Particle Swarm Optimization Driven by Evolving Elite Group

Ki-Baek Lee and Jong-Hwan Kim

***Abstract*— This paper proposes a novel hybrid algorithm of Particle Swarm Optimization (PSO) and Evolutionary Programming (EP), named Particle Swarm Optimization driven by Evolving Elite Group (PSO-EEG) algorithm. The hybrid algorithm combines the movement update property of canonical PSO with the evolutionary characteristics of EP. It is processed in two stages; elite group stage by EP and ordinary group stage by PSO. For the former group, a novel concept of Evolving Elite Group (EEG) is introduced, which consists of relatively superior particles in a population. The elite particles are evolved by mutation and selection scheme of EP. The other ordinary particles refer to the closest elite particle as well as the global best and the personal best, to update their location. Simulation results demonstrate the proposed PSO-EEG is highly competitive in terms of robustness, accuracy and convergence speed for five well-known complex test functions.**

I. Introduction

Evolutionary Algorithm (EA) and Particle Swarm Optimization (PSO) are both population based algorithms that have known to be successful in solving various complex optimization problems [1]-[4]. Both algorithms have strengths and weaknesses such that EAs are robust but require considerable computing power and slow in converging, while PSOs are relatively fast, simple and, in the same breath, easily converge to local solution.

Comparisons between EAs and PSOs were reported [5], [6], both of which concluded that a hybrid of the EA and PSO could yield further advances. Recently, researches on hybridizing EAs and PSOs have been studied. PSO-ES hybrid optimization was studied [7] and PSO embedding evolutionary programming (EP) was presented, which added EP perturbation to the velocity update process of PSO [8]. A PSO-GA hybrid optimization was studied and applied to recurrent network designs [9]. GA-PSO based hybrid algorithm was also provided by parallel processing both algorithms and mixing two populations periodically [10]. Another GA-PSO hybrid algorithm was presented by separating the population and carrying out optimization with each algorithm [11].

These hybrid algorithms show outstanding improvement in optimization problems, which are more robust and accurate than PSOs and also faster than classical EAs. These properties come from the facts that EA helps particles trapped in local minima jump out of there and PSO enhances the convergence property. However, the increase of evaluation time is unavoidable because EAs basically accompany lots of computation time for evaluation. Moreover, as PSOs use the global best position and the personal best position, jumps by EAs could be ignored unless those jumps surpass the global best position and the personal best positions.

In this paper, a hybrid algorithm of PSO and EP, named Particle Swarm Optimization driven by Evolving Elite Group (PSO-EEG), is proposed. The computational cost of evaluation of EA is far less than that of former hybrid algorithms by proposing a novel concept of Evolving Elite Group (EEG). The particles in the EEG are evolved by the mutation and selection schemes of EP, rather than all the particles in the population. Thus, the number of the evaluations is relatively small. EP is more convenient to solve numerical optimization problems than GA because GA requires encoding and decoding processes [12], [13]. The other ordinary particles refer to the nearest elite particle as well as the global best and the personal best, to update their velocity and position, similar to neighbor best concept. Following the analysis that, in our human society, people in elite group lead the other public and their decisions influence the behaviors of the public [14], [15], ordinary particles follow the elites and the elites evolve and compete with each other among themselves to survive in the elite group.

To demonstrate the effectiveness of PSO-EEG, computer simulations are carried out for five well-known complex test functions, some of which have critical local minima. Its performance is compared to that of the canonical PSO.

The remainder of this paper is organized as follows. In Section II, terminology and preliminaries are briefly reviewed. Section III describes PSO-EEG along with its advantages and procedure in detail. Section IV presents the simulation results on test functions including Sphere function, Griewangk's function, Rastrigin's function, Ackley's function, and Rosenbrock's valley function. The results are compared with those by canonical PSO. Finally, conclusions and further works follow in Section V.

II. Terminology and Preliminaries

A. Terminology

Population is a set of N particles, which have their own velocity, position and the fitness value. Velocity, $\mathbf{v}$, and position, $\mathbf{x}$, are defined as follows:

$$\mathbf{v} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^n$$

where n is the dimension of the space. Fitness function, f, as a function of the position, $\mathbf{x}$, is defined as follows:

$$f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}.$$

Evolving Elite Group (EEG) is a set of M elite particles. An elite particle is a particle whose fitness value is relatively larger than the other particles. In this paper, the elite particles

Ki-Baek Lee and Jong-Hwan Kim are with the Department of Electrical Engineering and Computer Science, KAIST, Republic of Korea. (phone: +82-42-350-8048; fax: +82-42-350-8877; email: {kblee, johkim}@rit.kaist.ac.kr).

978-1-4244-2959-2/09/$25.00 ©2009 IEEE

are selected by the q-tournament selection scheme as in Evolutionary Programming (EP). This q-tournament selection is one of the well known selection schemes, which selects the M individuals (here, particles) out of N elements by counting the number of wins of each particle in the matches with randomly chosen $q-1$ opponents.

Global best position is the one at which the fitness value is the largest all over the population including the past memories. Personal best position is the one at which the fitness value of the particle is the largest including the past memories. Nearest elite position to the particle is the one of the nearest particle of the current elite particles.

B. Preliminaries

1) Particle Swarm Optimization: In PSO, a population of particles is randomly generated initially. The update rules of the population in PSO at k-th step is described as follows:

$$\begin{cases} \mathbf{v}[k] = & w \cdot \mathbf{v}[k-1] \\ & +r_1 c_g(\mathbf{x}_{gBest} - \mathbf{x}[k-1]) \\ & +r_2 c_p(\mathbf{x}_{pBest} - \mathbf{x}[k-1]) \\ \mathbf{x}[k] = & \mathbf{x}[k-1] + \mathbf{v}[k] \end{cases} \tag{1}$$

where w, c_g, and c_p are the constants of canonical PSO, r_1 and r_2 are the random values between [0,1], $\mathbf{v}$ is the velocity of the particle, $\mathbf{x}$ is the position of the particle, $\mathbf{x}_{gBest}$ is the global best position of the particles, and $\mathbf{x}_{pBest}$ is the personal best position of the particle.

2) Evolutionary Programming: In EP, a population of N particles is randomly generated initially. Each particle is taken as a pair of real valued vectors, ($\mathbf{x}$, η), where $\mathbf{x}$ is the position and η is the standard deviation for Gaussian mutations. Each parent particle ($\mathbf{x}$, η) creates a single offspring ($\mathbf{x}'$, η') by the following equations:

$$\begin{cases} x'(i) = x(i) + \eta(i) N_i(0,1) \\ \\ \eta'(i) = \eta(i) exp(\tau' N(0,1) + \tau N_i(0,1)) \end{cases} \tag{2}$$

where $i = 1, ..., n$ and n is the dimension of the position of the particles, $x(i), x'(u), \eta(i)$ and $\eta'(i)$ denote the i-th component of the vectors $\mathbf{x}$, $\mathbf{x}'$, η and η', respectively. $N(0,1)$ denotes a normally distributed real random number with mean zero and standard deviation one. $N_i(0,1)$ indicates that the random number is newly generated for each value of i. The factor τ and τ' are commonly set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$. After then, by utilizing tournament selection, the N particles are selected out of $2N$ parents and offsprings [12].

III. Proposed PSO-EEG algorithm

A. Overall procedure

In the proposed PSO-EEG algorithm, a population of N particles is randomly generated initially and the fitness values of the particles are evaluated. After initialization, M elite particles are selected by the q-tournament selection method. And then, the selected elite particles are evolved using equation (2) and q-tournament selection of EP. By evaluating the fitness value of the particles, the global best position is determined. For each particle, the nearest elite particle is determined by the Euclidean distance. The velocity and the position of the particle are updated according to the global best position, the nearest elite position, and the personal best position. The update equations of the particles at k-th step is described as follows:

$$\begin{cases} \mathbf{v}[k] = & w \cdot \mathbf{v}[k-1] \\ & +r_1 c_g(\mathbf{x}_{gBest} - \mathbf{x}[k-1]) \\ & +r_2 c_n(\mathbf{x}_{nElite} - \mathbf{x}[k-1]) \\ & +r_3 c_p(\mathbf{x}_{pBest} - \mathbf{x}[k-1]) \\ \mathbf{x}[k] = & \mathbf{x}[k-1] + \mathbf{v}[k] \end{cases} \tag{3}$$

where w, c_g, c_n and c_p denote the weight factor, the constant of the global best, the constant of the nearest elite, and the constant of the personal best, respectively. r_1, r_2 and r_3 are the random values between [0,1]. $\mathbf{v}$ is the velocity of the particle and $\mathbf{x}$ is the position of the particle. $\mathbf{x}_{gBest}$, $\mathbf{x}_{nElite}$ and $\mathbf{x}_{pBest}$ denote the global best position, the nearest elite position, and the personal best position, respectively. For clarity, the flow of these operations is illustrated in Figure 1. The visualization of the elite and ordinary particles on 2-D plane is shown in Figure 2.

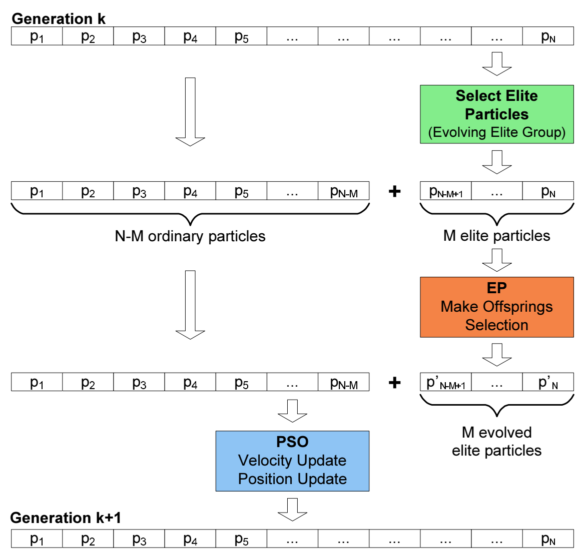


Fig. 1. Flow of the PSO-EEG Algorithm.

The pseudo-code of the proposed PSO-EEG algorithm is described as follows:

1) **initialize** the positions of N particles
   (evaluate the fitness values of the particles)
2) **repeat**
   a) select M elite particles
   b) evolve the elite particles by EP
   c) evaluate the fitness values of the particles
   d) determine the global best
   e) for each particle
      i) determine the the nearest elite particle
      ii) update velocity of the particle according to the global best position, the nearest elite position and the personal best position

TABLE I
THE FIVE BENCHMARK FUNCTIONS USED IN THE SIMULATION STUDY, WHERE n IS THE DIMENSION OF THE FUNCTION AND f_{min} IS THE MINIMUM OF THE FUNCTION VALUE.

Test function	Domain	f_{min}
$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$	0
$f_2(\mathbf{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$[-2.048, 2.048]^n$	0
$f_3(\mathbf{x}) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i))$	$[-5.12, 5.12]^n$	0
$f_4(\mathbf{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^n$	0
$f_5(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)} + 20 + e^1$	$[-32.768, 32, 768]^n$	0

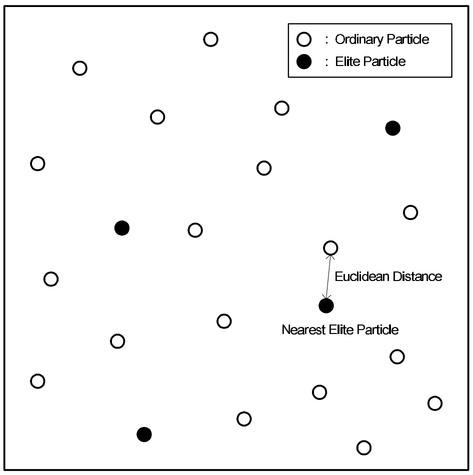


Fig. 2. Visualization of the elite and ordinary particles on 2-D plane.

iii) update position of the particle

3) **until** termination criterion is met.

B. Analysis and Discussion

The main contribution of this algorithm is that the calculation load is dramatically decreased while maintaining or even improving the exploitation and exploration ability by using Evolving Elite Group (EEG). Most of evaluations of EAs are needed to calculate the fitness values, to perform evolutionary operations, and to select next parents. The computational load is proportional to the number of particles or its square. In addition, the effects of genetic operations to the particles are not usually equal because as the dimension goes high well positioned particles are more likely to escape from the local minima by the generated jump by the genetic operations. In other words, applying genetic operations to the elite particles is more effective from the view point of avoiding local minima. As described earlier, only a few elites dominate the other public, and the choices made by the elites control the behaviors of the public in our human society [14], [15]. In the same manner, in the proposed algorithm, M particles are selected as elites for the EEG and EAs are applied to them to lead the rest of particles. Therefore, in the proposed algorithm, EAs are only used to the M particles in EEG, and the number of evaluations are reduced to less than 10 percent of full use of EAs. To propagate the evolutionary effect of the small elite group to the ordinary particles, in addition to the global best and the personal best, the nearest elite particle also influences the velocity update process.

IV. Simulation Results

A. Test functions

Five numerical minimization problems were chosen to compare the relative performance of the proposed PSO-EEG to canonical PSO. These functions are well known benchmark test functions and some of which have critical local minima (Table I).

Function f_1 is sphere function, which is very simple unimodal [16]. Function f_2 is Rosenbrock's valley function [17]. Rosenbrock's valley is a classic optimization problem, also known as Banana function. The global optimum is located inside as a long, narrow, parabolic shaped flat valley. f_3 is Rastrigin's function [18], [19], which is based on sphere function with the addition of cosine modulation to produce many local minima. Thus, the function is highly multimodal. f_4 is Griewangk's function [18]. Note that Griewangk's function is similar to Rastrigin's function and it has many widespread local minima. f_5 is Ackley's Path function [20], [21], which is a widely used multimodal test function.

B. Parameters

In the simulation, the number of particles, N was set to 100 for both of canonical PSO and PSO-EEG. In PSO-EEG, the number of elite particles, M was set to 10 and the selection scheme was 2-tournament. w, c_g and c_p in (1), the constants of update rules of the canonical PSO, were set to 0.7, 0.8, and 0.8, respectively. w, c_g, c_p and C_n in (3), the constants of update rules of PSO-EEG, were set to 0.7, 0.8, 0.8 and 0.4 respectively [22]. τ and τ' in (2), the factors of equation of the EP, were set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$, where n is the dimension of the position of the particles [23], [24], [25]. These parameters are summarized in Table II.

C. Results

In the simulation, comparison between the canonical PSO, PSO with only nearest elites (PSO+NE) without EP, and PSO-EEG on f_1-f_5 were performed to prove the effectiveness of the PSO-EEG in two ways; exploitation and exploration.

TABLE II

TABLE OF PARAMETERS FOR CANONICAL PSO AND PSO-EEG WHERE n IS THE DIMENSION OF THE POSITION OF THE PARTICLES.

Parameter	canonical PSO	PSO-EEG
N (Number of Particles)	100	100
M (Number of Elite Particles)	N/A	10
Selection type	N/A	tournament
Tournament size	N/A	2
w (constant for the inertia)	0.7	0.7
c_g (constant for the global best)	0.8	0.8
c_p (constant for the personal best)	0.8	0.8
c_n (constant for the nearest elite)	N/A	0.4
τ (factor of EP)	N/A	$(\sqrt{2\sqrt{n}})^{-1}$
τ' (factor of EP)	N/A	$(\sqrt{2n})^{-1}$

Figure 3 shows the results of each algorithm in 15 dimensions. Each figure shows the comparison of the best solution trajectories of each test function during 50 runs between the canonical PSO, PSO with only nearest elites(PSO+NE), and PSO-EEG on f_1-f_5. PSO with only nearest elites (PSO+NE) without EP searched faster than the canonical PSO for every test function. It means that in the PSO-EEG position update process, velocity increment by the nearest elite is helpful to increase searching speed. For every test function, PSO-EEG was not only able to find the most accurate solution, but also showed the fastest searching ability. This showed us that EEG, which was evolved by EP, generated the jumps to escape from the local minima successfully.

Table III shows the mean best (standard deviation), median best and the ratio of failures, which means the particles converge to local minima for each test function using uniformly random initialization. For every test function, while other algorithms failed to find the optimal solutions of test functions because of converging to local minima, PSO-EEG provided the dominant solution accuracy with only few failures.

V. CONCLUSIONS

In this paper, a hybrid algorithm, Particle Swarm Optimization Driven by Evolving Elite Group (PSO-EEG), was proposed using the movement update property of canonical Particle Swarm Optimization (PSO) and the evolutionary characteristics of EP. Following the update rules of particle movement in PSO, ordinary particles were driven by evolving elite ones as well as the global best and the personal best. The elite ones go through the evolutionary process by EP (Evolutionary Programming). The evaluation cost was not heavy because only the particles in the elite group were evolved by EP, 10 particles out of 100 in this paper. Simulation results demonstrated that the PSO-EEG could find the global or near global optimal solutions significantly faster than the canonical PSO and also reduce the probability of converging to local minima relatively lower than the canonical PSO. These are due to the EEG efficiently dispersing the particles by mutation operation and driving the particles fast and efficiently to the nearest elite.

An additional advantage of the PSO-EEG is that the partial algorithms, such as PSO and EP, can be customized, varied or extended according to the problems to be solved in order to allow quite large latitude of future researches. Future research includes an investigation into different evolutionary operations and selection schemes and their effects on performance. Moreover, application to the constrained optimization problems and the multi-objective problems should be studied.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *in Proc. IEEE Int. Conf. of Neural Networks*, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[2] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through simulated evolution," 1966.

[3] K.-B. Lee and J.-H. Kim, "Mass-spring-damper motion dynamics-based particle swarm optimization," in *in Proc. IEEE Congress of Evolutionary Computation*, vol. 1, Hong Kong, June 2008, pp. 2348–2353.

[4] H. Park and J.-H. Kim, "Potential and dynamics-based particle swarm optimization," in *in Proc. IEEE Congress of Evolutionary Computation*, vol. 1, Hong Kong, June 2008, pp. 2353–2359.

[5] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences," in *Evolutionary Programming VII*, ser. Lecture Notes in Computer Science, vol. 1447. Springer, 1998, pp. 601–610.

[6] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Evolutionary Programming VII*, ser. Lecture Notes in Computer Science, vol. 1447. Springer, 1998, pp. 611–616.

[7] Y. Shi and R. Krohling, "Co-evolutionary particle swarm optimization to solve min-max problems," in *in Proc. IEEE Congress of Evolutionary Computation*, vol. 2, 2002, pp. 1682–1687.

[8] N. Sinha and B. Purkayastha, "Pso embedded evolutionary programming technique for non-convex economic load dispatch," in *Power Systems Conference and Exposition, 2004. IEEE PES*, vol. 1, October 2004, pp. 66–71.

[9] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, vol. 34, pp. 997–1006, April 2004.

[10] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved ga and a novel pso-ga-based hybrid algoithm," *Information Processing Letters*, vol. 93, pp. 255–261, March 2005.

[11] M. Settles and T. Soule, "Breeding swarms: a ga/pso hybrid," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2005, pp. 161–168.

[12] D. B. Fogel, *System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling*. Needham Height MA: Ginn, 1991.

[13] J.-H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 119–128, May 1997.

[14] G. Mosca, *The Ruling Class*. McGraw-Hill, 1939.

[15] V. Pareto, A. S. Schwier, and A. N. Page, *Manual of Political Economy*. Kelley (Augustus M.) Publishers, 1971.

[16] K. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, 1975.

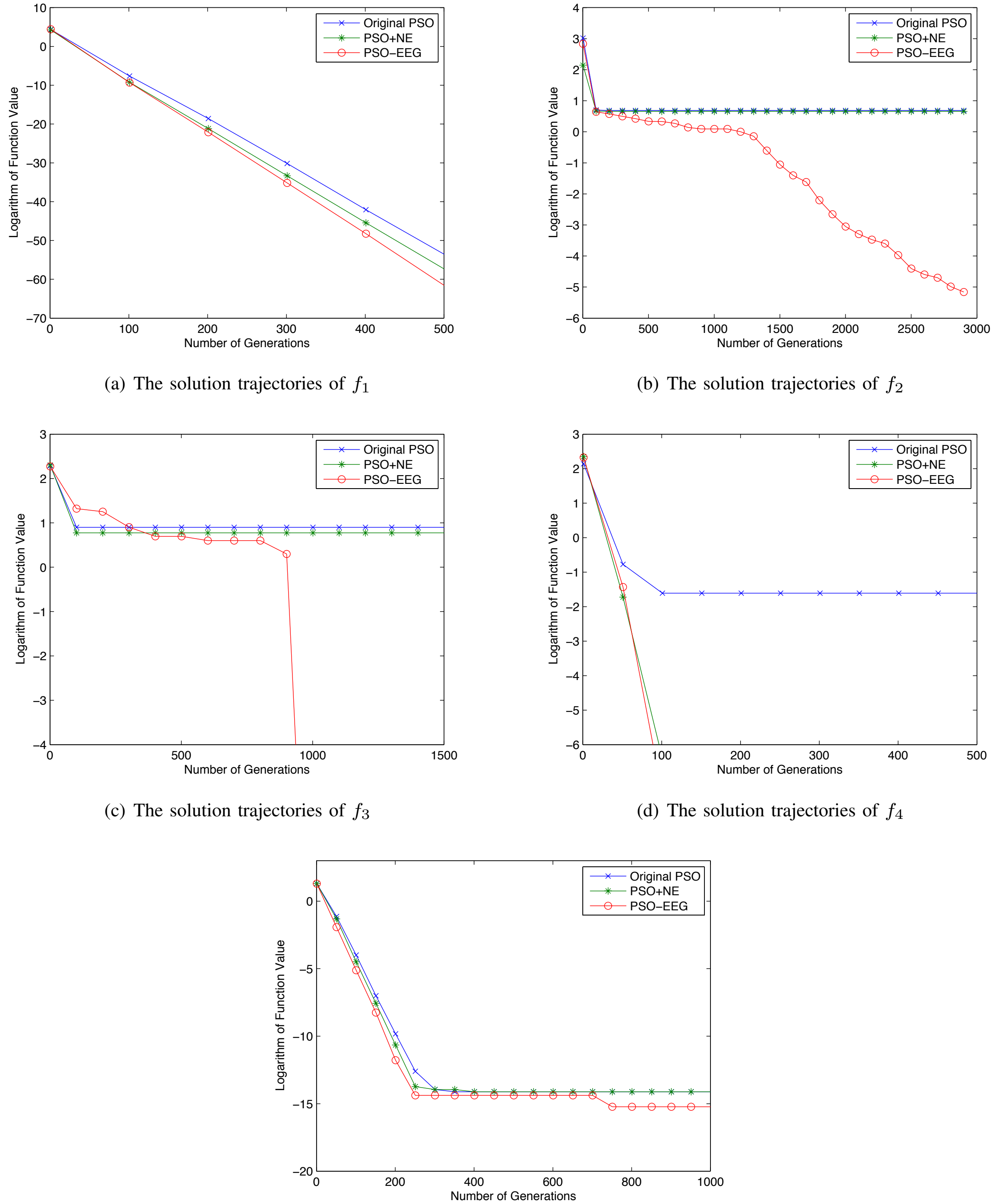
[17] H. Rosenbrock, "An automatic method for finding the greatest or the least value of a function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960.

[18] A. Törn and A. Zilinskas, "Global Optimization," *Lecture Notes in Computer Science*, vol. 350, 1989.

[19] H. Mühlenbein, D. Schomisch, and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," *Parallel Computing*, vol. 17, no. 6-7, pp. 619–632, 1991.

[20] D. H. Ackley, *A connectionist machine for genetic hillclimbing*. Boston: Kluwer, 1987.

[21] T. Bäck, *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.



(a) The solution trajectories of f_1

(b) The solution trajectories of f_2

(c) The solution trajectories of f_3

(d) The solution trajectories of f_4

(e) The solution trajectories of f_5

Fig. 3. Comparison of the best solution trajectories of 50 runs in 15 dimensions between the canonical PSO, PSO with only nearest elites (PSO+NE) without EP, and PSO-EEG on f_1-f_5.

TABLE III

COMPARISON BETWEEN THE CANONICAL PSO, PSO WITH ONLY NEAREST ELITES (PSO+NE) WITHOUT EP, AND PSO-EEG ON FUNCTIONS f_1-f_5. THE RESULTS ARE AVERAGED 50 RUNS, WHERE "MEAN BEST" AND "MEDIAN BEST" INDICATES THE MEAN AND MEDIAN BEST FUNCTION VALUES FOUND IN THE LAST GENERATION, "STD DEV" STANDS FOR THE STANDARD DEVIATION AND "ROF" IS THE RATIO OF FAILURES WHICH MEANS THAT THE PARTICLES CONVERGE TO LOCAL MINIMA.

			PSO			PSO+NE			PSO-EEG		
Problems	Dimensions	Number of Generations	Mean Best (Std Dev)	Median Best	ROF	Mean Best (Std Dev)	Median Best	ROF	Mean Best (Std Dev)	Median Best	ROF
f_1	5	200	9.58E-28	6.10E-28	0/50	9.34E-27	6.65E-27	0/50	4.69E-38	8.28E-41	0/50
			(9.76E-28)			(8.78E-27)			(2.39E-37)		
	10	400	1.19E-50	1.09E-50	0/50	4.71E-50	4.17E-50	0/50	1.18E-55	1.81E-56	0/50
			(1.41E-50)			(3.78E-50)			(2.36E-55)		
	15	600	1.11E-62	4.03E-65	0/50	4.29E-70	2.25E-70	0/50	2.82E-74	1.55E-74	0/50
			(2.42E-63)			(5.31E-70)			(3.59E-74)		
f_2	5	500	3.64E-01	2.43E-07	4/50	5.98E-01	2.47E-08	6/50	3.05E-11	1.72E-12	0/50
			(1.07E+00)			(1.33E+00)			(9.33E-11)		
	10	1000	6.02E+00	5.85E+00	50/50	5.87E+00	5.70E+00	49/50	3.25E-01	1.40E-03	4/50
			(1.40E+00)			(1.76E+00)			(1.09E+00)		
	15	1500	1.28E+01	1.16E+01	50/50	1.17E+01	1.15E+01	50/50	1.91E-01	1.25E-01	0/50
			(8.33E+00)			(1.66E+00)			(2.37E-01)		
f_3	5	500	1.19E+00	9.95E-01	41/50	2.25E+00	1.99E+00	48/50	0.00E+00	0.00E+00	0/50
			(8.04E-01)			(1.53E+00)			(0.00E+00)		
	10	1000	1.16E+01	1.09E+01	50/50	1.23E+01	1.19E+01	49/50	1.14E-05	0.00E+00	0/50
			(5.28E+00)			(5.48E+00)			(7.05E-05)		
	15	1500	2.44E+01	2.44E+01	50/50	2.52E+01	2.49E+01	50/50	6.02E-03	3.41E-07	0/50
			(7.29E+00)			(8.59E+00)			(1.90E-02)		
f_4	5	500	4.17E-02	3.45E-02	49/50	5.97E-02	5.42E-02	50/50	2.16E-02	1.11E-02	31/50
			(2.34E-02)			(3.49E-02)			(2.45E-02)		
	10	1000	1.12E-01	1.05E-01	50/50	1.05E-01	8.49E-02	50/50	3.92E-02	4.30E-02	36/50
			(5.68E-02)			(6.23E-02)			(3.31E-02)		
	15	1500	4.73E-02	3.45E-02	49/50	4.46E-02	3.94E-02	47/50	7.52E-03	0.00E+00	8/50
			(4.13E-02)			(3.25E-02)			(2.26E-02)		
f_5	5	500	9.44E-16	5.89E-16	0/50	5.89E-16	5.89E-16	0/50	5.89E-16	5.89E-16	0/50
			(1.08E-15)			(1.99E-31)			(1.99E-31)		
	10	1000	3.72E-15	4.14E-15	0/50	3.43E-15	4.14E-15	0/50	9.44E-16	5.89E-16	0/50
			(1.17E-15)			(1.44E-15)			(1.08E-15)		
	15	1500	7.88E-01	9.31E-01	27/50	2.66E-01	7.69E-15	10/50	3.36E-15	4.14E-15	0/50
			(7.99E-01)			(5.73E-01)			(1.49E-15)		

[22] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 58–73, Feb 2002.

[23] D. B. Fogel, "An introduction to simulated evolutionary optimization," *Neural Networks, IEEE Transactions on*, vol. 5, pp. 3–14, Jan 1994.

[24] T. Back and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.

[25] X. Yao, "Evolutionary programming made faster," *Evolutionary Computation, IEEE Transactions on*, vol. 3, pp. 82–101, July 1999.