2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Locating All the Global Minima Using Multi-Species Particle Swarm Optimizer: The Inertia Weight and The Constriction Factor Variants

Masao Iwamatsu, *Member, IEEE*

***Abstract*— This paper reports further simplification and improvement of a modified particle swarm optimizer (PSO) called the Multi-Species Particle Swarm Optimizer (MSPSO) proposed by the author. MSPSO extends the original PSO by dividing the particle swarm spatially into a multiple cluster called a species in a multi-dimensional search space. Each species explores a different area of the search space and tries to find out the global or local optima of that area. Therefore it can be used to locate all the global minima of multi-modal functions in parallel. The previous version of MSPSO relies strongly on the inertia-weight annealing and its performance depends on the annealing schedule. In this paper, instead, we use the constriction factor proposed by Clerc. Our new MSPSO could locate, for example, all 18 global optima of the two-dimensional Shubert function, yet it is free from annealing-schedule optimization of the inertia weight.**

I. Introduction

Global optimization of continuous function is the task of finding the lowest altitude points in a rugged landscape with high dimensions. Many heuristic methods based, in particular, on the population-based stochastic search have been proposed [1], [2]. Recently, Kennedy, Eberhart and Shi [2], [3] proposed a new heuristic method called the Particle Swarm Optimization (PSO) based on a socio-cognitive theory.

In the so-called g-PSO, which seems most popular now, the population consists of flying particles, and the particles (individuals or agents) search for the function space by random *flight*. They can memorize the location of their own best points *p-best* (personal best) where they got their best *fitness* (reward). They can also know the best point *g-best* (global best) among personal best points, by which they can exchange minimum information. By using the knowledge of own personal best (*p-best*) and the global best (*g-best*), each particle adjusts the direction and the magnitude of velocity of its flight. Then, the whole population moves toward the global minimum as a flock of bird and a school of fish move towards food. The performance of this original PSO has been shown to be comparable to those of the other population-based heuristics such as the genetic algorithm [2] or evolutionary programming [4], [5].

Like other meta-heuristics, PSO is designed to locate a unique single optimum solution. However, problems exist where several solutions or even an exhaustive search of all the multiple global optima are necessary. Such problems appear, for example, in engineering design [6].

In the previous paper [7], we focused on the problem of locating all the global optima of multi-modal functions using PSO, and proposed a new optimizer called the multi-species particle swarm optimization (MSPSO) algorithm. The proposed algorithm was shown to be successful to locate all the global minima of several multimodal functions. Unfortunately, a cumbersome annealing schedule of the inertia-weight [8], is necessary to tune the performance of the algorithm.

In this report, we propose a new variant of MSPSO that uses the constriction factor [2], [9] instead of the inertia-weight [8] and is free from the annealing scheduling. This paper is organized as follows: The next section consists of a brief review of the original PSO. Then we introduce speciation into PSO and propose two variants of MSPSO. In section 3, the new algorithm is tested against two-dimensional Shubert functions. We conclude this paper in section 4.

II. Speciation in a Particle Swarm

A. Original Particle Swarm Optimizer g-PSO

Suppose we have to find out the (single) global minimum of a multi-modal function $f(\mathbf{x}) = f(\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n)$ in n-dimensional space. In PSO, each particle i ($i = 1, \ldots N$) in the population P is characterized by three vectors $(\mathbf{x}_i, \mathbf{v}_i, \mathbf{p}_i)$ which represent their temporal position $\mathbf{x}_i = (\mathrm{x}_{i1}, \mathrm{x}_{i2}, \ldots, \mathrm{x}_{in})$, velocity $\mathbf{v}_i = (\mathrm{v}_{i1}, \mathrm{v}_{i2}, \ldots, \mathrm{v}_{in})$, and the best position $\mathbf{p}_i = (\mathrm{p}_{i1}, \mathrm{p}_{i2}, \ldots, \mathrm{p}_{in})$. The *fitness* of each particle is given by the function value $f(\mathbf{x}_i)$. Since we look at the minimization problem in this report, the lower the function value the better the fitness. Each particle stores its best position $\mathbf{p}_i$ called *personal best, p-best* which gives the best fitness in memory. They can also consult their neighbor's best position $\mathbf{p}_g$. Most simply, the neighbor is the whole population (fully connected topology), and therefore, the neighbor's best is the best position among personal bests of the whole population. Hence, the position $\mathbf{p}_g$ is called *global best, g-best* [2].

Now each particle i moves around the search space, and renews its velocity component j using its past experience (personal best) and the population's experience (global best). In the inertia-weight variant [2], [8], the velocity v is updated

Masao Iwamatsu is with the Faculty of Engineering, Musashi Institute of Technology, Tokyo 158-8557, JAPAN (phone: +81-3-3703-3111 ext.2382; fax: +81-3-5707-2222; e-mail: iwamatsu@ph.ns.musashi-tech.ac.jp).

0-7803-9487-9/06/$20.00/©2006 IEEE

by

$$\mathrm{v}_{ij} := w\mathrm{v}_{ij} + c_1 r_1 \left(\mathrm{p}_{ij} - \mathrm{x}_{ij}\right) + c_2 r_2 \left(\mathrm{p}_{gj} - \mathrm{x}_{ij}\right). \quad (1)$$

The parameter c_1 and c_2 are the acceleration or learning constant, and r_1 and r_2 are the uniform random numbers within the range $[0, 1]$. The parameter w is called inertia weight, which controls the exploration (global search)-exploitation (local search) tradeoff.

If v_{ij} is larger than a predefined velocity $\mathrm{V}_{\max}$ called maximum velocity, it is fixed to $\mathrm{V}_{\max}$. Similarly, if it is smaller than -$\mathrm{V}_{\max}$, it is fixed to -$\mathrm{V}_{\max}$. Then the particle changes its position by the "equation of motion":

$$\mathrm{x}_{ij} := \mathrm{x}_{ij} + \mathrm{v}_{ij}. \quad (2)$$

Therefore, in PSO, the particles move around the personal best positions $\mathbf{p}_i$ and global best positions $\mathbf{p}_g$ under the attractive random forces from these two points.

Both the inertia weight w [3], [8] and the maximum velocity $\mathrm{V}_{\max}$ control the exploration (global search) and exploitation (local search) abilities of each of the particles [10]. Naturally, the larger $\mathrm{V}_{\max}$ and the larger w favor the exploration, and smaller ones are suitable for the exploitation. Shi and Eberhart [10] recommended a time varying inertia weight with linearly decreasing inertia weight with $w_{\mathrm{int}} = 0.9$ at the initial step iter=0 of iteration and $w_{\mathrm{fin}} = 0.4$ at the final step iter=MAX:

$$w := (w_{\mathrm{int}} - w_{\mathrm{fin}}) \times (\mathrm{MAX} - \mathrm{iter}) / \mathrm{MAX} + w_{\mathrm{fin}}. \quad (3)$$

Therefore, we have to specify the annealing schedule (3) in advance, which is practically tedious and problematic.

In constriction-factor variant [2], [9], [11], the algorithm is basically the same as that for the inertia-weight variant, except the velocity is updated by

$$\mathrm{v}_{ij} := \chi \left(\mathrm{v}_{ij} + c_1 r_1 \left(\mathrm{p}_{ij} - \mathrm{x}_{ij}\right) + c_2 r_2 \left(\mathrm{p}_{gj} - \mathrm{x}_{ij}\right)\right). \quad (4)$$

instead of (1). Now the constriction factor χ is defined using c_1 and c_2 by

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (5)$$

with

$$\phi = c_1 + c_2 \quad (6)$$

Therefore the constriction-factor variant is free from tedious scheduling of inertia-weight annealing.

B. Multi-Species Particle Swarm Optimizer

Since this original *g*-PSO as it is or even more flexible local PSO (*l*-PSO) do not have a niching ability [12] and cannot find the multiple global minima of multimodal functions, we divide the whole population P into several subpopulations P_s ($s = 1, \ldots, M$) called species [13] so that each of species search for different minima. A similar multi-species particle swarm optimization algorithm (MS-PSO) was proposed by Chow and Tsui [14] recently. However, their species optimizes different functions and, thus, realizes the multiobjective optimization. In contrast, our algorithm aims at optimizing single function.

In order to cluster the population in n-dimensional search space, we define the distance $d(i, j)$ of two particles i and j by the Euclidean distance in n-dimensional search space with respect to their personal best positions $\mathbf{p}_i$ and $\mathbf{p}_j$ rather than their temporal positions $\mathbf{x}_i$ and $\mathbf{x}_j$ [7]:

$$d(i, j) = \sqrt{\sum_{k=1}^{n} \left(\mathrm{p}_{ik} - \mathrm{p}_{jk}\right)^2}. \quad (7)$$

The use of their best positions $\mathbf{p}$ rather than the temporal positions $\mathbf{x}$ makes the algorithm simple and efficient.

```
procedure Speciation
  begin
    S=φ;
    while (no more unmarked particles in population) do
      Search for the best (by personal best) unmarked i;
      Mark particle i as processed;
      found ← FALSE;
      for all species seeds s ∈ S do
        if(d(i, s) <= σ) then
          found← TRUE;
          Assign the particle i to species s;
          break;
        end(if);
      end(for)
      if(not found) then
        Particle i becomes the seed of new species;
        S← S ∪ i;
      end(if)
    end(while)
  end(procedure)
```

Fig. 1. Pseudo code of procedure (function) "Speciation" for finding a new species seed s or assigning each particle i to appropriate species s. S is the set of species seeds.

Each species P_s occupies a different area of the n-dimensional search space, and is characterized by *species seed* s whose personal best is at $\mathbf{p}_s$. The species seed is a dominating individual of the species in each area whose best fitness $f(\mathbf{p}_s)$ is the best fitness of that area. Now, each particle i belongs to one of the species P_s if the distance from species seed $\mathbf{p}_s$ is shorter than the threshold σ called species radius:

$$d(i, s) < \sigma \quad \rightarrow i \in \mathrm{P}_s. \quad (8)$$

Therefore the personal best $\mathbf{p}_s$ occupies the center of the hyper-sphere of radius σ that forms the species territory (Fig. 2).

The algorithm to select the species seeds and to assign each particle to one of species is shown in Fig. 1. It is superficially similar to the standard clustering algorithms [15].

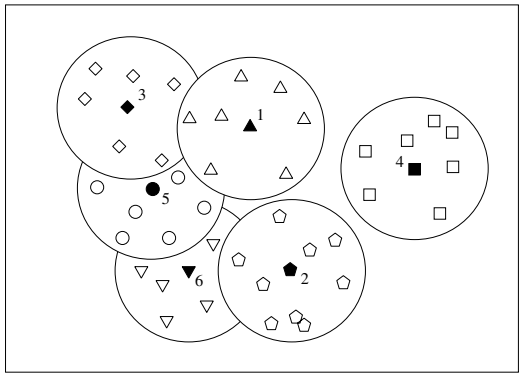

Fig. 2. Definition of species territory in search space. The species consists of particles that are enclosed by a hyper-sphere of radius σ centered at the species seed. The species seeds are indicated by black symbols, while the member of species are indicated by white symbols of the same shape. The orders of species seeds found are 1→2→3→ 4→5. Therefore the seed No.1 has the best fitness (the lowest function value). Note that there are some overlaps among the territory of species, however, the affiliation of each particle can be uniquely assigned according to the algorithm in Fig.1.

However, our clustering in Fig. 1 based on the algorithm of Li *et al.* [16] for their species conserving genetic algorithm (SCGA) is guided by the fitness rather than the density of the particles. Our species is also conceptually very similar to the neighborhood of the *l*-PSO [2], [17], [18]. Furthermore, in contrast to other algorithms [14], [19] neither the number of species nor the population of each species is pre-determined. Rather, the self-organization of species occurs spontaneously through the speciation algorithm in Fig. 1, and no further artificial procedure to cluster the population is necessary.

Now each particle i moves around the search space, and changes its velocity using the information of its personal best and the species best (seed's personal best) instead of the global best. In our Multi-Species PSO (MSPSO) eq. (1) of the original PSO is replaced by

$$\mathrm{v}_{ij} := w\mathrm{v}_{ij} + c_1 r_1 \left(\mathrm{p}_{ij} - \mathrm{x}_{ij}\right) + c_2 r_2 \left(\mathrm{p}_{sj} - \mathrm{x}_{ij}\right), \quad (9)$$

We refer this inertia-weight variant of MSPSO as MSPSO-i. Similarly, eq. (4) is replaced by

$$\mathrm{v}_{ij} := \chi \left(\mathrm{v}_{ij} + c_1 r_1 \left(\mathrm{p}_{ij} - \mathrm{x}_{ij}\right) + c_2 r_2 \left(\mathrm{p}_{sj} - \mathrm{x}_{ij}\right)\right). \quad (10)$$

in the constriction-factor variant which we refer as MSPSO-c. Here, the global best $\mathbf{p}_g$ is replaced by $\mathbf{p}_s$ of species seed that is the best position of the species to which the particle i belongs.

The structure of our MSPSO algorithm is basically the same as the original PSO algorithm except that the population P is partitioned into multiple species P_s using the speciation procedure (Fig. 1). The structure of our MSPSO algorithm is shown in Fig. 3.

III. Experimental Results

We have already used above inertia-weight variant of multi-species particle swarm optimizer (MSPSO-i) to locate all the global minima of several test functions examined previously by various optimization technique [20], in particular,

program Multi-Species Particle Swarm Optimizer (MSPSO)
begin
iter := 0;
Initialize positions of all particles
(x_{ij}:=rand($\mathrm{x}_{\min}$,$\mathrm{x}_{\max}$));
Initialize velocities of all particles
(v_{ij}=rand(-$\mathrm{V}_{\max}$,$\mathrm{V}_{\max}$));
while (not termination condition) **do**
Renew personal bests;
Speciation (clustering, see Fig. 1);
for all particles;
Calculate velocity **v** using species seed
(eqs.(9) or (10));
if $\mathrm{v}_{ij} > \mathrm{V}_{\max}$ **then**
v_{ij}:=$\mathrm{V}_{\max}$;
else if $\mathrm{v}_{\mathrm{ij}} < -\mathrm{V}_{\max}$ **then**
v_{ij}:=-$\mathrm{V}_{\max}$
end(if)
Move particles by **x**:=**x**+**v**
Renew personal positions;
Evaluate function;
end(for)
iter := iter+1;
end(while)
Output the species bests
end(program)

Fig. 3. Pseudo code of multi-species PSO (MSPSO). Speciation (clustering) and renewal of velocity using the species' best rather than global best are used. The procedure **Speciation** (clustering) is defined in Fig. 1.

by species conserving genetic algorithm (SCGA) based on the same speciation technique [16].

In this paper we will concentrate on the two-dimensional Shubert function, which is notoriously difficult multimodal function and has been studied by a relatively small number of researchers [7], [16], [21]

The two-dimensional Shubert function is defined by

$$\begin{aligned} f(x_1, x_2) &= \left(\sum_{i=1}^{5} i\cos\left[(i+1)x_1 + i\right]\right) \\ &\times \left(\sum_{i=1}^{5} i\cos\left[(i+1)x_2 + i\right]\right), \\ &(-10 \le x_{1,2} \le 10), \end{aligned} \quad (11)$$

which has 760 local minima, and only 18 of which are global minima with an object function value of -186.73. These 18 minima are further classified into 9 clusters each of which consists of two solutions separated only by a small distance 0.98. The minimum distance between two clusters, however, is much longer and is 5.65 as shown in Fig. 4.

We set the termination condition when the MSPSO algorithm can successfully find all the global minima within 99.99% accuracy. The program terminates only when all the 18 function values of the global minima found by all

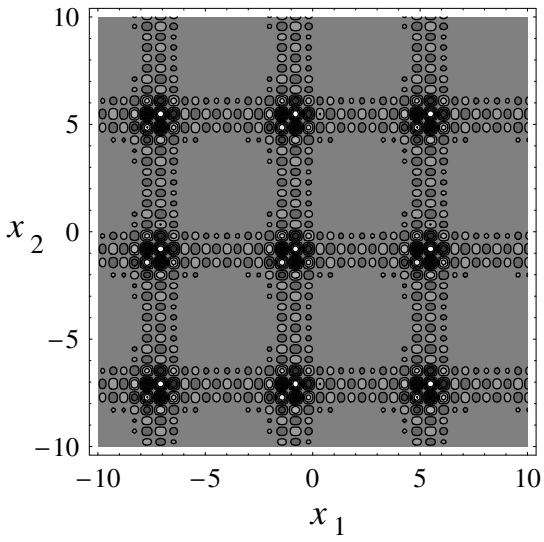

Fig. 4. Contour plot of the Shubert function. Because of the course resolution, 760 local minima are not distinguished. However, 9 clusters of global minima (dark area) which is separated by 5.65 are clearly distinguished. These 9 clusters consist of two global minima and two local minima that are separated only by a small distance 0.98. The exact location of 18 global minima are listed in Tab. I

TABLE I

THE LOCATIONS OF THE 18 GLOBAL MINIMA OF TWO-DIMENSIONAL SHUBERT FUNCTION. THE FUNCTION VALUE AT THOSE GLOBAL MINIMA IS $f_{\rm global} = -186.73$.

Number i	x_1	x_2
1	-7.7083	5.4828
2	-7.0835	4.8582
3	-1.4252	5.4828
4	-0.8003	4.8582
5	4.8582	5.4828
6	5.4828	4.8582
7	-7.7083	-0.8003
8	-7.0835	-1.4252
9	-1.4252	-0.8003
10	-0.8003	-1.4252
11	4.8582	-0.8003
12	5.4828	-1.4252
13	-7.7083	-7.0835
14	-7.0835	-7.7083
15	-1.4252	-7.0835
16	-0.8003	-7.7083
17	4.8582	-7.0835
18	5.4828	-7.7083

TABLE II

THE PARAMETERS USED FOR THE CONSTRICTION-FACTOR VARIANT (MSPSO-C) AND THE INERTIA-WEIGHT VARIANT (MSPSO-I) OF THE MULTI-SPECIES PARTICLE SWARM OPTIMIZATION.

Prameters	MSPSO-c	MSPSO-i
Population N	800	800
Coefficient c_1	2.05	2.05
Coefficient c_2	2.05	2.05
Maximum iteration (MAX)	150	150
Inertia weight ($w_{\rm int}$)	-	1.0
Inertia weight ($w_{\rm fin}$)	-	0.0
Constriction factor (χ)	0.729	-
Maximum velocity ($\mathrm{V}_{\max}$)	0.5-10	0.5-10
Species radisu (σ)	0.5-10	0.5-10

the species seeds are within 99.99% accuracy of the know function value $f_{\rm global} = 186.73$ of the global minima or when the maximum iteration MAX is reached. The significant figures of five digits are guaranteed for all the (multiple) global minima. Therefore the function values of all the 18 global minima are determined up to -186.73. This condition is slightly different and more stringent than others [16]. For example, Li *et al.* [16] used the solution acceptance threshold r_f to identify multiple global optima. However, the absolute accuracy of the global minima and the exact termination condition of their algorithm are not reported. Our termination condition is different from Li *et al* [16], but is easy to implement and is accurate enough to assess the performance of the algorithm.

The other algorithm parameters are: the maximum velocity $\mathrm{V}_{\max}$, the species radius σ, the two acceleration coefficients c_1 and c_2, the population N and the maximum number of iteration MAX. In addition, the initial and the final inertia weight $w_{\rm int}$, $w_{\rm fin}$ and the annealing schedule must be specified for MSPSO-i. Based on the detailed study of the algorithm parameters for MSPSO-i [7], we set those parameters such that the performance of the algorithm becomes optimum. Those parameters are listed in Tab. II

The necessary population to find out the solution of the single-global-minimum problems seems empirically to be 30 to 40 [2]. Therefore we use the population $N = 800$ because we have to find 18 global minima of the two-dimensional Shubert function. We use population 800 slightly larger than 40×18=720 because some particles may be trapped by local minima. The maximum number of iteration is set to MAX=150 from the previous experience [7]. For the two coefficients c_1 and c_2, we set the standard values $c_1 = c_2 = 2.05$ [9], [21] (Tab. II). The linear annealing schedule (3) with $w_{\rm int} = 1.0$ and $w_{\rm fin} = 0.0$ is used for MSPSO-i. Instead of tuning the final inertia weight $w_{\rm fin}$, we adjust the maximum iteration MAX and set MAX=150 in the previous paper [7]. Using these algorithm parameters in Tab. II, we will compare the performance of MSPSO-c and MSPSO-i by changing the maximum velocity $\mathrm{V}_{\max}$ and the species radius σ.

a) Effects of Maximum Velocity: The maximum velocity $\mathrm{V}_{\max}$ is usually set to the range of the variables [10]. Such a large $\mathrm{V}_{\max}$=10 in this case prefers a global search, and may not be effective because particles will easily move out from one species around one optimum to another which is separated only by 0.98. Similarly, too small $\mathrm{V}_{\max}$ will not be effective because information exchange between species is prohibited. In fact, the performance of MSPSO-i is shown to depend strongly on the value of $\mathrm{V}_{\max}$ used [7].

Table III compares the performance of MSPSO-c with that of MSPSO-i for different maximum velocity $\mathrm{V}_{\max}$. The success rate ("Success rate") and the average numbers of iterations ("Average iteration") necessary to reach all 18 global minima are compared for several values of the maximum velocity $\mathrm{V}_{\max}$. The averages are for 10 independent samples.

TABLE III

THE SUCCESS RATE ("SUCCESS RATE") AND THE AVERAGE NUMBERS OF ITERATIONS ("AVERAGE ITERATION") NECESSARY TO REACH ALL 18 GLOBAL MINIMA AND THEIR STANDARD DEVIATION IN PARENTHESIS (STD) FOR VARIOUS V_{max}. THE STANDARD DEVIATION STD=0.0 MEANS THAT THERE IS ONLY ONE SAMPLE.

V_{max}	MSPSO-c Success rate	Average Iteration (STD)	MSPSO-i Success rate	Average Iteration (STD)
0.5	0/10	- (-)	1/10	77.0 (-)
1.0	1/10	146.0 (0.0)	10/10	79.5 (5.1)
3.0	10/10	79.0 (28.5)	10/10	79.7 (3.7)
5.0	8/10	70.25 (12.2)	6/10	104.33 (10.2)
7.0	8/10	69.0 (7.7)	9/10	112.7 (9.7)
10.0	9/10	80.8 (14.7)	10/10	122.3 (7.3)

TABLE IV

THE AVERAGE NUMBER OF THE GLOBAL MINIMA FOUND ("GLOBAL MINIMA"), THE AVERAGE NUMBERS OF SPECIES FORMED ("SPECIES") AFTER MAX=150 ITERATIONS AND THEIR STANDARD DEVIATION IN PARENTHESIS (STD) FOR VARIOUS σ. THE STANDARD DEVIATION STD=0.0 MEANS THAT ALL SAMPLE HAS THE SAME VALUE AND THE STANDARD DEVIATION IS EXACTLY ZERO.

σ	MSPSO-c global minima	species	MSPSO-i global minima	species
0.5	17.8 (0.4)	392.5 (6.0)	17.6 (0.8)	360.9 (7.5)
1.0	9.0 (0.0)	125.8 (4.8)	9.0 (0.0)	107.4 (3.6)
3.0	9.0 (0.0)	23.6 (2.4)	9.0 (0.0)	23.8 (2.2)
5.0	9.0 (0.0)	9.2 (0.4)	9.0 (0.0)	9.2 (0.4)
10.0	3.0 (0.8)	3.8 (0.6)	3.2 (0.9)	4.0 (0.0)

The iteration is terminated when all the 18 global minima $f_{\text{global}} = 186.73$ is reached within the accuracy of five digits. The population is fixed to 800 and the species radius is fixed to $\sigma = 0.8$ and the other algorithm parameters are listed in Tab. II.

As intuitively expected, $V_{max} = 0.5$ which is smaller than 0.98 cannot locate 18 global optima. A further increase of the velocity from 1.0 degrades the performance for MSPSO-i. Since the minimum distance between two clusters is $\sim$5.65, the maximum velocity V_{max}=5 is not effective because particles around one cluster are easily moved out to the next cluster (success rate drops to 6/10 for MSPSO-i). Therefore, the performance of the MSPSO-i depends sensitively on the magnitude of maximum velocity V_{max}, while that of MSPSO-c is relatively insensitive to V_{max}. Even the simplest choice [2], [9] V_{max}= X_{max}=10 gives reasonably fast convergence. Although we used periodic boundary condition to make the search space a torus rather than an isolated square field ($-10 \leq x_1, x_2 \leq 10$), it is necessary to specify the maximum velocity V_{max} as MSPSO cannot locate all 18 solutions unless we specify the maximum velocity and limit the velocity of particles to increase infinitely.

b) Effects of species radius: In order to distinguish all 18 global minima, the species radius σ is the most important parameter. Since all 18 solutions are classified into 9 clusters, which are separated by a minimum distance 5.65, while the minimum distance of two solutions within the cluster is as small as 0.98, the species radius smaller than σ =0.98 with enough population will be necessary to locate all 18 solutions.

Table IV compares the performance of MSPSO-c with that of MSPSO-i for different species radius σ. The average number of the global minima found ("global minima") and the average numbers of species formed ("species") after MAX=150 iteration averaged over 10 samples for several values of the species radius σ for both MSPSO-c and MSPSO-i are compared. The population is fixed to 800 and the maximum velocity is fixed to $V_{max} = 3.0$ for MSPSO-c and $V_{max} = 1.0$ for MSPSO-i which seem to give the best performance from Tab. III.

The radius larger than 0.98 will make it difficult for both MSPSO-c and MSPSO-i to distinguish two solutions within a cluster, and both MSPSO-c and MSPSO-i will be expected to locate only 9 clusters or 9 minima. A further increase of the radius larger than σ =5.65 will make it difficult to distinguish even two clusters and our algorithm will be expected to fail to locate even 9 clusters. Table IV confirms our expectations. The performances of both MSPSO-c and MSPSO-i are comparable. Therefore, smaller species radius σ is necessary to locate all the global minima for both MSPSO-c and MSPSO-i.

From this exercise, we learn that the species radius σ is the most important parameter of both MSPOS-c and MSPSO-i for the hard optimization problem. It is certainly desirable that the species radius can be determined in advance without knowing the details of the objective function. Of course such knowledge about the species radius is almost equivalent to know the location of the global minima and, therefore, is almost equivalent to solve the global optimization problem itself. Modestly, we may interpret the species radius as the desired resolution or the desired quality of the solution.

Table IV also suggests the possibility of locating not only all the global minima but all the local minima. In fact, by using the small species radius $\sigma = 0.5$ and large population $N = 800$, whole population self-organized into as large as nearly 400 species. Since those species are centered at local minima, it will be possible to locate all 760 local minima of two-dimensional Shubert function by using sufficiently small species radius σ and sufficiently large population N.

c) Comparison to Multi-Start Standard PSO : It is probably fair to conduct experiments using the standard PSO with constriction factor using the multi-start technique. We use the standard PSO with constriction factor variant whose parameters are the same as those used by MSPCO-c in Tab. II

TABLE V

THE NUMBER OF SOLUTION FOUND BY STANDARD PSO WITH CONSTRICTION FACTOR USING MULTI-START TECHNIQUE. THE SOLUTION NUMBERS NOT FOUND (SEE TAB. I), THE NUMBER OF TRIAL ALLOWED FOR TOTAL 6000 ITERATIONS, AND THE AVERAGE ITERATION AND ITS STANDARD DEVIATION (STD) TO REACH THE SINGLE GLOBAL MINIMUM ARE ALSO TABULATED.

V_{max}	Number of Solution found	Soluiton not found (see Tab. I)	Number of Trial Allowed	Average Iteration (STD)
0.5	18/18	-	118	50.9 (14.4)
1.0	18/18	-	120	50.1 (10.9)
3.0	18/18	-	110	54.7 (12.7)
5.0	16/18	13,15	100	60.3 (14.5)
7.0	18/18	-	87	69.2 (15.4)
10.0	15/18	1,2,13	64	94.6 (124.0)

except the smaller population N=20 is used since we are searching for a single optimum.

Table V shows the results of experiment where the standard PSO with population N=20 runs for 6000 iteration. The number of function evaluation is, therefore, 20×6000=120000 which is equivalent to the number used by MSPSO-c with population N=800 and maximum iteration MAX=150 (800×150=120000). Table V clearly shows that the standard PSO does not necessarily find all 18 solutions when larger maximum velocity V_{max} is used. The average number of iteration required to reach single solution for standard PSO is slightly smaller than those cited in Tabs. III and IV, which is apparently due to the overhead of grouping the swarm into multiple species in MSPSO-c.

d) n-Dimensional Shubert Function : So far we have used two-dimensional Shubert function to demonstrate the ability and possibility of speciation using our MSPSO. It is possible to generalize our MSPSO in order to handle generalized *n*-dimensional Shubert functions which is defined by

$$f(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \sum_{j=1}^{5} j \cos\left[(j+1)x_i + j\right]. \quad (12)$$

It is conjectured [16] that the generalized Shubert function has $n \cdot 3^n$ global minima. Therefore 3-dimensional Shubert function has $3 \cdot 3^3 = 81$, and 4-dimensional one has $4 \cdot 3^4 = 324$ global minima. Then the population necessary to locate those global minima would be $(800/18) \times 81 = 3600$ for 3-dimensional and $(800/18) \times 324 = 14400$ for 4-dimensional Shubert function. Such a large population requires prohibitedly long CPU time and, therefore, the global optimization would be practically impossible.

In order to estimate the scalability of our MSPSO, we have generalized our MSPSO-c and optimized three-dimensional Shubert function using population N=3600. We carry out

TABLE VI

THE NUMBER OF SOLUTIONS FOUND BY OUR MSPOS FOR 3-DIMENSIONAL SHUBERT FUNCTION. IN ALL 10 SAMPLES, MSPSO COULD NOT LOCATE 81 GLOBAL MINIMA.

MAX	Numbers of solutions found in 10 samples
300	52, 62, 53, 51, 56, 49, 52, 53, 49, 59
2000	53, 55, 52, 58, 64, 55, 58, 57, 55, 48

10 independent trials using two sets of maximum iteration MAX=300 and 2000. The other algorithm parameters used are the species radius $\sigma = 0.8$ and the maximum velocity V_{max}=3.0 that seems to give the best performance in Tabs. III and IV. The other algorithm parameters are the same as those in Tab. II. Our preliminary results summarized in Tab. VI shows that our MSPSO could not locate all 81 global minima. Certainly, it is necessary to try finer tuning of algorithm parameter and/or to improve further the speciation technique in our MSPSO.

IV. CONCLUSIONS

In this work, we have extended the original particle swarm optimizer by including speciation and proposed the multi-species particle swarm (MSPSO) optimizer of constriction-factor variant (MSPSO-c) in addition to the previously developed inertia-weight variant (MSPSO-i). Whole population is partitioned into many species, which appear and disappear dynamically. Each species explore different optima in the search space independently by sharing information among the members of the species. Furthermore the immigration of particles from one species to another assists the information exchange between species. It should be noted that the particle swarm is self-organized into a number of different species whose numbe is *not* specified in advance by hand. Therefore, it is unnecessary to specify the number of global optima to be found in our algorithm.

We have compared the performance of MSPSO-c with that of previous MSPSO-i using the notoriously hard two-dimensional Shubert function. We have found that both MSPSO-c and MSPSO-i can successfully locate all the global minima simultaneously by tuning their algorithm parameters. The performance of MSPSO-c and that of MSPSO-i are almost comparable, which suggest that MSPSO-c is simpler to implement and easier to use than MSPSO-i since the former is free from the scheduling of inertia-weight annealing.

There have been several attempts to extend PSO to locate all the global optima [21], [22], [23]. However, some of these algorithms were based on the sequential search [20], and search for the global minima one after another by sequentially modifying the objective function [21], [22] and are tested only in much simpler benchmark functions. While our MSPSO based on the speciation of Li *et al.* [16] can search for all the global minima in parallel and is successful in locating all the global minima of much harder two-dimensional Shubert function. Very recently, an extension of PSO algorithm similar to ours which also adopts the

speciation of Li *et al.* [16] has been proposed by Parrott and Li [24]. They, however, considered only the dynamic multimodal problems in which the location of global optima changes dynamically. They also introduced the maximum species population to prevent the overcrowding of each species. Since they did not test their algorithm for the static problem, it is difficult to compare the performance of their algorithm with ours.

In conclusion, we have confirmed the effectiveness of multi-species particle swarm optimizer (MSPSO). It is found that the constriction variant (MSPSO-c) seems more effective than the previous inertia-weight variant (MSPSO-i). Our MSPSO seems to be the only one among various variants of PSO that can successfully used to solve such a complex problem of locating all 18 solutions of a two-dimensional Shubert function. Extensions to the dynamic environments and the higher dimensional problems or to locating all the local minima are left for the future investigations.

REFERENCES

[1] Z. Michalewicz, D. B. Fogel, *How to Solve It: Modern Heuristics,* Springer-Verlag, Berlin Heidelberg New York, 2000.

[2] J. Kennedy, R. C. Eberhart, *Swarm Intelligence,* Morgan Kaufmann, San Francisco, 2001.

[3] J. Kennedy, R. C. Eberhart, "Particle swarm optimization," *Proc. IEEE International Conference on Neural Networks*, IEEE Press, Piscataway NJ, 1995, pp. 1942–1945.

[4] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance difference," in *Evolutionary Programming V, Lecture Notes in Computer Science*, vol.1498, Edited by V.S. Porto et al., Springer-Verlag, Berlin, 1998, pp. 601–610.

[5] M. Iwamatsu, "Comparison of particle swarm and evolutionary programming as the global conformation optimizer of clusters," *International Journal of Modern Physics C*, vol. 16, No. 4, 2005, pp. 591–606.

[6] I. Parmee, "Exploring the design potential of evolutionary search, exploration and optimization," in *Evolutionary Design by Computers*, Edited by P.J. Bentley, Morgan Kaufmann, San Francisco, 1999, pp. 117–143.

[7] M. Iwamatsu, "Multi-species particle swarm optimizer for multimodal function optimization," *IEICE Transactions on Information and System* (English edition), vol. E89-D, No.3, 2006, pp. 1181-1187.

[8] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," *Proc. IEEE International Conference on Evolutionary Computation*, IEEE Press, Piscataway NJ, 1998, pp. 69–73.

[9] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation* vol.6, No.1, 2002, pp. 58–73.

[10] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming V, Lecture Notes in Computer Science*, vol.1498, Edited by V.S. Porto et al., Springer-Verlag, Berlin, 1998, pp. 591–600.

[11] R. C. Eberhart and Y. Shi, "Comparing inertia weight and constriction factors in particle swarm optimization," *Proc. 2002 IEEE Congress on Evolutionary Computation (CEC02)*, IEEE Press, Piscataway NJ, 2002, pp. 84–88.

[12] A. P. Engelbrecht, B. S. Masiye, and G. Pampará, "Niching ability of basic particle swarm optimization algorithms," *Proc. 2005 IEEE Swarm Intelligence Symposium (SIS05)*, 2005, pp. 397–400.

[13] K. Deb and W. M. Spears, "Speciation methods," in *Evolutionary Computation 2*, Edited by T. Bäck, D.B. Fogel, and Z. Michalewicz IOP Publishing, Bristol, 2000, pp. 93–100.

[14] C-K. Chow and H-T. Tsui, "Autonomous agent response learning by a multi-species particle swarm optimization," *Proc. IEEE Congress on Evolutionary Computation (CEC04)*, IEEE Press, Piscataway NJ, 2004, pp. 778–785.

[15] M. Pelikan and D. E. Goldberg, "Genetic algorithms, clustering, and the breaking of symmetry," in *Parallel Problem Solving from Nature-PPSN VI, Lecture Notes in Computer Science*, vol.1917, Edited by M. Shoenauer et al., Springer-Verlag, Berlin, 2000, pp. 385–394.

[16] J-P. Li, M. E. Balaz, G. T. Parks, P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evolutionary Computation*, vol.10, No.3, 2002, pp. 207–234.

[17] J. Kennedy, "Small worlds and mega-minds, effects of neighborhood topology of particle swarm performance," *Proc. 1999 IEEE Congress on Evolutionary Computation (CEC99)*, IEEE Press, Piscataway NJ, 1999, pp. 1931–1938.

[18] J. Kennedy, R. Mendes, "Population structure and particle swarm performance," *Proc. 2002 IEEE Congress on Evolutionary Computation (CEC02)*, IEEE Press, Piscataway NJ, 2002, pp. 1671–1676.

[19] J. J. Liang and P. N. Sunganthan, "Dynamic multi-swarm particle swarm optimizer," *Proc. 2005 IEEE Swarm Intelligence Symposium (SIS05)*, 2005, pp. 124–129.

[20] D. Beasley, D.R. Bull, R.R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation* vol.1, No.2, 1993, pp. 101–125.

[21] K. E. Parsopoulos and M. N. Vrahatis, "Modification of the particle swarm optimizer for locating all the global minima," in *Artificial Neural Networks and Genetic Algorithms*, Edited by V. Kurova, et al., Berlin: Springer-Verlag, 2001, pp. 324–327.

[22] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm Optimization," *IEEE Transactions on Evolutionary Computation* vol.8, No.3, 2004, pp. 211–224.

[23] R. Brits, A.P. Engelbrecht, and F. van den Bergh, "Scalability of niche PSO," *Proc. 2003 IEEE Swarm Intelligence Symposium (SIS03)*, 2003, pp. 228–234.

[24] D. Parrott and X. Li, "A Particle swarm model for tracking multiple peaks in a dynamic environment using speciation," *Proc. 2004 IEEE Congress on Evolutionary Computation (CEC04)*, IEEE Press, Piscataway NJ, 2004, pp. 98–103.