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). In this paper we present a method of multiple particle swarm optimizers with inertia weight (MPSOIWα), which belongs to a kind of the methods of cooperative particle swarm optimization. The crucial idea of the MPSOIWα, here, is to reinforce the search ability of the PSOIWα by the union's power of plural swarms. To demonstrate its effectiveness and search performance, computer experiments on a suite of 2-objective optimization problems are carried out by a weighted sum method. The resulting Pareto-optimal solution distributions corresponding to each given problem indicate that the linear weighted aggregation among the adopted three kinds of dynamically 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 PSOIWα and MPSOIW.**

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

I. Introduction

MULTI-objective optimization (MOO) is the processing of optimizing simultaneously two and more conflicting objectives subject to certain constraints [4], [6]. 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 and so on [8], [11], [23], how to efficiently deal with MOO becomes a live issue, and is centered on the development of the treatment technique.

Particle swarm optimization (PSO), which was created by Kennedy and Eberhart in 1995, is an adaptive, stochastic, and population-based optimization technique [15]. Based on the special features, i.e. information exchange, intrinsic memory, and directional search, the technique has higher latent search ability in optimization compared to some methods of evolutionary computation (EC) such as genetic algorithms and genetic programming [19], [20], [26], [27]. 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. Various kinds of the methods of cooperative PSO, for example, hybrid PSO, multi-layer PSO, multiple PSO with decision-making strategy etc. were published [2], [10], [17], [27].

In contrast to those methods running a single particle swarm, many attempts and strategies can be perfected with operating multiple particle swarms for more efficiently finding an optimal solution or near-optimal solutions [3], [14], [17], [28]. 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 [18].

An improved particle swarm optimizer with inertia weight (PSOIWα) was published [30]. For further upgrading its search performance to MOO, in this paper we propose to use a method of cooperative PSO, called multiple particle swarm optimizers with inertia weight (MPSOIWα). The crucial idea of the MPSOIWα, here, is to reinforce the search ability of the PSOIWα by the union's power of plural swarms. Although the search feature and performance of some PSO methods in MOO with fitness assignment manners such as criterion-based manner or dominance-based manner were studied and investigated [24], [25], there are insufficient results for systematically solving MOO problems by an aggregation-based manner, and analyzing the potential characteristics in details from the obtained experimental results [6], [16].

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 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 dynamically weighted aggregations (i.e. linear weighted aggregation, bang-bang weighted aggregation, and sinusoidal weighted aggregation[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 and performance of the MPSOIWα contrast with that of the PSOIWα and MPSOIW.

II. Basic Concepts

For explaining how to treat with MOO by a fitness assignment manner, some basic concepts and definitions on a general MOO problem, *Pareto*-optimal solution, front distance, cover rate, a weighted sum method, and three kinds of dynamically weighted aggregations are briefly described.

Manuscript received December 30, 2011; revised January 27, 2012. 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 sub-swarms) searching for a solution (serially or in parallel) and exchanging some information during the search according to some communication strategies.

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

A. MOO Problem

In general, the formulation of a MOO problem can be defined as follows.

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

where $f_i(\vec{x})$ is the *i*-th objective, $g_j(\vec{x})$ is the *j*-th inequality constraint, $h_m(\vec{x})$ is the *m*-th equality constraint, $\vec{x} = (x_1, x_2, \cdots, x_N)^T \in \Re^N = \Omega$ (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 at the same time. Consequently, the goal of handling the MOO problem is effectively to achieve a set of solutions that satisfy *Pareto* optimality for improvement of mental capacity.

B. Pareto-optimal Solution

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

C. Weighted Sum Method

There are some fitness assignment manners such as aggregation-based one, criterion-based one, and dominance-based one, which are used for MOO [7], [12]. As to be generally known, a conventional weighted sum (CWS) method is a straightforward approach applied to deal with MOO problems. In this case, the different objectives are summed up to a single scalar F_s (criterion) with some prescribed weights as follows.

$$F_s(\vec{x}) = \sum_{i=1}^{I} c_i f_i(\vec{x}) \tag{3}$$

where $c_i (i = 1, 2, \cdots, I)$ is the non-negative weight. During the optimization, usually, these weights are fixed by the constraint of $\sum_{i=1}^{I} c_i = 1$, and prior knowledge is also needed to specify these weights for obtaining good solutions.

To thoroughly conquer the weakness of the CWS method run, the following dynamically weighted sum (DWS) method is often used to MOO in practice. The criterion F_d of the method can be expressed as follows.

$$F_d(t, \vec{x}) = \sum_{i=1}^{I} c_i(t) f_i(\vec{x}) \tag{4}$$

where t is time-step to search, and $c_i(t) \geq 0$ is the dynamic weight. In order to present the method, a 2-objective optimization problem is considered as an example. Hence, the definitions of three kinds of the adopted dynamically weighted aggregations are expressed below.

- 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.

D. Front Distance

Front distance is expressed as a metric of 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 reflects the estimation accuracy of the optimizer used. Concretely, the definition of front distance (FD) is expressed as

$$FD = \frac{1}{Q}\sqrt{\sum_{q=1}^{Q} d_q^2},\ \ d_q = f_i(\vec{x}_q^*) - f_i(\vec{x}_q^o),\ \ \forall i \in I \tag{5}$$

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 true *Pareto*-optimal solution set.

E. 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* front. This is because the estimation accuracy is insufficiency to reveal the distribution status of the obtained *Perato*-optimal solutions and their possibility for dealing with the given problem.

Here, the formulation of CR is mathematically expressed by

$$CR = \frac{1}{I}\sum_{i=1}^{I} CR_i \tag{6}$$

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

$$CR_i = \frac{\sum_{l=1}^{\Gamma} \gamma_l}{\Gamma} \tag{7}$$

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.

III. ALGORITHMS

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 weak convergence of the original PSO [1], [5], Shi et al. modified the update rule of the i-th particle's velocity by constant reduction of the inertia coefficient over time-step [9], [21]. 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 as same as the original PSO.

Owing to the bigger difference between the two boundary values of the variable inertia weight, it is obvious that the search behavior of the PSOIW achieves a search shift which smoothly changes from exploratory mode to exploitative one in the whole optimization process. Hence, this way is very simple and useful for conquering the weakness of the PSO in convergence and enhancing the solution accuracy. On the other hand, the shortcoming of the PSOIW is easily to fall into a local minimum and hardly to escape from that place in dealing with multimodal problems because the terminal value w_e is set to small.

B. The PSOIWα

For alleviating the weakness of the PSOIW search, we introduce the LRS [22], [29] into the PSOIW to form a hybrid search optimizer (called PSOIWα). Implementing the PSOIWα, here, is to enable a particle swarm search escapes from local minimum sooner for efficiently obtaining an optimal solution or near-optimal solutions.

The PSOIWα's procedure is implemented as follows.

```
step-1: Give the terminating condition, $U$ (the number
        of random data) of the PSOIW$\alpha$ 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 $\sigma$ 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 $u \le U$ 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 improving the search ability of the existent PSOIWα to MOO, we propose to use multiple particle swarm optimizers with inertial weight, MPSOIWα. Figure 1 illustrates a flowchart of the MPSOIWα.

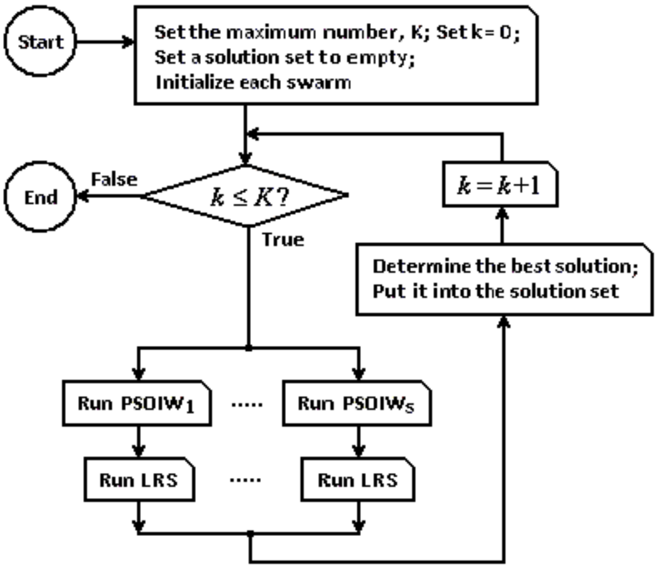
Start
Set the maximum number, K; Set k = 0;
Set a solution set to empty;
Initialize each swarm
End
False
$k \le K?$
True
$k = k+1$
Determine the best solution;
Put it into the solution set
Run PSOIW$_1$
.....
Run PSOIW$_S$
Run LRS
.....
Run LRS

Fig. 1. A flowchart of the MPSOIWα.

The most difference between the PSOIWα and MPSOIWα in composition is just to implement the plural PSOIWα ($S \ge 2$) in parallel for finding the most suitable solution or near-optimal solutions. Concretely, the best solution of the multi-swarm search, i.e. $\vec{x}_k^o = arg \max_{i=1,2,\cdots,S} \{g(\vec{q}_k^i)\}$, is determined from the solutions obtained by each PSOIWα run at time-step k, and then put it into a solution set which is the storage memory of the multi-swarm.

It is obvious that the MPSOIWα is the use of swarm intelligence to search by the union's power of plural swarms for enforcing the search ability of the PSOIWα. It is to be noted that if the LRS 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	$f_{11}(\vec{x}) = x_1,\ g(\vec{x}) = 1 + \dfrac{9}{N-1}\sum_{n=2}^{N} x_n,\ f_{12}(\vec{x}) = g(\vec{x})\left(1 - \sqrt{\dfrac{f_{11}(\vec{x})}{g(\vec{x})}}\right)$	$\Omega \in [0,1]^N$
ZDT2	$f_{21}(\vec{x}) = x_1,\ f_{22}(\vec{x}) = g(\vec{x})\left(1 - \left(\dfrac{f_{21}(\vec{x})}{g(\vec{x})}\right)^2\right)$	$\Omega \in [0,1]^N$
ZDT3	$f_{31}(\vec{x}) = x_1,\ f_{32}(\vec{x}) = g(\vec{x})\left(1 - \sqrt{\dfrac{f_{31}(\vec{x})}{g(\vec{x})}} - \left(\dfrac{f_{31}(\vec{x})}{g(\vec{x})}\right) sin\big(10\pi f_{31}(\vec{x})\big)\right)$	$\Omega \in [0,1]^N$

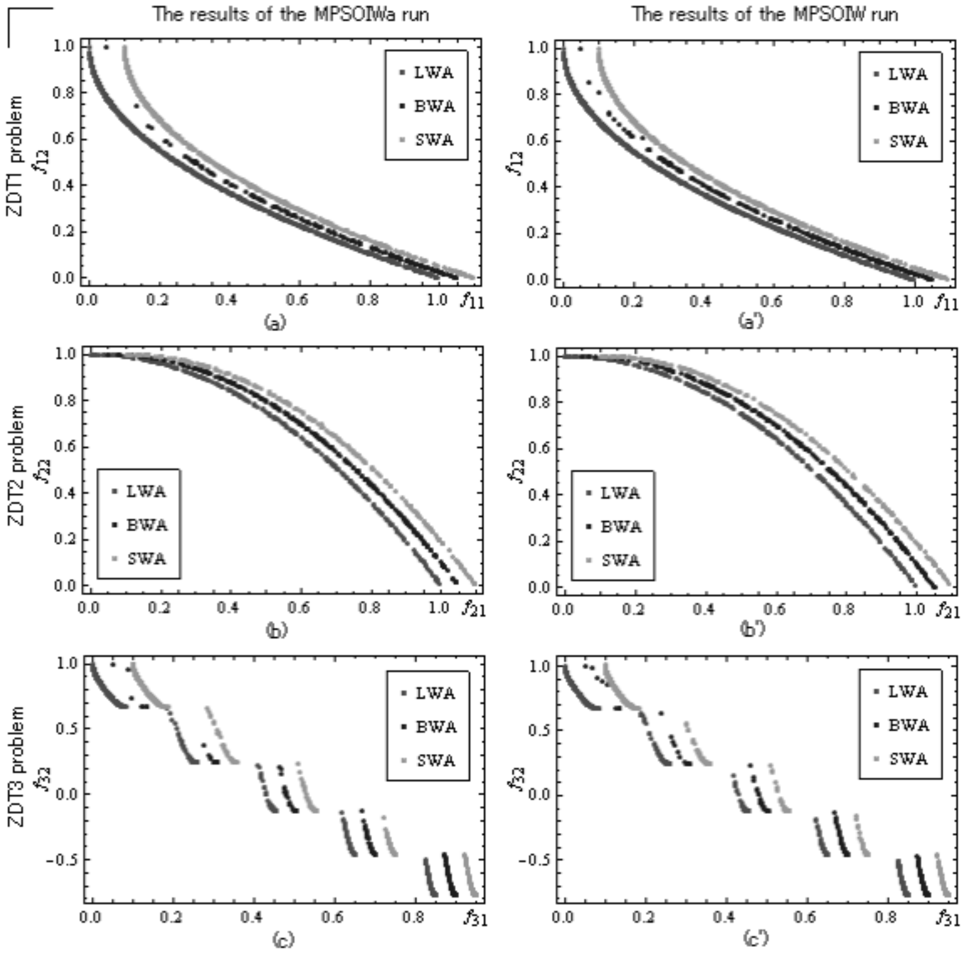

Fig. 2. Solution distributions of the MPSOIWα and MPSOIW by using the LWA (red-point), BWA (blue-point) and SWA (green-point), respectively. Notice: the distance between the experimental data sets for each subgraph is 0.05 (shift only in horizontal direction).

optimization problems [31] in Table I is used in the next computer experiments. The characteristics of the *Pareto* fronts of the given problems include the convex (*ZDT1*), concave (*ZDT2*), and discontinuous multimodal (*ZDT3*), respectively.

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

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.

A. Performance Comparison

For the sake of observation, Figure 2 shows the resulting solution distributions of the MPSOIWα and MPSOIW by using the LWA, BWA, and SWA, respectively. According to the distinction of each solution distribution corresponding to these given problems, the analytical judgment can be described as follows.

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.

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**$\times10^{-8}$	**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**$\times10^{-8}$	**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**$\times10^{-7}$	**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.

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 Figure 2, 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.

The following features can be observed from Table III. Firstly, there is the most number of solutions obtained by using the LWA for the given problems even for the *ZDT2* one in where a large number of *Pareto*-optimal solutions are in unstable position [13]. 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.

Therefore, the effectiveness and search ability of the MPSOIWα are roughly confirmed by the above analytical results. Furthermore, better solution distribution and higher solution accuracy can be observed as well by using either the LWA or SWA. Our 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 dynamically weighted aggregation can be expressed as follows.

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

The relationship of the above domination 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 *Perato*-optimal solution to the maximum under the condition of implementing the same optimizer. Due to this reason, more good solutions can be easily obtained during the short search cycle, $K = 25$.

B. Effect of Multi-swarm Search

For equal treatment in search, the number of particles used in a swarm is the same to the total number of particles used in the 3-swarms. As an example, Figure 3 shows the resulting solution distributions of both the MPSOIWα and PSOIWα (i.e. $P = 30$) by using the LWA. We can see that the density of solution distributions of the MPSOIWα are higher than that of the PSOIWα for each given problem.

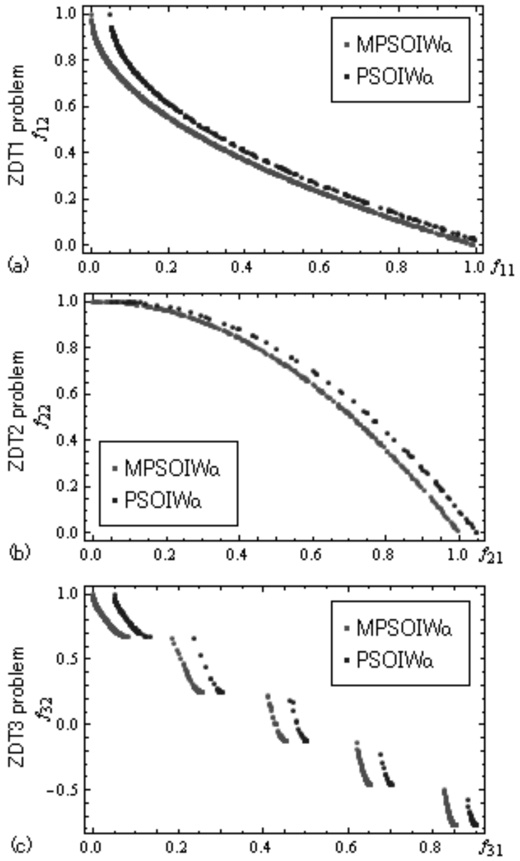

Fig. 3. The solution distributions of the MPSOIWα (red-point) and PSOIWα (blue-point) by using the LWA.

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 performance indexes with the MPSOIWα and PSOIWα, the big difference between the both experimental results clearly indicate the strong points of the multi-swarm search in dealing with the given MOO problems under the condition of the same number of particles

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

problem	method	solution	*FD*	*CR* (%)
ZDT1	MPSOIWα	1254	2.234×10^{-8}	99.5
	PSOIWα	522	6.661×10^{-8}	91.0
ZDT2	MPSOIWα	272	1.198×10^{-8}	94.0
	PSOIWα	231	9.938×10^{-8}	61.5
ZDT3	MPSOIWα	1231	8.961×10^{-7}	46.0
	PSOIWα	432	4.496×10^{-6}	41.0

used. It is demonstrated that the MPSOIWα is a powerful method of cooperative PSO to MOO.

V. CONCLUSIONS

In this paper, multiple particle swarm optimizers with inertia weight, MPSOIWα, has been presented to MOO. Based on the composition of the MPSOIWα, 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 resulting experimental data by respectively using three kinds of dynamically weighted aggregations, it is observed that the search performance of the MPSOIWα is superior to that of 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 experimental results could offer an important evidence, i.e. choosing the dynamically weighted sum method with the LWA for efficiently dealing with complex MOO problems.

It is left for further study to apply the MPSOIWα to 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 and powerful cooperative PSO algorithms [2], [10], [30] will be discussed for MOO in near future.

REFERENCES

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

[12] E. J. Hughes, "Multiobjective Problem Solving from Nature," *Natural Computing Series*, Part IV, pp.307-329, 2008.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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, pp.874-883, 2005.

[18] 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.

[19] R. Poli, J. Kennedy and T. Blackwell, "Particle swarm optimization — An overview," *Swarm Intell*, vol.1, pp.33-57, 2007.

[20] 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.

[21] 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.

[22] 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.

[23] 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).

[24] 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.

[25] 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.

[26] H. Zhang and M. Ishikawa, "Evolutionary Particle Swarm Optimization (EPSO) – Estimation of Optimal PSO Parameters by GA," *Proceedings of the IAENG International MultiConference of Engineers and Computer Scientists 2007* (IMECS2007), vol.1, pp.13-18, Hong Kong, China, 2007.

[27] H. Zhang and M. Ishikawa, "Characterization of particle swarm optimization with diversive curiosity," *Journal of Neural Computing & Applications*, pp.409-415, Springer London, 2009.

[28] 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.

[29] 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.

[30] 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.

[31] E. Zitzler, K. Deb and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol.8, no.2, pp.173-195, 2000.