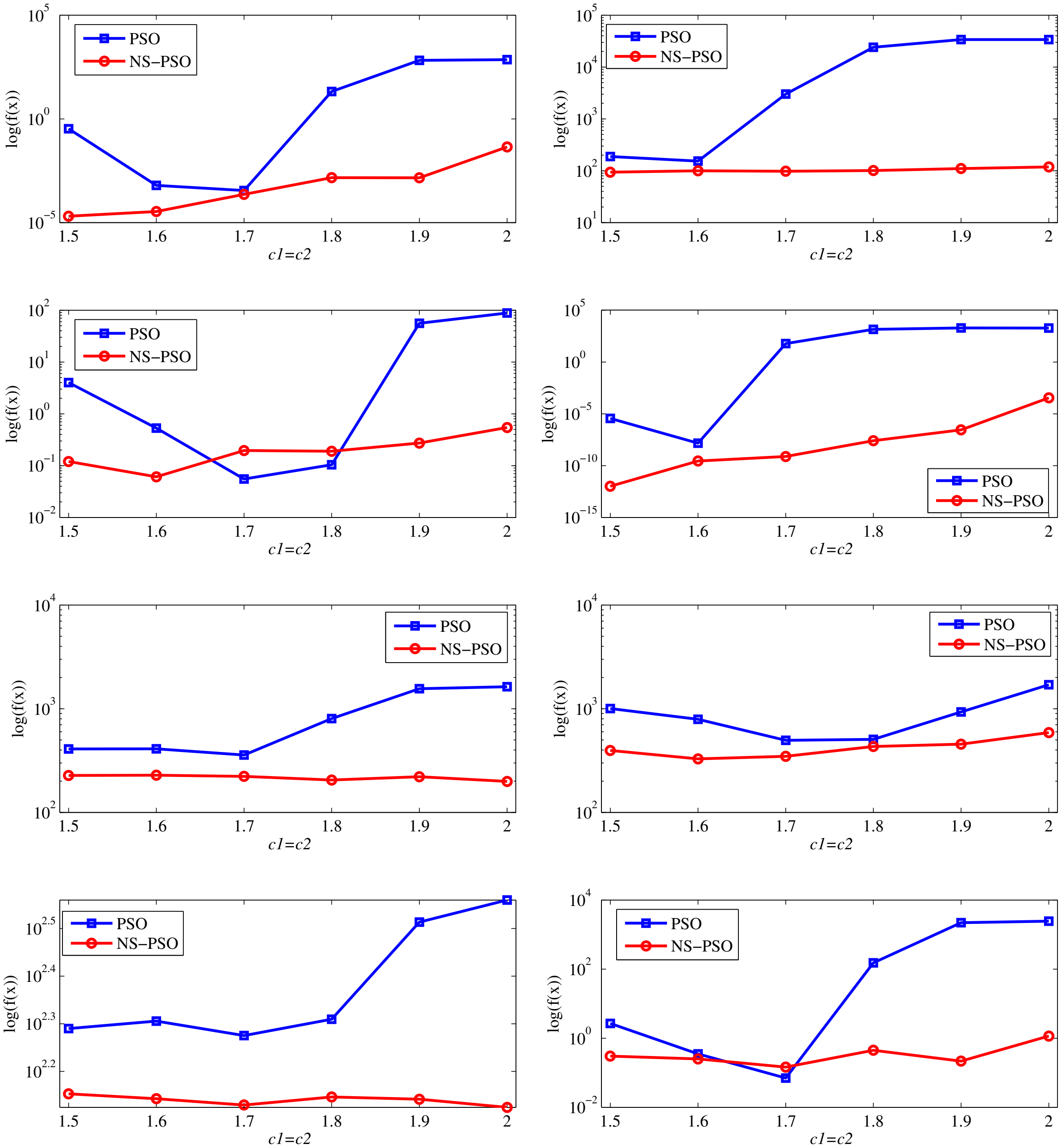
In the feature works, we should investigate the behaviors of NS-PSO with various network topologies and its relevance between network structure and performance in detail.

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE. Int. Conf. Neural Netw.*, pp. 1942–1948, 1995.

[2] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proc. of Cong. on Evolut. Comput.*, pp. 1931–1938, 1999.

[3] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. of Cong. on Evolut. Comput.*, pp. 1671–1676, 2002.



(g) (h)

Fig. 5. Mean *gbest* value with different $c_1 = c_2$ for 100-dimensional eight functions. Fixed parameters; $w = 0.7$. $\sigma = 3$ for Ackley's function f_6 and 1.5 for other all functions. (a) Sphere function. (b) Rosenbrock's function. (c) 3^{rd} De Jong's function. (d) 4^{th} De Jong's function. (e) Rastrigin's function. (f) Ackley's function. (g) Stretched V sine wave function. (h) Griewank's function.

[4] R. Mendes, J. Kennedy and J. Neves, "The Fully Informed Particle Swarm: Simpler, Maybe Better," in *IEEE Trans. Evolut. Comput.*, vol. 8, no.3, pp. 204–210, June 2004.

[5] S. B. Akat and V. Gazi, "Particle Swarm Optimization with Dynamic Neighborhood Topology: Three Neighborhood Strategies and Preliminary Results," in *Proc. of IEEE Swarm Intelligence Symposium*, 2008.

[6] X. Xie, W. Zang and Z. Yang, "A dissipative swarm optimization," in *Proc. of Cong. on Evolut. Comput.*, pp. 1456–1461, 2002.