# The Performance Analysis of an Improved PSOIW for Multi-objective Optimization

Hong Zhang, *Member, IAENG*

***Abstract*—In this paper, we investigate the search performance of an improved particle swarm optimizer with inertia weight (PSOIW$\alpha$) for multi-objective optimization. The basic idea of the proposed PSOIW$\alpha$, here, is to introduce a localized random search into the original PSOIW to reinforce its search ability for improvement of search performance. To demonstrate the effectiveness and search effect of the proposal, computer experiments on a suite of 2-objective optimization problems are carried out by an aggregation-based manner. We show the distribution status of the resulting Pareto-optimal solutions corresponding to each given problem by respectively using three kinds of dynamically weighted aggregations, point out which one is the most suitable for acquiring good search result among them, and clarify the search characteristics and performance effect of the PSOIW$\alpha$.**

***Index Terms*—multi-objective optimization, swarm intellingence, particle swarm optimizer with inertia weight, hybrid search, weighted sum method, dynamically weighted aggregation.**

## I. Introduction

THE process of optimizing simultaneously two and more conflicting objectives subject to certain constraints is called multi-objective optimization (MOO) [1], [3]. The treatment technique is very important to indicate rational potentiality factors in a given MOO problem even if any complicated constitution and evaluation. So far, MOO has been widely applied in various areas of science, technology, industry, finance, automobile design, aeronautical engineering, daily living and so on [5], [21].

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 search multiple optimal solutions. Normally, a MOO problem has to be converted to a single-objective optimization (SOO) problem before the optimization. Thus the search generates a single *Pareto* optimum for each run of the optimization, and that the obtained solutions are highly sensitive to the weight vector used in the converting process. Nevertheless, the issue of adopting the way is to how to ensure that the obtained solutions satisfy *Pareto* optimality[1].

Since the optimization methods of evolutionary computation (EC) can obtain plural candidate solutions, i.e. individuals in a population, it seems naturally to use them in MOO for finding a group of *Pareto*-optimal solutions simultaneously. According to the distinguishing features of group search, the use of EC methods for dealing with MOO problems has significantly grown and has been found to be successful over the last decade [4], [14].

As a new member of EC, particle swarm optimization (PSO) [13] is an adaptive, stochastic, and population-based optimization technique. Based on the special information exchange, intrinsic memory, and directional search of the technique, it has higher search ability in optimization compared to the other members such as genetic algorithms and genetic programming [16], [17], [28], [29]. Although the search performance and effect of some PSO methods in MOO has been studied and investigated [22], [23], 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 by using different dynamically weighted aggregations, respectively.

For resolving the above situation, in this paper we investigate the search performance of an improved particle swarm optimizer with inertia weight (PSOIW$\alpha$) to MOO. The basic idea of the proposed PSOIW$\alpha$, here, is to introduce a localized random search (LRS) [30] into the original PSOIW [18] to reinforce its search ability for improvement of search performance. The construction and execution of the algorithm are the most simple and easy-to-operation in which a hybrid search (i.e. a compound made up of the PSOIW and LRS) is implemented to find *Pareto*-optimal solutions corresponding to a given MOO problem.

To demonstrate the effectiveness and performance effect of the PSOIW$\alpha$, computer experiments on a suite of 2-objective optimization problems are carried out by a weighted sum method. We show the distribution status 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[2]), clarify the search characteristics and performance effect of the PSOIW$\alpha$ by comparing the search performance with the original PSOIW, and indicate that which one is the most suitable for acquiring better search results to the given MOO problems among the used dynamically weighted aggregations.

The rest of the paper is organized as follows. Section II briefly introduces some basic concepts and definitions for completely dealing with a general MOO problem. Section III describes the algorithms of the PSOIW and the proposed PSOIW$\alpha$. Section IV provides the experimental results cor-

Manuscript received September 30, 2011; revised December 20, 2011. 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.

[1]Pareto optimality, named after Italian sociologist and economist Vilfredo Pareto (1848-1923), is a situation which exists when economic resources and output have been allocated in such a way that no-one can be made better off without sacrificing the well-being of at least one person.

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

(Advance online publication: 27 February 2012)

responding to a suite of 2-objective optimization problems, and analyzes the potential characteristics of the PSOIW$\alpha$ in the technical details. Finally, the concluding remarks appear in Section V.

## II. Basic Concepts

For understanding the whole process of treating with MOO by a fitness assignment manner, in this section, we gradually describe some basic concepts and definitions on a general MOO problem, *Pareto* optimality, front distance, a weighted sum method and three kinds of the used dynamically weighted aggregations.

### A. MOO Problem

In general, the formulation of a MOO problem can be defined by

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

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 basic condition of $I \geq 2$, as experiential knowledge, the $I$-objectives may be conflicting with each other. Under this circumstance, it is difficult to obtain the global optimum corresponding to each objective 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 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 *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* front ($PF$).

### C. Weighted Sum Method

So far, there are many fitness assignment manners such as aggregation-based one, criterion-based one, and dominance-based one which are used for treating with MOO problems [4], [10]. As to be generally known, a conventional weighted sum (CWS) method is a straightforward approach applied for MOO [9]. In this case, the different objectives are summed up to a single scalar $F$ (criterion) with some prescribed weights as follows.

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

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 efficiently obtaining good solutions.

Because all of values of the weights in Eq.(3) are constants, it is clear that only one *Pareto*-optimal solution can be obtained with one run of the optimization. This matter means that an experimenter who has to implement an optimizer many times for obtaining different *Pareto*-optimal solutions corresponding to the given MOO problem.

To thoroughly overcome the disadvantage of the CWA method, some dynamically weighted aggregations are applied to MOO in practice. For explaining how to deal with a 2-objective optimization problem, as an example, the definitions of three kinds of the used dynamically weighted aggregations are expressed as follows.

- Linear weighted aggregation (LWA):

$$\begin{cases} c_1^l(t) = mod\Big(\dfrac{t}{T}, 1\Big), \\ c_2^l(t) = 1 - c_1^l(t) \end{cases} \tag{4}$$

- Bang-bang weighted aggregation (BWA):

$$\begin{cases} c_1^b(t) = \dfrac{sign\big(sin(2\pi t/T)\big) + 1}{2}, \\ c_2^b(t) = 1 - c_1^b(t) \end{cases} \tag{5}$$

- Sinusoidal weighted aggregation (SWA):

$$\begin{cases} c_1^s(t) = \Big|sin\big(\dfrac{\pi t}{T}\big)\Big|, \\ c_2^s(t) = 1 - c_1^s(t) \end{cases} \tag{6}$$

where *T* is a period of the variable weights, and *t* is the time-step in Eqs. (4), (5), and (6).

According to the change characteristics of the dynamically weighted aggregations in Figure 1, as usual, the value of the criterion $F$ of the used LWA or SWA smoothly changes with the growth of time-step $t$ in the period, $T = 20$. Contrast to this, the value of the criterion $F$ of the used BWA changes discontinuously. Moreover, such abrupt movement is just only one time in the same period. Through the above observation, it is considered that different characteristics and process of variations in the criteria with the growth of time-step $t$ will greatly reflect the search performance and effect of using each aggregation corresponding to a given MOO problem.

### 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 reflects the accuracy of estimation of the optimizer used. Concretely, the definition of front distance ($FD$) is expressed by

$$FD = \frac{1}{Q}\sqrt{\sum_{q=1}^{Q} d_q^2},\ \ d_q = f_i(\vec{x}_q^*) - f_i(\vec{x}_q^o),\ \ \forall i \in I \tag{7}$$

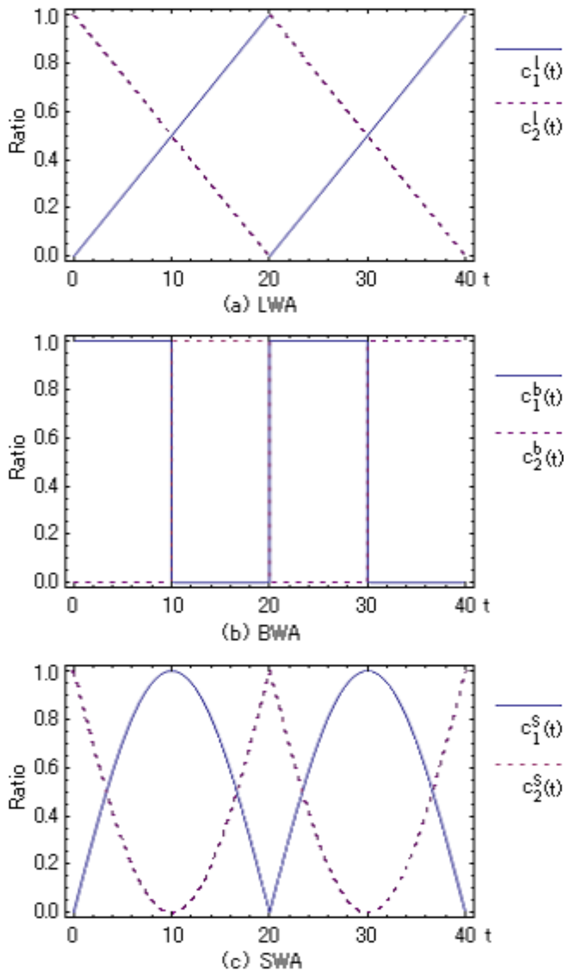


Fig. 1. The change characteristics of three types of the adopted dynamically weighted aggregations under the condition of period $T = 20$. (a) Linear weighted aggregation, (b) Bang-bang weighted aggregation, (c) Sinusoidal weighted aggregation.

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 showing the coverage of the elements being in the set of non-dominated solutions found to the *Pareto* front. This is because the accuracy of estimation is insufficiency to reveal the distribution of the obtained *Perato*-optimal solutions and their possibility for dealing with the given problem.

Here, the formulation of the CR is mathematically defined by

$$CR = \frac{1}{I}\sum_{i=1}^{I} CR_i \tag{8}$$

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

$$CR_i = \frac{\sum_{l=1}^{\Gamma} \gamma_l}{\Gamma} \tag{9}$$

where $\Gamma$ 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 status of existence of the obtained *Pareto*-optimal solutions in the $l$-th subdivision for the $i$-th objective.

It is obvious that the higher value of CR means the bigger of dominated volume in the objective space. However, the divided number to a designated objective space is given by an experimenter, so it goes without saying that $CR$ is just a relative metric depending on the value of $\Gamma$.

## III. Algorithms

For the convenience of the following description to the used each particle swarm 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

As to be generally known, the weak convergence is a disadvantage of the original PSO [25], [2]. To cope with this difficulty, Shi et al. modified the update rule of the $i$-th particle's velocity by constant reduction of the inertia coefficient over time-step [6], [18]. Concretely, the formulation of the particle swarm optimizer with inertia weight (PSOIW) is defined as

$$\begin{cases} \vec{x}_{k+1}^i = \vec{x}_k^i + \vec{v}_{k+1}^i \\ \vec{v}_{k+1}^i = w(k)\,\vec{v}_k^i + w_1\vec{r}_1 \otimes (\vec{p}_k^i - \vec{x}_k^i) + w_2\vec{r}_2 \otimes (\vec{q}_k - \vec{x}_k^i) \end{cases} \tag{10}$$

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 a 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 $k$ as follows.

$$w(k) = w_s + \frac{w_e - w_s}{K} \times k \tag{11}$$

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

It is obvious that owing to the difference between the boundary values of the variable inertia weight, the search behavior of the PSOIW achieves a search shift which smoothly changes from exploratory mode to exploitative one in optimization. Hence, the manner is very simple and useful for conquering the weakness of the PSO in convergence and enhancing the solution accuracy. However, the shortcoming of the PSOIW is that its search easily to fall into a local minimum and hardly to escape from that place in dealing with multidomal problems because the terminal value, $w_e$, is set to small.

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* | $f_{31}(\vec{x}) = x_1,\ f_{32}(\vec{x}) = g(\vec{x})\left(1 - \sqrt{\frac{f_{31}(\vec{x})}{g(\vec{x})}} - \left(\frac{f_{31}(\vec{x})}{g(\vec{x})}\right) sin\left(10\pi f_{31}(\vec{x})\right)\right)$ | $\Omega \in [0,1]^N$ |

### B. The PSOIW$\alpha$

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 [19], [20]. In order to alleviate the weakness of the PSOIW and to improve the search ability and search efficiency of the PSOIW, we introduce the LRS [31] into the PSOIW (called PSOIW$\alpha$) to explore for efficiently obtaining an optimal solution or near-optimal solutions.

The PSOIW$\alpha$ is a hybrid search method and its 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.

Based on the complementary characteristics of the used hybrid search (i.e. a kind of memetic algorithm [15]), the correctional function seems to be close to the HGAPSO [12] in search effect, which implements a plain GA [8] and the PSO with the mixed operations for improving the adaptation to treat with various blended distribution problems.

## IV. COMPUTER EXPERIMENTS

To facilitate comparison and analysis of the search performance of the proposed POSIW$\alpha$, the suite of 2-objective optimization problems [34] in Table I is used in the following 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 gives the major parameters of the PSOIW$\alpha$ to solve the given problems in Table I. The choice of their values is referred to the results of partial preliminary experiments.

TABLE II
MAJOR PARAMETERS OF THE PSOIW$\alpha$ 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, $U$ | 20 |
| the search range of the LRS, $\sigma$ | 0.1 |

For understanding the search process of the PSOIW$\alpha$ run in which how to deal with a MOO problem by using a dynamically weighted aggregation, as an example, Figure 2 shows the changes of fitness values of the top-one particle for dealing with the *ZDT3* problem by using the LWA.

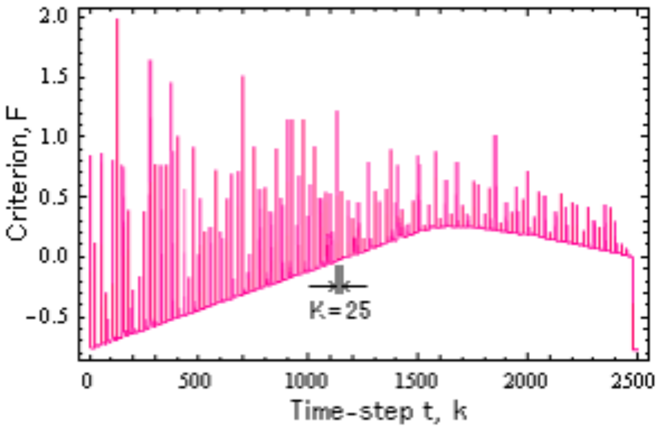


Fig. 2. The fitness change of top-one particle in the search process of the PSOIW$\alpha$ for the *ZDT3* problem by using the LWA.

According to the definition of MOO in Section II-A, the smaller the fitness values are, the better the obtained solutions are. We can clearly see from Figure 2 that the convergence of the PSOIW$\alpha$ run is faster in the whole optimization process. The smooth variation of the best criterion (top-one particle) suggests that the optimal solutions $\vec{x}^o$ can be continuously obtained during one short search cycle ($K = 25$). The vibration of the best fitness occurs with the change of variable criterion. And the vibration range of the fitness reflects the influence receiving from the change of the criterion $F$ overall time-step. Needless to say, the movement features of the fitness in progress are not unique according to deal with the given different problems.

### A. Performance Comparison

Figure 3 shows the resulting solution distributions of the PSOIW$\alpha$ and PSOIW by using the LWA, BWA, and

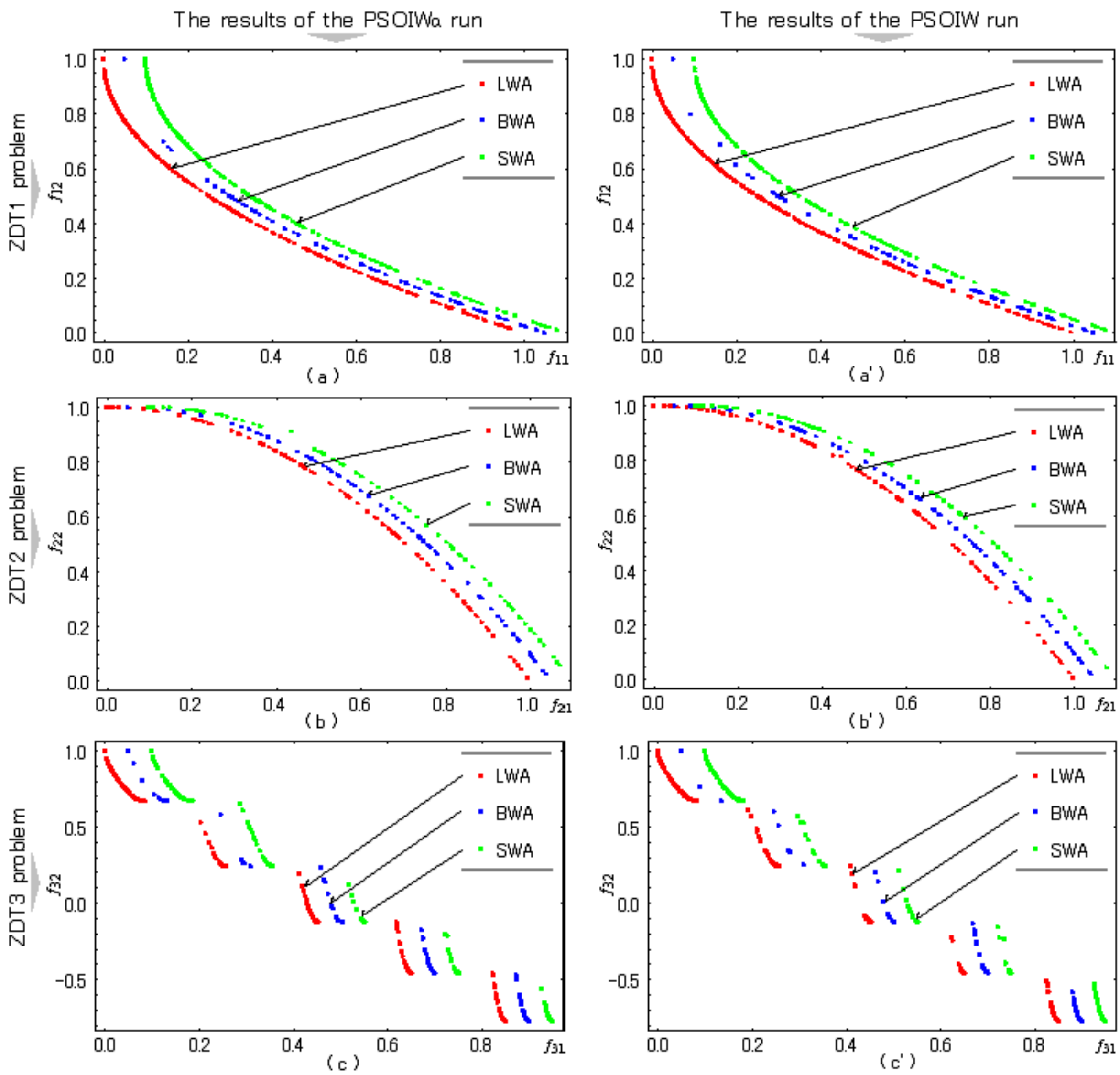


Fig. 3. Solution distributions of the PSOIW$\alpha$ and PSOIW by using the LWA, BWA and SWA, respectively. Notice: the distance between the experimental data sets for each problem is 0.05 (shift only in horizontal direction).

SWA, respectively. By observing the distinction of each solution distribution corresponding to the given problems, the analytical judgment can be described as follows.

1) Regardless of the used methods either the PSOIW$\alpha$ or PSOIW, and the characteristic of each given problems, the resulting features and solution distribution are nearly same.
2) Regardless of the used methods and the characteristics of the given problems, the solution distributions of using the BWA are relatively worse than that by using the LWA and BWA.
3) In comparison with the solution distributions of using the LWA for both the *ZDT1* (convex) and ZDT2 (concave) problems, the former is in the high density.

For quantitative analysis to these experimental results in Figure 3, Table III gives the performance indexes, i.e. the number of the obtained optimal solutions, $\vec{x}^o$, and the corresponding *FD* and *CR*, of the PSOIW$\alpha$ and PSOIW by respectively using the LWA, BWA, and SWA for each given problem.

From the resulting statistical data shown in Table III, we can see the following features: Firstly, there is the most number of solutions obtained by using the LWA in the given problems even for the ZDT2 one in where a large number of *Pareto*-optimal solutions are in unstable position [11]. Secondly, the solution accuracy of the PSOIW$\alpha$ is superior to that of the PSOIW for each problem. Thirdly, the obtained results of using the LWA in *CR* is the best than that of using the 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.

Based on the above analytical results, the effectiveness and good search ability of the PSOIW$\alpha$ are roughly confirmed. Furthermore, better solution distribution and higher solution accuracy can be observed as well by using either the LWA or SWA. This fact gives an example, i.e. smooth change of their criteria with the growth of time-step can make that the probability of finding good solutions greatly goes up in the same period, $T = 2500$, as evidence.

According to the above comparison and observation, the relationship of domination reflecting the search ability of the PSOIW$\alpha$ with the used dynamically weighted aggregations

TABLE III
PERFORMANCE COMPARISON OF BOTH THE PSOIW$\alpha$ AND PSOIW BY USING THE LWA, BWA, AND SWA, RESPECTIVELY ($\Gamma$ IS SET TO 100).

| problem | aggregation | PSOIW$\alpha$ | | | PSOIW | | |
|---|---|---|---|---|---|---|---|
| | | solution | *FD* | *CR* (%) | solution | *FD* | *CR* (%) |
| *ZDT1* | LWA | **422** | **3.7047** $\times 10^{-7}$ | 89.5 | 418 | $7.5439 \times 10^{-7}$ | **90.0** |
| | BWA | 78 | $4.1231 \times 10^{-4}$ | 48.0 | 63 | $4.3378 \times 10^{-4}$ | 40.0 |
| | SWA | 405 | $2.1356 \times 10^{-5}$ | 81.0 | 334 | $2.7462 \times 10^{-5}$ | 78.0 |
| *ZDT2* | LWA | **102** | **4.3382** $\times 10^{-8}$ | **60.0** | 101 | $1.3421 \times 10^{-4}$ | **60.0** |
| | BWA | 71 | $5.4060 \times 10^{-4}$ | 52.5 | 76 | $6.3600 \times 10^{-4}$ | 55.0 |
| | SWA | 75 | $5.2670 \times 10^{-5}$ | 50.0 | 81 | $2.2592 \times 10^{-4}$ | 51.5 |
| *ZDT3* | LWA | **423** | **6.7482** $\times 10^{-6}$ | **45.5** | 413 | $7.5439 \times 10^{-5}$ | 41.5 |
| | BWA | 165 | $1.7861 \times 10^{-4}$ | 30.0 | 137 | $2.2592 \times 10^{-4}$ | 27.0 |
| | SWA | 321 | $2.1958 \times 10^{-4}$ | 39.0 | 322 | $4.6655 \times 10^{-4}$ | 40.5 |

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

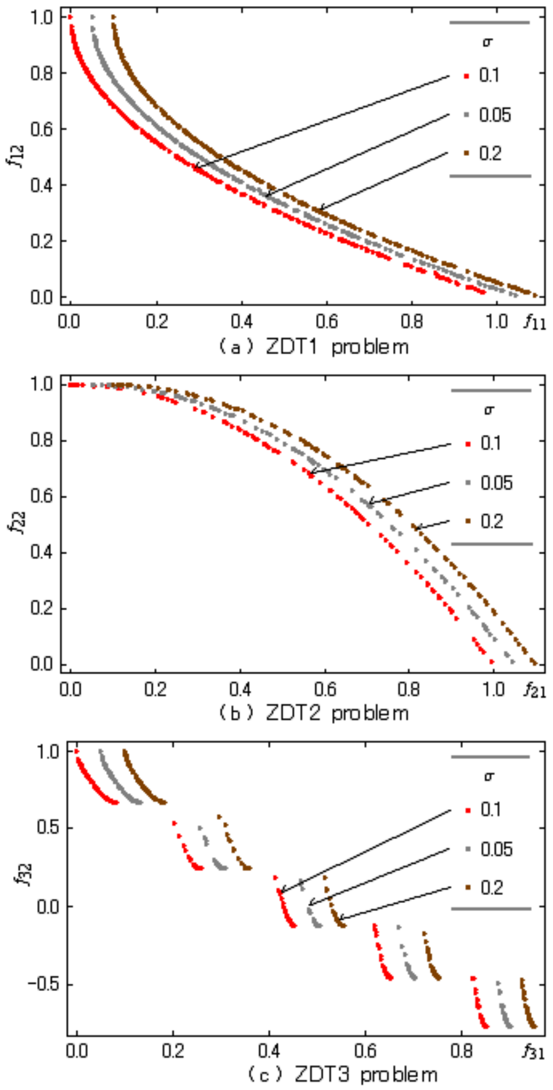


Fig. 4. The solution distributions of the PSOIW$\alpha$ by using the LWA with tuning the parameter $\sigma$.

can be expressed as follows.

$$LWA \succ SWA \succ BWA$$

The relationship of domination means 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.

### B. Effect of the LRS

For investigating the search effect of the LRS used in the PSOIW$\alpha$, the range of the parameter $\sigma$ was adjusted in the following computer experiments. Figure 4 shows the resulting search results of the PSOIW$\alpha$ by using the LWA under the condition of the fixed the number of random points with tuning the parameter value, $\sigma = 0.05, 0.1, 0.2$, respectively. However, the remarkable features of the LRS run cannot be observed with tuning the parameter $\sigma$ from the solution distributions shown in Figure 4.

TABLE IV
THE PERFORMANCE COMPARISON OF THE PSOIW$\alpha$ BY USING THE LWA WITH TUNING THE PARAMETER $\sigma$.

| problem | $\sigma$ | solution | *FD* | *CR* (%) |
|---|---|---|---|---|
| *ZDT1* | 0.1 | **422** | $3.7742 \times 10^{-7}$ | 89.5 |
| | 0.05 | 406 | $2.3279 \times 10^{-5}$ | **92.0** |
| | 0.2 | 373 | **9.1633** $\times 10^{-8}$ | 90.0 |
| *ZDT2* | 0.1 | 102 | **4.3382** $\times 10^{-8}$ | 60.0 |
| | 0.05 | 106 | $4.6589 \times 10^{-8}$ | 63.0 |
| | 0.2 | **125** | $4.4246 \times 10^{-8}$ | **68.5** |
| *ZDT3* | 0.1 | 423 | $6.7482 \times 10^{-6}$ | **45.5** |
| | 0.05 | 420 | **3.2998** $\times 10^{-6}$ | 38.5 |
| | 0.2 | **453** | $7.1231 \times 10^{-5}$ | 44.0 |

Furthermore, Table IV gives the search performance indexes of the PSOIW$\alpha$ in Figure 4. By comparing these statistical data, we cannot decide that whether which case is good or not as yet. This matter means that the random property on the search performance is clearly certified according to the uncertain rule found from the obtained best search results. In other words, this fact suggests that the adopted value of the parameter, $\sigma = 0.1$, is still suitable for dealing with the given problems, and parameter selection to obtain an opportune parameter $\sigma$ is necessary.

### C. Computation Cost

To compare the computation costs of both the PSOIW$\alpha$ and PSOIW run, as an example, the computer experiments[3] were carried out by using the LWA with increasing the dimensional number $n$ of the variable vector for the *ZDT1* problem. The measured running times (RT) of them are shown in Figure 5.

Furthermore, the conformity of the $RT$ with respect to the dimensional number $n$ for both is shown as follows.

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

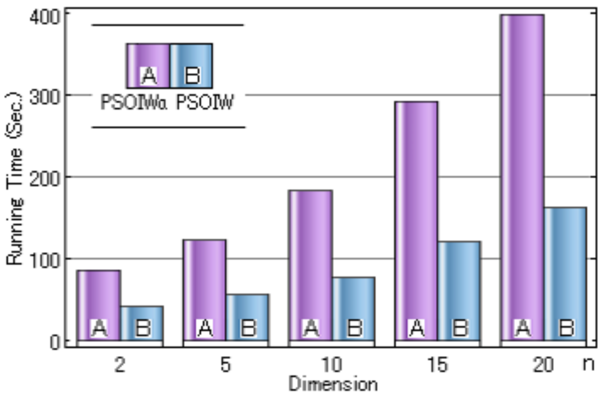


Fig. 5. Running time of both the PSOIW$\alpha$ and PSOIW for dealing with the $ZDT1$ problem in different-dimension, $n$.

$$\begin{cases} RT_{PSOIW\alpha} = 69.951 + 7.641n + 0.448n^2 \\ RT_{PSOIW} = 35.646 + 2.701n + 0.183n^2 \end{cases}$$

Accordingly, comparing with the values of the first-degree and second-degree coefficients in the above two approximate equations, all of the proportional rates between the PSOIW$\alpha$ and PSOIW are more than double.

On the basis of the wide margin between them, it is easily reminded of that the experimental result fits in with "no free lunch" (NFL) theorem [27]. As an application of meta-optimization technique, for example, the method of evolutionary particle swarm optimizer with inertia weight (EPSOIW) [33] can be used to improve the search performance of the original PSOIW. This is because the computation cost of an optimized PSOIW is similar to the original PSOIW except the computation cost of estimating appropriate parameter values of the PSOIW to the give problem.

On the other hand, we can also say that although the computation cost of the PSOIW$\alpha$ extremely depends on the number $U$ of random points, the computation cost is not proportion to the search performance in direct. For interpreting the issue, carrying a lot of experiments out to determine a suitable number to explore is necessary. This is a hot topic to make the use of stochastic optimization methods for efficient search. For the sake of whole structural ingredient, more detailed inspection and analysis on it is omitted here.

## V. Conclusions

In this paper, an improved particle swarm optimizer with inertia weight, called PSOIW$\alpha$, has been presented to multi-objective optimization, MOO. Based on the composition of the PSOIW$\alpha$, it is the most simple expansion of the original PSOIW, which has the advantages of easy-to-operation to realize a hybrid search.

Applications of the PSOIW$\alpha$ 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 using three kinds of dynamically weighted aggregations, respectively, it is observed that the search performance of the proposed PSOIW$\alpha$ is superior to the original PSOIW, and the comparative analysis of the PSOIW$\alpha$ shows that the search performance of using the LWA is better than that of using the BWA and SWA for the given problems. Therefore, it is no exaggeration to say that obtained experimental results offer an important evidence for choosing the dynamically linear weighted sum method to efficiently deal with MOO problems.

It is left for further study to apply the PSOIW$\alpha$ to complex MOO problems in the real-world and to compare the search performance with other EC methods. In order to enhance the adaptability, efficiency, and solution accuracy of the PSOIW$\alpha$, the search strategies and attempts on prediction, intelligent and powerful cooperative PSO algorithms [24], [7], [32] 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] A. Chinchuluun and P. M. Pardalos, "A survey of recent developments in multiobjective optimization," *Ann Oper Res*, vol.154, pp.29-50, 2007.

[2] 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.

[3] 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.

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

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

[6] 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.

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

[8] D. E. Goldberg, *Genetic Algorithm in Search Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1989.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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 91125, USA, 1989.

[16] 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.

[17] 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.

[18] 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.
[19] 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.
[20] J. C. Spall, “Stochastic Optimization,” In: J. Gentle et al. (Eds.) *Handbook of Computational Statistics*, Springer Heidelberg, pp.169-197, 2004.
[21] 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).
[22] 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.
[23] 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.
[24] 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.
[25] F. van den Bergh, “An Analysis of Particle Swarm Optimizers,” *Ph.d thesis*, University of Pretoria, South Africa, 2002.
[26] 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.
[27] 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.
[28] 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.
[29] 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.
[30] H. Zhang, “Multiple Particle Swarm Optimizers with Diversive Curiosity,” *Proceedings of the IAENG International MultiConference of Engineers and Computer Scientists 2010* (IMECS2010), vol.1, pp.174-179, Hong Kong, China, 2010.
[31] 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.
[32] 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.
[33] 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, pp.1746-1753, DOI:10.1109/CEC.2011.594827, New Orleans, LA, USA, 2011.
[34] 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. He has been obtained approximately one hundred thousand dollars of competitive research funding for hosting two research projects to develop the algorithms on inverse optimization and swarm intelligence.