An Analysis of Multiple Particle Swarm Optimizers with Inertia Weight for Multi-objective Optimization

Hong Zhang, *Member, IAENG*

***Abstract*—An improved particle swarm optimizer with inertia weight (PSOIWα) was applied to multi-objective optimization (MOO). For further improving its search performance, in this paper, we propose to use a cooperative PSO method called multiple particle swarm optimizers with inertia weight (MPSOIWα) to search. The crucial idea of the MPSOIWα, here, is to reinforce the search ability of the PSOIWα by the union's power of plural swarms, i.e. distributed processing. To demonstrate the search performance and effect of the proposal, computer experiments on a suite of 2-objective optimization problems are carried out by an aggregation-based manner. The resulting Pareto-optimal solution distributions corresponding to each given problem indicate that the linear weighted aggregation among the adopted three kinds of dynamic weighted aggregations is the most suitable for acquiring better search results. Throughout quantitative analysis to experimental data, we clarify the search characteristics and performance effect of the MPSOIWα contrast with that of the original PSOIW, PSOIWα, and MPSOIW.**

***Index Terms*—particle swarm optimization, cooperative PSO, swarm intelligence, hybrid search, multi-objective optimization, Pareto optimality, weighted sum method.**

I. Introduction

MULTI-objective optimization (MOO), also known as multi-criteria or multi-performance optimization, is the processing of optimizing simultaneously two and more conflicting objectives subject to certain constraints [5], [7], [30]. Since many practical problems are involved in MOO, which can be mainly found in different domains of science, technology, industry, finance, automobile design, aeronautical engineering etc. [1], [9], [31], how to efficiently deal with MOO becomes a live issue, and is centered on the development of the treatment technique.

As to be generally known, traditional optimization methods such as many gradient-based methods are difficult to treat with the true multi-objective case, because they were not designed to find plural optimal solutions. In effect, a MOO problem has to be converted to a single-objective optimization (SOO) one before the problem-solving. In this situation, the search generates a single optimal solution by each run of the optimization, and that the obtained optimal solutions are highly sensitive to the weight vector used in the converting process. Nevertheless, the issue of adopting the way is how to ensure that the obtained every optimal solution satisfies *Pareto* optimality.

Since the methods of evolutionary computation (EC) can generate plural candidate solutions, i.e. individuals in a population, it seems naturally to use them in coping with MOO for finding a *Pareto*-optimal solution set simultaneously. According to the distinguishing features of group search, the use of EC methods for dealing with MOO problems has significantly grown over the last decade, and has achieved valuation results [8], [11], [21].

Particle swarm optimization (PSO), which was proposed by Kennedy and Eberhart in 1995, is an adaptive, stochastic, and population-based optimization technique [19]. Based on the three unique features: information exchange, intrinsic memory, and directional search, the technique has higher latent search ability in optimization compared to other methods of EC such as genetic algorithms and genetic programming [25], [26], [36], [37]. Especially, in recent years, a large number of studies and investigations on cooperative PSO[a] in relation to symbiosis, group behavior, and synergy are in the researcher's spotlight. Consequently, various methods of cooperative PSO, e.g. hybrid PSO, multi-layer PSO, multiple PSO with decision-making strategy etc. were successively published [3], [12], [23], [37], [41].

Compared with those methods running a single particle swarm, different attempts and strategies can be clearly perfected by implementing multiple particle swarms for more efficiently finding an optimal solution or near-optimal solutions [4], [18], [23], [39]. Owing to the plain advantage, utilizing the techniques of group searching, parallel and intelligent processing has become one of extremely important approaches to optimization, and a lot of publications and reports have been shown that the methods of cooperative PSO have better adaptability and higher search performance than ones of uncooperative PSO in dealing with various optimization and practical problems [14], [24].

An improved particle swarm optimizer with inertia weight (PSOIWα) was applied to MOO [43], which can provide a basic structure and framework for testing different methods of PSO. In order to further upgrade the search performance of the PSOIWα, in this paper, we propose to use a method of cooperative PSO called multiple particle swarm optimizers with inertia weight (MPSOIWα) to search. The crucial idea of the MPSOIWα, here, is to reinforce the search ability of the PSOIWα by the union's power of plural swarms,

Manuscript received April 11, 2012.

It was originally presented at IMECS 2012 [44], which is a substantially extended version. This work was supported in part by Grant-in-Aid Scientific Research (C)(22500132) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

H. Zhang is with the Department of Brain Science and Engineering, Graduate School of Life Science & Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan. phone/fax: +81-93-695-6112; e-mail: zhang@brain.kyutech.ac.jp.

[a]Cooperative PSO is generally considered as multiple swarms (or subswarms) searching for a solution (serially or in parallel) and exchanging some information during the search according to some communication strategies.

i.e. distributed processing. In spite of the search behaviors and performance of various PSO methods in MOO with some fitness assignment manners such as criterion-based one or dominance-based one were studied and investigated [32], [33], however, there is a lack of detailed analysis and sufficient results for systematically solving MOO problems by the methods of cooperative PSO using an aggregation-based manner, and analyzing the potential characteristics in details from the obtained experimental results [7], [20].

To demonstrate the effectiveness and performance effect of the MPSOIWα, computer experiments on a suite of 2-objective optimization problems are carried out by a well-known weighted sum method. For interpreting the information treatment and search effect of the method, we show the distributions of the obtained *Pareto*-optimal solutions corresponding to each given problem by respectively using three kinds of dynamic weighted aggregations (i.e. linear weighted one, bang-bang weighted one, and sinusoidal weighted one[b]), point out that which one of them is the most suitable for acquiring good search results to the given MOO problems, and clarify the search characteristics, performance effect, and computation cost of the MPSOIWα compared with the search performance of the original PSOIW, PSOIWα, and MPSOIW.

The rest of the paper is organized as follows. Section II briefly introduces some basic concepts and definitions for dealing with a general MOO problem. Section III describes the search methods of the PSOIW, PSOIWα, and MPSOIWα. Section IV provides the obtained experimental results corresponding to a suite of 2-objective optimization problems, and analyzes the potential characteristics of the MPSOIWα in the technical details. Finally, the concluding remarks appear in Section V.

II. Basic Concepts

For finely explaining how to treat with MOO by a fitness assignment manner, in this section, some basic concepts and definitions on a general MOO problem, *Pareto* optimality, front distance, a weighted sum method, and three kinds of dynamic weighted aggregations are described.

A. MOO Problem

Without loss of generality, a MOO problem can be expressed as follows.

$$\begin{array}{ll} \underset{\vec{x}}{Minimize} & \left(f_1(\vec{x}), f_2(\vec{x}), \cdots, f_I(\vec{x})\right)^T \\ s.t. & g_j(\vec{x}) \geq 0,\ j = 1, 2, \cdots, J \\ & h_m(\vec{x}) = 0,\ m = 1, 2, \cdots, M \\ & x_n \in [x_{nl}, x_{nu}],\ \ n \in (1, 2, \cdots, N) \end{array} \tag{1}$$

where $f_i(\vec{x})$ is the *i*-th objective or criterion, $g_j(\vec{x})$ is the *j*-th inequality constraint, $h_m(\vec{x})$ is the *m*-th equality constraint, $\vec{x} = (x_1, x_2, \cdots, x_N)^T \in \Re^N$ (= Ω search space) is the vector of decision variable, x_{nl} and x_{nu} are the superior boundary value and the inferior boundary value of each component x_n of the vector $\vec{x}$, respectively.

Due to the given condition of $I \geq 2$, the I-objectives may be conflicting with each other. Under this circumstance, it is difficult to obtain the global optimum corresponding to each objective by traditional optimization methods such as Newton's method, steepest descent method, and BFGS method etc. at the same time. Consequently, the major aim of handling the MOO problem is effectively to achieve a set of solutions that satisfy *Pareto* optimality for improvement of mental capacity and interpretation of decision making.

[b]Many researchers call sinusoidal weighted aggregation (SWA) as dynamic weighted aggregation (DWA).

B. Pareto Optimality

A solution $\vec{x}^* \in \Omega$ is said to be *Pareto*-optimal solution if and only if there does not exist another solution $\vec{x} \in \Omega$ so that $f_i(\vec{x})$ is dominated by $f_i(\vec{x}^*)$. The formula of the above relationship is expressed as

$$f_i(\vec{x}) \not\leq f_i(\vec{x}^*)\ \ \forall i \in I \quad iif \quad f_i(\vec{x}) \not< f_i(\vec{x}^*)\ \ \exists i \in I \tag{2}$$

In other words, this definition says that $\vec{x}^*$ is a *Pareto*-optimal solution if there exists no feasible solution (vector) $\vec{x}$ which would decrease some criteria without causing a simultaneous increase in at least one other criterion.

Furthermore, all of the *Pareto*-optimal solutions for a given MOO problem constitute the *Pareto*-optimal solution set (P^*), or the *Pareto*-optimal front (PF).

C. Front Distance

Front distance is a metric for checking how far the elements are in the set of non-dominated solutions found from those in the true *Pareto*-optimal solution set. It directly reflect the estimation accuracy of the optimizer used. Concretely, the definition of front distance (FD) is expressed as

$$FD = \frac{1}{Q}\sqrt{\sum_{q=1}^{Q} d_q^2},\ \ d_q = f_i(\vec{x}_q^*) - f_i(\vec{x}_q^o),\ \ \forall i \in I \tag{3}$$

where Q is the number of the elements in the set of non-dominated solutions found, and d_q is the Euclidean distance (measured in objective space) between each of these obtained optimal solutions, $\vec{x}^o$, and the nearest member $\vec{x}^*$ of the *Pareto*-optimal solution set.

D. Cover Rate

Cover rate (*CR*) is an other metric for checking the coverage of the elements being in the set of non-dominated solutions found to the *Pareto*-optimal front. This is because the estimation accuracy is insufficiency to reveal the distribution status of the obtained optimal solutions and their possibility for dealing with the given problem.

Here, the formulation of CR is mathematically expressed as

$$CR = \frac{1}{I}\sum_{i=1}^{I} CR_i \tag{4}$$

where CR_i is the partial cover rate corresponding to the i-th objective, which is defined by

$$CR_i = \frac{\sum_{l=1}^{\Gamma} \gamma_l}{\Gamma} \tag{5}$$

where Γ is the number of dividing the i-th objective space which is from the minimum to the maximum of the fitness value, i.e. $[f_i(\vec{x})^{min}, f_i(\vec{x})^{max}]$, and $\gamma_l \in (0, 1)$ indicates the existence status of the obtained optimal solutions in the l-th subdivision for the i-th objective.

Since the divided number to a designated objective space is given by an experimenter, it goes without saying that CR is just a relative metric depending on the value of Γ.

E. Weighted Sum Method

There are some fitness assignment manners, for example, aggregation-based one, criterion-based one, and dominance-based one, which are used for MOO [8], [15]. It is well-known that a conventional weighted sum (CWS) method is a straightforward approach applied to deal with MOO problems [13]. Concretely, the different objectives are summed up to a single scalar F_s (i.e. criterion) with some prescribed weights as follows.

$$F_s(\vec{x}) = \sum_{i=1}^{I} c_i f_i(\vec{x}) \tag{6}$$

where $c_i (i = 1, 2, \cdots, I)$ is the i-th non-negative weight. During the optimization, generally, all of the weights are fixed by the constraint of $\sum_{i=1}^{I} c_i = 1$.

As usual, prior knowledge is also needed to specify appropriate weights for obtaining good solutions. However, the following issues exist: (1) Result depends on the used weights, (2) Some solutions cannot be reached, (3) Multiple runs of the optimizer are required in order to obtain the whole solutions. To thoroughly conquer the shortcoming of the CWS method, a dynamic weighted sum (DWS) method is often used to MOO in practice [17], [45].

The criterion F_d of the DWS method is expressed as

$$F_d(t, \vec{x}) = \sum_{i=1}^{I} c_i(t) f_i(\vec{x}) \tag{7}$$

where t is the time-step to search, and $c_i(t) \geq 0$ is the dynamic weight satisfying the constraint of $\sum_{i=1}^{I} c_i(t) = 1$ at time-step t.

In order to fully investigate the search effect of the DWS method, as an example, a 2-objective optimization problem is employed. Hence, the definitions of three kinds of the adopted dynamic weighted aggregations are expressed as follows.

- Linear weighted aggregation (LWA):

$$c_1^l(t) = mod\Big(\frac{t}{T}, 1\Big), \;\; c_2^l(t) = 1 - c_1^l(t)$$

- Bang-bang weighted aggregation (BWA):

$$c_1^b(t) = \frac{sign\big(sin(2\pi t/T)\big) + 1}{2}, \;\; c_2^b(t) = 1 - c_1^b(t)$$

- Sinusoidal weighted aggregation (SWA):

$$c_1^s(t) = \Big|sin\big(\frac{\pi t}{T}\big)\Big|, \;\; c_2^s(t) = 1 - c_1^s(t)$$

where T is a period of the variable weights in the above equations.

For the sake of detailed observation, Fig. 1 illustrates the change characteristics of the above mentioned dynamic weighted aggregations. We can clearly see that the weight values of the LWA or SWA smoothly change with the growth of time-step t in the period T=20. Contrast to this case, the weight values of the BWA change discontinuously. Moreover, such abrupt movement of them is just only one time in the same period. It is considered that different characteristics and process of variations in the criteria $F_d(t, \vec{x})$ with the growth of time-step t will greatly reflect the search performance and search effect of using each weighted aggregation corresponding to a given MOO problem [16].

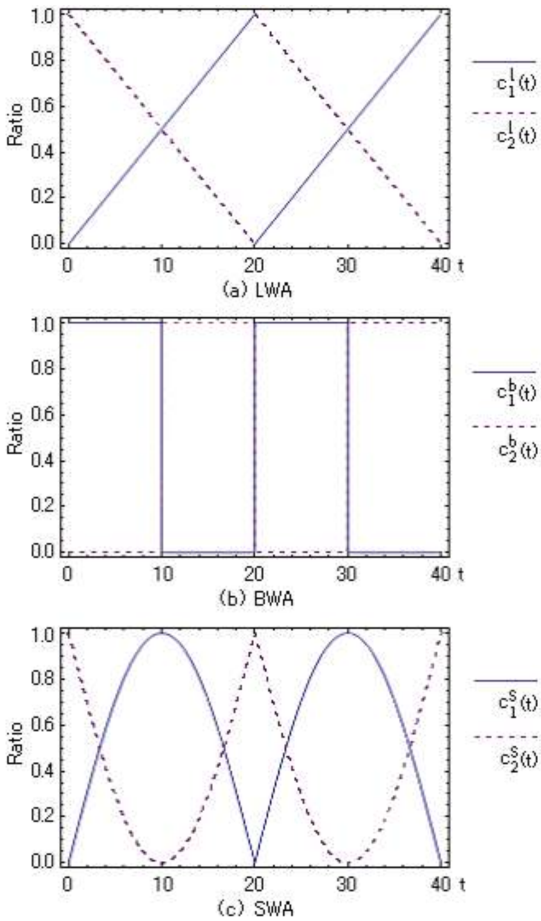

Fig. 1. The change characteristics of three kinds of the adopted dynamic weighted aggregations under the condition of period $T = 20$. (a) Linear weighted aggregation, (b) Bang-bang weighted aggregation, (c) Sinusoidal weighted aggregation.

III. Search Methods

For the convenience of the following description to the used every optimizer, let the search space be N-dimensional, the number of particles of a swarm be P, the position of the i-th particle be $\vec{x}^i = (x_1^i, x_2^i, \cdots, x_N^i)^T \in \Omega$, and its velocity be $\vec{v}^i = (v_1^i, v_2^i, \cdots, v_N^i)^T \in \Omega$, respectively.

A. The PSOIW

To overcome the weakness of the original PSO [2], [6] in convergence, Shi et al. modified the update rule of the i-th particle's velocity by constant reduction of the inertia coefficient over time-step [10], [27]. Concretely, the formulation of the particle swarm optimizer with inertia weight (PSOIW) is defined as

$$\begin{cases} \vec{x}_{k+1}^i = \vec{x}_k^i + \vec{v}_{k+1}^i \\ \vec{v}_{k+1}^i = w(k)\, \vec{v}_k^i + w_1 \vec{r}_1 \otimes (\vec{p}_k^i - \vec{x}_k^i) + w_2 \vec{r}_2 \otimes (\vec{q}_k - \vec{x}_k^i) \end{cases} \tag{8}$$

where w_1 and w_2 are coefficients for individual confidence and swarm confidence, respectively. $\vec{r}_1, \vec{r}_2 \in \Re^N$ are two random vectors, each element of which is uniformly distributed on the interval $[0,1]$, and the symbol $\otimes$ is an element-wise operator for vector multiplication. $\vec{p}_k^i$ $(=arg \max_{k=1,2,\cdots} \{g(\vec{x}_k^i)\}$, where $g(\vec{x}_k^i)$ is the criterion value of the i-th particle at time-step k) is the local best position of the i-th particle up to now, $\vec{q}_k (=arg \max_{i=1,2,\cdots} \{g(\vec{p}_k^i)\})$ is the global best position among the whole particles at time-step k. $w(k)$ is the following variable inertia weight which is linearly reduced from a starting value w_s to a terminal value w_e with the increment of time-step k.

$$w(k) = w_s + \frac{w_e - w_s}{K} \times k \tag{9}$$

where K is the number of iteration for the PSOIW run. In the original PSOIW, two terminal values, w_s and w_e, are set to 0.9 and 0.4, respectively, and $w_1 = w_2 = 2.0$ are used.

This is a simple and useful way for conquering the weak convergence and enhancing the solution accuracy of the PSO. However, the shortcoming of the PSOIW search is easily to fall into a local minimum and hardly to escape from that solution in dealing with multimodal problems because the terminal value w_e is set to small.

B. The PSOIWα

As a matter of common knowledge, random search methods are the simplest ones of stochastic optimization with non-directional search, and are effective in handling many complex optimization problems [28], [29].

For obtaining better search results, we introduce the LRS [40] into the PSOIW to create a hybrid search optimizer (called PSOIWα). Implementing the PSOIWα, here, is to enable a particle swarm search escapes from local minimum sooner for efficiently finding an optimal solution or near-optimal solutions. Concretely, the PSOIWα's procedure is implemented as follows.

- $step$-1: Give the terminating condition U of the PSOIWα run, and set the counter $u = 1$.
- $step$-2: Implement PSOIW and determine the best solution $\vec{q}_k$ at time-step k, and set $\vec{q}_{now} = \vec{q}_k$.
- $step$-3: Generate a random data, $\vec{z}_u \in \Re^N \sim N(0, \sigma^2)$ (where σ is a small positive value given by user, which determines the small limited space). Check whether $\vec{q}_k + \vec{z}_u \in \mathbf{\Omega}$ is satisfied or not. If $\vec{q}_k + \vec{z}_u \notin \mathbf{\Omega}$ then adjust $\vec{z}_u$ for moving $\vec{q}_k + \vec{z}_u$ to the nearest valid point within $\mathbf{\Omega}$. Set $\vec{q}_{new} = \vec{q}_k + \vec{z}_u$.
- $step$-4: If $g(\vec{q}_{new}) > g(\vec{q}_{now})$ then set $\vec{q}_{now} = \vec{q}_{new}$.
- $step$-5: Set $u = u + 1$. If $u \leq U$ then go to the $step$-2.
- $step$-6: Set $\vec{q}_k = \vec{q}_{now}$ to correct the solution found by the particle swarm at time-step k. Stop the search.

C. The MPSOIWα

For further improving the search ability of the above mentioned PSOIWα to MOO, we propose to use multiple particle swarm optimizers with inertial weight, called MPSOIWα, to search. As a matter of course, the most difference between the PSOIWα and MPSOIWα in composition is just to implement the plural PSOIWα ($S \geq 2$) in parallel for finding the most suitable solution or near-optimal solutions.

Concretely, the best solution of the s-th PSOIWα in the MPSOIWα run at time-step k is obtained by

$$\vec{q}_{k+1}^s = \begin{cases} \arg \max_{u=1,2,\cdots} \{g(\vec{q}_k^s + \vec{z}_u)\}, & if\ g(\vec{q}_k^s + \vec{z}_u) \geq g(\vec{q}_k^s) \\ \vec{q}_k^s, & otherwise \end{cases} \tag{10}$$

After implementing each PSOIWα in parallel, the best solution of the whole multi-swarm search at time-step k is determined by

$$\vec{x}_k^o = arg \max_{s=1,2,\cdots,S} \{g(\vec{q}_k^s)\} \tag{11}$$

Then put the best solution $\vec{x}_k^o$ into a solution set which is the storage memory of the multi-swarm search.

Owing to the use of the operations of parallel processing and maximum selection, it is self-explanatory the search ability of the MPSOIWα is superior to that of the PSOIWα. This is just the result by the use of union's power of plural swarms, i.e. each particle swarm cooperates with each other on the decision of the best solution.

For understanding how to deal with a MOO problem by a aggregation-based DWS method, Fig. 2 illustrates a flowchart of the MPSOIWα to show the operation processing and information control in the whole MOO process.

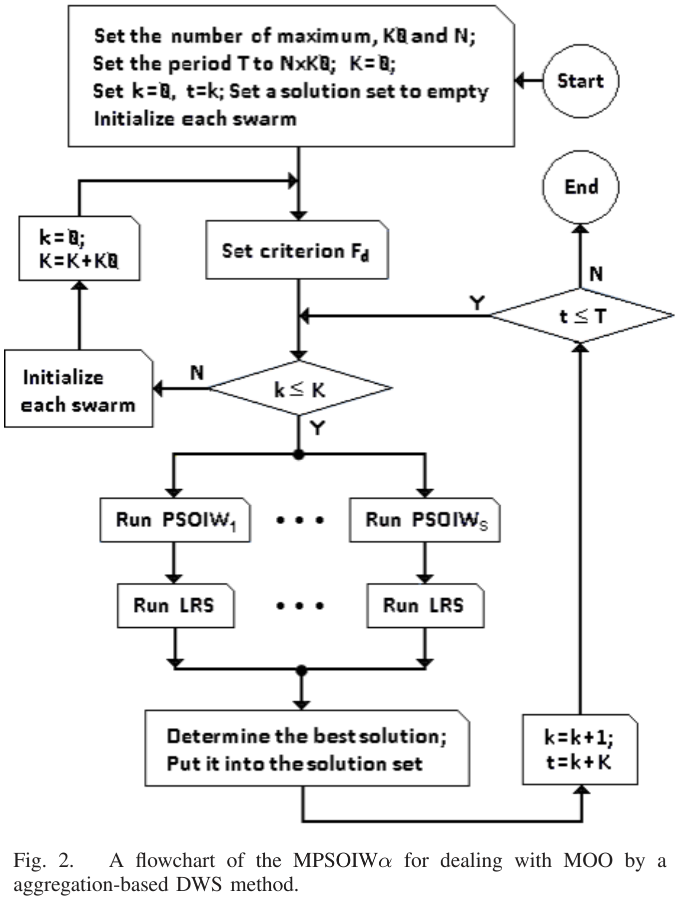

Fig. 2. A flowchart of the MPSOIWα for dealing with MOO by a aggregation-based DWS method.

Here, it is to be noted that if the LRS in the MPSOIWα is not implemented after each PSOIW run, the method will be called as MPSOIW.

IV. Computer Experiments

To facilitate comparison and analysis of the search performance of the proposed MPOSIWα, the suite of 2-objective

TABLE I
A SUITE OF 2-OBJECTIVE OPTIMIZATION PROBLEMS

problem	objective	search range
ZDT1	$f_{11}(\vec{x}) = x_1,\ g(\vec{x}) = 1 + \frac{9}{N-1}\sum_{n=2}^{N} x_n,\ f_{12}(\vec{x}) = g(\vec{x})\left(1 - \sqrt{\frac{f_{11}(\vec{x})}{g(\vec{x})}}\right)$	$\Omega \in [0,1]^N$
ZDT2	$f_{21}(\vec{x}) = x_1,\ f_{22}(\vec{x}) = g(\vec{x})\left(1 - \left(\frac{f_{21}(\vec{x})}{g(\vec{x})}\right)^2\right)$	$\Omega \in [0,1]^N$
ZDT3	$f_{31}(\vec{x}) = x_1,\ f_{32}(\vec{x}) = g(\vec{x})\left(1 - \sqrt{\frac{f_{31}(\vec{x})}{g(\vec{x})}} - \left(\frac{f_{31}(\vec{x})}{g(\vec{x})}\right) sin\left(10\pi f_{31}(\vec{x})\right)\right)$	$\Omega \in [0,1]^N$

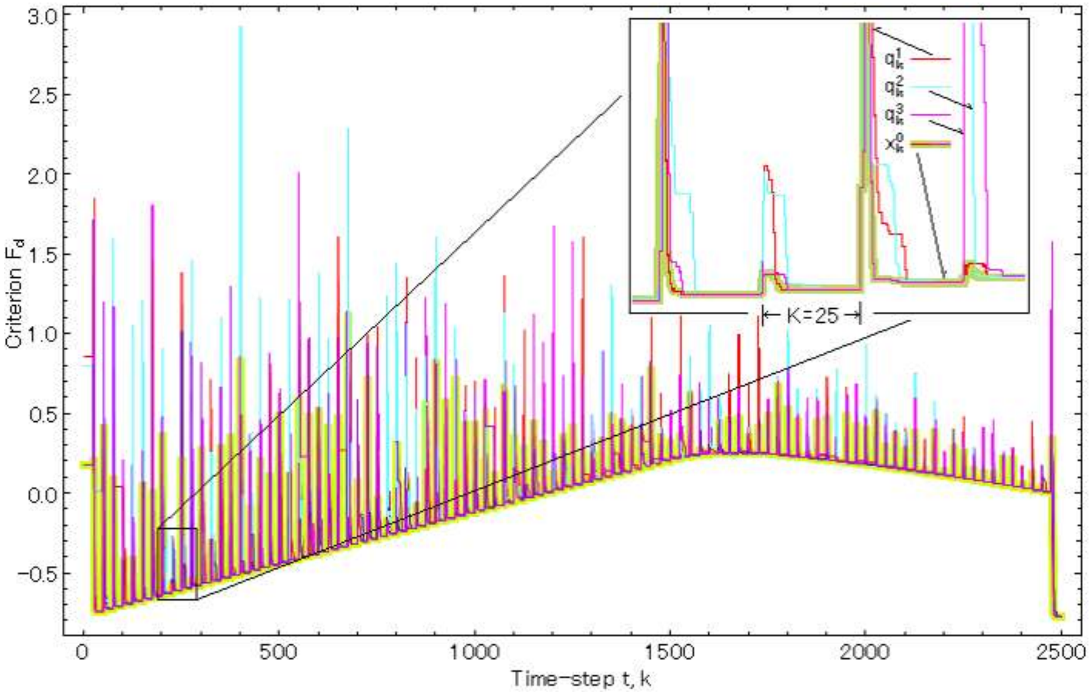

Fig. 3. The change of fitness values of the top-particles of three swarms in the search process of the MPSOIWα for the *ZDT3* problem by using the LWA.

optimization problems [46] in Table I is used in the next computer experiments. The characteristics of the *Pareto* fronts of these given benchmark problems include the convex (*ZDT1* problem), concave (*ZDT2* problem), and discontinuous multimodal (*ZDT3* problem), respectively.

Table II gives the major parameters of the MPSOIWα for solving the given problems in Table I. The choice of their values is referred to the results of some preliminary experiments, which can be satisfied to rapidly converge to a certain solution during a short search cycle.

TABLE II
MAJOR PARAMETERS OF THE MPSOIWα RUN

parameter	value
the number of particles, P	10
the number of iterations, K	25
the number of period, T	2500
the number of random points, U	10
the search range of the LRS, σ	0.1
the number of multiple particle swarms, S	3

For investigating the search process and situation of the MPSOIWα run in which how to deal with a MOO problem by using a dynamic weighted aggregation, as an example, Fig. 3 shows the change of fitness values of the top-particles of three swarms for dealing with the *ZDT3* problem by using the LWA.

According to the definition of MOO mentioned in Section II-A, the smaller the fitness values are, the better the obtained solutions are. We can see from Fig. 3 that the convergence of the MPSOIWα run is faster, and three particle swarms play complementary best fitness value with each other in the whole optimization process. The smooth variation of the best criterion (the best solution of the multi-swarm) suggests that the *Pareto*-optimal solutions can be continuously obtained during one short search cycle (K=25). The vibration of the best fitness found by each swarm occurs with the change of variable criterion. And the vibration range of the best fitness reflects the influence receiving from the change of the criterion F_d overall time-step. Needless to say, the movement features of the fitness in progress are not unique for the given different problems.

A. Performance Comparison

Due to observation, Fig. 4 shows the resulting solution distributions of the MPSOIWα and MPSOIW by using the LWA, BWA, and SWA, respectively. Based on the distinction of each solution distribution corresponding to these given problems, the analytical judgment can be described as follows.

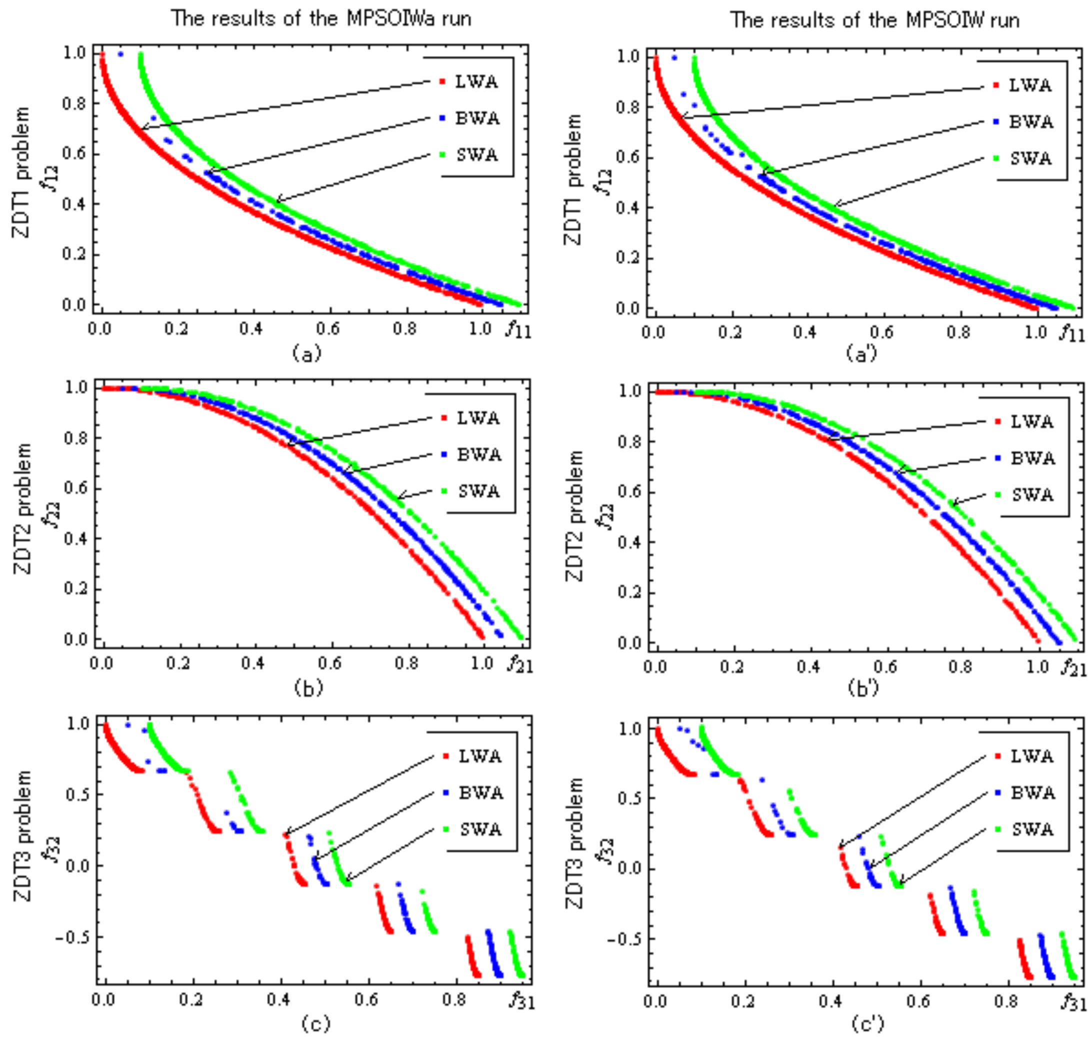

Fig. 4. Solution distributions of the MPSOIWα and MPSOIW by using the LWA (red-point), BWA (blue-point) and SWA (green-point), respectively. Notice: the distance between the experimental data sets for each subgraph is 0.05 (shift only in horizontal direction).

TABLE III
PERFORMANCE COMPARISON OF BOTH THE MPSOIWα AND MPSOIW BY USING THE LWA, BWA, AND SWA, RESPECTIVELY (Γ IS SET TO 100).

problem	aggregation	MPSOIWα			MPSOIW		
		solution	*FD*	*CR* (%)	solution	*FD*	*CR* (%)
ZDT1	LWA	**1254**	**2.234**$\times 10^{-8}$	**99.5**	1191	3.948×10^{-8}	**99.5**
	BWA	187	9.809×10^{-5}	52.0	227	1.107×10^{-4}	53.0
	SWA	988	4.511×10^{-8}	99.5	1016	7.355×10^{-8}	99.0
ZDT2	LWA	272	**1.198**$\times 10^{-8}$	**94.0**	**283**	1.992×10^{-7}	**94.0**
	BWA	259	3.692×10^{-7}	92.0	228	8.852×10^{-7}	91.5
	SWA	229	7.604×10^{-8}	93.5	219	3.381×10^{-7}	93.0
ZDT3	LWA	**1231**	**8.961**$\times 10^{-7}$	**46.0**	1107	9.245×10^{-7}	45.5
	BWA	396	1.655×10^{-4}	40.5	421	6.551×10^{-5}	40.0
	SWA	949	9.433×10^{-7}	42.5	1018	1.092×10^{-6}	42.0

The values in bold signify the best result for each given problem.

1) Regardless of the used methods either the MPSOIWα, or MPSOIW, and the characteristic of each given problems, the resulting features and solution distributions are nearly same.
2) Regardless of the used methods and the characteristics of the given problems, the conditions of solution distributions by using the BWA are worse than that by using the LWA or SWA special for the *ZDT1* and *ZDT3* problems.
3) In comparison with the solution distributions of using the LWA for both the *ZDT1* (convex) and *ZDT2* (concave) problems, the former is relatively in the higher density.

For quantitative analysis to the experimental results of the MPSOIWα and MPSOIW in Fig. 4, Table III gives the statistical data, i.e. the number of the obtained optimal solutions $\vec{x}^o$, and the corresponding *FD* and *CR* for each given problem.

By comparing with the performance indexes of two methods, the following marked features can be observed. Firstly, there is the most number of solutions obtained by using the LWA for the given problems even for the *ZDT2* one

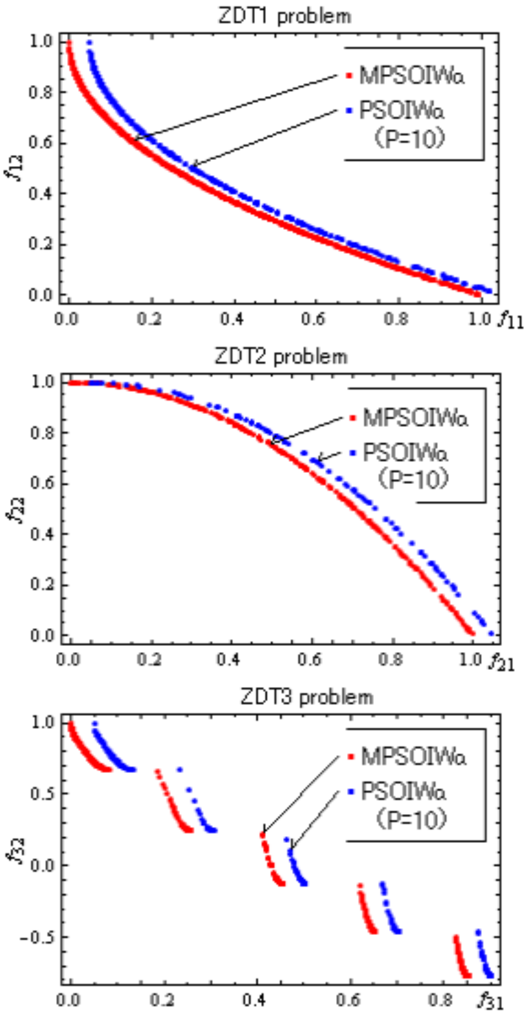

Fig. 5. The solution distributions of the MPSOIWα and PSOIWα ($P = 10$) by using the LWA. Notice: the distance between the experimental data sets for each subgraph is 0.05 (shift only in horizontal direction).

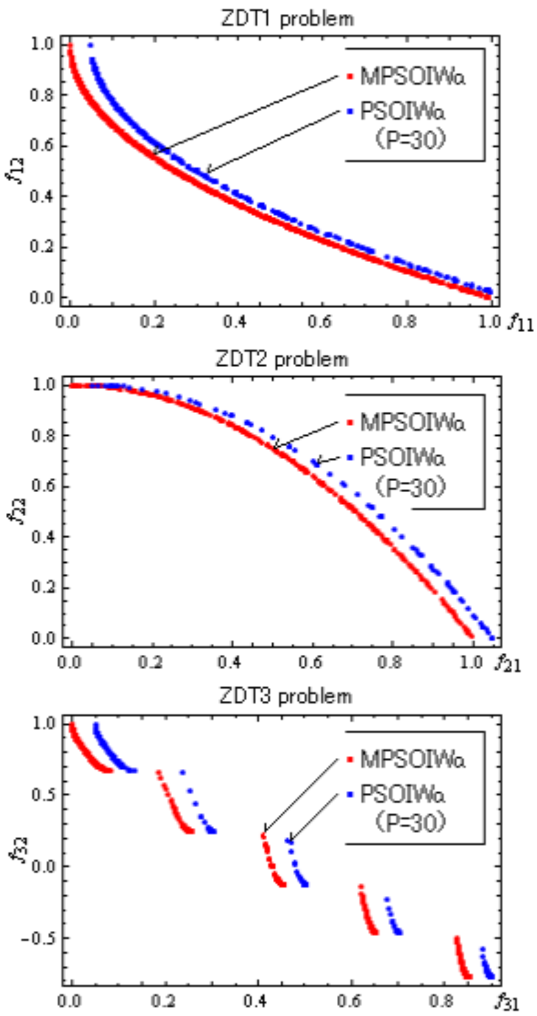

Fig. 6. The solution distributions of the MPSOIWα and PSOIWα ($P = 30$) by using the LWA. Notice: the distance between the experimental data sets for each subgraph is 0.05 (shift only in horizontal direction).

in where a large number of *Pareto*-optimal solutions are in unstable position [16]. Secondly, the solution accuracy of the MPSOIWα is superior to that of the MPSOIW for each given problem. Thirdly, the obtained results of using the LWA in *CR* index are the best than that of using BWA and SWA, respectively. Fourthly, the search performance of using the LWA is not only much better than that of using the BWA, but also is relatively better than that of using the SWA as a whole.

According to the above consideration, the effectiveness and search ability of the MPSOIWα are roughly demonstrated. Furthermore, better solution distribution and higher solution accuracy can be observed as well by using either the LWA or SWA. The obtained experimental results indicate that smooth change of their criteria with the growth of time-step t can make that the probability finding good solutions greatly goes up in the same period, $T = 2500$, as evidence.

Based on the above mentioned comparison and observation, the relationship of domination reflecting the search performance (SP) of the MPSOIWα by using each dynamic weighted aggregation is expressed as follows.

$$SP_{LWA} \succ SP_{SWA} \succ SP_{BWA}$$

The above relationship of dominance in SP plainly indicates that the uniform change of the weights can make the moving process of variable criterion to be equalization which raises the probability finding the *Pareto*-optimal solution to the maximum under the condition of implementing the same optimizer. Due to this reason and characteristics, more good solutions can be easily obtained during the short search cycle, $K = 25$.

B. Effect of Multi-swarm Search

For identifying the effect and characteristics of multi-swarm search, the following computer experiments on contrasting the different number of the particles used in the PSOIWα are carried out.

1) Unequal in Number of Particles: In this case, the number of particles used in the PSOIWα is just set to one-third of the total number of particles used in the MPSOIWα, $P = 10$.

As an example, Fig. 5 shows the resulting solution distributions of both the MPSOIWα and PSOIWα by using the LWA. We can clearly see that the density of solution distributions of the MPSOIWα are higher than that of the PSOIWα for each given problem under the condition of same number of period, $T = 2500$.

2) Equal in Number of Particles: The number of particles used in the PSOIWα is set to the same to the total number of particles used in the MPSOIWα, $P=30$. As an example, Fig. 6 shows the resulting solution distributions of both the MPSOIWα and PSOIWα by using the LWA. In this case, we also can see that the density of solution distributions of the MPSOIWα are higher than that of the PSOIWα for each given problem, in spite of the different results obtained by the PSOIWα (P=10) compared to the solution distribution in Fig. 5.

For quantitative analysis to the obtained experimental results in Fig. 5 and Fig. 6, Table IV gives the performance indexes, i.e. the number of the optimal solutions $\vec{x}^o$ obtained by using the LWA, and the corresponding *FD* and *CR* for the given problems. By directly comparing the statistical results of the MPSOIWα and PSOIWα, the big difference between the both experimental results can be confirmed. It clearly indicates the strong points of the multi-swarm (MPSOIWα) search in dealing with the given MOO problems, and the method is applicable to cope with MOO, which is not only to efficiently find a large number of *Pareto*-optimal solutions, but also to find them with high-accuracy.

TABLE IV
SEARCH PERFORMANCE OF BOTH THE MPSOIWα AND PSOIWα ($P=10,30$) BY USING THE LWA (Γ IS SET TO 100).

problem	method	solution	*FD*	*CR* (%)
	MPSOIWα	**1254**	**2.234**$\times10^{-8}$	**99.5**
ZDT1	PSOIWα(P=10)	422	3.704×10^{-7}	89.5
	PSOIWα(P=30)	522	6.661×10^{-8}	91.0
	MPSOIWα	**272**	**1.198**$\times10^{-8}$	**94.0**
ZDT2	PSOIWα(P=10)	102	4.338×10^{-8}	60.0
	PSOIWα(P=30)	231	9.938×10^{-8}	61.5
	MPSOIWα	**1231**	**8.961**$\times10^{-7}$	**46.0**
ZDT3	PSOIWα(P=10)	423	6.748×10^{-6}	45.0
	PSOIWα(P=30)	432	4.496×10^{-6}	41.0

The values in bold signify the best result for each given problem.

On the other hand, by directly comparing the performance indexes of the PSOIWα run in Table IV, the increment of the number of particles can cause the improvement of performance in the number of solutions and solution accuracy, but not in the cover rate for the *ZDT3* problem. This reflects the basic feature of the PSOIWα run to MOO, i.e. the increment of particles used is not in proportion to the increment of the *CR*.

C. Computation Cost

To investigate the computation costs of the MPSOIWα, MPSOIW, PSOIWα and PSOIW run, as an example, the computer experiments[c] were carried out by using the LWA with increasing the dimensional number n of the variable vector for the *ZDT1* problem. The resulting average number of running times (RT) for implementing these methods are shown in Fig. 7.

Furthermore, the conformity of RT with respect to the dimensional number n for the used four methods is shown as follows.

$$\begin{cases} RT_{MPSOIW\alpha} = 120.864 + 8.1010n + 0.4959n^2 \\ RT_{PSOIW\alpha} \quad = 35.0395 + 4.8905n + 0.0654n^2 \\ RT_{MPSOIW} \quad = 63.2933 + 6.8871n + 0.1174n^2 \\ RT_{PSOIW} \quad\;\; = 23.1463 + 2.0206n + 0.0567n^2 \end{cases}$$

[c]Computing environment: Intel(R) Xeon(TM) CPU 3.40GHz, 2.00GB RAM; Computing tool: Mathematica 8.0.

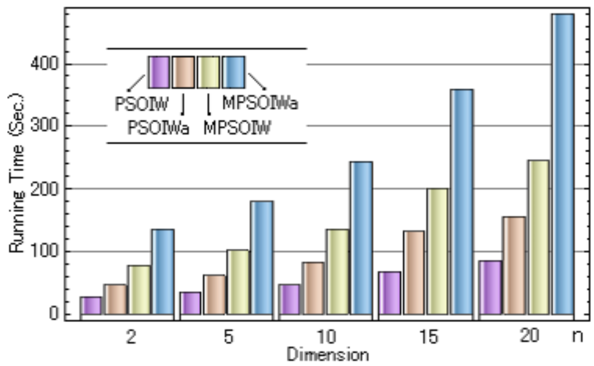

Fig. 7. Running time of implementing the PSOIW, PSOIWα, MPSOIW, and MPSOIWα for dealing with the $ZDT1$ problem with increasing the dimensional number n.

Accordingly, comparing with the values of the first-degree and second-degree coefficients in the above two pairs of approximate equations, all of the proportional rates between the MPSOIWα and PSOIWα, and between MPSOIW and PSOIW are more than double. On the basis of the wide margin between them, it is easily reminded of that the experimental results fit in with "no free lunch" (NFL) theorem [35]. As an application of meta-optimization technique, for example, the method of evolutionary particle swarm optimizer with inertia weight (EPSOIW) [42] could be used for improving the search performance of the original PSOIW used in MPSOIWα. This is because the computation cost of an optimized PSOIW is similar to that of the original PSOIW except the computation cost of estimating appropriate parameter values of the PSOIW to the given MOO problem.

V. CONCLUSIONS

In this paper, multiple particle swarm optimizers with inertia weight, called MPSOIWα, has been presented to MOO. It is the most simple expansion of the existent PSOIWα, which has the advantages of a hybrid search with easy-to-operation as a method of cooperative PSO.

Applications of the MPSOIWα to the given suite of 2-objective optimization problems well demonstrated its effectiveness by the aggregation-based manner. Owing to the obtained experiment results respectively by using three kinds of dynamic weighted aggregations, it is observed that the search performance of the MPSOIWα is superior to that of both the PSOIWα and MPSOIW, and the comparative analysis of the MPSOIWα shows that the search performance of using the LWA is better than that of using the BWA or SWA for the given MOO problems. Therefore, it is no exaggeration to say that our empirical analysis offers an important evidence, i.e. choosing the dynamic weighted sum method with the LWA for efficiently dealing with complex MOO problems.

It is left for further study to apply the MPSOIWα to complex MOO problems in the real-world. Furthermore, in order to enhance the adaptability, efficiency, and solution accuracy of the MPSOIWα, the search strategies and attempts on prediction, intelligent, cooperativeness, and other powerful methods of cooperative PSO [3], [12], [41] will be discussed for MOO in near future.

Acknowledgment

The author would like to thank Prof. M. Nagamatu for his helpful suggestions and discussions on the performance analysis of MOO and other topics.

References

[1] M. R. Alrashidi and M. E. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Power System Operations," *Electric Power Components and Systems*, vol.34, Issue 12, pp.1349-1357, 2006.

[2] F. van den Bergh, "An Analysis of Particle Swarm Optimizers," *Ph.D thesis*, University of Pretoria, Pretoria, South Africa, 2002.

[3] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol.8, Issue 3, pp.225-239, 2004.

[4] J. F. Chang, S. C. Chu, J. F. Roddick and J. S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of information science and engineering*, no.21, pp.809-818, 2005.

[5] A. Chinchuluun and P. M. Pardalos, "A survey of recent developments in multiobjective optimization," *Ann Oper Res*, vol.154, pp.29-50, 2007.

[6] 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, pp.58-73, 2000.

[7] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," *Proceedings of Congress Evolutionary Computation* (CEC 2002), vol.1, pp.1051-1056, Honolulu, HI, USA, 2002.

[8] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd, New York, 2001.

[9] K. Deb, "Multi-Objective Optimization," in E. K. Burke et al. (Eds.), *Search Methodologies — Introductory Tutorials in Optimization and Decision Support Techniques*, Springer, 2005.

[10] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particleswarm optimization," *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, vol.1, pp.84-88, La Jolla, CA, USA, 2000.

[11] M. Eghbal, N. Yorino, Y. Zoka and E. E. El-Araby, "Application of Multi-Objective Evolutionary Optimization Algorithms to Reactive Power Planning Problem," *IEEJ Transactions on Electrical and Electronic Engineering*, vol.4, pp.625-632, 2009.

[12] M. El-Abd and M. S. Kamel, "A Taxonomy of Cooperative Particle Swarm Optimizers," *International Journal of Computational Intelligence Research*, ISSN 0973-1873, vol.4, no.2, pp.137-144, 2008.

[13] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Journal of Operational Research*, vol.4, pp.99-107, 1992.

[14] C. R. Hema, M. P. Paulraj, R. Nagarajan, S. Yaacob and A. H. Adom, "Application of Particle Swarm Optimization for EEG Signal Classification," *Biomedical Soft Computing and Human Sciences*, vol.13, no.1, pp.79-84, 2008.

[15] E. J. Hughes, "Multiobjective Problem Solving from Nature," *Natural Computing Series*, Part IV, pp.307-329, DOI:10.1007/978-3-540-72964-8_15, 2008.

[16] Y. Jin, M. Olhofer and B. Sendhoff, "Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?" *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO2001), pp.1042-1049, San Francisco, CA, USA, 2001.

[17] Y. Jin and B. Sendhoff, "Constructing Dynamic Optimization Test Problems Using the Multi-Objective Optimization Concept," in G. Raidl et al. (Eds.) *Applications of Evolutionary Algorithms*, LNCS 3005, pp.525-536, 2004.

[18] C.-F. Juang, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol.34, no.2, pp.997-1006, 2004.

[19] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp.1942-1948, Piscataway, NJ, USA, 1995.

[20] X. Li, J. Branke and M. Kirley. "On Performance Metrics and Particle Swarm Methods for Dynamic Multiobjective Optimization Problems," *Proceedings of IEEE Congress of Evolutionary Computation* (CEC), pp.1635-1643, Singapore, 2007.

[21] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct Multidisc Optim*, vol.26, pp.369-395, 2004.

[22] P. Moscato, "On Evolution, Search Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," *Technical Report Caltech Concurrent Computation Program*, Report 826, California Institute of Technology, Pasadena, CA, USA, 1989.

[23] B. Niu, Y. Zhu and X. He, "Multi-population Cooperation Particle Swarm Optimization," in M. Capcarrere et al. (Eds.), *Advances in Artificial Life*, LNCS 3630, (Springer Heidelberg, 2005), pp.874-883, 2005.

[24] H. Piao, Z. Wang and H. Zhang, "Cooperative-PSO-based PID neural network integral control strategy and simulation research with asynchronous motor controller design," *Journal of WSEAS Transactions on Circuits and System*, vol.8, Issue 8, pp.696-708, 2009.

[25] R. Poli, J. Kennedy and T. Blackwell, "Particle swarm optimization — An overview," *Swarm Intell*, vol.1, pp.33-57, DOI:10.1007/s11721-007-0002-0, 2007.

[26] M. Reyes-Sierra and C. A. Coello Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol.2, no.3, pp.287-308, 2006.

[27] Y. Shi and R. C. Eberhart, "A modified particle swarm optimiser," *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp.69-73, Anchorage, Alaska, USA, 1998.

[28] F. J. Solis and R. J.-B. Wets, "Minimization by Random Search Techniques," *Mathematics of Operations Research*, vol.6, no.1, pp.19-30, 1981.

[29] J. C. Spall, "Stochastic Optimization," in J. Gentle et al. (Eds.) *Handbook of Computational Statistics*, Springer Heidelberg, pp.169-197, 2004.

[30] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computations, and Application*, John Wiley & Sons, Ltd, New York, 1986.

[31] S. Suzuki, "Application of Multi-objective Optimization and Game Theory to Aircraft Flight Control Problem," *Institute of Systems, Control and Information Engineers*, vol.41, no.12, pp.508-513, 1997 (in Japanese).

[32] P. K. Tripathi, S. Bandyopadhyay and S. K. Pal, "Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficient," *Information Sciences* 177, pp.5033-5049, 2007.

[33] C.-S. Tsou, S.-C. Chang and P.-W. Lai, "Using Crowing Distance to Improve Multi-Objective PSO with Local Search," in Felix T. S. Chan et al. (Eds.), *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, pp.77-86, 2007.

[34] D. A. van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., Wright-Patterson AFB, OH, Tech. Rep. TR-98-03, 1998.

[35] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, pp.67-82, 1997.

[36] H. Zhang and M. Ishikawa, "Evolutionary Particle Swarm Optimization (EPSO) – Estimation of Optimal PSO Parameters by GA," Lecture Notes in Engineering and Computer Science: *Proceedings of The International MultiConference of Engineers and Computer Scientists 2007*, IMECS 2007, 21-23 March, 2007, Hong Kong, pp.13-18.

[37] H. Zhang and M. Ishikawa, "Characterization of particle swarm optimization with diversive curiosity," *Journal of Neural Computing & Applications*, pp.409-415, Springer London, DOI:10.1007/s00521-009-0252-4, 2009.

[38] H. Zhang, "Multiple Particle Swarm Optimizers with Diversive Curiosity," Lecture Notes in Engineering and Computer Science: *Proceedings of The International MultiConference of Engineers and Computer Scientists 2010*, IMECS2010, 17-19 March, 2010, Hong Kong, pp.174-179.

[39] H. Zhang and M. Ishikawa, "The performance verification of an evolutionary canonical particle swarm optimizers," *Neural Networks*, vol.23, no.4, pp.510-516, 2010.

[40] H. Zhang and J. Zhang, "The Performance Measurement of a Canonical Particle Swarm Optimizer with Diversive Curiosity," *Proceedings of International Conference on Swarm Intelligence* (ICSI'2010), LNCS 6145, Part I, pp.19-26, Beijing, China, 2010.

[41] H. Zhang, "Multiple Particle Swarm Optimizers with Inertia Weight with Diversive Curiosity and Its Performance Test," *IAENG International Journal of Computer Science*, vol.38, no.2, pp.134-145, 2011.

[42] H. Zhang, "Assessment of An Evolutionary Particle Swarm Optimizer with Inertia Weight," *Proceedings of 2011 IEEE Congress on Evolutionary Computation* (CEC), ISBN: 978-1-4244-7834-7, DOI:10.1109/CEC.2011.594827, pp.1746-1753, New Orleans, LA, USA, 2011.

[43] H. Zhang, "The Performance Analysis of an Improved PSOIW for Multi-objective Optimization," *IAENG International Journal of Computer Scientists*, ISSN: 1819-9224, vol.39, no.1, pp.34-41, 2012.

[44] H. Zhang, "Multiple Particle Swarm Optimizers with Inertia Weight for Multi-objective Optimization," Lecture Notes in Engineering and Computer Science: *Proceedings of The International MultiConference of Engineers and Computer Scientists* 2012, IMECS2012, 14-16 March, 2012, Hong Kong, pp.23-28.

[45] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff and E. Tsang, "Prediction-based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization," *Proceedings of The 4th International Conference on Evolutionary Multi-Criterion Optimization*, LNCS 4403, Springer, pp.832-846, 2007.
[46] E. Zitzler, K. Deb and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol.8, no.2, pp.173-195, 2000.

Hong Zhang was born in Beijing, China, who received his M.Eng degree from the Division of Engineering, Graduate School, Kyushu University, Japan in 1991 and D.Eng degree from the Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology (KIT), Japan in 2001, respectively. Now he is an assistant professor at the Graduate School of Life Science and Systems Engineering, KIT. His interest includes neural computation, genetic and evolutionary computation, data mining, pattern recognition, system identification, optimization, inverse optimization, swarm intelligence, mobile robot and other applications.

Dr. H. Zhang is a member of IEEJ, IEICE, and IAENG, and has published over 25 refereed journals papers and 50 refereed conference proceeding papers and book chapters. His paper entitled "Improving the Performance of Particle Swarm Optimization with Diversive Curiosity" has been selected for the *Best Paper Award* of the 2008 IAENG International Conference on Artificial Intelligence and Applications. Until now, he has been obtained approximately one hundred thousand dollars of competitive research funding, and hosted two research projects to develop the algorithms on inverse optimization and swarm intelligence.