Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 194706, 14 pages
http://dx.doi.org/10.1155/2014/194706

Research Article

Human Behavior-Based Particle Swarm Optimization

Hao Liu,[1,2] Gang Xu,[3] Gui-yan Ding,[2] and Yu-bo Sun[2]

[1] *School of Mathematics and Statistics, Beijing Institute of Technology, Beijing 100081, China*
[2] *School of Science, University of Science and Technology Liaoning, Anshan 114051, China*
[3] *Department of Mathematics, Nanchang University, Nanchang 330031, China*

Correspondence should be addressed to Hao Liu; liuhao123@ustl.edu.cn

Received 3 December 2013; Accepted 17 March 2014; Published 17 April 2014

Academic Editors: P. Agarwal, V. Bhatnagar, and Y. Zhang

Particle swarm optimization (PSO) has attracted many researchers interested in dealing with various optimization problems, owing to its easy implementation, few tuned parameters, and acceptable performance. However, the algorithm is easy to trap in the local optima because of rapid losing of the population diversity. Therefore, improving the performance of PSO and decreasing the dependence on parameters are two important research hot points. In this paper, we present a human behavior-based PSO, which is called HPSO. There are two remarkable differences between PSO and HPSO. First, the global worst particle was introduced into the velocity equation of PSO, which is endowed with random weight which obeys the standard normal distribution; this strategy is conducive to trade off exploration and exploitation ability of PSO. Second, we eliminate the two acceleration coefficients c_1 and c_2 in the standard PSO (SPSO) to reduce the parameters sensitivity of solved problems. Experimental results on 28 benchmark functions, which consist of unimodal, multimodal, rotated, and shifted high-dimensional functions, demonstrate the high performance of the proposed algorithm in terms of convergence accuracy and speed with lower computation cost.

1. Introduction

Particle swarm optimization (PSO) [1] is a population-based intelligent algorithm, and it has been widely employed to solve various kinds of numerical and combinational optimization problems because of its simplicity, fast convergence, and high performance.

Researchers have proposed various modified versions of PSO to improve its performance; however, there still are premature or lower convergence rate problems. In the PSO research, how to increase population diversity to enhance the precision of solutions and how to speed up convergence rate with least computation cost are two vital issues. Generally speaking, there are four strategies to fulfill these targets as follows.

(1) Tuning control parameters. As for inertial weight, linearly decreasing inertial weight [2], fuzzy adaptive inertial weight [3], rand inertial weight [4], and adaptive inertial weight based on velocity information [5], they can enhance the performance of PSO. Concerning acceleration coefficients, the time-varying acceleration coefficients [6] are widely used. Clerc and Kennedy analyzed the convergence behavior by introducing constriction factor [7], which is proved to be equivalent to the inertial weight [8].

(2) Hybrid PSO, which hybridizes other heuristic operators to increase population diversity. The genetic operators have been hybridized with PSO, such as selection operator [9], crossover operator [10], and mutation operator [11]. Similarly, differential evolution algorithm [12], ant colony optimization [13], and local search strategy [14] have been introduced into PSO.

(3) Changing the topological structure. The global and local versions of PSO are the main type of swarm topologies. The global version converges fast with the disadvantage of trapping in local optima, while the local version can obtain a better solution with slower convergence [15]. The Von Neumann topology is helpful for solving multimodal problems and may perform better than other topologies including the global version [16].

(4) Eliminating the velocity formula. Kennedy proposed the bare-bones PSO (BPSO) [17] and variants of BPSO [18, 19]. Sun et al. proposed quantum-behaved PSO (QPSO) and relative convergence analysis [20, 21].

Initialize Parameters:
$N \Leftarrow$ population size;
$D \Leftarrow$ the dimensionality of search space;
$T \Leftarrow$ the number of maximum iteration;
$w \Leftarrow$ the inertial weight;
$\left[x_{\min}^{d}, x_{\max}^{d}\right] \Leftarrow$ the allowable position boundaries, $d = 1, 2, \ldots, D$;
$\left[v_{\min}^{d}, v_{\max}^{d}\right] \Leftarrow$ the allowable velocity boundaries, $d = 1, 2, \ldots, D$;
Initialize Population: $V_i = \left(v_i^1, v_i^2, \ldots, v_i^D\right)$, $X_i = \left(x_i^1, x_i^2, \ldots, x_i^D\right)$, $\quad i = 1, 2, \ldots, N$;
$v_i^d \Leftarrow v_{\min}^d + \text{rand}_i^d \cdot (v_{\max}^d - v_{\min}^d)$;
$x_i^d \Leftarrow x_{\min}^d + \text{rand}_i^d \cdot (x_{\max}^d - x_{\min}^d)$;
Initialize *P*best**,** *G*best **and** *G*worst**:**
Evaluate fitness of all particles in $X = \{X_1, X_2, \ldots, X_N\}$;
*P*best $\Leftarrow X$;
*G*best $\Leftarrow \arg\min\{f(P\text{best}_1), f(P\text{best}_2), \ldots, f(P\text{best}_N)\}$;
*G*worst $\Leftarrow \arg\max\{f(P\text{best}_1), f(P\text{best}_2), \ldots, f(P\text{best}_N)\}$;
For $t = 1, 2, \ldots, T$
For each particle $i = 1, 2, \ldots, N$
Update velocity according to (5) and check the boundaries;
Update position according to (3) and check the boundaries;
Endfor
Evaluate fitness of all particles in $\{X\}$;
Update *P*best, *G*best ***and*** *G*worst;
Endfor.
Return the best solution.

Algorithm 1: HPSO.

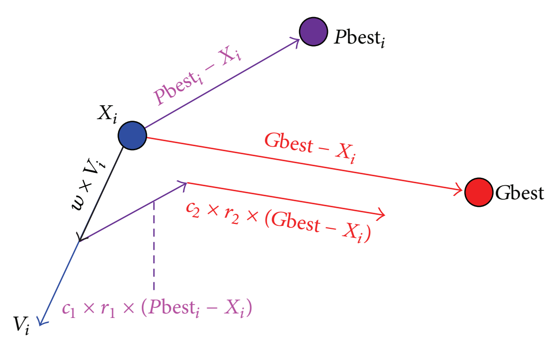

Figure 1: Cognition and social terms in PSO.

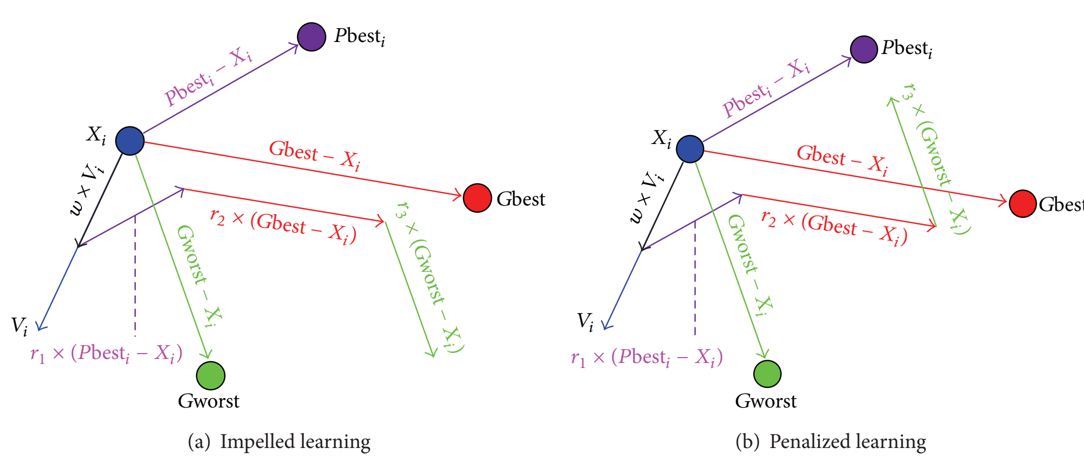

Figure 2: Impelled/penalized term in HPSO.

Table 1: Functions' names, dimensions, ranges, and global optimum values of benchmark functions used in the experiments.

Number	Function name	Dimension (D)	$[\text{Range}]^D$	F_{opt}
F_1	Sphere model	30/50/100	$[-100, 100]^D$	0
F_2	Schwefel's problem 2.22	30/50/100	$[-10, 10]^D$	0
F_3	Schwefel's problem 1.2	30/50/100	$[-100, 100]^D$	0
F_4	Schwefel's problem 2.21	30/50/100	$[-100, 100]^D$	0
F_5	Step function	30/50/100	$[-100, 100]^D$	0
F_6	Quartic function, that is, noise	30/50/100	$[-1.28, 1.28]^D$	0
F_7	Rosenbrock's function	30/50/100	$[-10, 10]^D$	0
F_8	Schwefel's function	30/50/100	$[-500, 500]^D$	0
F_9	Generalized Rastrigin's function	30/50/100	$[-5.12, 5.12]^D$	0
F_{10}	Noncontinuous Rastrigin's function	30/50/100	$[-5.12, 5.12]^D$	0
F_{11}	Ackley's function	30/50/100	$[-32, 32]^D$	0
F_{12}	Generalized Griewank's function	30/50/100	$[-600, 600]^D$	0
F_{13}	Weierstrass's function	30/50/100	$[-0.5, 0.5]^D$	0
F_{14}	Generalized penalized function	30/50/100	$[-50, 50]^D$	0
F_{15}	Cosine mixture problem	30/50/100	$[-1, 1]^D$	$-0.1 \times D$
F_{16}	Rotated elliptic function	30/50/100	$[-1.28, 1.28]^D$	0
F_{17}	Rotated Schwefel's function	30/50/100	$[-500, 500]^D$	0
F_{18}	Rotated Ackley's function	30/50/100	$[-32, 32]^D$	0
F_{19}	Rotated Griewank's function	30/50/100	$[-600, 600]^D$	0
F_{20}	Rotated Weierstrass's function	30/50/100	$[-0.5, 0.5]^D$	0
F_{21}	Rotated Rastrigin's function	30/50/100	$[-5.12, 5.12]^D$	0
F_{22}	Rotated Salomon's function	30/50/100	$[-100, 100]^D$	0
F_{23}	Rotated Rosenbrock's function	30/50/100	$[-100, 100]^D$	0
F_{24}	Shifted Rosenbrock's function	30/50/100	$[-100, 100]^D$	390
F_{25}	Shifted Rastrigin's function	30/50/100	$[-5, 5]^D$	−330
F_{26}	Shifted Schwefel's problem 2.21	30/50/100	$[-100, 100]^D$	−450
F_{27}	Shifted rotated Ackley's function	30/50/100	$[-32, 32]^D$	−140
F_{28}	Shifted rotated Weierstrass's function	30/50/100	$[-0.5, 0.5]^D$	90

In recent years, some modified PSO have extremely enhanced the performance of PSO. For example, Zhan et al. proposed adaptive PSO (APSO) [22] and Wang et al. proposed so-called diversity enhanced particle swarm optimization with neighborhood search (DNSPSO) [23]. The former introduces an evolutionary state estimation (ESE) technique to adaptively adjust the inertia weight and acceleration coefficients. The later ones, a diversity enhancing mechanism and neighborhood-based search strategies, were employed to carry out a tradeoff between exploration and exploitation.

Though all kinds of variants of PSO have enhanced performance of PSO, there are still some problems such as hardly implement, new parameters to just, or high computation cost. So it is necessary to investigate how to trade off the exploration and exploitation ability of PSO and reduce the parameters sensitivity of the solved problems and improve the convergence accuracy and speed with the least computation cost and easy implementation. In order to carry out the targets, in this paper, the global worst position (solution) was introduced into the velocity equation of the standard PSO (SPSO), which is called impelled/penalized learning according to the corresponding weight coefficient. Meanwhile, we eliminate the two acceleration coefficients c_1 and c_2 from the SPSO to reduce the parameters sensitivity of the solved problems. The so-called HPSO has been employed to some nonlinear benchmark functions, which compose unimodal, multimodal, rotated, and shifted high-dimensional functions, to confirm its high performance by comparing with other well-known modified PSO.

The remainder of the paper is structured as follows. In Section 2, the standard particle swarm optimization (SPSO) is introduced. The proposed HPSO is given in Section 3. Experimental studies and discussion are provided in Section 4. Some conclusions are given in Section 5.

2. Standard PSO (SPSO)

The PSO is inspired by the behavior of bird flying or fish schooling; it is firstly introduced by Kennedy and Eberhart in 1995 [1] as a new heuristic algorithm. In the standard PSO (SPSO) [2], a swarm consists of a set of particles, and each particle represents a potential solution of an optimization problem. Considering the ith particle of the swarm with N particles in a D-dimensional space, its position and velocity at iteration t are denoted by $X_i(t) = (x_i^1(t), x_i^2(t), \ldots, x_i^D(t))$ and $V_i(t) = (v_i^1(t), v_i^2(t), \ldots, v_i^D(t))$. Then, the new velocity

Table 2: Experimental results obtained by SPSO and HPSO on function from F_1 to F_{10}.

Fun	Dim		Best	Worst	Meadian	Mean	SD	Significant
F_1	30	SPSO	1.1992e − 04	1.0000e + 04	9.9690e − 04	666.6686	2.5371e + 03	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	9.4288e − 04	1.0000e + 04	0.0078	3.6667e + 03	3.6667e + 03	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	1.0013e + 04	7.0017e + 04	4.0087e + 04	4.0698e + 04	2.0974e + 04	
		HPSO	**0**	**10000**	**0**	**333.3333**	**1.8257e + 03**	+
F_2	30	SPSO	6.8555e − 04	30.0018	10.0017	11.3364	10.0777	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	0.0329	70.0010	40.0006	37.3438	15.2918	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	51.0214	181.4054	110.5934	114.3039	29.0723	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_3	30	SPSO	6.4613e + 03	3.7311e + 04	2.2333e + 04	2.1337e + 04	6.7035e + 03	
		HPSO	**0**	**5.1779e + 03**	**0**	**172.5975**	**945.3557**	+
	50	SPSO	4.0023e + 04	1.0191e + 05	6.5660e + 04	7.0328e + 04	1.7603e + 04	
		HPSO	**0**	**6.9787e + 03**	**0**	**232.6222**	**1.2741e + 03**	+
	100	SPSO	1.7694e + 05	3.0086e + 05	2.4789e + 05	2.4752e + 05	3.6623e + 04	
		HPSO	**0**	**2.6987e + 04**	**0**	**3.8008e + 03**	**6.9150e + 03**	+
F_4	30	SPSO	8.6091	21.2711	12.9945	13.3502	3.5341	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	24.2031	39.5127	31.0562	31.1715	4.2886	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	54.1172	75.3686	64.7834	64.2358	4.2202	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_5	30	SPSO	**0**	10001	**0**	1.0005e + 03	3.0512e + 03	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	**0**	20004	4.5000	5.0028e + 03	6.8230e + 03	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	127	90040	40068	4.3086e + 04	2.2747e + 04	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_6	30	SPSO	0.0344	18.8556	0.0959	3.5587	5.1400	
		HPSO	**1.4522e − 04**	**0.0030**	**0.0012**	**0.0012**	**8.5738e − 04**	+
	50	SPSO	0.0780	72.6594	13.6489	19.6604	19.3860	
		HPSO	**7.4623e − 05**	**0.0017**	**5.3645e − 04**	**6.3534e − 04**	**4.7283e − 04**	+
	100	SPSO	86.7855	381.9209	200.8146	211.9720	88.3159	
		HPSO	**3.5210e − 05**	**0.0019**	**2.9387e − 04**	**4.0826e − 04**	**3.5395e − 04**	+
F_7	30	SPSO	**14.3237**	1.0083e + 04	140.5176	2.4686e + 03	4.2581e + 03	
		HPSO	28.6353	**28.9456**	**28.8793**	**28.8461**	**0.0932**	+
	50	SPSO	97.0317	9.4285e + 05	376.2306	3.4093e + 04	1.7169e + 05	
		HPSO	**48.4886**	**48.8766**	**48.7600**	**48.7513**	**0.0875**	+
	100	SPSO	706.1328	2.8333e + 06	9.4375e + 05	8.8851e + 05	8.9157e + 05	
		HPSO	**98.4280**	**98.8373**	**98.7133**	**98.7129**	**0.0818**	+
F_8	30	SPSO	**2.0226e + 03**	**4.8935e + 03**	**3.5787e + 03**	**3.6128e + 03**	**733.1063**	
		HPSO	3.5886e + 03	8.0516e + 03	6.6047e + 03	6.3505e + 03	1.0893e + 03	−
	50	SPSO	**5.8499e + 03**	**9.7913e + 03**	**7.8862e + 03**	**7.7139e + 03**	**1.0101e + 03**	
		HPSO	6.5496e + 03	1.4460e + 04	1.1191e + 04	1.0866e + 04	2.1757e + 03	−
	100	SPSO	1.8110e + 04	**2.4259e + 04**	**2.0949e + 04**	**2.1084e + 04**	**1.7384e + 03**	
		HPSO	**1.2615e + 04**	3.1402e + 04	2.4302e + 04	2.4077e + 04	4.9510e + 03	−

Table 2: Continued.

Fun	Dim		Best	Worst	Meadian	Mean	SD	Significant
F_9	30	SPSO	28.7299	160.3815	87.6754	92.5142	32.6994	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	175.2643	351.6480	260.4359	258.0518	48.4078	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	555.8950	993.3887	750.1694	749.1658	749.1658	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_{10}	30	SPSO	61.4129	221.0445	132.7694	134.5414	33.8073	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	157.1020	440.0897	324.2632	310.3595	64.3675	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	623.5658	$1.0433e+03$	804.6981	813.3435	88.5932	
		HPSO	**0**	**25**	**0**	**0.8333**	**4.5644**	+

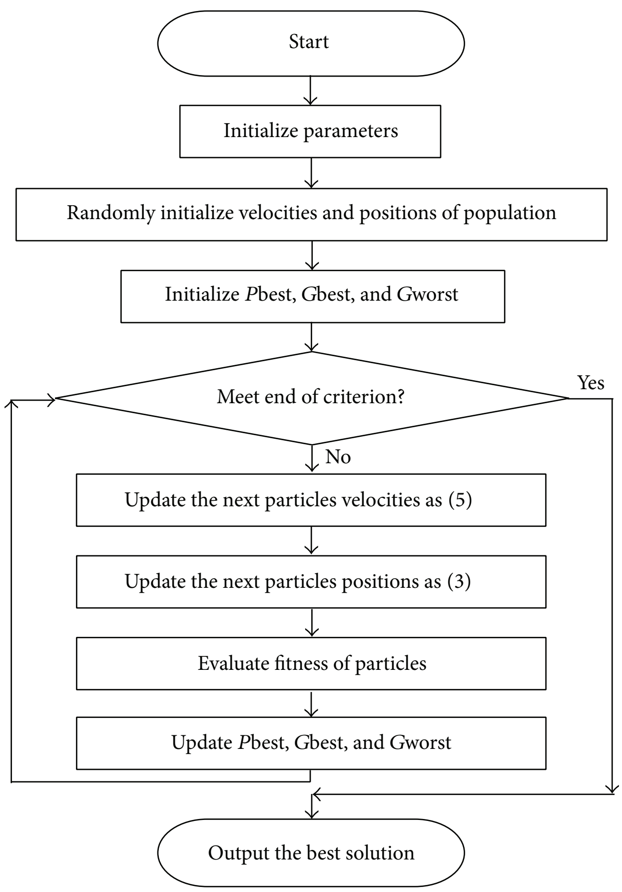

Figure 3: HPSO flowchart.

and position on the d-dimension of this particle at iteration $t+1$ will be calculated by using the following:

$$v_i^d(t+1) = w \cdot v_i^d(t) + c_1 \cdot r_1^d(t) \cdot \left(P\mathrm{best}_i^d(t) - x_i^d(t)\right) + c_2 \cdot r_2^d(t) \cdot \left(G\mathrm{best}^d(t) - x_i^d(t)\right), \quad (1)$$

where $i = 1, 2, \ldots, N$, and N is the population size; $d = 1, 2, \ldots, D$, and D is the dimension of search space; r_1^d and r_2^d are two uniformly distributed random numbers in the interval $[0, 1]$; acceleration coefficients c_1 and c_2 are nonnegative constants which control the influence of the cognitive and social components during the search process. $P\mathrm{best}_i(t) = (P\mathrm{best}_i^1(t), \ldots, P\mathrm{best}_i^D(t))$, called the personal best solution, represents the best solution found by the ith particle itself until iteration t; $G\mathrm{best}(t) = (G\mathrm{best}^1(t), \ldots, G\mathrm{best}^D(t))$, called the global best solution, represents the global best solution found by all particles until iteration t. w is the inertial weight

Table 3: Experimental results obtained by SPSO and HPSO on functions from F_{11} to F_{20}.

Fun	Dim		Best	Worst	Median	Mean	SD	Significant
F_{11}	30	SPSO	0.0043	19.9630	0.0595	2.3935	5.4041	
		HPSO	**8.8818e − 16**	**8.8818e − 16**	**8.8818e − 16**	**8.8818e − 16**	**0**	+
	50	SPSO	0.0598	19.9646	12.6912	10.5673	6.3042	
		HPSO	**8.8818e − 16**	**8.8818e − 16**	**8.8818e − 16**	**8.8818e − 16**	**0**	+
	100	SPSO	15.4237	20.2143	19.5200	19.4135	0.8672	
		HPSO	**8.8818e − 16**	**8.8818e − 16**	**8.8818e − 16**	**8.8818e − 16**	**0**	+
F_{12}	30	SPSO	7.0274e − 04	90.8935	0.0178	12.0794	31.2763	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	0.0014	270.8170	0.0415	45.1971	70.1274	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	1.1140	721.0594	361.0858	376.1758	158.6584	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_{13}	30	SPSO	0.1403	4.3952	0.3210	1.0567	1.4863	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	0.8657	15.2389	7.5828	8.2388	3.6607	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	27.6235	64.4826	49.3984	47.7138	10.0126	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_{14}	30	SPSO	6.4114e − 05	2.2031	0.4202	0.5373	0.5730	
		HPSO	0.0710	0.2803	0.1301	0.1444	0.0513	+
	50	SPSO	0.1882	6.9784	2.2774	2.3889	1.5688	
		HPSO	0.1016	0.3137	0.1652	0.1702	0.0438	+
	100	SPSO	32.5063	5.1200e + 08	457.9143	7.6801e + 07	1.5257e + 08	
		HPSO	0.1866	0.5097	0.2703	0.2736	0.0653	+
F_{15}	30	SPSO	−3.0000	−2.8522	−3.0000	−2.9507	0.0709	
		HPSO	**−3**	**−3**	**−3**	**−3**	**0**	+
	50	SPSO	−5.0000	−2.3044	−4.4827	−4.2127	0.6865	
		HPSO	**−5**	**−5**	**−5**	**−5**	**0**	+
	100	SPSO	−7.9165	4.7637	−5.2127	−4.6977	2.8465	
		HPSO	**−10**	**−10**	**−10**	**−10**	**0**	+
F_{16}	30	SPSO	2.3604e + 03	3.8233e + 04	3.8233e + 04	1.2375e + 04	9.2463e + 03	
		HPSO	0	5.8369e + 03	0	390.6710	1.2756e + 03	+
	50	SPSO	7.1213e + 03	1.0427e + 05	3.3195e + 04	3.4891e + 04	2.2914e + 04	
		HPSO	0	4.0529e + 03	0	224.6749	873.6249	+
	100	SPSO	6.2317e + 04	2.7386e + 05	1.4222e + 05	1.4697e + 05	5.7699e + 04	
		HPSO	0	1.9403e + 04	0	1.0583e + 03	3.8088e + 03	+
F_{17}	30	SPSO	6.7986e + 03	9.7587e + 03	8.3387e + 03	8.2508e + 03	739.7223	
		HPSO	8.3590e + 03	9.8803e + 03	9.0866e + 03	9.0790e + 03	442.4330	−
	50	SPSO	1.3020e + 04	1.7080e + 04	1.4999e + 04	1.5149e + 04	1.0581e + 03	
		HPSO	1.5003e + 04	1.7349e + 04	1.6514e + 04	1.6310e + 04	669.3538	−
	100	SPSO	2.7400e + 04	2.7400e + 04	3.1087e + 04	3.1149e + 04	2.1654e + 03	
		HPSO	3.0329e + 04	3.5493e + 04	3.4226e + 04	3.3586e + 04	1.5320e + 03	−
F_{18}	30	SPSO	20.7888	21.0951	21.0053	20.9848	0.0712	
		HPSO	8.8818e − 16	21.1210	20.9931	11.2354	10.6894	+
	50	SPSO	21.0515	21.2478	21.1455	21.1436	0.0536	
		HPSO	8.8818e − 16	21.2404	21.1366	12.0016	10.6745	+
	100	SPSO	21.2367	21.3931	21.3368	21.3358	0.0364	
		HPSO	8.8818e − 16	21.3949	21.3545	15.6658	9.6084	+

Table 3: Continued.

Fun	Dim		Best	Worst	Median	Mean	SD	Significant
F_{19}	30	SPSO	1.0517	495.3131	273.6408	243.6176	154.3551	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	265.0558	1.4393e + 03	798.8065	786.0782	289.8401	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	1.9937e + 03	4.0158e + 03	2.9388e + 03	2.9263e + 03	543.9053	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_{20}	30	SPSO	22.5705	34.8494	28.6842	28.8734	3.5028	
		HPSO	0	39.9834	0	3.1393	9.7817	+
	50	SPSO	45.9462	70.7399	55.5532	55.6014	5.7839	
		HPSO	0	66.4051	0	2.2135	12.1239	+
	100	SPSO	106.4483	139.8394	120.6118	121.4481	7.8030	
		HPSO	0	129.4941	0	8.3487	31.7918	+

to balance the global and local search abilities of particles in the search space, which is given by

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \times t, \tag{2}$$

where $w_{\max}$ is the initial weight, $w_{\min}$ is the final weight, t is the current iteration number, and T is the maximum iteration number. Then, update particle's position using the following:

$$x_i^d (t+1) = x_i^d (t) + v_i^d (t+1) \tag{3}$$

and check $x_{\min}^d \le x_i^d(t+1) \le x_{\max}^d$, where $x_{\min}^d$ and $x_{\max}^d$ represent lower and upper bounds of the dth variable, respectively.

3. Human Behavior-Based PSO (HPSO)

In this section, a modified version of SPSO based on human behavior, which is called HPSO, is proposed to improve the performance of SPSO. In SPSO, all particles only learn from the best particles Pbest and Gbest. Obviously, it is an ideal social condition. However, considering the human behavior, there exist some people who have bad habits or behaviors around us, at the same time, as we all known that these bad habits or behaviors will bring some effects on people around them. If we take warning from these bad habits or behaviors, it is beneficial to us. Conversely, if we learn from these bad habits or behaviors, it is harmful to us. Therefore, we must give an objective and rational view on these bad habits or behavior.

In HPSO, we introduce the global worst particle, who is of the worst fitness in the entire population at each iteration. It is denoted as Gworst and defined as follows:

$$\begin{aligned} &G\text{worst}(t) \\ &\quad = \operatorname{argmax}\left\{f\left(P\text{best}_1\right), f\left(P\text{best}_2\right), \ldots, f\left(P\text{best}_N\right)\right\}, \end{aligned} \tag{4}$$

where $f(\cdot)$ represents the fitness value of the corresponding particle.

To simulate human behavior and make full use of the Gworst, we introduce a learning coefficient r_3, which is a random number obeying the standard normal distribution; that is, $r_3 \in N(0,1)$. If $r_3 > 0$, we consider it as an impelled learning coefficient, which is helpful to enhance the "flying" velocity of the particle; therefore, it can enhance the exploration ability of particle. Conversely, if $r_3 < 0$, we consider it as a penalized learning coefficient, which can decrease the "flying" velocity of the particle; therefore, it is beneficial to enhance the exploitation. If $r_3 = 0$, it represents that these bad habits or behaviors have not effect on the particle. Meanwhile, in order to reduce the parameters sensitivity of the solved problems, we take place of the two acceleration coefficients c_1 and c_2 with two random learning coefficients r_1 and r_2, respectively. Therefore, the velocity equation has been changed as follows:

$$\begin{aligned} v_i^d(t+1) = {} & w \cdot v_i^d(t) + r_1(t) \cdot \left(P\text{best}_i^d(t) - x_i^d(t)\right) \\ & + r_2(t) \cdot \left(G\text{best}^d(t) - x_i^d(t)\right) \\ & + r_3(t) \cdot \left(G\text{worst}^d(t) - x_i^d(t)\right), \end{aligned} \tag{5}$$

where r_1 and r_2 are two random numbers in range of $[0,1]$ and $r_1 + r_2 = 1$. The random numbers r_1, r_2, and r_3 are the same for all $d = 1, 2, \ldots, D$ but different for each particle, and they are generated anew in each iteration. If $v_i^d(t+1)$ overflows the boundary, we set boundary value to it. Consider

$$v_i^d(t+1) = \begin{cases} v_{\min}^d, & \text{if } v_i^d(t+1) < v_{\min}^d, \\ v_{\max}^d, & \text{if } v_i^d(t+1) > v_{\max}^d, \\ v_i^d(t+1), & \text{otherwise}, \end{cases} \tag{6}$$

where $v_{\min}^d$ and $v_{\max}^d$ are the minimum and maximum velocity of the d-dimensional search space, respectively. Similarly, if $x_i^d(t+1)$ flies out of the search space, we limit it to the corresponding bound value.

In SPSO, the cognition and social learning terms move particle i towards good solutions based on $P\text{best}_i$ and Gbest in the search space as shown in Figure 1. This strategy makes a particle fly fast to good solutions, so it is easy to trap in

TABLE 4: Experimental results obtained by SPSO and HPSO on functions from F_{21} to F_{28}.

Fun	Dim		Best	Worst	Median	Mean	SD	Significant
F_{21}	30	SPSO	67.1541	307.3070	213.8939	203.8842	61.8125	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	158.2955	715.0245	518.1705	500.5593	135.5998	
		HPSO	0	269.3463	0	8.9782	49.1757	+
	100	SPSO	$1.0850e+03$	$1.9021e+03$	$1.5793e+03$	$1.5669e+03$	190.5584	
		HPSO	0	582.0882	0	35.5882	136.0270	+
F_{22}	30	SPSO	0.7999	14.9999	1.2522	2.9025	4.3553	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	50	SPSO	2.0999	26.0999	13.9628	12.8291	6.9033	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
	100	SPSO	16.5013	41.9999	35.4551	33.9791	6.3075	
		HPSO	**0**	**0**	**0**	**0**	**0**	+
F_{23}	30	SPSO	81.0577	$4.0119e+09$	$2.0685e+08$	$6.8745e+08$	$1.0469e+09$	
		HPSO	28.8214	28.9856	28.9323	28.9252	0.0421	+
	50	SPSO	$3.7253e+03$	$2.1495e+10$	$3.6515e+09$	$3.6515e+09$	$5.3957e+09$	
		HPSO	48.7069	48.8900	48.8205	48.8139	0.0479	+
	100	SPSO	$6.7997e+09$	$9.2655e+10$	$3.3160e+10$	$3.8223e+10$	$2.0050e+10$	
		HPSO	98.6590	98.8846	98.8109	98.7983	0.0545	+
F_{24}	30	SPSO	$6.2312e+08$	$2.3418e+10$	$4.9110e+09$	$5.8767e+09$	$5.6099e+09$	
		HPSO	$5.9432e+05$	$6.2859e+09$	$7.6373e+06$	$3.7982e+08$	$1.2316e+09$	+
	50	SPSO	$4.3540e+09$	$3.3195e+10$	$1.3961e+10$	$1.6077e+10$	$8.3270e+09$	
		HPSO	$3.9454e+06$	$8.9387e+09$	$3.1766e+07$	$7.0962e+08$	$1.9565e+09$	+
	100	SPSO	$4.9031e+10$	$1.5465e+11$	$9.1986e+10$	$9.7151e+10$	$2.8460e+10$	
		HPSO	$2.0551e+08$	$5.4553e+09$	$6.7593e+08$	$1.1373e+09$	$1.2367e+09$	+
F_{25}	30	SPSO	−229.5551	−78.6646	−176.9746	−174.7148	35.8633	
		HPSO	−204.3636	−100.1465	−148.1389	−149.7299	27.1636	−
	50	SPSO	−77.4305	156.8323	22.8512	24.6168	62.2086	
		HPSO	−102.9219	132.8077	−16.6107	−4.1921	58.2317	+
	100	SPSO	475.3838	860.0386	612.6947	632.8693	100.6069	
		HPSO	394.3532	805.2473	581.1779	590.3932	80.6175	+
F_{26}	30	SPSO	−425.5452	−331.1195	−385.1191	−387.6682	22.2647	
		HPSO	−439.6877	−399.0205	−423.4928	−422.5533	11.3496	+
	50	SPSO	−399.6029	−326.6739	−379.4869	−370.8387	18.7600	
		HPSO	−415.6822	−391.7124	−401.4635	−400.8395	6.5162	+
	100	SPSO	−358.3688	−300.6930	−322.8060	−324.4641	15.5861	
		HPSO	−380.3478	−360.8031	−369.0319	−370.4683	5.1369	+
F_{27}	30	SPSO	−119.2212	−118.8710	−119.0179	−119.0258	0.0866	
		HPSO	−119.1100	−118.8700	−118.9469	−118.9589	0.0545	−
	50	SPSO	−119.0222	−118.7656	−118.8316	−118.8535	0.0603	
		HPSO	−118.9117	−118.7327	−118.7780	−118.7911	0.0421	−
	100	SPSO	−118.7259	−118.6013	−118.6485	−118.6537	0.0310	
		HPSO	−118.6872	−118.5986	−118.6231	−118.6289	0.0204	−
F_{28}	30	SPSO	113.2663	126.0977	118.5782	119.4693	3.6330	
		HPSO	114.4722	132.2305	124.3094	124.5205	4.3399	−
	50	SPSO	137.8303	153.5400	145.1433	145.1503	4.2018	
		HPSO	141.9493	162.4008	153.9547	153.1087	5.4273	−
	100	SPSO	194.1222	232.4306	215.9257	215.9174	8.6772	
		HPSO	212.5258	245.0126	229.4886	230.4426	7.4650	−

Table 5: Some well-known PSOs algorithms in the literature.

Algorithm	Year	Topology	Parameter settings
GPSO	1998	Global star	w: 0.9 − 0.4, $c_1 = c_2 = 2.0$
LPSO	2002	Local ring	w: 0.9 − 0.4, $c_1 = c_2 = 2.0$
FIPS	2004	Local Uring	$\chi = 0.729$, $\sum c_i = 4.1$
HPSO-TVAC	2004	Global star	w: 0.9 − 0.4, c_1 : 2.5 − 0.5, and c_2: 0.5 − 2.5
UPSO	2004	Global star	w: 0.9 − 0.4, $c_1 = c_2 = 2.0$, and $U = 0.5$
DMS-PSO	2005	Dynamic multiswarm	w: 0.9 − 0.2, $c_1 = c_2 = 2.0$, $m = 3$, and $R = 5$
VPSO	2006	Local Von Neumann	w: 0.9 − 0.4, $c_1 = c_2 = 2.0$
CLPSO	2006	Comprehensive learning	w: 0.9 − 0.4, $c = 1.49445$, and $m = 7$
QIPSO	2007	Global star	w: 0.9 − 0.4, $c_1 = c_2 = 2.0$
APSO	2009	Global star	w: 0.9, $c_1 = c_2 = 2.0$; δ : random in $[0.05, 0.1]$, σ: 1 − 0.1
AFPSO	2011	Global star	w: 0.9 − 0.4, c_1, and c_2 are based on fuzzy rule
AFPSO-QI	2011	Global star	w: 0.9 − 0.4, c_1, and c_2 are based on fuzzy rule

local optima. From Figure 2, we can clearly observe that both impelled learning term and penalized term provide a particle with the chance to change flying direction. Therefore, the impelled/penalized term plays a key role in increasing the population diversity, which is beneficial in helping particles to escape from the local optima and enhance the convergence speed. In HPSO, the impelled/penalized learning term performs a proper tradeoff between the exploration and exploitation.

To sum up, Figure 3 illustrates the flowchart of HPSO. Meanwhile, the pseudocodes of implementing the HPSO are listed as shown in Algorithm 1.

4. Experimental Studies and Discussion

To evaluate the performance of HPSO, 28 minimization benchmark functions are selected [22, 24, 25] as detailed in Section 4.1. HPSO is compared with SPSO in different search spaces and the results are given in Section 4.2. In addition, HPSO is compared with some well-known variants of PSO in Section 4.3.

4.1. Benchmark Functions. In the experimental study, we choose 28 minimization benchmark functions, which consist of unimodal, multimodal, rotated, shifted, and shifted rotated functions. Table 1 lists the main information; please refer to papers [22, 24, 25] to obtain further detailed information about these functions. Among these functions, F_1–F_6 are unimodal functions. F_7 is the Rosenbrock function, which is unimodal for $D = 2$ and $D = 3$ but may have multiple minima in high dimension cases. F_8–F_{15} are unrotated multimodal functions and the number of their local minima increases exponentially with the problem dimension. F_{16}–F_{23} are rotated functions. F_{24}–F_{26} are shifted functions and F_{27} and F_{28} are shifted rotated multimodal functions and $O = (o^1, o^2, \ldots, o^D)$ is a randomly generated shift vector located in the search space. To obtain a rotated function, an orthogonal matrix M [26] is considered and the rotated variable $y = M \times x$ is computed. Then, the vector y is used to evaluate the objective function value.

4.2. Comparison of HPSO with SPSO. The performance on the convergence accuracy of HPSO is compared with that of SPSO. The test functions listed in Table 1 are evaluated. For a fair comparison, we set the same parameters value. Population size is set to 30 ($N = 30$), upper bounds of velocity $V_{\max} = 0.5 \times (X_{\max} - X_{\min})$, and the corresponding lower bounds $V_{\min} = -V_{\max}$, where $X_{\min}$ and $X_{\max}$ are the lower and upper bounds of variables, respectively. Inertia weight w is linearly decreased from 0.9 to 0.4 in SPSO and HPSO. Acceleration coefficients c_1 and c_2 in SPSO are set to 2. The two algorithms are independently run 30 times on the benchmark functions. The results in terms of the best, worst, median, mean, and standard deviation (SD) of the solutions obtained in the 30 independent runs by each algorithm in different search spaces are as shown in Tables 2, 3, and 4. At the same time, the maximum iteration T is 1000 for $D = 30$, 2000 for $D = 50$, and 3000 for $D = 100$, respectively.

From Tables 2–4, we can clearly observe that the convergence accuracy of HPSO is better than SPSO on the most benchmark functions. An interesting result is that HPSO can find the global optimal solutions on functions F_2, F_4, F_5, F_9, F_{12}, F_{13}, F_{15}, F_{19}, and f_{22} in all search spaces; that is to say, HPSO can obtain the 100% success rate on the listed functions. Considering F_1 and F_{10}, though HPSO can find the global optimal solutions in all different search ranges, it only obtained the mean values 333.3333 and 0.8333, respectively, in 100-dimensional space. At the same time, HPSO offers the higher convergence accuracy on functions F_3, F_6, F_7, F_{11}, F_{14}, F_{16}, F_{20}, F_{21}, F_{23}, and F_{26}. However, we must observe that SPSO has higher performance on function F_8. As for F_{25}, SPSO has better performance in 30-dimensional search space, but HPSO has better performance in 50- and 100-dimensional search spaces. As for shifted rotated functions F_{27} and F_{28}, both SPSO and HPSO have worst convergence accuracy. As seen, the dimension of the selected functions has great effect on SPSO. For example, considering function F_1, SPSO has mean value 666.6686, 3.6667e + 03, and 4.0698e + 04 in 30-dimensional, 50-dimensional, and 100-dimensional search spaces, respectively, while HPSO has mean values 0, 0, and 333.333 in the corresponding search space. Therefore, we

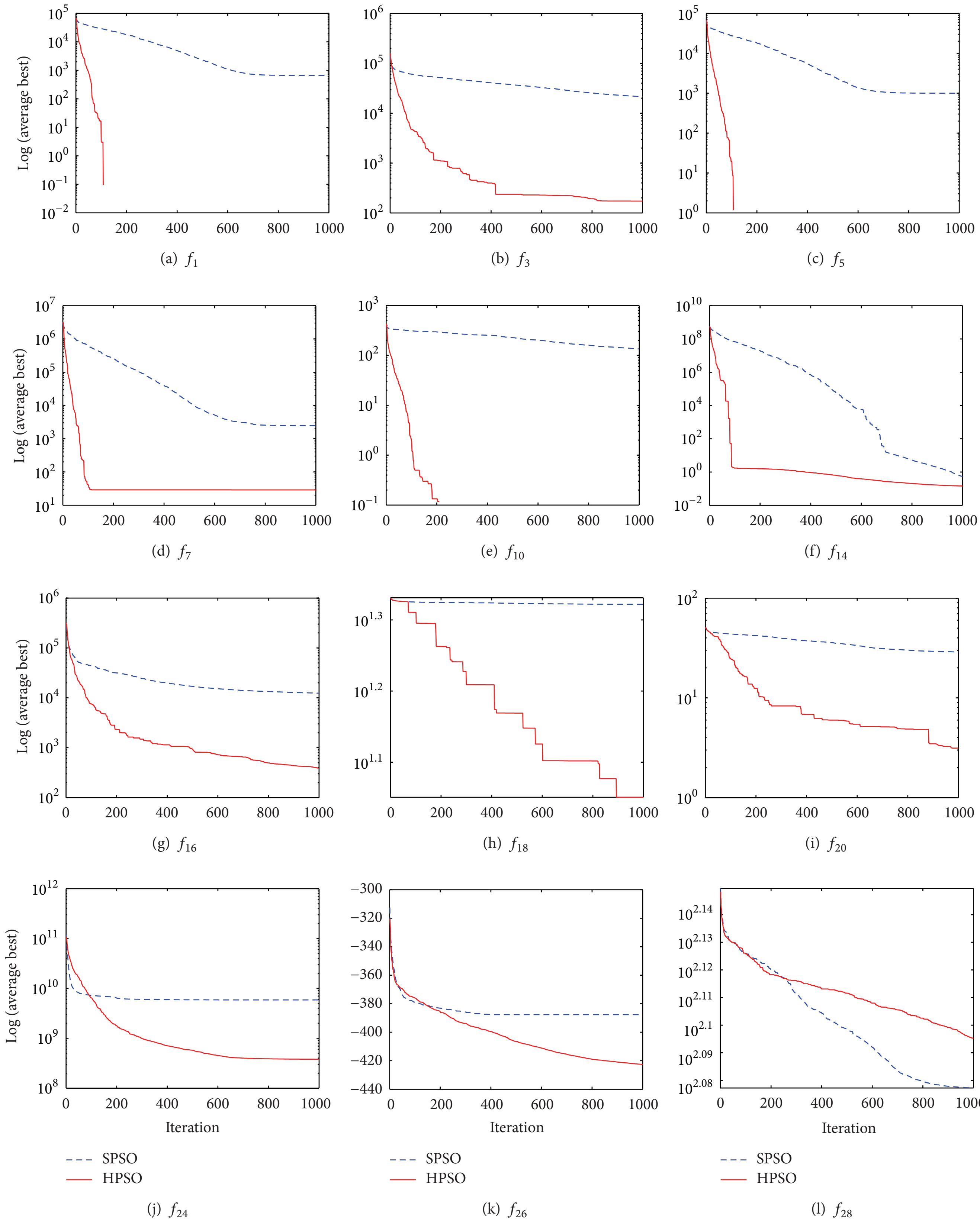

FIGURE 4: Convergence comparison of HPSO and SPSO on the selected test functions with $D = 30$, $N = 30$, and $T = 1000$.

Table 6: Comparison results of eight PSO algorithms [22] with HPSO on 10 functions ($N = 20$, $D = 30$, and FEs $= 2 \times 10^5$).

Function	GPSO	LPSO	VPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	APSO	HPSO
F_1									
Mean	$1.98e-53$	$4.77e-29$	$5.11e-38$	$3.21e-30$	$3.38e-41$	$3.85e-54$	$1.89e-19$	$1.45e-150$	**0**
SD	$7.08e-53$	$1.13e-28$	$1.91e-37$	$3.60e-30$	$8.50e-41$	$1.75e-53$	$1.49e-19$	$5.73e-150$	**0**
Rank	4	8	6	7	5	3	9	2	**1**
F_2									
Mean	$2.51e-34$	$2.03e-20$	$6.29e-27$	$1.32e-17$	$6.9e-23$	$2.61e-29$	$1.01e-13$	$5.15e-84$	**0**
SD	$5.84e-34$	$2.89e-20$	$8.68e-27$	$7.86e-18$	$6.89e-23$	$6.6e-29$	$6.51e-14$	$1.44e-83$	**0**
Rank	3	7	5	8	6	4	9	2	**1**
F_3									
Mean	$6.45e-2$	18.60	1.44	0.77	$2.89e-7$	47.5	395	**1.0e − 10**	167
SD	$9.45e-2$	30.71	1.55	0.86	$2.97e-7$	56.4	142	**2.13e − 10**	913
Rank	3	6	5	4	2	7	9	**1**	8
F_5									
Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SD	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F_6									
Mean	$7.77e-3$	$1.49e-2$	$1.08e-2$	$2.55e-3$	$5.54e-2$	$1.1e-2$	$3.92e-3$	$4.66e-3$	**1.03e − 04**
SD	$2.42e-3$	$5.66e-3$	$3.24e-3$	$6.25e-4$	$2.08e-2$	$3.94e-3$	$1.14e-3$	$1.7e-3$	**8.99e − 05**
Rank	5	8	6	2	9	7	3	4	**1**
F_9									
Mean	30.7	34.90	34.09	29.98	2.39	28.1	$2.57e-11$	$5.8e-15$	**0**
SD	8.68	7.25	8.07	10.92	3.71	6.42	$6.64e-11$	$1.01e-14$	**0**
Rank	7	9	8	6	4	5	3	2	**1**
F_{10}									
Mean	15.5	30.40	21.33	35.91	1.83	32.8	0.167	$4.14e-16$	**0**
SD	7.4	9.23	9.46	9.49	2.65	6.49	0.379	$1.45e-15$	**0**
Rank	5	7	6	9	4	8	3	2	**1**
F_{11}									
Mean	$1.15e-14$	$1.85e-14$	$1.4e-14$	$7.69e-15$	$2.06e-10$	$8.52e-15$	$2.01e-12$	$1.11e-14$	**8.88e − 16**
SD	$2.27e-15$	$4.80e-15$	$3.48e-15$	$9.33e-16$	$9.45e-10$	$1.79e-15$	$9.22e-13$	$3.55e-15$	**0**
Rank	5	7	6	2	9	3	8	4	**1**
F_{12}									
Mean	$2.37e-2$	$1.10e-2$	$1.31e-2$	$9.04e-4$	$1.07e-2$	$1.31e-2$	$6.45e-13$	$1.67e-2$	**0**
SD	$2.57e-2$	$1.60e-2$	$1.35e-2$	$2.78e-3$	$1.14e-2$	$1.73e-2$	$2.07e-12$	$2.41e-2$	**0**
Rank	9	5	6	3	4	7	2	8	**1**
F_{14}									
Mean	$1.04e-2$	$2.18e-30$	$3.46e-3$	$1.22e-31$	$7.07e-30$	**2.05e − 32**	$1.59e-21$	$3.76e-31$	$1.70e-2$
SD	$3.16e-2$	$5.14e-30$	$1.89e-2$	$4.85e-32$	$4.05e-30$	**8.12e − 33**	$1.93e-21$	$1.2e-30$	$1.42e-2$
Rank	8	4	7	2	5	**1**	6	3	9
Average rank	5	6.2	5.6	4.4	4.9	4.6	5.3	2.9	**2.5**
Final rank	6	9	8	3	5	4	7	2	**1**

also conclude that HPSO has better stability than SPSO from the data in different search spaces.

In the 9th columns of Tables 2–4, we report the statistical significance level of the difference of the means of the two algorithms. Note that here "+" indicates that the t value is significant at a 0.05 level of significance by two-tailed test, and "−" stands for the difference of means that is not statistically significant.

Figure 4 graphically presents the comparison in terms of convergence characteristics of the evolutionary processes in solving the selected benchmark functions in 30-dimensional search space with $N = 30$ and $T = 1000$.

4.3. Comparison of HPSO with Other PSO Algorithms. In this section, a comparison of HPSO with some well-known PSO algorithms which are listed in Table 5 is performed to evaluate the efficiency of the proposed algorithm.

At first, we choose 10 unimodal and multimodal test functions for this evaluation. According to [22], the algorithms

Table 7: Comparison results of seven PSO algorithms [25] with HPSO on six functions ($N = 30$, $D = 30$, and $T = 10{,}000$).

Function	SPSO	QIPSO	UPSO	FIPS	CLPSO	AFSO	AFSO-Q1	HPSO
F_9								
Mean	52.30	25.61	59.40	106.1	74.39	17.93	15.69	**0**
SD	27.35	15.98	58.05	30.54	9.77	5.63	4.47	**0**
Rank	5	4	6	8	7	3	2	**1**
F_{13}								
Mean	0.534	36.38	8.70	6.40	$1.39e-03$	$4.52e-03$	$1.50e-03$	**0**
SD	1.74	4.66	3.08	3.04	$3.28e-04$	$9.20e-03$	$3.48e-03$	**0**
Rank	5	8	7	6	2	4	3	**1**
F_{21}								
Mean	320.2	317.5	309.5	434.1	263.3	266.3	253.3	**0**
SD	14.70	23.24	25.88	34.99	11.96	12.00	12.63	**0**
Rank	7	6	5	8	3	4	2	**1**
F_{22}								
Mean	17.03	15.20	14.29	26.60	11.94	10.38	8.46	**0**
SD	2.55	1.32	2.15	1.42	1.37	1.38	0.948	**0**
Rank	7	6	5	8	4	3	2	**1**
F_{27}								
Mean	−119.10	−119.10	−119.10	−119.90	−119.00	−119.70	−119.80	−119.05
SD	$7.09e-02$	$5.68e-02$	$3.24e-02$	$3.78e-02$	$4.28e-02$	$3.85e-02$	$5.45e-02$	$5.50e-02$
Rank	4	4	4	1	6	3	2	5
F_{28}								
Mean	115.90	121.90	**113.20**	113.60	118.30	123.20	123.10	117.32
SD	2.90	4.90	6.14	3.63	2.40	**2.25**	3.01	3.65
Rank	3	6	**1**	**2**	**5**	8	7	4
Average rank	5.17	5.67	4.67	5.50	4.50	4.17	3.00	**2.17**
Final rank	6	8	5	7	4	3	2	**1**

GPSO [2], LPSO [16], VPSO [27], FIPS [28], HPSO-TVAC [6], DMS-PSO [29], CLPSO [24], and APSO [22] are considered as detailed in Table 5. The experimental results of the algorithms are directly from [22] as shown in Table 6. In this trial, the population size $N = 20$, the dimension $D = 30$, and the maximum fitness evaluations (FEs) were set to 2×10^5 also. The parameter configurations of the selected algorithms have been set according to their corresponding references. The inertia weight w is linearly decreased from 0.9 to 0.4 in HPSO. HPSO is independently run 30 times and the mean and SD are shown in Table 6. As seen, HPSO has the first rank among the algorithms and obtains the global minimum on functions F_1, F_2, F_5, F_9, F_{10}, and F_{12} and gives the good near-global optima on functions F_6 and F_{11}. Meanwhile, HPSO has the worst performance on functions F_3 and F_{14}. As for F_3, APSO has the best convergence accuracy, and HPSO only wins CLPSO. Considering F_{14}, DMS-PSO has the best performance.

Then, in the next step, we choose six functions from [25] and seven algorithms of GPSO, QIPSO [30], UPSO [31], FIPS, AFSO [25], and AFSO-Q1 [25] as detailed in Table 5. For a fair comparison, the population size $N = 30$, the dimension $D = 30$, and the maximum iteration $T = 10,000$ also in HPSO, and the inertia weight w is linearly decreased from 0.9 to 0.4. HPSO is independently run 30 times and the mean and SD are shown in Table 7. As seen, HPSO shows better performance and has the first rank. HPSO finds the global optimal solution on functions F_9, F_{13}, F_{21}, and F_{22}. FIPS and UPSO have better convergence accuracy on functions F_{27} and F_{28}, respectively.

Therefore, it is worth saying that the proposed algorithm has considerably better performance than the other well-known PSO algorithms in unimodal and multimodal high-dimensional functions.

5. Conclusion

In this paper, a modified version of PSO called HPSO has been introduced to enhance the performance of SPSO. To simulate the human behavior, the global worst particle was introduced into the velocity equation of SPSO, and the learning coefficient which obeys the standard normal distribution can balance the exploration and exploitation abilities by changing the flying direction of particles. When the coefficient is positive, it is called impelled leaning coefficient, which is helpful to enhance the exploration ability. When the coefficient is negative, it is called penalized learning coefficient, which is beneficial for improving the exploitation ability. At the same time, the acceleration coefficients c_1 and c_2 have been replaced with two random numbers, whose sum is equal to 1 in [0, 1]; this strategy decreases the dependence on parameters of the solved problems. The proposed algorithm has been evaluated on 28 benchmark functions including unimodal, unrotated multimodal, rotated, shifted, and shifted rotated functions, and the experimental results confirm the high performance of HPSO on the main functions. However, as seen, HPSO has the worst performance on shifted rotated functions, so it is worth researching how to enhance the performance of HPSO on shifted rotated functions in the future. Meanwhile, applying HPSO to solve real-world problems is also a research field.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The project is supported by the National Natural Science Foundation of China (Grant no. 61175127) and the Science and Technology Project of Department of Education of Jiangxi Province China (Grant no. GJJ12093).

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[2] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.

[3] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation*, pp. 101–106, May 2001.

[4] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the Congress on Evolutionary Computation*, pp. 94–100, May 2001.

[5] G. Xu, "An adaptive parameter tuning of particle swarm optimization algorithm," *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4560–4569, 2013.

[6] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[7] 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, 2002.

[8] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, pp. 84–88, July 2000.

[9] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 84–89, May 1998.

[10] Y.-P. Chen, W.-C. Peng, and M.-C. Jian, "Particle swarm optimization with recombination and dynamic linkage discovery," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 6, pp. 1460–1470, 2007.

[11] P. S. Andrews, "An investigation into mutation operators for particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1044–1051, July 2006.

[12] W.-J. Zhang and X.-F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in *Proceedings of the IEEE International Conference on System Security and Assurance*, pp. 3816–3821, October 2003.

[13] M. S. Kıran, M. Gündüz, and K. Baykan, "A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum," *Applied Mathematics and Computation*, vol. 219, no. 4, pp. 1515–1521, 2012.

[14] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 522–528, September 2005.

[15] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1931–1938, 1999.

[16] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1671–1676, 2002.

[17] J. Kennedy, "Bare bones particle swarms," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 80–87, 2003.

[18] R. A. Krohling and E. Mendel, "Bare bones particle swarm optimization with Gaussian or cauchy jumps," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 3285–3291, May 2009.

[19] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Bare bones differential evolution," *European Journal of Operational Research*, vol. 196, no. 1, pp. 128–139, 2009.

[20] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu, "Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point," *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3763–3775, 2011.

[21] J. Sun, X. Wu, V. Palade, W. Fang, C.-H. Lai, and W. Xu, "Convergence analysis and improvements of quantum-behaved particle swarm optimization," *Information Sciences*, vol. 193, pp. 81–103, 2012.

[22] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.

[23] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.

[24] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[25] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu, "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions," *Information Sciences*, vol. 181, no. 20, pp. 4539–4549, 2011.

[26] R. Salomon, "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms," *BioSystems*, vol. 39, no. 3, pp. 263–278, 1996.

[27] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and bestof-neighborhood particle swarms," *IEEE*

Transactions on Systems, Man, and Cybernetics, Part C, vol. 36, no. 4, pp. 515–519, 2006.

[28] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.

[29] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 127–132, June 2005.

[30] M. Pant, T. Radha, and V. P. Singh, "A new particle swarm optimization with quadratic interpolation," in *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '07)*, pp. 55–60, December 2007.

[31] K. E. Parsopoulos and M. N. Vrahatis, "UPSO: a unified particle swarm scheme," in *Proceedings of the International Conference of Computational Methods in Sciences and Engineering*, vol. 1 of *Lecture Series on Computer and Computational Sciences*, pp. 868–873, 2004.