Modified Particle Swarm Optimization Combined with Trigonometric Function and Variable Neighborhood Search

Wenxue Zhang
School of Science, Ningxia Medical University, Yinchuan 750004, China
Email: wxzhang@163.com

Abstract—**Based on the inertia weight and acceleration coefficients of trigonometric function and variable neighborhood search, a modified particle swarm optimization is discussed. The modified PSO with the parameters of the trigonometric function has powerful global exploration at the beginning and ending of the evolution, and strongly local exploitation in the interim of run. Through analyzing the behavior of the PSO, three novel heuristic neighborhood search rules are suggested to enhance the robustness and convergence. The proposed approach demonstrates its superiority in convergence, robustness and solution quality with the experiment tested on the benchmark multimodal function.**

Index Terms—**particle swarm optimization, trigonometric function, variable neighborhood search, inertia weight, acceleration coefficients**

I. INTRODUCTION

The particle swarm optimization (PSO), inspired by social behavior of bird flocking and fish schooling, is a stochastic population-based powerful optimization technique for solving global optimization problems, firstly proposed by Kennedy and Eberhart [1,2]. PSO has many similarities with evolutionary computation techniques such as genetic algorithm (GA). However, unlike GA, PSO has no evolution operators such as crossover and mutation. The advantages of PSO are fewer memory requirement, shorter computational time and easy to implement. So it has gained popularity lately and has already been applied to different fields widely, such as function optimization, artificial neural network training, data mining, scheduling problem, pattern classification and fuzzy system control [3-5]. However, as many evolutionary algorithms, PSO does exhibit some disadvantages: it faces the problem of the appropriate parameters adjustment [6,7]; sometimes it is easy to trap in local optima and severely limited by the high computational cost of the slow convergence rate, especially in complex multimodal optimal problems [8-10].

Fortunately, many researchers are devoted to improve the performance of standard PSO and rich harvests have been obtained in three aspects.

Firstly, a few theoretical studies of particle trajectories can be found, which facilitated the derivation of heuristics to select parameter values for guaranteed convergence to a stable point [11-12].

Secondly, many approaches and strategies are proposed to enhance the performance of PSO without sacrificing the speed of solution significantly by adjusting inertia weight and acceleration coefficients. Namely, there are linearly decreasing inertia weight, increasing inertia weight, randomized inertia weight, exponential decreasing inertia weight, fuzzy adaptive inertia weight, and linearly decreasing acceleration coefficients [7,8,13].

Finally, in order to improve the performance of PSO, various neighborhood search techniques are proposed to increase information sharing among the neighbor particles. Mohais [14] proposed a dynamically adjusted neighborhood. Janson [15] used dynamic hierarchy to define the neighborhood structure. Fan [9] addressed the method of combing PSO with a local simplex search technique. Liu [16] presented a center particle swarm optimization. Unlike other ordinary particles, the center particle has no explicit velocity, and is set to the center of the swarm at every iteration. Furthermore, hybridization of PSO with others evolutionary algorithms has been investigated in many studies [17,18].

In this paper, an improved particle swarm optimization is considered based on the inertia weight and acceleration coefficients of trigonometric function and variable neighborhood search, called TFVNS-PSO. This work differs from the existing ones at least in two aspects: firstly, it proposes the dynamic inertia weight and the dynamic acceleration coefficients that are the trigonometric functions of iterations, which change dynamically in accordance with sigmoid. Therefore, it has powerful global exploration at the beginning and ending of the evolution, and strongly local exploitation in the interim of run. Secondly, it presents several novel heuristic neighborhood search strategies to balance the convergence and diversity of particle swarm. Thus it overcomes premature convergence and improves the rate of convergence simultaneously.

The remaining of this paper is organized as follows: The next section reviews the standard PSO. Section □ presents the structure and algorithm of TFVNS-PSO. The robustness and efficiency of TFVNS-PSO is illustrated in solving benchmark function in Section IV. Finally, section V summarizes and draws conclusions.

doi:10.4304/jcp.7.6.1377-1384

II. Standard Particle Swarm Optimization

The standard PSO is proposed by Shi and Eberhart in 1998 [19]. Theoretically, given a d-dimension minimum optimization problem $f(X)$, for the i th particle in the t th generation, let $v_{id}(t)\in[V_{\min},V_{\max}]$, where $V_{\min}$ and $V_{\max}$ are constants depended on given problem; let $x_{id}(t)\in[X_{\min},X_{\max}]$ be the location of the particle, where $X_{\min}$ and $X_{\max}$ are the boundaries of the search space. The location of the best fitness achieved so far by the i th particle is denoted as $p_{id}(t)$ (memorized by every particle) and the index of the global best fitness by the whole population, as $p_{gd}(t)$ (memorized in a common repository). The velocity and position update equations are given as:

$$v_{id}(t+1)=\omega v_{id}(t)+c_1\xi(p_{id}(t)-x_{id}(t)) \\ +c_2\eta(p_{gd}(t)-x_{id}(t)) \tag{1}$$

$$x_{id}(t+1)=v_{id}(t+1)+x_{id}(t), i=1,2,...,m \tag{2}$$

where m is the population size; c_1 and c_2, known as the cognitive and social parameters respectively, are named acceleration coefficients; ξ and η are elements from two uniform random sequences in the range $[0,1]$, namely, $\xi,\eta\in U(0,1)$.

The best positions are updated at every iteration step as follows:

$$P_i(t+1)=\begin{cases}X_i(t+1), if\ f(X_i(t+1))<f(P_i(t))\\ P_i(t), \quad otherwise\end{cases} \tag{3}$$

In recent versions, the parameters are selected such that the relation [12-14],

$$\omega>\tfrac{1}{2}(c_1+c_2)-1 \tag{4}$$

where, $-1\le\omega\le1$ (5)

and $0\le c_1+c_2\le4$ (6)

The procedure of PSO is given as follows:

```
Step1: Initialization
Step1.1 $t=0$, $v_i(0)$ and $x_i(0)$;
Step1.2 Calculate $p_i(0)$ and $p_g(0)$;
Step2: While termination criteria is satisfied do
Step2.1 $for(i=1;i<=m;i++)\{$
        Update $v_i(t)$ using (1);
        Update $x_i(t)$ using (2);
        Update $p_i(t)$ using (3);
        if ( $f(p_i(t))<f(p_g(t))$ ) then $p_g(t)=p_i(t)$;
        } //end for
Step2.2 $t=t+1$;
Step3: Output.
```

III. TFVNS-PSO

This section proposes the dynamic inertia weight and the dynamic acceleration coefficients that are the trigonometric functions of iterations, such that TFVNS-PSO has powerful global exploration at the beginning and ending of the evolution, and strong local exploitation in the interim of run. In order to balance the convergence and diversity of particle swarm, several novel heuristic neighborhood search strategies are presented in following.

A. The inertia weight of trigonometric function (TFIW)

A vital issue of the intelligent optimization methods is to balance global exploration and local exploitation. Generally, in order to avoid trapping in local optimum at the very start, the coarse exploration is necessary in the initial iterations; in order to turn away sightless searching whole space, the stronger exploitation is required in the medium-term iterations; in order to jump out local optimum, the powerful exploration is the only way in later iterations.

As far as PSO goes, the balance between global exploration and local exploitation is achieved chiefly by adjusting inertia weight. A higher value of the inertia weight offers sharp changes in velocity, which means better exploration. However, smaller inertia weight implies less variation in velocity to provide better exploitation.

But the change trend of inertia weight is only in one direction and the decreasing direction is popular in the existing studies [7,8,13]. A novel inertia weight of trigonometric function (TFIW), therefore, is proposed to address this problem as follows:

$$\omega'=\left|\cos(\tfrac{t}{T}\pi)\right|, \tag{7}$$

$$\omega=\begin{cases}\omega' & \omega'\ge\alpha\\ \omega'+\alpha & \omega'<\alpha\end{cases} \tag{8}$$

where, T is a maximum number of iterations; t is a iterative counter, $t\in\{0,1,...,T\}$; α is a constant, $\alpha\in[0.1,0.4]$, $\alpha=0.25$ is proposed. The corresponding TFIW curve is depicted in Figure 1 for $\alpha=0.25$ and $T=2000$.

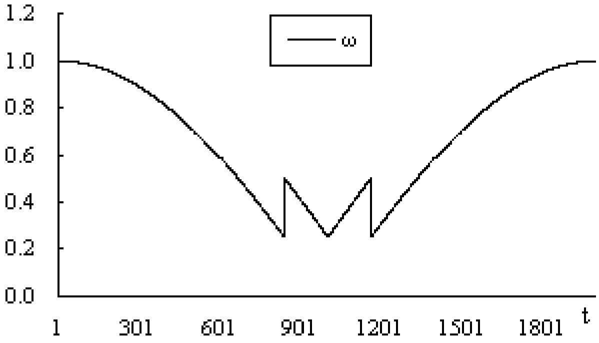

Figure 1. The TFIW curve.

Farther, TFIW is a nonlinear function of the present iteration number t and maximum number T of iterations. TFIW slowly decreases in the initial search (the trend of the function slope), relatively sharper in the mid-iteration and relatively slow in the later iterations. Thus, the exploration and exploitation are balanced dynamically.

B. The acceleration coefficients (c1 and c2) of trigonometric function (TFAC)

The higher value of cognitive parameter c_1 and social parameter c_2 may provide powerful exploration and strong exploitation, respectively [20]. Therefore, the higher c_1 and smaller c_2 must be satisfied in the initial and later of the iterations, and the smaller c_1 and higher c_2 must be satisfied in the middle of the iterations, which should help the PSO to approach the optimum solution quickly. Consequently, this paper proposes the novel dynamic acceleration coefficients of trigonometric function for improved performance of PSO algorithm as follows:

$$c_1^{'} = b \times \omega^{'} \times \omega^{'} \tag{9}$$

$$c_1 = \begin{cases} c_1^{'} & c_1^{'} \geq \beta \\ c_1^{'} + \beta & c_1^{'} < \beta \end{cases} \tag{10}$$

$$c_2^{'} = b - c_1^{'} \tag{11}$$

$$c_2 = \begin{cases} c_2^{'} & c_2^{'} \geq \gamma \\ c_2^{'} + \gamma & c_2^{'} < \gamma \end{cases} \tag{12}$$

where, b is a constant, $b \in (0,4)$, $b = 2$ is proposed; β is a constant, $\beta \in [0.1, 0.4]$, $\beta = 0.25$ is proposed; γ is a constant, $\gamma \in [0.1, 0.4]$, $\gamma = 0.25$ is proposed. The corresponding TFAC curves are depicted in Figure 2 for $b = 2$, $\alpha = 0.25$, $\beta = 0.25$, $\gamma = 0.25$ and $T = 2000$.

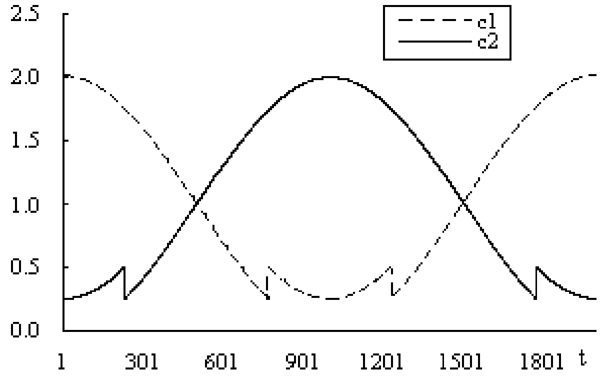

Figure 2. The TFAC curves.

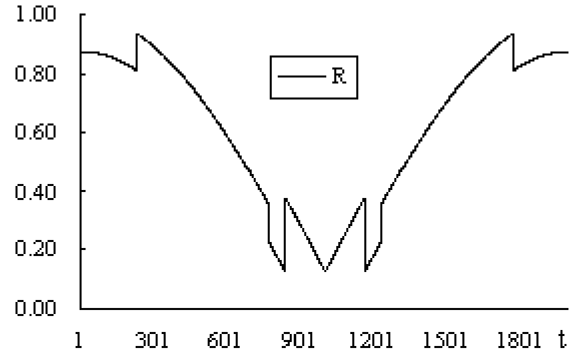

Figure 3. The R curve.

Let, $R = \omega - \frac{1}{2}(c_1 + c_2) + 1$ (13)

The corresponding R curve is depicted in Figure 3 for $b = 2$, $\alpha = 0.25$, $\beta = 0.25$, $\gamma = 0.25$ and $T = 2000$.

Figure 3 shows the R curve with $b = 2$, $\alpha = 0.25$, $\beta = 0.25$, $\gamma = 0.25$, and $T = 2000$, from which it's clearly demonstrated that inequality (4) can be always satisfied with proposed dynamic parameter selection mechanism.

C. Variable neighborhood search

A neighborhood search algorithm typically starts from a solution (that is, an assignment of values to its decision variables) and moves from solutions to neighboring solutions in hope of improving a function f. The function f measures the quality of solutions to the problem at hand. The main operation of a neighborhood search algorithm amounts to moving from a solution s to one of its neighbors. The set of neighboring solutions of s, denoted by $N(s)$, is called the *neighborhood* of s. At a specific computation step, some of these neighbors may be *legal*, in which case they may be selected, or they may be *forbidden*. Once the legal neighbors are identified (by operation L), the local search selects one of them and decides whether to move to this neighbor or to stay at s (operation S). These concepts, which define the *moves* in local search, are illustrated as follows [21].

The basic neighborhood search template

```
Function NeighborhoodSearch{
    s:=GenerateInitalSolution();
    $s^* = s$;
    for $k = 1$ to MaxTrials do{
        if $satisfiable(s) \wedge f(s) < f(s^*)$
            then { $s^* = s$; } //end if
        $s = S(L(N(s), s), s)$;
    }//end for
    Return $s^* = s$;
    }//end function NeighborhoodSearch
```

Variable neighborhood search (VNS) is a recent metaheuristic for solving combinatorial and global optimization problems whose basic idea is systematic change of neighborhood both within a descent phase to find a local optimum and in a perturbation phase to get out of the corresponding valley [22]. Its development has been rapid, mainly to allow solution of large problem instances, such as highly-constrained nurse rostering problems, multi-resource generalized assignment problem, jobshop scheduling problems [23-25].

A key feature of particle swarm optimization algorithm is the fitness information shared with individuals in a particle's neighborhood. The kind of neighborhood structure that is used affects the rate at which information is disseminated throughout the population [20]. This paper presents several entirely new heuristic neighborhood search strategies to improve the performance of convergence and robustness.

Definition 1: The set of *neighborhood of the particle* s, denoted by $N(s)$, is defined as

$$N(s) = \{s^* \mid x^{s^*} = v^{s^*} + x^s\}, \tag{14}$$

where, $v^{s^*} = \omega v^s + c_1\xi(p^s - x^s) + c_2\eta(p^s_{gd} - x^s)$ and $v^{s^*} \in [V_{\min}, V_{\max}]$.

Definition 2: The set of *legal neighborhood of the particle* s, denoted by $L(N(s), s)$, is defined as

$$L(N(s), s) = \{s^* \mid s^* \in N(s) \wedge x^{s^*} \in [X_{\min}, X_{\max}]\}. \tag{15}$$

D. Heuristic neighborhood search for the original particle (HNSOP)

As one point of view, in order to improve the rate of convergence, when a new particle, denoted s, is created randomly, $f(s) < f(s^0)$ (minimal optimization function f, original particle s^0, $s \in L(N(s^0), s^0)$) is expected. As another point of view, in order to avoid premature convergence, a degrading particle is accepted with probability. To this addressed problem, a wholly new heuristic neighborhood search for the original particle (HNSOP) is proposed.

Definition 3: The set of *neighborhood of the original particle,* denoted by $N_{OP}(s^0)$, is defined as

$$\begin{aligned} N_{OP}(s^0) = \{s^i \mid & ((s^i \in L(N(s^{i-1}), s^{i-1}), f(s^i) < f(s^0), \\ & i \in \{1,2,...,m-1\}) \wedge (f(s^{j-1}) \geq f(s^0), \\ & j \in \{1,2,...,i\})) \\ & \vee ((s^i \in L(N(s^{i-1}), s^{i-1}), f(s^i) \geq f(s^0), i = m) \\ & \wedge (f(s^i) > f(s^0), i \in \{1,2,...,m-1\}))\} \end{aligned} \tag{16}$$

The heuristic neighborhood search for the original particle

```
Function HNSOP ( $f, s^0, t, T$ ) {
        Updated $\omega, c_1$ and $c_2$;
        Selected $s \in N_{OP}(s^0)$ ;
        Return $s^0 := s$;
        }//end Function HNSOP
```

E. Heuristic neighborhood search for the taboo particle (HNSTP)

By analyzing the behavior of the PSO, particle s should be avoided in population repeatedly to improve the efficiency of PSO and to jump out the local optimum. But particle s emerges in the population in the high-frequency, which may imply particle s with the more information of the optimal solution. Thus, if particle s is retained in the population and particle s shares information with others particles, the efficiency of finding the optimal solution will be increased acutely. Contrary, particle s emerges in the population in the high-frequency, which may mean be trapped in local optimum. So in order to jump out the local optimum and augment swarm's diversity, the particle s must be mutated. Therefore, an entirely new heuristic neighborhood search for the taboo particle s (HNSTP) is put forward.

Definition 4: A *good population* S_{good} is a set of all particles of the best fitness achieved so far by each particle.

Definition 5: A set $N_{TP1}(s, S_{good})$ is defined as

$$\begin{aligned} N_{TP1}(s, S_{good}) = \{s^{'} \mid x_i^{s'} \in \{x_i^s, x_i^{s*}\} \wedge (\forall i, \neg\, x_i^{s'} = x_i^s) \\ \wedge (\forall i, \neg\, x_i^{s'} = x_i^{s^*}), s^* \in S_{good}, \\ i \in \{1,2,...,d\}\} \end{aligned} \tag{17}$$

where, $x^s = (x_1^s, x_2^s, ..., x_d^s)$.

Definition 6: A set $N_{TP2}(s)$ is defined as

$$\begin{aligned} N_{TP2}(s) = \{s^{'} \mid x_i^{s'} \in \{x_i^s\}, i \in \{d_1, d_2\}, d_1 \in \{1,2,...,d-1\}, \\ d_2 \in \{2,...,d-1,d\}, d_1 < d_2, x_k^{s'} \notin \{x_k^s\} \wedge\ x_k^{s'} \\ \text{is legal}, k \in \{1,...,d_1-1\} \cup \{d_2+1,...,d_1\}\} \end{aligned} \tag{18}$$

Definition 7: The set of *neighborhood of the taboo particle,* denoted by $N_{TP}(s, S_{good})$, is defined as

$$N_{TP}(s, S_{good}) = N_{TP1}(s, S_{good}) \cup N_{TP2}(s) \tag{19}$$

The heuristic neighborhood search for the taboo particle

```
Function HNSTP ( $f, \&s, S_{good}, \&s^{gb}$ ) {
        // $s^{gb}$ is the global best particle
        Select best $s^*$ from $\mathrm{N_{TP1}}(s, S_{good})$;
        if ( $x^s == x^{s^*}$ ) then
                Select best $s^*$ from $\mathrm{N_{TP2}}(s,)$ ; //end if
        if ( $f(s^*) < f(s)$ ) then
                $s := s^*$; //end if
        if ( $f(s) < f(s^{gb})$ ) then
                $s^{gb} := s$; //end if
        }//end Function HNSTP
```

F. Heuristic neighborhood search for the global best particle (HNSGBP)

On the one hand, the global best particle, denoted s^{gb}, has excellent information related to global optimum solution. On the other hand, the good population S_{good} also has finer experience. So sharing knowledge among them is likely to advance the convergence rate and the robustness of PSO. Therefore, an entirely new heuristic neighborhood search for the global best particle s^{gb} (HNSGBP) is presented.

The heuristic neighborhood search for the global best particle

```
Function HNSGBP ( $f, m, \&S, S_{good}, \&s^{gb}$ ) {
        // $S$ is current swarm, $m$ is swarm size.
        Select $m$ better particles from
        $\mathrm{N_{TP1}}(s^{gb}, S_{good}) \cup S$ ;
        Assign these $m$ better particles to $S$;
        Update $s^{gb}$;
        }//end Function HNSGBP
```

G. *The structure of TFVNS-PSO*

```
*The procedure of TFVNS-PSO is given as follows:*
**Step1:** Initialization
Step1.1 Initialize iterative counter be $t=0$, the $m$
        random velocities $v_i(0)$ and the $m$ random
        positions $x_i(0)$ of the particles, maximum
        number of iterations be $T$;
Step1.2 Initialize $\omega$, $c_1$ and $c_2$;
Step1.3 Calculate $p_i(0)$ and $p_g(0)$;
Step1.4 Initialize the TabuList of the $x_i(t)$ be empty;
**Step2:** While termination criteria is satisfied do
Step2.1 Update $\omega$, $c_1$ and $c_2$;
Step2.2 $for(i=1;i<=m;i++)\{$
                Update $v_i(t)$ using (1);
                Update $x_i(t)$ using (2);
        //Using heuristic neighborhood search for the
        // original particle;
        //to accept a bad particle by a certain probability;
        Function HNSOP ( $f,s^0,t,T$ );
        } //end Step2.2
Step2.3 $for(i=1;i<=m;i++)\{$
        if $x_i(t)$ in TabuList then{
        //Using heuristic neighborhood search for the
        //taboo particle;
        //to enhance the escaping local optimization
        //ability of the particle by augmenting swarm's
        //diversity;
                Function HNSTP ( $f,\&s,S_{good},\&s^{gb}$ );}
        else {
                Update $p_i(t)$ using (3);
                if( $f(p_i(t))<f(p_g(t))$ ) then
                $p_g(t)=p_i(t)$ ;} //end if
        } //end Step2.3
Step2.4 $for(i=1;i<=m;i++)\{$
        //Using guided neighborhood search by global
        //best particle
        //to increase the rate of swarm's convergence
        Function HNSGBP ( $f,m,\&S,S_{good},\&s^{gb}$ );
        } //end Step2.4
Step2.5 $t=t+1$;
**Step3:** Output.
*//end the procedure of TFVNS-PSO*
```

IV. EXPERIMENT AND DISCUSSION

A. *Experiment*

To illustrate the effectiveness and robustness of TFVNS-PSO algorithm for optimization problems, a set of 5 representative benchmark functions was employed to evaluate it in comparison with LPSO, FPSO, and RPSO [7].

The same initialization range $[-5,5]$ and the constants $\mathrm{V}_{\min}=\mathrm{X}_{\min}=-10$ and $\mathrm{V}_{\max}=\mathrm{X}_{\max}=10$ were used. At each algorithm, the number of particles was 20. The max iterations of these algorithms were all 300. A total of 30 repeated for each experimental setting were conducted and the averages as well as the standard deviations of these optimization results are computed.

The table I lists the testing results on the functions of f_1, f_2, f_3, f_4 and f_5.

TABLE I.
THE AVERAGES OF THE OPTIMIZATION RESULTS FOR THE FIVE BENCHMARK FUNCTIONS AND THEIR CORRESPONDING STANDARD DEVIATIONS IN THE PARENTHESES.

f	n	LPSO	FPSO	RPSO	TFVNS-PSO
f_1	20	1039 (719.9)	690.1 (647.0)	20.80 (9.240)	**2.9094** **(1.4055)**
	80	3403 (20294)	20011 (14785)	484.8 (1726)	**38.9435** **(5.6327)**
	140	62010 (30544)	66989 (44570)	1719 (8026)	**123.5730** (36.6839)
	200	124344 (94505)	85569 (72089)	10288 (36593)	**251.6320** **(102.2045)**
f_2	20	94.34 (23.28)	67.30 (34.96)	6.749 (9.819)	**3.1193** **(1.6784)**
	80	694.4 (80.38)	376.6 (243.8)	65.22 (118.1)	**51.8321** **(10.3386)**
	140	1200 (243.2)	928.5 (459.5)	164.2 (175.7)	**131.1524** **(24.4465)**
	200	1859 (373.2)	1005 (613.5)	241.4 (396.5)	**216.5155** **(29.9613)**
f_3	20	6.874 (3.491)	7.910 (4.948)	9.037 (6.255)	**0.0000** **(0.0000)**
	80	91.55 (49.25)	65.34 (41.21)	10.76 (15.74)	**0.2341** **(0.2927)**
	140	195.8 (73.12)	95.04 (75.84)	18.39 (45.49)	**1.1845** **(0.6650)**
	200	261.9 (93.90)	193.6 (120.0)	25.86 (44.27)	**3.9817** **(1.9490)**
f_4	20	0.45 (0.19)	0.27 (0.14)	0.02 (0.09)	**0.0018** **(0.0043)**
	80	0.87 (0.11)	0.66 (0.28)	0.13 (0.29)	**0.0013** **(0.0025)**
	140	0.93 (0.14)	0.79 (0.25)	0.11 (0.31)	**0.0040** **(0.0033)**
	200	1.00 (0.06)	0.90 (0.21)	0.05 (0.13)	**0.0255** **(0.0307**)
f_5	20	0.06 (0.02)	0.05 (0.02)	**0.03** (0.01)	0.0372 (**0.0000**)
	80	0.16 (0.03)	0.15 (0.04)	**0.07** (0.06)	0.0781 (**0.0223**)
	140	0.22 (0.02)	0.21 (0.06)	**0.08** (0.07)	0.1526 (**0.0256**)
	200	0.27 (0.04)	0.25 (0.04)	**0.08** (0.07)	0.1847 (**0.0212**)

Benchmark functions

$\min f_1(x)=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$,
$-10\le x_i\le 10, i\in\{1,2,...,n\}$.

$$\min f_2(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10),$$
$$-10\le x_i\le 10, i\in\{1,2,...,n\}.$$
$$\min f_3(x)=\sum_{i=1}^{n}x_i^2, -10\le x_i\le 10, \quad i\in\{1,2,...,n\}.$$
$$\min f_4(x)=\tfrac{1}{4000}\sum_{i=1}^{n}x_i^2+\prod_{i=1}^{n}\cos(\tfrac{x_i}{\sqrt{i}})+1,$$
$$-10\le x_i\le 10, i\in\{1,2,...,n\}.$$
$$\min f_5(x)=0.5+\frac{(\sin\sqrt{\sum_{i=1}^{n}x_i^2})^2-0.5}{(1+0.001\sum_{i=1}^{n}x_i^2)^2},$$
$$-10\le x_i\le 10, i\in\{1,2,...,n\}.$$

From table I, the TFVNS-PSO is obviously excellent in both convergence and robustness for f_1, f_2, f_3 and f_4. For f_5, the TFVNS-PSO shows little efficiency, but it improves the robustness considerably.

B. Discussion

The TFIWAC denotes without employing HNSOP, HNSTP and HNSGBP, namely, only improved PSO based on the TFIW and TFAC. A modified TFIWAC using HNSOP is denoted by HNSOP. And a modified TFIWAC using HNSTP is denoted by HNSTP. And a modified TFIWAC using HNSGBP is denoted by HNSGBP.

Further, to demonstrate the efficiency of TFIWAC, HNSOP, HNSTP and HNSGBP, the benchmark function f_6 was employed. Many authors tested algorithm using it widely.

Benchmark function

$$\min f_6(x)=\tfrac{1}{100}\sum_{i=1}^{100}(x_i^4-16x_i^2+5x_i),$$
$$-10\le x_i\le 10, i\in\{1,2,...,100\}.$$

The function f_6 has about 2^{100} the local optimums in the feasible solution space, and the global optimum is -78.3323.

With respect to the expected number of required function evaluations, the terminate criteria is $\left|f_6^{\min}-f_6^*\right|<10^{-3}$ ($f_6^*=-78.3323$). The PSO parameters are set to their default values, $\omega=0.729$, $c_1=2.0$, $c_2=1.8$. The TFIWAC, HNSOP, HNSTP and HNSGBP parameters are set $b=2$, $\alpha=0.25$, $\beta=0.25$, $\gamma=0.25$ and $T=6000$. The swarm size is 100 and the maximum generation is 6000 in five algorithms. Each group test includes 30 independent experiments. The experimental results are shown in the figure 4 and figure 5.

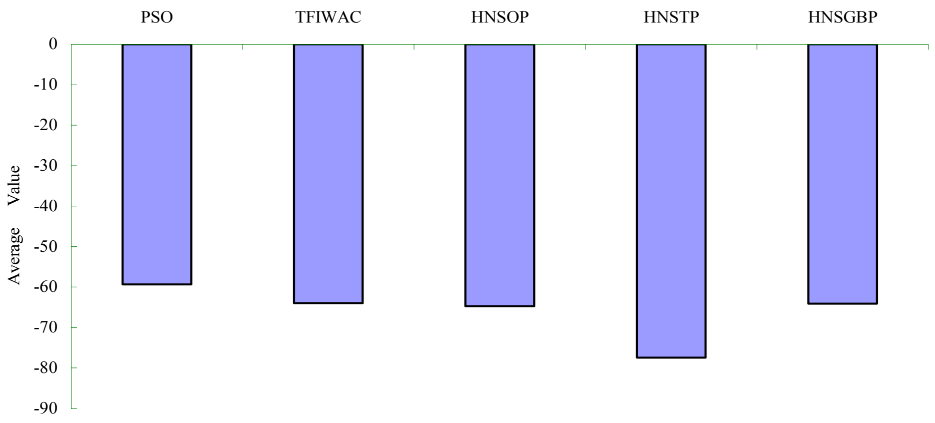

Figure 4. The average value.

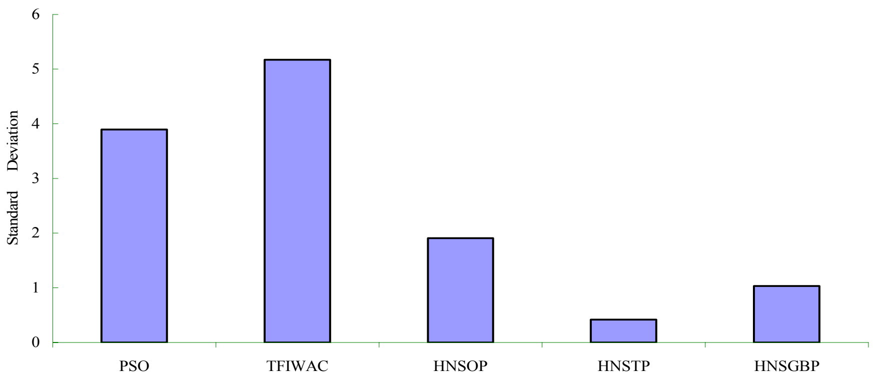

Figure 5. The standard deviation.

Obviously, the several conclusions can be got from the figure 4 and figure 5. Firstly, the TFIWAC is superior to PSO in the efficiency (the smaller average, the better efficiency; and the smaller standard deviation, the better robustness.). Secondly, the HNSTP is the most excellent in both convergence and robustness. Finally, the HNSOP and the HNSGBP improve the efficiency of the TFIWAC little, but they enhance the robustness of the TFIWAC remarkably.

V. Conclusions

A novel improved PSO algorithm is proposed in this paper. Our main contribution is the modification of the parameters update rule and the heuristic variable neighborhood search strategies of particle swarm optimization. On the one hand, dynamic inertia weight based on the trigonometric functions of iterations is discussed, and then the dynamic acceleration coefficients based on the trigonometric functions of iterations is considered under the inertia weight of trigonometric function. On the other hand, through analyzing the behavior of the PSO, three novel heuristic variable neighborhood search rules are suggested to enhance the robustness and convergence. Compared with PSO and the other algorithms, the TFVNS-PSO demonstrates its superiority in convergence, robustness and solution quality, because the TFVNS-PSO has powerful global exploration at the beginning and ending of the evolution, and strongly local exploitation in the interim of run.

In the future, the sensitivity of TFVNS-PSO parameters: b, α, β, γ and T should be discussed. The proposed approach should be used to solve other discrete combinatorial optimization problems such as flow shop scheduling problem and job shop scheduling problem.

Acknowledgment

The authors would like to thank anonymous reviewers and the editors for constructive comments and suggestions. This work is supported by the Special Talent Foundation of Ningxia Medical University in 2011.

References

[1] J. Kennedy, R. Eberhart, "Particle Swarm optimization," *IEEE Int. Conf. on Neural Networks*, Perth, WA, Australia, vol.4, pp. 1942–1948, Nov/Dec.1995.

[2] J. Kennedy, R.C. Eberhart, "A new optimizer using particle swarm theory," *Sixth Int. Symp. on Micro Machine and Human Science*, Nagoya, Japan, pp. 39–43, Oct. 1995.

[3] C. J. Lin, M. H. Hsieh, "Classification of mental task from EEG data using neural networks based on particle swarm optimization," *Neurocomputing,* vol.72 Issue 4/6, pp. 1121-1130, January 2009.

[4] G. Moslehi, M. Mahnam, "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search," *International Journal of Production Economics*, Vol. 129, Issue 1, pp. 14-22, January 2011.

[5] C. Karakuzu, "Retraction notice to: Fuzzy controller training using particle swarm optimization for nonlinear system control," ISA Transactions, Vol. 48, Issue 2, pp. 245, April 2009.

[6] P. Chakraborty, S. Das, G. G. Roy, et al., "On convergence of the multi-objective particle swarm optimizers," *Information Sciences*, Vol. 181, Issue 8, pp. 1411-1425, April 2011.

[7] Q. Luo, D.Y. Yi, "A co-evolving framework for robust particle swarm optimization," *Applied Mathematics and Computation,* Vol. 199, Issue 2, pp. 611–622, June 2008.

[8] X.M. Yang, J.S. Yuan, J.Y Yuan, H.N. Mao, "A modified particle swarm optimizer with dynamic adaptation." *Applied Mathematics and Computation*, Vol. 189, Issue 2, pp. 1205–1213, June 2007.

[9] S.F. Fan, E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, Vol. 181, Issue 2, pp. 527–548, September 2007.

[10] Y. Jiang, T.S. Hu, C.C. Huang, X.N. Wu, "An improved particle swarm optimization algorithm," *Applied Mathematics and Computation*, Vol.193, Issue 1, pp. 231–239, October 2007.

[11] F. van den Bergh, A.P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Sciences*, Vol. 176, Issue 1, pp. 937–971, April 2006.

[12] M. Jiang, Y.P. Luo, S.Y. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, Vol.102, Issue 1, pp. 8–16, April 2007.

[13] B. Jiao, Z.G. Lian, X.S. Gu, "A dynamic inertia weight particle swarm optimization algorithm," *Chaos, Solitons and Fractals*, Vol.37, Issue 3, pp. 698–705, August 2008.

[14] A. S. Mohais, C. Ward, C. Posthoff, "Randomized directed neighborhoods with edge migration in particle swarm optimization," *Proceedings of the IEEE Congress on Evolutionary Computation,* Vol.1, pp.548–555, June 2004.

[15] S. Janson, M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Trans. on Syst., Man, and Cybern., Part B: Cybern.,* Vol. 35, Issue 6, pp. 1272–1282, Dec. 2005.

[16] Y. Liu, Z. Qin, Z.W. Shi, J. Lu, "Center particle swarm optimization," *Neurocomputing,* Vol. 70, Issue 4/6, pp. 672–679, January 2007.

[17] X.H. Shi, Y.C. Liang, H.P. Lee, *et al.*, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Information Processing Letters*, Vol. 93, Issue 5, pp. 255–261, March 2005.

[18] M. Senthil Arumugam, M.V.C. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing*, Vol. 8, Issue 1, pp. 324–336, January 2008.

[19] Y.H. Shi, R. Eberhart, "A modified particle swarm optimizer," *IEEE Congress on Evolutionary Computation,* pp. 69–73, May 1998.

[20] A. Ratnaweera, S.K. Halgamuge, H.C. Watson. "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE transactions on evolutionary computation*, Vol.8, Issue 3, pp. 240–255, June 2004.

[21] P.V. Hentenryck, L. Michel, *Constraint-based local search*, the MIT Press, Cambridge, 2009.

[22] Z. Naji-Azimi, M. Salari, B. Golden, et al., "Variable neighborhood search for the cost constrained minimum label spanning tree and label constrained minimum spanning tree problems," *Computers & Operations*

Research, Vol. 37, Issue 11, pp. 1952-1964, November 2010.

[23] E. K. Burke, J.P. Li, R. Qu, "A hybrid model of integer programming and variable neighborhood search for highly-constrained nurse rostering problems," *European Journal of Operational Research*, Vol. 203, Issue 2, pp. 484-493, June 2010.

[24] S. Mitrović-Minić, A. P. Punnen, "Local search intensified: Very large-scale variable neighborhood search for the multi-resource generalized assignment problem," *Discrete Optimization*, Vol. 6, Issue 4, pp. 370-377, November 2009.

[25] A. Bagheri, M. Zandieh, "Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—Variable neighborhood search approach," *Journal of Manufacturing Systems*, Vol. 30, Issue 1, pp. 8-15, January 2011.

Wenxue Zhang was born in Ningxa province, China, in 1981. He received the B.S. degree in applied mathematics from Ningxia University, Yinchuan, PRC, in 2004 and the M.S. degree in computer software theory from Ningxia University, Yinchuan, PRC, in 2007, and the Ph.D. degree in management science and engineering from the University of Science and Technology Beijing, Beijing, PRC, in 2011. Currently, he is a teacher of school of science, Ningxia Medical University, Yinchuan, PRC. His research interests are soft computing, production planning and scheduling, logistics and supply chain management, and emergency management.