Evolutionary Particle Swarm Optimization – A Meta-Optimization Method with GA for Estimating Optimal PSO Models

Hong Zhang[1] and Masumi Ishikawa[2]

[1] Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan. zhang@brain.kyutech.ac.jp
[2] Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan. ishikawa@brain.kyutech.ac.jp

1 Introduction

Particle Swarm Optimization (PSO) is an algorithm for swarm intelligence based on stochastic and population-based adaptive optimization inspired by social behavior of bird flocks and fish swarms [5, 10].

In recent years, many variants of PSO have been proposed [11] and successfully applied in many research and application disciplines, because of its intuitive understandability, ease of implementation, and the ability to efficiently solve large-scale and highly nonlinear optimization problems prevalent in social sciences, computer science and complex engineering systems [9, 14, 15].

Although the original PSO is very simple with only a few parameters to adjust, it provides better performance in computing speed, computing accuracy, and memory size compared with other methods such as machine learning, neural network learning and genetic computation. Needless to say, each parameter in PSO greatly affects the performance of PSO. However, how to determine appropriate values of parameters in PSO is yet to be found.

How to determine appropriate values of parameters in PSO can be regarded as meta-level optimization. Hence many researchers have paid much attention to this challenging problem. One approach to their determination is to try various values of parameters randomly to find a proper parameter set for PSO which handles many kinds of optimization problems reasonably well [2, 3]. Since this is an exhaustive way, the computing cost is heavy in high-dimensional parameter space.

The other approach is to implement a Composite PSO (CPSO) [13], in which the Differential Evolution (DE) algorithm [16] handles the PSO heuristics in parallel during optimization. The instantaneous fitness of the global optimal particle is used in CPSO for evaluating the performance of PSO. Its experimental results indicated that CPSO could surpass the success ratio [1] of the original PSO for some benchmark test problems. As an expansion of the CPSO, Optimazed Particle Swarm Optimization (OPSO) has been proposed and was applied to training of artificial neural networks [12]

On the other hand, for specifically determining the values of parameters in PSO, Eberhart et al. proposed to use an inertia weight using a constriction factor [6]. It concludes that the best approach is to use the constriction factor while limiting the maximum velocity, v_{max}, of each particle within the dynamic range of the variable, x_{max}, in each dimension. However, the selection of the inertia parameter and maximum velocity are problem dependent [18].

In order to systematically determine the values of parameters in PSO, this paper proposes a meta-optimization method called Evolutionary Particle Swarm Optimization (EPSO). Specifically, EPSO estimates appropriate values of parameter in PSO by a Real-coded Genetic Algorithm with Elitism strategy (RGA/E). This is a new meta-optimization method which is different from CPSO and OPSO. Firstly, the values of parameters in PSO are directly estimated by RGA/E in EPSO. The elitism strategy is performed to improve the performance and convergence behavior of EPSO. Secondly, a temporally cumulative fitness of the best particle is used in EPSO for effectively evaluating the performance of PSO. This is a key idea of this paper for successfully accomplishing meta-optimization.

As is well known, PSO is a stochastic adaptive optimization algorithm. Sometimes particle swarm searches well by chance, even if parameter values are not appropriate for solving a given optimization problem. Since the temporally cumulative fitness proposed is reflects the sum of instantaneous fitness, its variance is inversely proportional to the time span. Therefore, applying the fitness to EPSO is expected to systematically determine appropiate values of parameters in PSO by RGA/E for various optimization problems reasonably well, and greatly increase the success ratio of the original PSO for efficiently finding the global optimal solution corresponding to a given optimization problem.

To demonstrate the effectiveness of the proposed EPSO method, computer experiments on a 2-dimensional optimization problem are carried out. We show experimental results, comfirm the characteristics of dependency on initial conditions, and analyze the resulting PSO models.

The rest of the paper is organized as follow. Section 2 briefly describes the original PSO and RGA/E. Section 3 presents the proposed EPSO method and a key idea about the temporally cumulative fitness that we used in the method. Section 4 discusses the results of computer experiments applied to a 2-dimensional optimization problem. Section 5 gives conclusions.

[1] Success refers to cases where particles reach globally optimal solution. Success ratio is defined as the relative frequency of success.

2 The original PSO and RGA/E

2.1 The original PSO

The original PSO is modeled by particles with position and velocity in a high dimensional space. Particles move in hyperspace and remember the best position that they have found. In general, members of a swarm communicate the information of good positions with each other and adjust their own position and velocity based on (1) a global best position that is known to all, and (2) local best positions that are known by neighboring particles.

Let $\boldsymbol{x}_k^i$ and $\boldsymbol{v}_k^i$ denote the position and velocity of the ith particle at time k, respectively. The position and velocity of ith particle is updated by

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \boldsymbol{v}_{k+1}^i \tag{1a}$$
$$\boldsymbol{v}_{k+1}^i = c_0 \boldsymbol{v}_k^i + c_1 r_1 (\boldsymbol{x}_l^i - \boldsymbol{x}_k^i) + c_2 r_2 (\boldsymbol{x}_g - \boldsymbol{x}_k^i), \tag{1b}$$

where c_0 is an inertial factor, c_1 is an individual confidence factor and c_2 is a swarm confidence factor, and $r_1, r_2 \in U[0,1]$ [2] are random values. $\boldsymbol{x}_l^i (= arg \max_{k=1,2,\cdots} \{g(x_k^i)\}$, where $g(x_k^i)$ is the fitness value of the ith particle at time k.) is the local best position corresponding to the ith particle up to now, $\boldsymbol{x}_g (= arg \max_{i=1,2,\cdots} \{g(\boldsymbol{x}_l^i)\})$ is the global best position of particles of the entire swarm, respectively. Note that a constant, v_{max}, is used to arbitrarily limit the velocity of each particle and to improve the resolution of the search.

The procedure of the original PSO is described as follows.

step 1 Initialize the position, $\boldsymbol{x}_0^i$, and the velocity, $\boldsymbol{v}_0^i$, of each particle randomly. Their range is domain specific. Set counter $k=1$ and the maximum number, K, of iterations in search.

step 2 Calculate fitness value, $g(\boldsymbol{x}_k^i)$, of each particle, $\boldsymbol{x}_k^i$, for determining the local best position, $\boldsymbol{x}_l^i$, and the global best position, $\boldsymbol{x}_g$, up to the current time.

step 3 Update $\boldsymbol{v}_{k+1}^i$ and $\boldsymbol{x}_{k+1}^i$ by Eq.(1) for obtaining new position and velocity of each particle. It is to be noted that $\boldsymbol{v}_{k+1}^i$ has a upper bound of velocity, v_{max}, then $v_{k+1}^i = v_{max}$. If the current fitness value of $\boldsymbol{x}_l^i$ is larger than the previous one, replace $\boldsymbol{x}_l^i$ with the current one. If the current fitness value of $\boldsymbol{x}_g$ is larger than the previous one, replace $\boldsymbol{x}_g$ with the current one. If $k<K$ then $k=k+1$ and go to **step 2**, else stop the search.

The each parameter, i.e., c_0, c_1, and c_2, in the original PSO has a role in exploration, respectively. How to determine them is to greatly affect the performance of PSO. So far, these values usually are determined by a designer based on a large number of experimental results. The original PSO uses a set of parameter values, (c_0, c_1, c_2)=(1, 2, 2), for various optimization problems [10].

[2] $U[0,1]$ denotes that the probability distribution is a continuous uniform one. The support of the interval is defined by the two parameters, 0 and 1, which are its minimum and maximum values.

2.2 RGA/E

A real-coded genetic algorithm with elitism strategy, RGA/E, is applied to simulate the survival of the fittest among individuals over consecutive generation for solving real-valued optimization problems. The fitness function used in it, which is always problem dependent, is defined over the genetic representation and measures the quality of the represented solution. Fig. 1 indicates the flowchart of RGA/E.

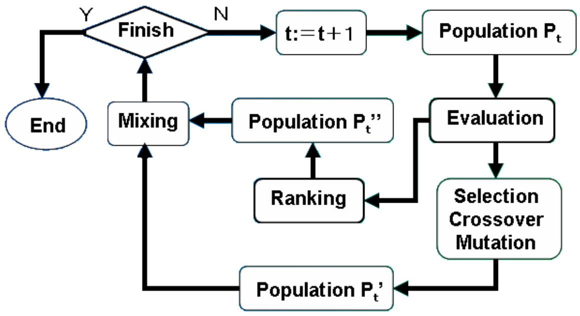

Fig. 1. Flowchart of RGA/E

For convenience, the individual, $\boldsymbol{w}^i$, in a population is represented by a K-dimensional vector of real numbers:

$$\boldsymbol{w}^i = (w_1^i, w_2^i, \cdots, w_K^i).$$

Concretely, the following genetic operations, i.e., selection, crossover, mutation, and rank algorithm are used in RGA/E.

Roulette wheel selection

Roulette wheel selection is a very popular deterministic operator nowadays, but in most implementations it has random components. The probability of selecting the individual, $\boldsymbol{w}^i$, operation can be expressed as follows,

$$p[\boldsymbol{w}^i] = \frac{g(\boldsymbol{w}^i)}{\sum_{i=1}^{M} g(\boldsymbol{w}^i)}$$

where $g(\cdot)$ is the fitness value of the individual, $\boldsymbol{w}^i$, and M is the number of individuals in the population.

BLX-α crossover

BLX-α crossover [7] reinitializes the values of offspring $\boldsymbol{w}^{ij}$ with values from a extended range given by the parents ($\boldsymbol{w}^i$, $\boldsymbol{w}^j$), where $w_k^{ij} = U(w_k^{ij_{min}} -$

$I\times\alpha, w_k^{ij_{max}}+I\times\alpha)$ [3] with $w_k^{ij_{min}}=\mathrm{Min}(w_k^i, w_k^j)$, $w_k^{ij_{max}}=\mathrm{Max}(w_k^i, w_k^j)$ and $I=w_k^{ij_{max}}-w_k^{ij_{min}}$. The parameter α is a predetermined constant.

Random mutation

For a randomly chosen gene k of an individual $\boldsymbol{w}^i$, the allel w_k^i is added by a randomly chosen value, $\Delta w_k^i \in U[-\Delta w_b, \Delta w_b]$ [17]. Note that Δw_b is the given boundary value for a mutant variant in operation of the random mutation. Therefore, the allele, $w_k^{i\prime}$, of the offsprings, $\boldsymbol{w}^{i\prime}$, can be obtained as follows:

$$w_k^{i\prime} = w_k^i + \Delta w_k^i.$$

Ranking algorithm

The elitism strategy is adopted for improving convergence behavior of genetic algorithms. For executing the strategy, the individuals in the current population, P_t, are sorted according to their fitness. Then the sorted population is referred to as a new one. Therefore, the index of individuals in the population will be used to determine that individuals are allowed to reproduce directly to the next generation by mixing operation. Here, s_n is defined as the predetermined number of the superior individuals from the population $P_t{''}$.

3 The proposed EPSO method

In this section, we describe the proposed EPSO method that efficiently estimates appropriate values of parameters in PSO corresponding to a given optimization problem by RGA/E and the temporally cumulative fitness of the best particle that we adopted.

Fig. 2 illustrates the basic concept of EPSO. An iterative procedure of EPSO is composed of two parts: One is an outer loop using RGA/E. The other is an inner loop using PSO. For a given set of parameters provided by RGA/E, while PSO finds a global optimal solution, the resulting fitness value according to each particle is provided to RGA/E for on-line evolutionary computation. Owing to evolutionary computation, it is expected that the values of parameters in PSO, c_0, c_1, c_2, will be improved, and the corresponding fitness value generated by PSO will increase as iteration proceeds.

Usually, a fitness function, $f(c_0, c_1, c_2)$, in RGA/E can be defined by

[3] $U[a, b]$ denotes that the probability distribution is a continuous uniform one. The support of the interval is defined by the two parameters, a and b, which are its minimum and maximum values.

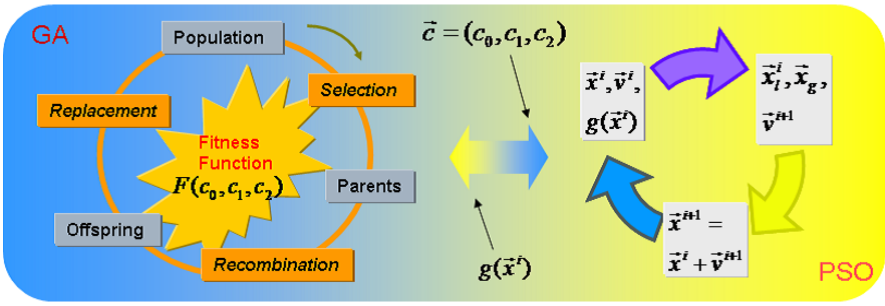

Fig. 2. Basic concept of the proposed EPSO method

$$\begin{aligned} f(c_0, c_1, c_2) &= \max_{k=1,2,\cdots} \{ \max_{i=1,2,\cdots} \{ g(\boldsymbol{x}_k^i)|_{c_0,c_1,c_2} \} \} \\ &= \max_{k=1,2,\cdots} \{ g(\boldsymbol{x}_k^b)|_{c_0,c_1,c_2} \} \\ &= g(\boldsymbol{x}_g)|_{c_0,c_1,c_2}, \end{aligned} \tag{2}$$

where $\boldsymbol{x}_k^b$ is the best position in the entire swarm at time k. However, since the fitness, $f(c_0, c_1, c_2)$, is an instantaneous one, it is unstable for evaluating the performance of PSO. Accordingly, it is not suitable for a reliable evaluation to the stochastic object.

The goal of EPSO is to find appropriate values of parameters in PSO for efficiently solving a given optimization problem. In order to change the temporary stability of the fitness as well as possible, we propose to adopt a temporally cumulative fitness instead of the above instantaneous fitness in RGA/E for effectually evaluating the performance of PSO.

The goodness of a particle swarm searches can be used by the best particle, $\boldsymbol{x}_k^b$, as a practical measure. For eliminating the influence of the mentioned temporary unstable of the fitness, $f(c_0, c_1, c_2)$, we use the temporally cumulative fitness of the best particle for evaluating the performance of PSO. Specifically, the fitness function used in EPSO can be expressed by

$$F(c_0, c_1, c_2) = \sum_{k=1}^{K} g(\boldsymbol{x}_k^b)|_{c_0,c_1,c_2}. \tag{3}$$

It is obvious that the temporally cumulative fitness, $F(c_0, c_1, c_2)$, evaluates the entire dynamic process of the best particle, $\boldsymbol{x}_k^b$. Therefore, it can basically eliminate the influence from the instantaneous fitness, $f(c_0, c_1, c_2)$, which just be determined by an instantaneous value. Based on the property of the proposed fitness, $F(c_0, c_1, c_2)$, the goal of EPSO, which efficiently estimates appropriate values of parameter in PSO, should be easily achieved.

In order to express the property of the fitness, Fig. 3 illustrates the relationship between the instantaneous fitness, $f(c_0, c_1, c_2)$, of the best particle at time k and its temporally cumulative fitness, $F(c_0, c_1, c_2)$. We observed that the values of the fitness, $F(c_0, c_1, c_2)$, almost linearly increase without receiving the influence according to the variation of the fitness, $f(c_0, c_1, c_2)$, even if the value of the fitness, $f(c_0, c_1, c_2)$, greatly increase or decrease during the particle swarm searches.

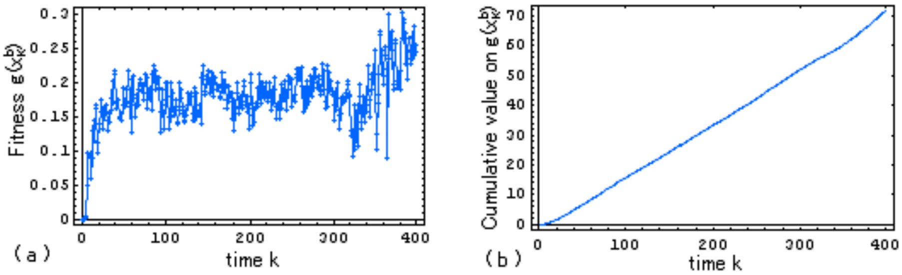

Fig. 3. The relationship between the instantaneous fitness, $f(c_0, c_1, c_2)$, and the temporally cumulative fitness, $F(c_0, c_1, c_2)$. (a) Instantaneous fitness, $f(c_0, c_1, c_2)$, of the best particle at time k, (b) The temporally cumulative value of the instantaneous fitness, $F(c_0, c_1, c_2)$, over time.

It is to be noted that the temporally cumulative fitness, $F(c_0, c_1, c_2)$, of the best particle is used to express the performance of the estimated PSO model as a swarm representative. This is a new trial in evaluating a stochastic objective, and as far as we known, the fitness, $F(c_0, c_1, c_2)$, has never been proposed, and it is for the first time applied for evaluating the performance of PSO by RGA/E.

4 Computer experiments

4.1 Task and parameters in experiment

To demonstrate the effectiveness of EPSO, computer experiments with a 2-dimensional optimization problem are carried out. Based on the obtained experimental results, we investigate the characteristics of dependency on different initial conditions, and analyze the resulting PSO models.

The search space of the optimization problem we used is 60×60 (i.e., $x_{max} = 30$), the fitness value regarding the search environment in Fig. 4 is definded by

$$g(\boldsymbol{x}) = 0.4\, e^{-\frac{(x_1-20)^2+(x_2-3)^2}{2\times 3^2}} + 0.25\, e^{-\frac{(x_1-15)^2+(x_2+20)^2}{2\times 4^2}} + 0.2\, e^{-\frac{(x_1+10)^2+(x_2+15)^2}{2\times 4^2}}$$
$$+0.2\, e^{-\frac{(x_1-10)^2+(x_2-20)^2}{2\times 4^2}} + 0.25\, e^{-\frac{(x_1-0)^2+(x_2+1)^2}{2\times 4^2}} + 0.05\, e^{-\frac{(x_1+20)^2+(x_2+5)^2}{2\times 6^2}} .$$

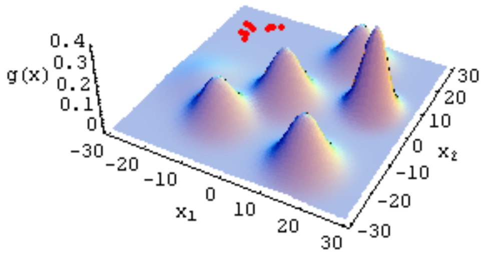

Fig. 4. Search environment

About the definition of initial conditions we used: particles start from a designated region, (z_s, z_c). z_s denotes the area of the region, and z_c denotes the central coordinates of the region, respectively. In order to clarify how these conditions affect the performance of PSO, we use the following 4 cases, i.e., case 1: $z_{s_1} = 10 \times 10$, $z_{c_1} = (-20, 20)$; case 2: $z_{s_2} = 10 \times 10$, $z_{c_2} = (0, 0)$; case 3: $z_{s_3} = 30 \times 30$, $z_{c_3} = (0, 0)$; case 4: $z_{s_4} = 60 \times 60$, $z_{c_4} = (0, 0)$ for solving the given optimization problem.

The computational cost for each iteration is proportional to the number of particles. Due to heavy computational cost for simultaneously executing RGA/E and PSO, EPSO should be carried out by using small number of particles. To reduce the computational cost, we have to reduce the number of particles. On the other hand, the number of iterations might increase for small number of particles, by taking the trade-off into account, we decide the number of particles is 10 in our computer experiments for finding a global optimal solution [1].

Table 1. Major parameters used in EPSO

Items	Parameters
The number of individuals	M = 100
The number of generation	G = 20
The number of superior individuals	$s_n = 2$
Roulette wheel selection	–
Probability of BLX-2.0 crossover	$p_c = 1.0$
Probability of random mutation	$p_m = 1.0$
Boundary value of the mutation	$\Delta x_b = 1.5$
The number of particle	P = 10
The number of iterations	K = 400
The maximum velocity	$v_{max} = 20$

Table 1 gives the major parameters used in EPSO for solving the given optimization problem shown in Fig. 4 in the following experiments. Note that since the search space of the given optimization problem is just a 2-

dimensional, a global and effective search must be kept by increasing the diversity of individuals at each generation. So we set the parameters on the probability of crossover and mutation operators to be 1.0 for keeping high diversity of individuals in a population at each generation.

4.2 Experimental results

Firstly, we observe the varying status of the values of parameters in PSO with implementing EPSO. For example, Fig. 5 shows the variations of the temporally cumulative fitnesses ($F(c_0, c_1, c_2)$), which are the superior six ranks among the entire population and the corresponding to the parameter values of these individuals (c_0, c_1, c_2) at a search process (different generations).

As seen in Fig. 5(a), the result of the obtained highest fitness, $F(\cdot) = 0.400008$, indicates that EPSO can find appropriate values of parameters in PSO for efficiently solving the given optimization problem, and the resulting parameter values of PSO arriving at the global optimal solution are not unique. This means that the proposed EPSO method is effective in estimating the optimal PSO model, and the obtained optimal PSO model is not only one for solving the given optimization problem.

Table 2. The resulting appropriate values of parameters in PSO by starting particles from the different initial conditions, i.e., cases 1, 2, 3, and 4.

Case	Parameter c_0	c_1	c_2	Model type	Success ratio
1	-	4.65±0.0	2.22±0.0	b	9.10%
	0.69±0.2	-	5.45±4.7	c	36.4%
	1.01±0.2	1.37±0.8	1.62±0.8	d	54.5%
2	-	-	11.6±0.7	a	15.4%
	-	1.84±0.0	7.86±0.0	b	7.70%
	0.87±0.1	-	3.58±3.2	c	30.8%
	1.15±0.3	1.65±0.4	1.70±0.6	d	46.1%
3	-	-	2.91±1.9	a	23.5%
	-	0.66±0.0	3.12±0.0	b	5.90%
	0.74±0.4	-	1.31±0.5	c	11.7%
	1.05±0.3	0.92±0.6	1.66±1.1	d	58.8%
4	-	4.96±4.2	4.37±2.1	b	16.7%
	0.81±0.1	-	2.22±1.1	c	33.3%
	1.00±0.2	1.91±0.9	1.77±0.9	d	50.0%

Table 2 indicates the resulting values of parameters in PSO for finding the global optimal solution by starting particles with the different initial conditions, i.e., cases 1, 2, 3, and 4 [4].

[4] Computing enviroment: Intel(R) Xeon(TM); CPU 3.40GHz; Memory 2GB RAM; Computing tool: Mathematica 5.2; Computing time: about 3 min per case.

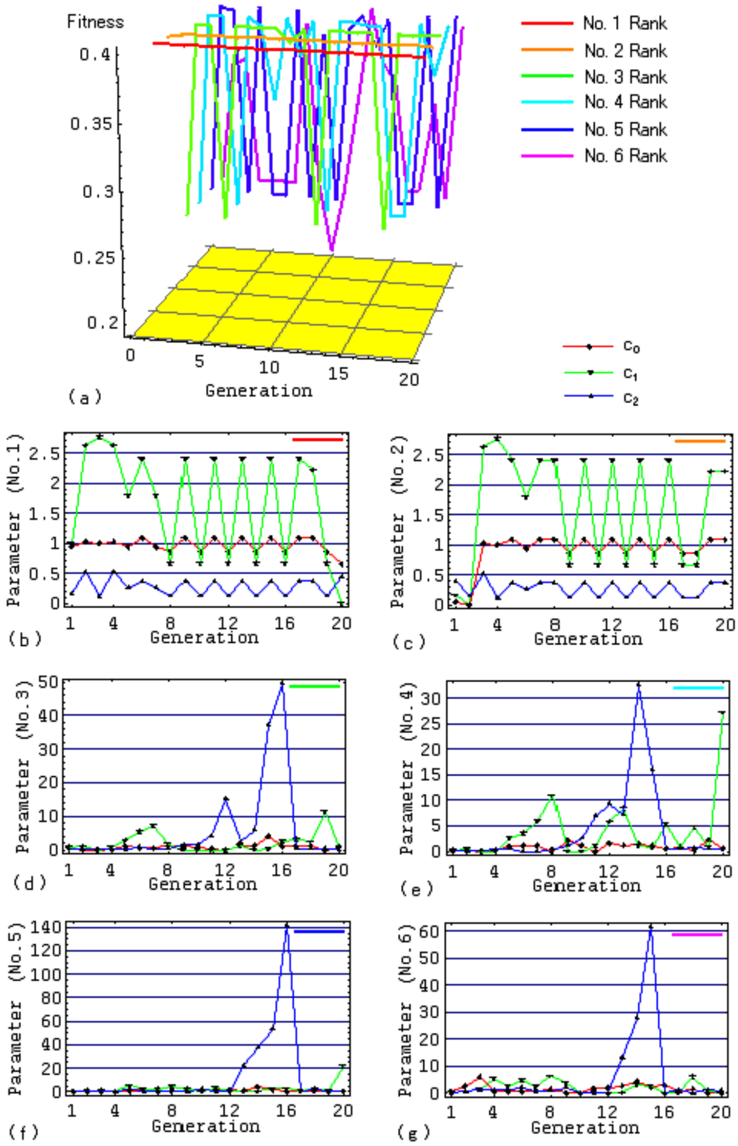

Fig. 5. The variations of the fitness which are superior six ranks among the entire population and the corresponding to the parameter values in a search process for case 1, (a) The variations of the fitnesses with implementing EPSO, (b-g) The variations of the corresponding to parameter values, c_0, c_1, c_2.

The "type" in Table 2 stands for the following velocity equations in PSO.

$$\begin{cases} type\ a: \ \boldsymbol{v}^i_{k+1} = & & c_2 r_2(\boldsymbol{x}_g - \boldsymbol{x}^i_k) \\ type\ b: \ \boldsymbol{v}^i_{k+1} = & c_1 r_1(\boldsymbol{x}^i_l - \boldsymbol{x}^i_k) & + c_2 r_2(\boldsymbol{x}_g - \boldsymbol{x}^i_k) \\ type\ c: \ \boldsymbol{v}^i_{k+1} = c_0 \boldsymbol{v}^i_k & & + c_2 r_2(\boldsymbol{x}_g - \boldsymbol{x}^i_k) \\ type\ d: \ \boldsymbol{v}^i_{k+1} = c_0 \boldsymbol{v}^i_k & + c_1 r_1(\boldsymbol{x}^i_l - \boldsymbol{x}^i_k) & + c_2 r_2(\boldsymbol{x}_g - \boldsymbol{x}^i_k) \end{cases}$$

Secondly, we observed that the estimated mean value of parameter, c_2, is always not zero under any initial condition. It declares that the swarm confidence factor plays an important role in finding a global optimal solution. The fact perfectly agrees with the role of the swarm communication in PSO, i.e., the members of the swarm communicate the information of the best positions with each other.

We also observed that the mean values of parameters, c_1 and c_2, becomes small and the mean value of parameter, c_0, becomes bigger (approaches to 1.0) with the extending the area of the square zone (comparison with the resulting statistical data of cases 2, 3, 4). This means that the effect of individual confidence factor, c_1, and swarm confidence factor, c_2, becomes weaker and the effect of the inertia factor, c_0, becomes stronger, respectively, for enhancing performance of PSO.

Table 3. The statistical data of the fitness by the resulting PSO models with different type (mean±standard deviation) for case 1, 2, 3, and 4. (PR: The proportion ratio for the success)

Items	Model				Case
	type a	type b	type c	type d	
Fitness	–	0.213±0.149	0.218±0.193	0.290±0.121	1
PR	–	22.73%	36.36%	40.91%	
Fitness	0.287±0.066	0.302±0.073	0.362±0.066	0.349±0.073	2
PR	17.86%	21.43%	21.43%	39.29%	
Fitness	0.241±0.043	0.260±0.049	0.260±0.049	0.362±0.066	3
PR	5.0%	10.0%	10.0%	75.0%	
Fitness	–	0.347±0.073	0.340±0.094	0.384±0.045	4
PR	–	27.91%	32.56%	39.53%	

Thirdly, we implement the resulting each PSO model, respectively, for investigating the convergence behavoirs and characteristics on them. For example, Table 3 gives the obtained experimental data shown in Fig. 6 (with a box-and-whisker plot [5]) and the corresponging to the proportion ratio of the

[5] A box-and-whisker plot is histogram-like method of displaying data. The range is simply the difference between the maximum and minimum values in the data. The quartiles divide the data into quarters like the median divides the data into the halves. There are three quartiles: $\{Q_1, Q_2, Q_3\}$. The first quartile, Q_1, is the

success by using the mean values of each parameter in each model for case 1, 2, 3, and 4.

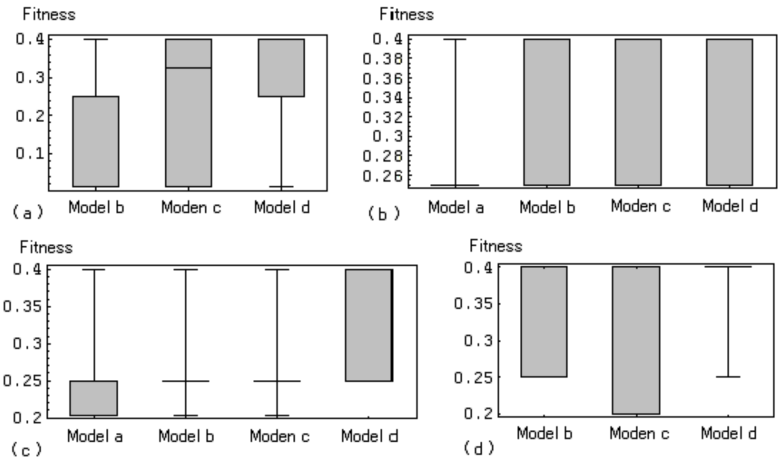

Fig. 6. Distribution of the fitness corresponding to each model for case 1, 2, 3, and 4.

Comparision with the resulting data of the proportion ratio of the success shown in Table 3, we observed that the type d of PSO models has better performance than those of other types of PSO model in finding the global optimal solution, that is, the mean value of the fitness is higher and the standard deviation value is smaller than those of others, respectively. This experimental result sufficiently confirms that the structure of the original PSO model is a rational and effective one for efficiently finding the global optimal solution.

As an example, Fig. 7 shows a search process of PSO with the resulting mean values of parameters in PSO (model d) by EPSO for case 1. We can see that the result indicates that these particles arrived the positions in the search space and distribution of these positions at the search process, and the effect of EPSO in exploration. Note that the pattern of the distribution is deeply dependent on the given optimization problem and the parameters used in the PSO.

4.3 Discussions

We are interested in finding optimal parameter sets that provide the PSO with reasonable exploration and convergence capability with applying a restriction

median of the lower part of the data, the second quartile, Q_2, is the median of the entire set of the data, the third quartile, Q_3, is the median of the upper part of the data.

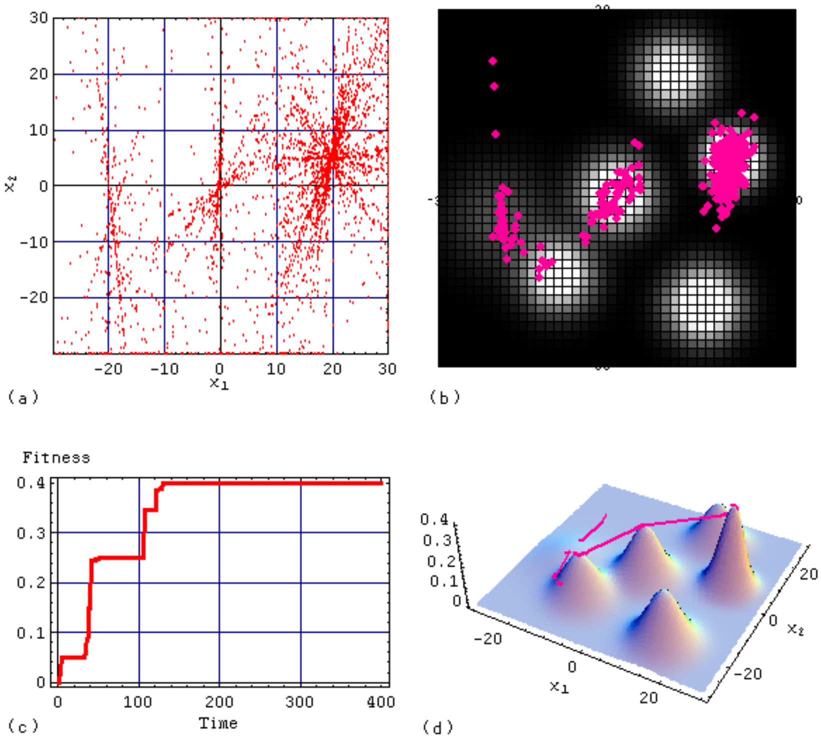

Fig. 7. A search process of PSO with the resulting average values of parameters by EPSO (model *d*) for case 1. (a) The distribution of positions for all particles, (b) The distribution of positions for the best particle, (c) The variation of fitness for the global best particle, (d) The moving track of the global best particle in the search environment.

for the velocity. It is obvious that the success ratio of using the structure of the original PSO (model *d*) is significantly bigger than those of other models (model *a*, *b*, and *c*) irrespective of various initial conditions. Because the value of parameter, c_2 exists in each kind of the velocity equations, so that the best search strategy in PSO for finding the global optimal solution is to follow the best particle which is nearest to the global optimal solution. And through implementing EPSO, the optimal structure of PSO models and their parameter values corresponding to the given optimization problem are simultaneously obtained.

However, how about the effect of EPSO in comparison with other methods? For answering the question, in the following experiments we compared the performance of EPSO, PSO and RGA/E for the given optimization problem. Fig. 8 illustrates the obtained experimental results for case 1. The parameters in RGA/E shown in Table 1 were used.

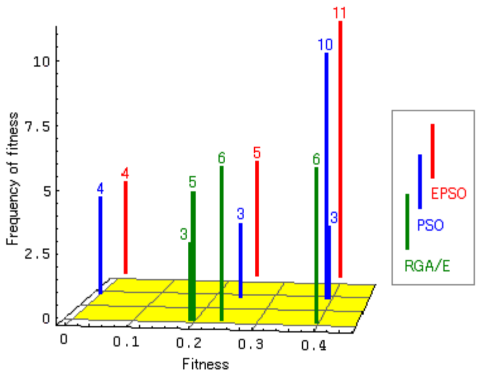

Fig. 8. Performance comparison among EPSO (red), PSO (blue), and RGA/E (green) for case 1.

As seen in Fig. 8, EPSO has the best performance in the frequency of fitness, i.e., the particles found the global optimal solution (EPSO: 11, PSO: 3, RGA/E: 6). The obtained results and the goal of the use of the fitness, $F(c_0, c_1, c_2)$, accord well together. From the viewpoint of finding the global optimal solution, the performance of RGA/E is inferior to EPSO and is superior to PSO. However, comparing the distributions of frequency of fitness, we observed that the search of EPSO or PSO relatively falls into local minima than those of RGA/E.

Table 4. The statistical data of the fitness by EPSO, PSO, and RGA/E (mean±standard deviation) for case 1.

	EPSO	PSO	RGA/E
Fitness	0.284±0.149	0.299±0.152	0.275±0.083

As contrasted with the above explaination, Table 4 gives the mean and standard deviation of the fitness values by EPSO, PSO, and RGA/E for case 1. The mean values of the frequency distribution indicate that PSO is superior to EPSO and RGA/E. The reason is because the total number of the frequency of fitness arriving the global optimal solution and near the global optimal solution for PSO is higher than those of EPSO and RGA/P. However, comparing with the number ratio (11 : 3) of the frequency of fitness reaching the global best position between EPSO and PSO, we found that EPSO is superior to PSO in the accuracy of statistics. This result sufficientlly reflects the characteristics of the temporally cumulative fitness, $F(c_0, c_1, c_2)$, that we proposed in EPSO.

Of course, we should also point out that the performance of RGA/E depends upon given parameters too. Their parameter values also need to be optimized for the fair comparison.

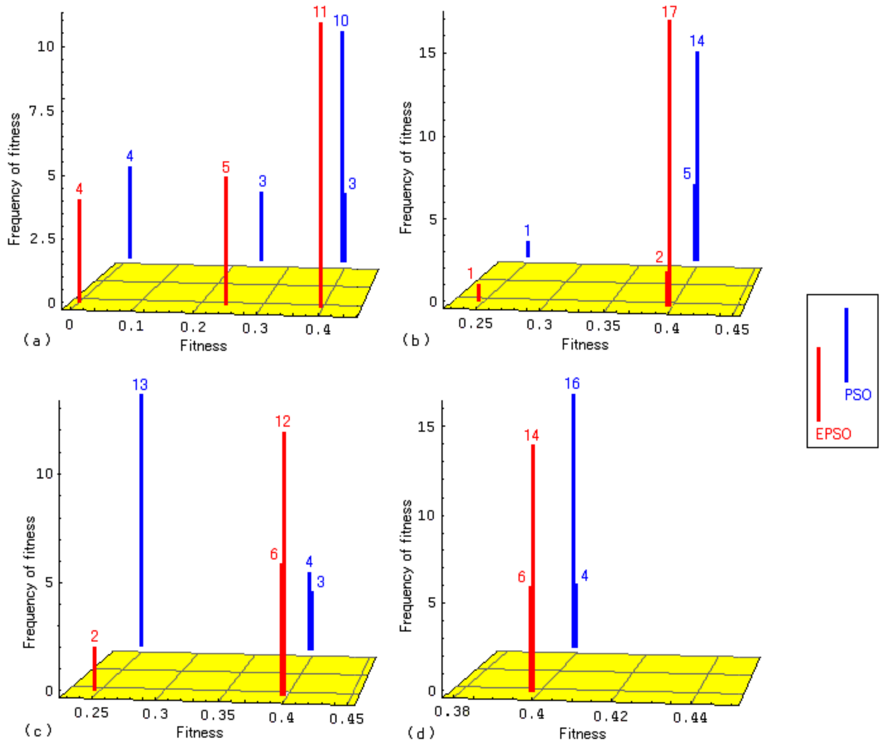

Fig. 9. Frequency distribution of fitness on EPSO and the original PSO. The fitness value for the global optimal solution is 0.400008. (a) case1; Histogram with the values of parameters in the original PSO, histogram with the mean value of parameters in the PSO (type d) estimated by EPSO, and comparison with them. (b) case 2. (c) case 3. (d) case 4.

To further examine the effectiveness of EPSO, Fig. 9 illustrates the frequency distributions of the fitness by implementing the original PSO and by implementing EPSO. We observed that the proposed EPSO method has superior performance for finding the global optimal solution than that the original PSO has in the case 1, 2, 3, and 4. The experimental results demonstrate the effectiveness of the proposed EPSO method, and indicate the variation of the performance of the estimated models according to different initial conditions. The results clearly exhibit that the temporally cumulative fitness, $F(c_0, c_1, c_2)$, is effective, and suitable for finding appropriate values of parameters in PSO to efficiently solve the given optimization problem.

The use of the fitness, $F(c_0, c_1, c_2)$, provides a useful way that systematically evaluates performance of stochastic and population-based systems. Based on the property of the fitness, it is considered that the proposal is not only applicable to PSO, but also is applicable to other population-based adaptive optimization methods such as Ant Colony Optimization (ACO) [4], Genetic Algorithms (GAs) [8] and so on. So it can be applied for efficiently estimating optimal models as a general fitness function of meta-optimization method.

5 Conclusion

In this paper, we have proposed a meta-optimization method, Evolutionary Particle Swarm Optimization, EPSO, that efficiently determines the values of parameters in PSO for finding a global optimal solution. It adopts a temporally cumulative fitness, which evaluates the varying position of the best particle at a search process, for evaluating the performance of PSO. This is the first proposal for estimating the appropriate values of parameters in PSO by a real-coded genetic algorithm with elitism strategy. Furthermore, it is considered that the fitness is not only applicable to evaluate PSO, but also extendable to various stochastic adaptive optimization algorithms with adjustable parameters for evaluating dynamic process.

Our experimental results show the effectiveness of the proposed method, and declare that the values of parameters in PSO are correctly estimated for finding the global optimal solution to the given 2-dimensional optimization problem. The proposed EPSO method exhibits that it has higher success ratio than those of the original PSO and RGA/E in exploration. Most importantly, the results suggest that the proposed method successfully provides a new paradigm for designing dynamic objective.

Even though better experimental results have been obtained, only a small scale optimization problem with various initial conditions has so far been carried out. In order to enhance the efficiency and exactness in model selection and parameter identification, applications of the proposed EPSO method to complex and high dimensional optimization problems are left for near the future studies.

Acknowledgment This research was partially supported by the 21st century COE (Center of Excellence) program (#19) granted to Kyushu Institute of Technology from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

1. Bergh F, Engelbrecht AP (2001) Effects of Swarm Size on Cooperative Partical Swarm Optimisers *Proceedings of the Genetic and Evolutionary Computation*

Conference (GECCO2001), Morgan Kaufmann 892-899 San Francisco California USA

2. Beielstein T, Parsopoulos KE, Vrahatis MN (2002) Tuning PSO Parameters Through Sensitivity Analysis Technical Report of the Collaborative Research Center 531 Computational Intelligence CI-124/02, University of Dortmund.
3. Carlisle A, Dozier G (2001) An Off-The-Shelf PSO. *Proceedings of the Workshop on Particle Swarm Optimization Indianapolis* 1-6
4. Dorigo M, Maniezzo V and Colorni A. (1996) The Ant System: Optimization by a colony of cooperating agents. *IEEE Transaction on Systems, Man, and Cybernetics-Part B* 1:1-13
5. Eberhart RC and Kennedy J (1995) A new optimizer using particle swarm theory. *Proceedings of the sixth International Symposium on Micro Machine and Human Science* 39-43 Nagoya Japan
6. Eberhart RC and Shi Y (2000) Comparing inertia weights and constriction factors in particleswarm optimization. *Proceedings of the 2000 IEEE Congress on Evolutionary Computation* 1: 84-88 La Jolla CA
7. Eshelman LJ and Schaffer JD (1993) Real-Coded Genetic Algorithms and Interval-Schemata. *Foundations of Genentic Algorithms* Morgan Kaufman Publishers, San Mateo 2:187-202
8. Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston
9. Gudise VG, Venayagamoorthy GK (2003) Evolving digital circuits using particle swarm. Neural Networks, *Proceedings of the International Joint Conference on Special Issue* 1:468-472
10. Kennedy J and Eberhart RC (1995) Particle swarm optimization. *Proceedings of the 1995 IEEE International Conference on Neural Networks* 1942-1948 Piscataway, New Jersey.
11. Kennedy J (2006) In Search of the Essential Particle Swarm *Proceedings of 2006 IEEE Congress on Evolutionary Computations*, 6158-6165 Vancouver BC Canada.
12. Meissner M, Schmuker M and Schneider G (2006) Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics* 7:125-135
13. Parsopoulos KE and Vrahatis MN (2002) Recent approaches to global optimization problems through Particle Swarm Optimization *Natural Computing* 1:235-306
14. Margarita Reyes-Sierra and Carlos A. Coello Coello (2006) Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art *International Journal of Computational Intelligence Research* 2(3):287-308
15. Spina R (2006) Optimisation of injection moulded parts by using ANN-PSO approach. *Journal of Achievements in Materials and Manufacturing Engineering* 15(1-2):146-152
16. Storn R and Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization* 11:341-359.
17. Zhang H and Ishikawa M (2005) A Hybrid Real–Coded Genetic Algorithm with Local Search. *Proceedings of the 12th International Conference on Neural Information Processing* (ICONIP2005) 732-737 Taipei Taiwan R.O.C..

18. Xie XF, Zhang WJ and Yang ZL (2002) A Disspative Particle Swarm Optimiztion *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC2002) 1456-1461 Honolulu Hawaii USA