SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Hybrid quantum particle swarm optimization algorithm and its application

Yukun WANG[1,2] & Xuebo CHEN[2*]

[1]*School of Chemical Engineering, University of science and technology Liaoning, Anshan* 114051*, China;*
[2]*School of Electronic and Information Engineering, University of science and technology Liaoning, Anshan* 114051*, China*

Appendix A Generation method of *Lévy* vector

In the HQPSO algorithm, the *Lévy* vector is generated using the Mantegna rule with *Lévy* distribution characteristics. In the Mantegna rule, the step length s is designed as follows:

$$s = \frac{u}{|v|^{\frac{1}{\beta}}}. \tag{A1}$$

where,$u \sim V(0, \sigma_u^2)$,$v \sim N(0, \sigma_v^2)$, σ_u and σ_v are defined as follows:

$$\sigma_v = \left(\frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right)^{1/\beta}. \tag{A2}$$

where,$\beta = 1.5$. A three-dimensional(3D) figure of the *Lévy* flight process is shown in Figure A1.

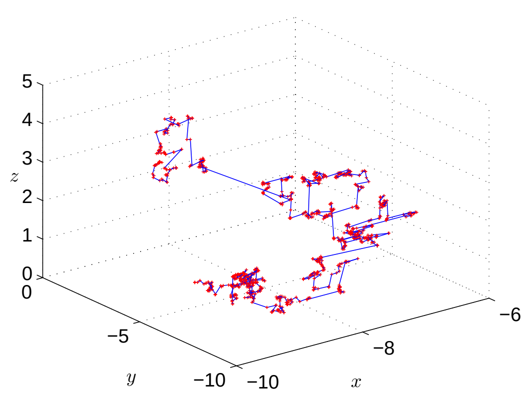


Figure A1 3D figure of *Lévy* flight process.

The figure shows that frequently small steps and occasional large steps occur alternately during the process of *Lévy* flight, which helps the algorithm jump out of local minima.

* Corresponding author (email: xuebochen@126.com)

Appendix B Curve of $\psi(t)$ with the increasing number of iterations ((The maximum iteration number is 1000).

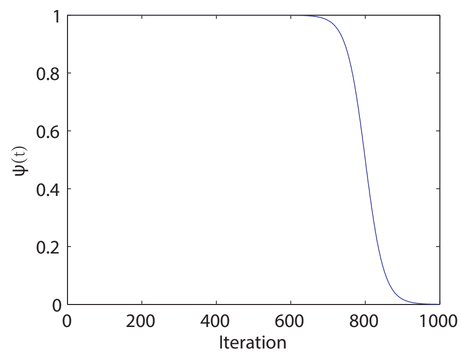


Figure B1 Curve of $\psi(t)$.

Appendix C Experimental results and statistical analysis of HQPSO algorithm and other comparison algorithms.

Appendix C.1 Benchmark functions

To evaluate the performance of HQPSO and compare it with QPSO-based evolutionary (i.e., QPSO [1],DIR-QPSO [2],HC-QPSO [4],and ESH-CQPSO [5]) and other recently improved evolutionary algorithms(i.e., SLPSOA [9], LGWO [10], E-BA [11],and NoCuSa [12]), the benchmark functions listed in Table C1 were tested using the above algorithms. The functions are divided into three classes: f_1 to f_7 are unimodal functions, f_8 to f_{13} are multimodal functions, and f_{14} to f_{18} are rotated and shifted functions. The corresponding dimensions D, range, global minimum values f_{opt}, and acceptance values of the functions are also listed in Table C1 as well.

Table C1 Benchmark functions

Name	Function	D	Range	f_{opt}	Acceptance
Shpere	$f_1(x)=\sum_{i=1}^{D}x_i^2$	30	$[-100,100]^D$	0	10^{-6}
DeJongF4	$f_2(x)=\sum_{i=1}^{D}ix_i^4$	30	$[-10,10]^D$	0	10^{-6}
Sum Square	$f_3(x)=\sum_{i=1}^{D}ix_i^2$	30	$[-100,100]^D$	0	10^{-6}
Schwefel 2.21	$f_4(x)=\max(abs(x))$	30	$[-100,100]^D$	0	10^{-6}
Schwefel 2.22	$f_5(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30	$[-10,10]^D$	0	10^{-10}
Zakharov	$f_6(x)=\sum_{i=1}^{D}x_i^2+\sum_{i=1}^{D}0.5ix_i^2+\sum_{i=1}^{D}0.5ix_i^4$	30	$[-10,10]^D$	0	10^{-6}
Rosenbrock	$f_7(x)=\sum_{i=1}^{D}[100(x_{i+1}-x_i)^2+(x_i-1)^2]$	30	$[-100,100]^D$	0	28
Ackley	$f_8(x)=-20\exp(-\frac{1}{5}\sqrt{1/\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e$	30	$[-32,32]^D$	0	10^{-6}
Schwefel	$f_9(x)=418.9829D-\sum_{i=1}^{D}x_i\sin(\lvert x_i\rvert^{\frac{1}{2}})$	30	$[-500,500]^D$	0	2500
Alpin	$f_{10}(x)=\sum_{i=1}^{D}abs(x_i\sin x_i+0.1i)$	30	$[-50,50]^D$	0	1
Rastrigin	$f_{11}(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	30	$[-5.12,5.12]^D$	0	10^{-6}
Griewank	$f_{12}(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2+\prod_{i=1}^{D}\cos(x_i/\sqrt{i})+1$	30	$[-600,600]^D$	0	10^{-6}
Michalewicz	$f_{13}(x)=\sum_{i=1}^{D}\sin(x_i)\sin(ix_i^2/\pi)^{20}$	30	$[0,\pi]^D$	-	-18
Shifted Sum Square	$f_{14}(x)=\sum_{i=1}^{D}i(x_i-\sqrt{i})^2$	30	$[-100,100]^D$	0	10^{-6}
Shifted Rosenbrock	$f_{15}(x)=\sum_{i=1}^{D}[100(y_{i+1}-y_i)^2+(y_i-1)^2]+390$, $y_i=x_i-\sqrt{i}$	30	$[-100,100]^D$	390	490
Shifted Rastrigin	$f_{16}(x)=\sum_{i=1}^{D}(y_i^2-10\cos(2\pi y_i)+10)$, $y_i=x_i-\frac{i}{D}$	30	$[-5.12,5.12]^D$	0	10^{-6}
Rotated Griewank	$f_{17}(x)=\frac{1}{4000}\sum_{i=1}^{D}y_i^2+\prod_{i=1}^{D}\cos(y_i/\sqrt{i})+1$, where $y=Mx$,M is an orthogonal matrix	30	$[-600,600]^D$	0	10^{-6}
Rotated Rastrigin	$f_{18}(x)=\sum_{i=1}^{D}(y_i^2-10\cos(2\pi y_i)+10)$, where $y=Mx$,M is an orthogonal matrix	30	$[-5.12,5.12]^D$	0	10^{-6}

Appendix C.2 Comparisons with other QPSOs

Appendix C.2.1 *Parameter settings of HQPSO and other QPSOs*

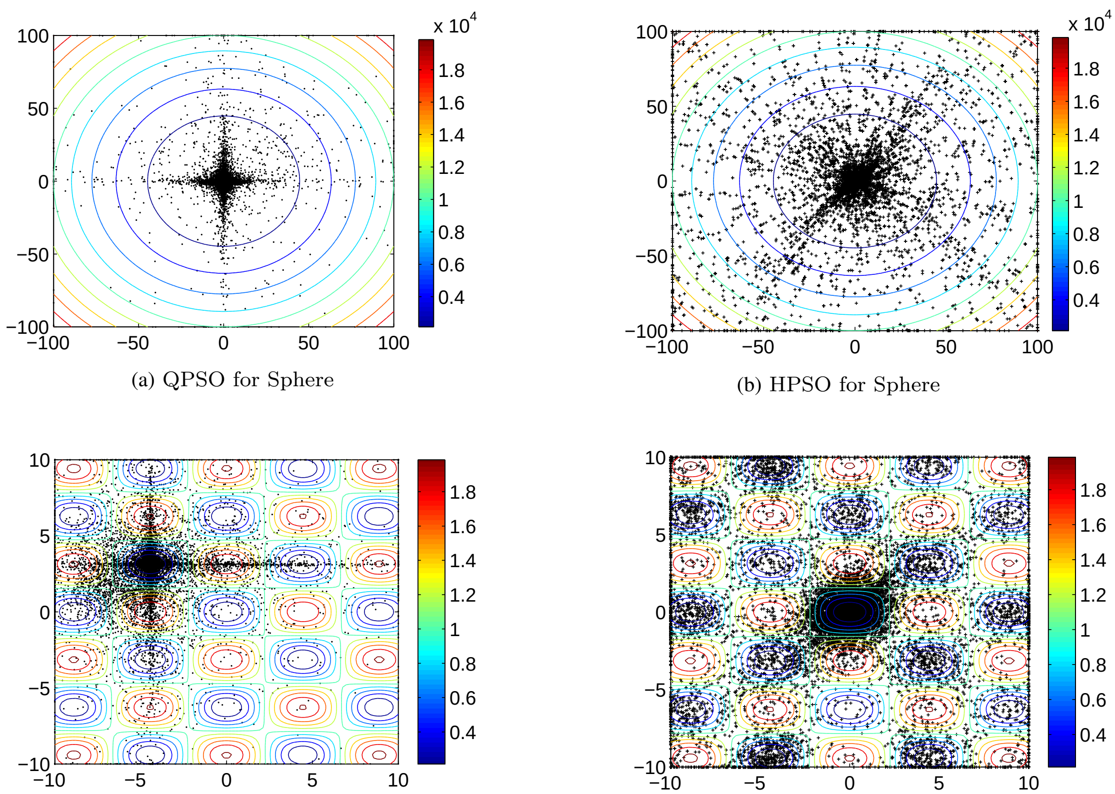
The parameter settings of HQPSO and other QSPOs are given in Table C2, where the parameter settings of the comparison algorithms are based on the suggestions in the corresponding references.

Table C2 Parameters settings of algorithms used in the comparisons

Algorithm	Reference	Parameters
QPSO	Ref.[1]	$\alpha_{max} = 1, \alpha_{min} = 0.5$
DIR-QPSO	Ref.[2]	$\alpha_{max} = 1, \alpha_{min} = 0.5, N_{s1} = N_{s2}$
HCQPSO	Ref.[4]	$\alpha_{max} = 1, \alpha_{min} = 0.5$
ESH-CQPSO	Ref.[5]	$\alpha_{max} = 1, \alpha_{min} = 0.5$
HQPSO	Present	$\lambda = 3, L = 10$

Appendix C.2.2 *Population diversity analysis of HQPSO and QPSO*

To illustrate the details of the population diversity in QPSO and HQPSO, we take two two-dimensional(2D) test functions as an example. One is the unimodal Sphere function (f_2), and the other is the multimodal Griewank function (f_{12}, where the space range is $[-10, 10]$). Firstly, we obverse the population diversity of QPSO and HQPSO through the distributions of the points during the search process. For the two minimization problems, the population size is 30 and the number of maximum iterations is 1000 for both algorithms. The distributions of the points during the search process are observed and shown in Figure C1.



(a) QPSO for Sphere

(b) HPSO for Sphere

(c) QPSO for Griewank

(d) HQPSO for Griewank

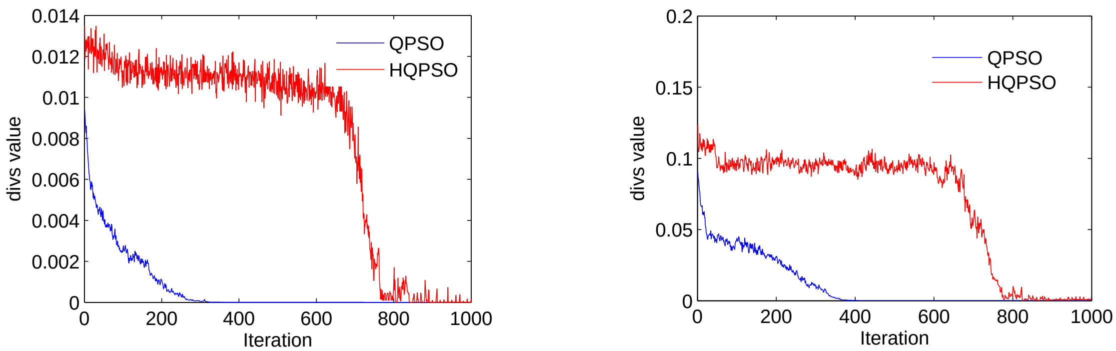
Figure C1 Comparison of distributions between QPSO and HQPSO algorithms.

It can be seen from Figure C1 that HDPSO produces a wider range of points than that of QPSO. Therefore, HQPSO has stronger abilities to lead the swarm escape from local minima.

Second, we use a diversity measure [8]to analyze the changes in diversity during the QPSO and HPSO search process. In this letter, the diversity measure is a new distance to best point measure defined as follows:

$$divs = \frac{1}{M(X_{max} - X_{min})^2} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{ij} - x_j^*)^2} \tag{C1}$$

where M denotes population size; X_{max} and X_{min} the are upper and lower limits of the search spaces, respectively; D denotes problem dimension; x_{ij} is the value of the jth dimension of the ith individual, and x_j^* is the value of the jth dimension of the current best point. A low divs value indicates that the swarm has clustered in a small region. Conversely, a high divs value indicates that the swarm has scattered over in a wide region. Hence, the diversity could be considered together with the problem and the search process of the algorithm. In this test , for the QPSO and HQPSO algorithms, the population size N is 50, problem dimension is 30 and the number of maximum iterations is 1000.The change in diversity during the search process is recorded in Figure C2.



(a) Sphere Function (b) Griewank Function

Figure C2 Diversity comparison between QPSO and HQPSO algorithms.

Figure C2 shows that the diversity of QPSO decreases rapidly with increasing iteration number, so the algorithm can very easily fall into local minima. The diversity of HQPSO can be maintained at a certain level, except in the later stages of the search process, and it has stronger abilities to lead swarm escape from local minima. In the later stages, the rapidly decreasing diversity beneficial for enhancing the convergence precision of HQPSO.

Appendix C.2.3 *Experimental results and discussion*

In this subsection, the proposed HQPSO is compared with QPSO, DIR-QPSO, HCQPSO, and ESH-CQPSO. For fair comparison, all algorithms were tested using the same population size $M = 50$ and the same maximum number of iterations $T = 1000$. In each trial, each algorithm was run 30 times independently on each test function.

1. Comparisons of convergence precision

The comparison results of convergence precision between HQPSO and other QPSOs are listed in Table C3, where "Mean" indicates the average of best fitness values averaged over 30 runs. "Std. Dev" denotes the standard deviation. The best values are highlighted in bold font. From the results in Table C3, we can draw the following conclusions:

(1) In the case of the unimodal functions, HQPSO and ESH-CQPSO could obtain the global best solution of f_2 in each trial. On functions other than f_2, HQPSO outperformed the other algorithms. QPSO performed badly on all unimodal functions, and loses in f_6 and f_7. As for f_7, only HQPSO achieved satisfactory performance. To sum up, HQPSO performed better than the other algorithms on the selected unimodal functions.

(2) As for multimodal functions, HQPSO, HCQPSO, and ESH-CPQSO can obtain the global best solution on f_{11}. DIR-QPSO, HQPSO, HCQPSO, and ESH-CPQSO can obtain the global best solution on f_{12}. As for f_7, f_{10}, and f_{13}, the best performance was achieved by HQPSO. As for f_8, the best performance was achieved by DIR-QPSO, ESH-CQPSO, and HQPSO, and their best values, worst values, mean values, and standard deviations reached 8.8817e-016, 8.8817e-016, 8.8817e-016, and 0, respectively. As for f_9, HQPSO lost only to DIR-QPSO in terms of Best and mean values, and its performance was considerably better than that of the other algorithms. Therefore, we can conclude that HQPSO has better performance than the other algorithms on the selected multimodal functions.

(3) On the rotated and shifted functions, HQPSO performed satisfactorily on f_{14} to f_{18} and lost out only to ESH-CQPSO on f_{17}. From Table C3, the best Std.Dev was achieved by HQPSO on all test functions except f_{17}. The lower Std.Dev illustrates that HQPSO has better stability. Meanwhile, convergence comparisons of all the algorithms are shown graphically in Figure C3. These comparisons confirm the results presented in Table C3.

From Table C3, the best Std.Dev is achieved by HQPSO on all test functions except f_{17}. The lower Std.Dev illustrates that HQPSO is of better stability. Meanwhile, the convergence comparisons of all algorithms are graphically shown in Figure C3 . It confirms the results in Table C3.

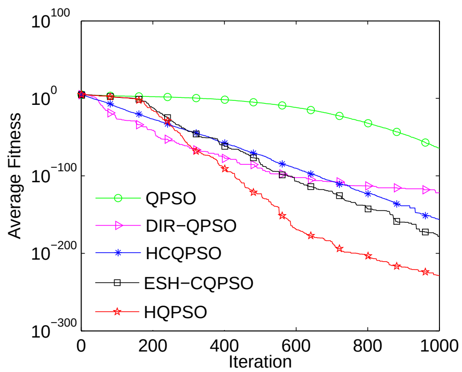
2. Comparisons on convergence speed

In solving real-world problems, fitness evaluation (FE) time overwhelms the algorithm overhead. Hence, for comparing convergence speed, the mean FE(FEs) required to reach an acceptable accuracy level would be considerably more interesting than CPU time [6]. In this letter, the FEs required to arrive at acceptable solutions in the successful runs are listed in Table C4. The best result is shown in bold font. If an algorithm can not reach its acceptable value in the search process, its FEs is denoted by "×".

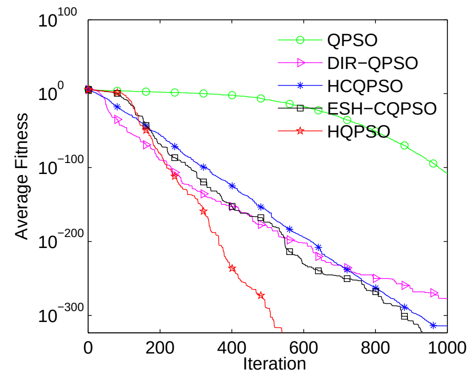
Table C3 Statistical results on test functions

Algorithm	Statistics	f_1	f_2	f_3	f_4	f_5	f_6
QPSO	Best	7.8660e-069	9.2646e-115	3.4242e-068	5.7739e-022	8.3403e-040	6.7463e-010
	Worst	8.6563e-065	6.0899e-107	5.2459e-060	2.5132e-019	3.2618e-038	9.2301e-006
	Mean	4.4739e-065	9.1765e-108	4.7531e-063	3.7184e-020	7.7575e-039	1.8392e-006
	Std.Dev	5.0442e-130	4.2097e-214	6.4574e-126	5.4133e-039	1.2058e-076	6.4335e-012
DIR-QPSO	Best	3.6258e-162	1.4535e-323	8.2134e-196	1.6902e-086	7.1579e-100	1.4245e-161
	Worst	1.1534e-121	7.0284e-258	7.4245e-104	1.9809e-070	2.2965e-078	4.2981e-125
	Mean	1.1543e-122	6.0315e-259	6.3532e-132	2.0627e-071	2.0917e-079	5.7302e-132
	Std.Dev	5.8404e-243	**0**	7.3760e-238	3.5387e-141	4.7929e-157	3.2421e-261
HCQPSO	Best	1.2953e-184	1.5363e-311	3.3224e-183	1.1059e-146	7.1333e-082	3.4294e-301
	Worst	6.4249e-156	1.7571e-309	4.5324e-150	5.4612e-093	7.1482e-072	4.5204e-202
	Mean	6.4249e-157	1.0123e-310	3.5356e-151	4.9697e-094	6.5098e-073	3.5294e-203
	Std.Dev	4.1279e-312	**0**	2.5363e-300	2.7108e-186	4.6436e-144	**0**
ESH-CQPSO	Best	3.7912e-216	**0**	4.2243e-207	1.4883e-105	2.0712e-120	2.3572e-218
	Worst	3.8951e-178	**0**	3.6743e-168	4.6182e-082	1.1572e-101	1.1435e-168
	Mean	3.8951e-179	**0**	3.6352e-169	4.1992e-082	1.0731e-102	2.3452e-169
	Std.Dev	**0**	**0**	**0**	1.9388e-162	1.2127e-203	2.0471e-317
HQPSO	Best	**1.3035e-309**	**0**	**0**	**7.0380e-150**	**3.4242e-227**	**0**
	Worst	**4.8932e-230**	**0**	**3.5643e-237**	**4.8441e-106**	**4.2542e-163**	**3.5631e-215**
	Mean	**4.9061e-231**	**0**	**3.4234e-238**	**4.4055e-107**	**3.2424e-171**	**4.5674e-241**
	Std.Dev	**0**	**0**	**0**	**2.1331e-212**	**2.9174e-336**	**0**
		f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
QPSO	Best	28.08669	8.8547e-016	8.6889e+003	5.97397	5.4352	9.3196e-006
	Worst	28.3092	4.4408e-015	9.3076e+003	73.3201	1.3246e+002	0.0231
	Mean	28.1614	4.1179e-015	8.9635e+003	18.3914	32.2323	0.01153
	Std.Dev	0.0038	1.1474e-030	5.2643e+004	3.5620e+002	1.6436e+003	2.6579e-004
DIR-QPSO	Best	27.2546	**8.8817e-016**	**3.8182e-004**	7.6479e-006	2.4225	**0**
	Worst	27.6334	**8.8817e-016**	3.7949e+003	0.7000	9.5354e+001	**0**
	Mean	27.4395	**8.8817e-016**	**1.1686e+003**	0.1083	21.6896	**0**
	Std.Dev	0.0168	0	2.0859e+006	0.0644	1.9342e+003	**0**
HCQPSO	Best	26.5518	4.4408-015	7.4634e+003	0.3000	**0**	**0**
	Worst	26.8177	7.9936e-015	8.5963e+003	11.3000	**0**	**0**
	Mean	26.6965	6.0557e-015	8.1559e+003	3.7916	**0**	**0**
	Std.Dev	0.0077	3.4423e-030	1.2726e+005	8.9990	**0**	**0**
ESH-CQPSO	Best	9.9025e-014	8.8817e-016	3.9583e+003	0.2000	**0**	**0**
	Worst	22.86823	8.8817e-016	5.2797e+003	4.7000	**0**	**0**
	Mean	2.0789	8.8817e-016	4.5014e+003	1.8833	**0**	**0**
	Std.Dev	47.5413	**0**	2.0521e+005	1.8815	**0**	**0**
HQPSO	Best	**4.4674e-018**	**8.8817e-016**	1.6367e+003	**4.7220e-006**	**0**	**0**
	Worst	**2.0489e-009**	**8.8817e-016**	**2.3065e+003**	**4.5278e-004**	**0**	**0**
	Mean	**2.5250e-010**	**8.8817e-016**	1.9470e+003	**1.4736e-004**	**0**	**0**
	Std.Dev	**4.0164e-019**	**0**	**4.0381e+004**	**1.3962e-008**	**0**	**0**
		f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
QPSO	Best	-20.6514	1.5538e-008	4.1480e+002	9.5448	2.0144e-011	16.1655
	Worst	-12.5972	3.2287e-006	1.1922e+003	41.2121	0.0467	32.9471
	Mean	-15.0762	4.6135e-007	6.4062e+002	26.3916	0.0139	23.9673
	Std.Dev	7.9384	8.7674e-013	1.1197e+005	73.6934	2.9293e-004	21.0433
DIR-QPSO	Best	-20.9575	1.7543e-007	4.0439e+002	19.8995	5.5178e-014	3.1942e-004
	Worst	-13.5859	9.2656e-006	4.1726e+002	43.7786	2.0838e-013	0.0011
	Mean	-18.5097	1.4439e-006	4.0714e+002	26.7981	1.1562e-013	5.9164e-004
	Std.Dev	4.3585	6.8737e-012	13.1782	36.8351	2.0960e-027	5.4644e-008
HCQPSO	Best	-15.3225	4.3099e-008	4.1982e+002	0.0353	1.2741e-009	0.0055
	Worst	-9.5057	2.7781e-007	4.9759e+002	3.01185	2.9893e-007	0.0195

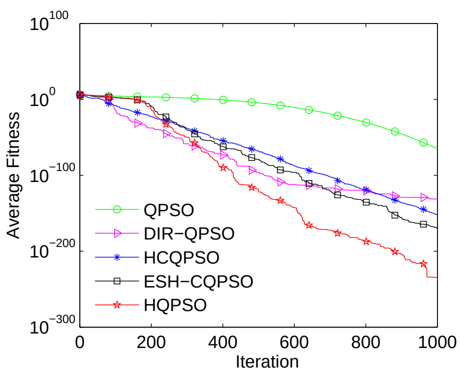
Algorithm	Statistics	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
	Mean	-11.4797	1.0232e-007	4.4284e+002	1.3936	4.2209e-008	0.0106
	Std.Dev	2.6859	4.7773e-015	4.8276e+002	0.8775	6.7702e-015	9.7929e-006
ESH-CQPSO	Best	-20.5971	5.0467e-004	4.0242e+002	0.0188	0	0.0112
	Worst	-18.0571	0.0033	5.2325e+002	0.0648	0	0.0556
	Mean	-19.0912	0.0015	4.2604e+002	0.0383	0	0.0321
	Std.Dev	0.8222	5.6207e-0076	1.3187e+003	1.6321e-004	0	1.6129e-004
HQPSO	Best	**-21.5724**	**1.8009e-011**	**4.0038e+002**	**2.1232e-013**	**0**	**4.5474e-013**
	Worst	**-18.6207**	**1.1681e-009**	**4.1159e+002**	**4.6361e-010**	2.2204e-016	**6.7643e-012**
	Mean	**-20.5760**	**2.6934e-010**	**4.0652e+002**	**7.6537e-011**	3.7007e-017	**1.6029e-012**
	Std.Dev	**0.6765**	**1.6578e-019**	**8.2104**	**2.3526e-022**	5.2291e-033	**2.7202e-024**



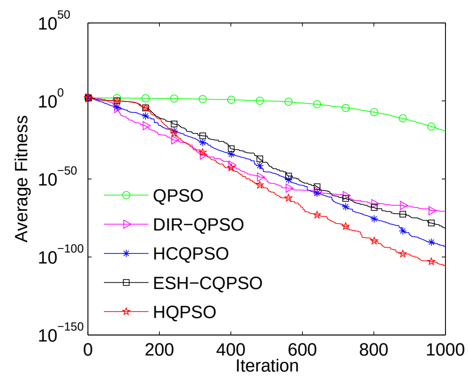
(a) f_1



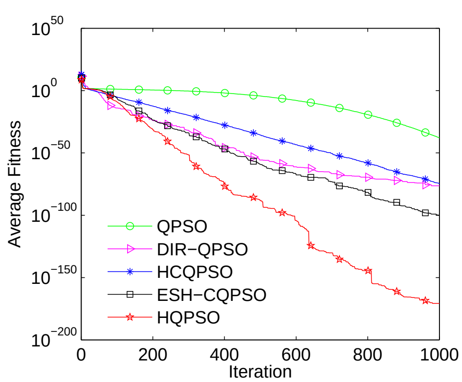
(b) f_2



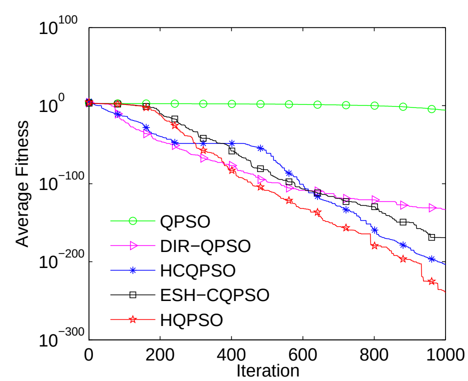
(c) f_3



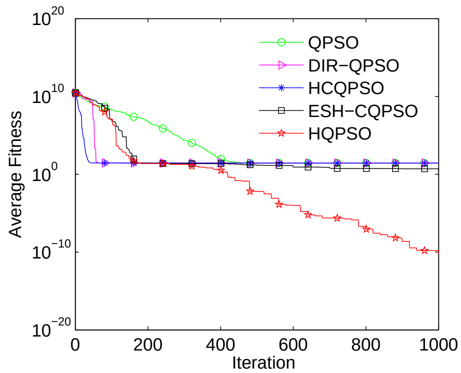
(d) f_4



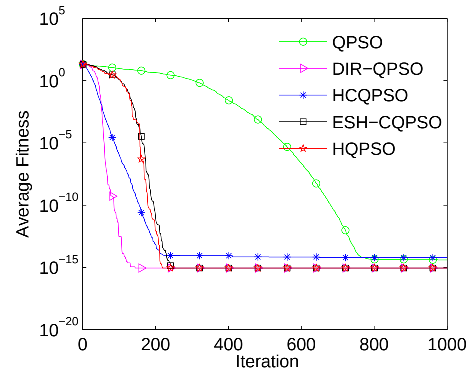
(e) f_5



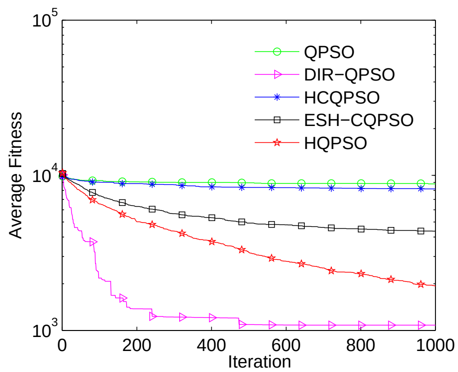
(f) f_6



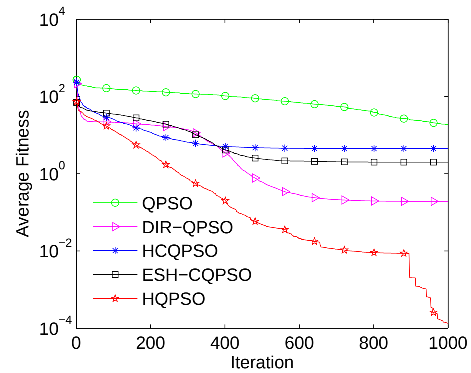
(g) f_7



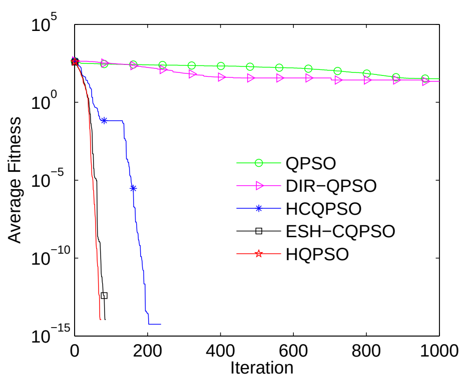
(h) f_8



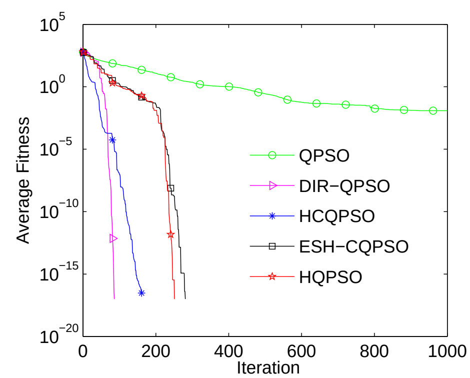
(i) f_9



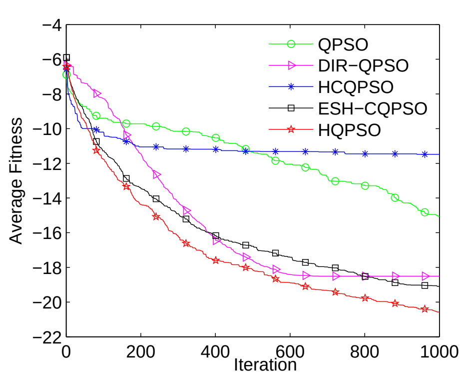
(j) f_{10}



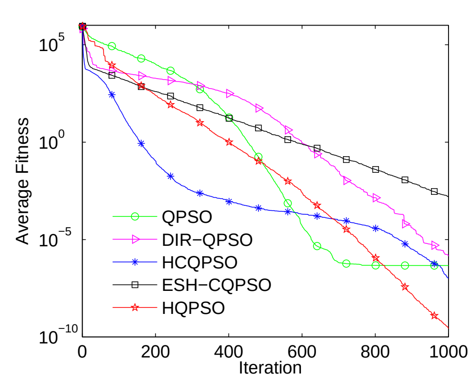
(k) f_{11}



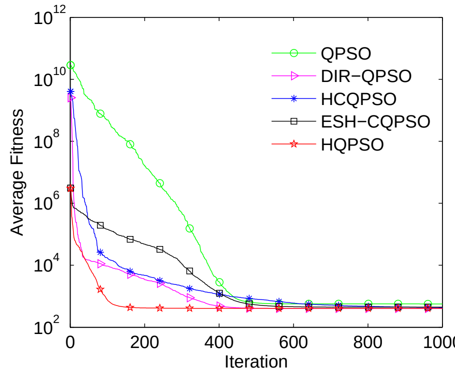
(l) f_{12}



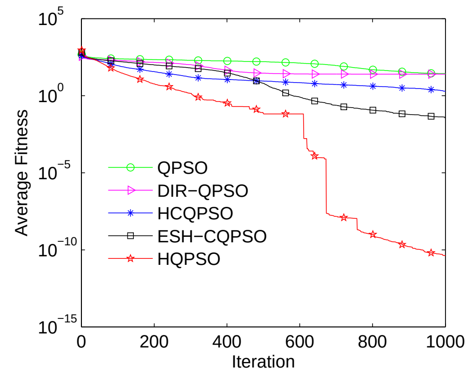
(m) f_{13}



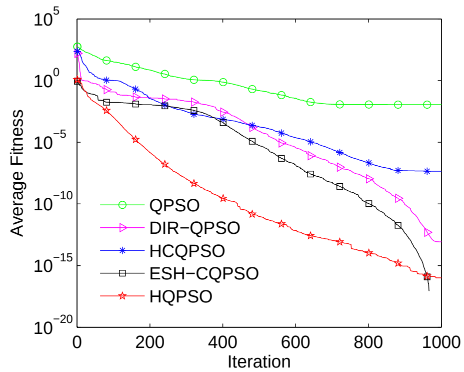
(n) f_{14}



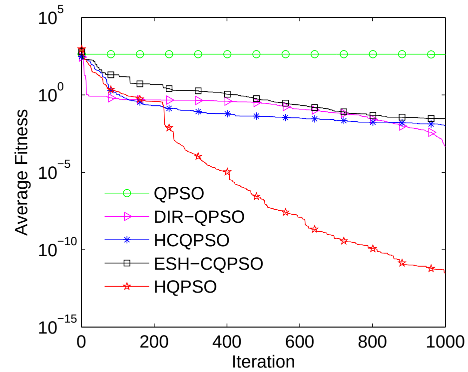
(o) f_{15}



(p) f_{16}



(q) f_{17}



(r) f_{18}

Figure C3 Comparisons of convergence speed.

From the results in Table C4, the convergence speed of the improved QPSO algorithms is significantly better than that of QPSO. DIR-QPSO has the best convergence speed on f_1,f_2,f_3,f_4,f_5, f_8, f_9, and f_{12}. HQPSO has the best convergence speed on f_7, f_{10}, f_{11}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, and f_{18}. HCQPSO has the best convergence speed only on f_6. HQPSO successfully reached the acceptance value on all test functions, DIR-QPSO lost in f_1, f_{12}, f_{16}, and f_{18}, and HCQPSO lost in f_{11}, f_{12}, f_{16}, and f_{18}. QPSO had the worst performance in the comparison algorithms, and its success rate on all test functions was only 67%. The most interesting result was that HQPSO converged faster on multimodal, shifted, and rotated functions; thus, DIR-QPSO has better convergence speed on unimodal functions. In real-world problems, multimodal, shifted, and rotated functions are more common, so HQPSO is more suitable for solving real-world optimization problems.

Table C4 Statistical results of FEs on test functions

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
QPSO	25246	22811	25413	25004	26334	31647	×	29524	×
DIR-QPSO	**3414**	**2263**	**3645**	**3424**	**3346**	9147	48051	**3435**	**6061**
HCQPSO	3607	3341	4157	3864	4327	**4645**	47775	16566	×
ESH-CQPSO	8352	4342	8962	10057	4711	11611	19534	9252	×
HQPSO	7643	4118	8014	8433	4432	10323	**14439**	8948	39547
	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
QPSO	×	×	×	×	×	47109	×	×	×
DIR-QPSO	21824	×	4213	29613	×	34333	×	44102	×
HCQPSO	×	8322	4542	×	48217	43404	×	37632	×
ESH-CQPSO	×	3314	10150	29441	×	46301	×	31229	×
HQPSO	**10137**	**3071**	9264	**26642**	**42338**	**33639**	**260570**	**10233**	**20341**

3. Comparison on other indicators

To further illustrate the superiority of the proposed HQPSO algorithm, comparison results of success rate (SR, the percentage of successful runs in which acceptable solutions are obtained), "$w/t/l$" and ranking performances (ranking by mean value listed in table C3) among HQPSO and the other compared algorithms are listed in Table C5. Here, SR stands for the probability that the tested algorithm has successfully reached the acceptance value for each tested function, and

"$w/t/$" means that the HQPSO wins against w algorithms, ties with t algorithms, and loses against l algorithms, compared to other QPSO algorithms.

Table C5 Statistical results of SR, "$w/t/l$" and ranking performance

Algorithm		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
	SR(%)	100	100	100	100	100	63	0	100	0
QPSO	$w/t/l$	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	1/0/3	0/0/4
	Rank	5	5	5	5	5	5	5	4	5
	SR(%)	100	100	100	100	100	100	100	100	80
DIR-QPSO	$w/t/l$	2/0/1	1/0/3	3/0/1	3/0/1	2/0/2	3/0/1	1/0/3	2/0/2	4/0/0
	Rank	4	4	2	2	3	2	4	3	1
	SR(%)	100	100	100	100	100	100	100	0	0
HCQPSO	$w/t/l$	0/0/3	2/0/2	2/0/2	1/0/3	1/0/3	1/0/3	2/0/2	0/0/4	1/0/3
	Rank	3	3	3	4	4	4	3	5	4
	SR(%)	100	100	100	100	100	100	100	100	0
ESH-CQPSO	$w/t/l$	1/0/2	3/1/0	1/0/3	2/0/2	3/0/1	2/0/2	3/0/1	3/1/0	2/0/2
	Rank	2	1	4	3	2	3	2	1	3
	SR(%)	100	100	100	100	100	100	100	100	100
HQPSO	$w/t/l$	4/0/0	3/1/0	4/0/0	4/0/0	4/0/0	4/0/0	4/0/0	3/1/0	3/0/1
	Rank	1	1	1	1	1	1	1	1	2
		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
	SR(%)	0	43	0	0	73	40	0	80	0
QPSO	$w/t/l$	0/0/4	0/0/4	0/0/4	0/4/0	2/0/2	0/0/4	1/0/3	0/0/4	0/0/4
	Rank	5	5	5	4	3	5	4	5	5
	SR(%)	100	83	100	50	53	100	0	100	0
DIR-QPSO	$w/t/l$	3/0/1	1/0/3	1/3/0	2/0/2	1/0/3	3/0/1	0/0/4	2/0/2	3/0/1
	Rank	2	4	1	3	4	2	5	3	2
	SR(%)	0	100	100	0	100	93	0	100	0
HCQPSO	$w/t/l$	1/0/3	2/2/0	1/3/0	0/0/4	3/0/1	1/0/3	2/2/0	1/0/3	2/0/2
	Rank	4	1	1	5	2	4	3	4	3
	SR(%)	0	100	100	100	0	77	0	100	0
ESH-CQPSO	$w/t/l$	2/0/2	2/2/0	1/3/0	3/0/1	0/0/4	2/0/2	3/1/0	4/0/0	1/0/3
	Rank	3	1	1	2	5	3	2	1	4
	SR(%)	100	100	100	100	100	100	100	100	100
HQPSO	$w/t/l$	4/0/0	2/2/0	1/3/0	4/0/0	4/0/0	4/0/0	4/0/0	3/0/1	4/0/0
	Rank	1	1	1	1	1	1	1	1	2

Form table C5, HQPSO has a 100% success rate on all test functions. HQPSO has rank "1" on all test functions, except f_9 and f_{917}. As forf_9 and f_{917}, HQPSO lost only against DIR-QPSO and ESH-CQPSO, respectively. TableC6 provides an overall comparison among all QPSOs. The total rank and final rank of each comparison algorithm are listed in TableC6.

Table C6 Total and final rank of each comparison algorithm

Rank	Algoritm				
	QPSO	DIR-QPSO	HCQPSO	ESH-CQPSO	HQPSO
Total rank	85	51	60	42	20
Final rank	5	3	4	2	1

From the results presented in Table C6, HQBPSO achieved the lowest total rank, that is, "20", and the best final rank, that is, "1". This indicates that HQPSO is superior in terms of overall performance on all test functions than the other algorithms. We can obviously observe that the total rank value of HQPSO is considerably less than that of the other algorithms. This indicates that the overall performance of HQPSO is considerably better than that of the other algorithms.

Appendix C.3 Comparisons with other recently improved evolutionary algorithms

To further illustrate the superiority of the proposed HQPSO algorithm, we compare it with the other recently improved evolutionary algorithms listed in Table C7.

Table C7 Comparison algorithms and their parameters settings

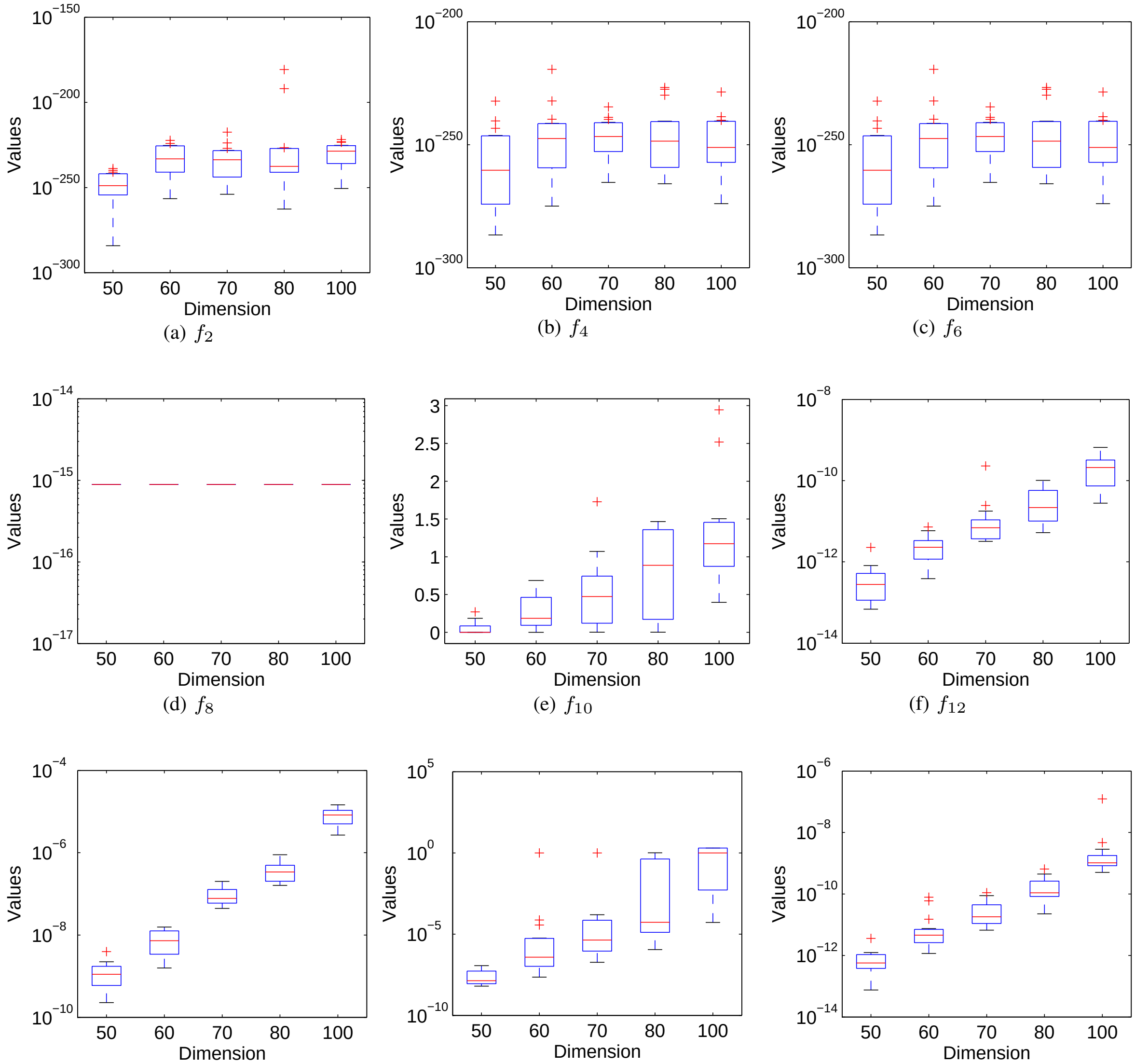
Algorithm	Reference	year	parameters
SLPSOA	Ref.[9]	2017	$M = 10, \alpha = 0.1, G = 5$
LGWO	Ref.[10]	2017	$\alpha_0 = 2, \beta \sim U(0,1), p \sim U(0,1),$
EBA	Ref.[11]	2017	$r_0 = 0.1, A_0 = 0.95, f_{min} = 0, f_{max} = 2, Limit1 = 0.8, Limit2 = 0.5$
NoCuSa	Ref.[12]	2017	$pa = 0.3, \alpha = 1.1, \beta = 1.7, \delta = 1.6$
HQPSO	Present	–	$\lambda = 3, L = 10$

These algorithms include scatter learning particle swarm optimization algorithm (SLPSOA) [9], grey wolf optimization algorithm with Lvy flight (LGWO) [10], enhanced bat algorithm with mutation operator (EBA) [11], and nonhomogeneous cuckoo search algorithm based on quantum Mechanism (NoCuSa) [12]. For a fair comparison, all algorithms were tested using the same population size $M = 50$, except NoCuSa (NoCuSa evaluates fitness twice per iteration, and the population size $M = 25$), and the same maximum number of iterations $T = 1000$. In each trial, each algorithm was run 30 times independently, and the dimension of each selected function was 40. The acceptance value of f_{13} was -24, and the other selected functions are listed based on their acceptance values in TableC1.

Table C8 Comparison results with other evolutionary algorithms

Algorithm	Statistics	f_1	f_3	f_5	f_7	f_9	f_{11}
SLPSOA	Mean	3.4524e-036	1.5309e-041	1.0391e-63	0.8371	**21.4562**	7.3928e-06
	Std.Dev	1.4292e-071	3.3940e-080	3.8374e-122	0.2109	**4.2983**	4.5729e-10
	FEs	21409	23502	11093	15072	**27832**	10352
LGWO	Mean	2.3442e-028	5.3492e-032	5.4534e-e41	12.3343	2.9203e+003	0.1235
	Std.Dev	1.3424e-41	3.4933e-056	3.2040e-78	3.2349	4.4245e+005	1.0134
	FEs	9327	8429	12764	15357	×	×
EBA	Mean	2.3031e-201	5.3924e-223	**0**	6.3456	2.6434e+003	**0**
	Std.Dev	0	0	**0**	1.4029	1.4535e+005	**0**
	FEs	8164	8915	**4319**	13641	×	**4165**
NoCuSa	Mean	3.0931e-136	3.0493e-153	3.9284e-114	4.2134	3.3294e+003	4.5920e-006
	Std.Dev	1.3423e-270	5.3824e-301	1.9238e-225	1.2904	6.3123e+005	1.3947e-011
	FEs	9103	**8130**	8279	16837	×	9712
HQPSO	Mean	**2.4713e-227**	**7.5374e-261**	3.9204e-151	**1.7886e-08**	2.2465e+003	**0**
	Std.Dev	**0**	**0**	1.1037e-300	**8.6573e-017**	4.0483e+004	**0**
	FEs	**8148**	8313	4764	**14952**	41546	4354
		f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
SLPSOA	Mean	-24.2943	1.4829e-007	4.1984e+002	1.2938e-007	3.4342e-007	1.6464e-010
	Std.Dev	1.2901	2.1947e-013	17.3903	3.2298e-011	1.3985e-014	6.3019e-020
	FEs	41094	48293	34013	27439	15432	23410
LGWO	Mean	-17.4452	4.5367e-007	4.3436e+002	4.4214	2.3552e-011	0.1343
	Std.Dev	4.2983	1.3940e-012	17.4354	2.4512	1.3435e-010	1.1372
	FEs	× 45354	37548	×	13563	×	
EBA	Mean	-9.3247	4.9204e-008	4.6453e+002	9.3942	1.3944e-014	3.4920e-009
	Std.Dev	0.9345	7.3534e-015	15.7453	4.6361	2.3492e-027	1.0293e-015
	FEs	×	45635	35245	×	11204	23532
NoCuSa	Mean	-25.4521	2.3903e-007	4.8214e+002	5.2942	7.3054e-08	1.3948e-007
	Std.Dev	2.4124	1.4920e-014	23.4209	2.2391	1.3428e-015	5.2304e-013
	FEs	43357	47292	41045	×	13871	23492
HQPSO	Mean	**-25.6128**	**4.3491e-09**	**4.1703e+002**	**7.2014e-010**	**1.5829e-016**	**9.4903e-012**
	Std.Dev	**2.3620**	**6.4920e-017**	**12.9203**	**1.4830e-020**	**3.5929e-031**	**1.0949e-023**
	FEs	**39463**	**45564**	**33902**	**26305**	**10552**	**20737**

Form Table C8, HQPSO and SLPSOA have 100% success rate on all selected functions. LGWO loses on f_9, f_{11}, f_{13}, f_{16}, and f_{18}; EBA loses on f_9, f_{13}, and f_{16}; And NoCuSa loses on f_9 and f_{16}. HQPSO achieved the best performance of convergence precision on f_1, f_2, f_7, and f_{13} to f_{18}, and achieved the best convergence speed on all test functions except f_3, f_5, f_9, and f_{11}. HQPSO has the most promising performance among these algorithms on the selected functions.



(a) f_2 (b) f_4 (c) f_6

(d) f_8 (e) f_{10} (f) f_{12}

(g) f_{14} (h) f_{16} (i) f_{18}

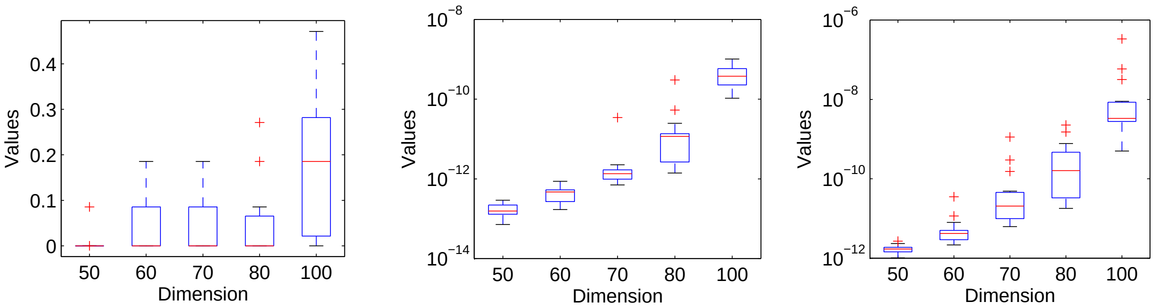
Figure C4 Statistical results of HQPSO on high-dimensional functions.

Appendix C.3.1 *Adaptability of HQPSO to high-dimensional functions*

In this subsection, the adaptability of HQPSO to high-dimensional functions is shown in Figure C4. In Figure C4, all selected functions were tested using the same population size $M = 50$ and the same maximum number of iterations $T = 2000$. For all selected functions, f_2, f_4, and f_6 are unimodal functions; f_8, f_{10}, and f_{12} are multimodal functions; and f_{14}, f_{16}, and f_{18} are rotated and shifted functions. To better illustrate the adaptability of HQPSO to high-dimensional functions, five different dimensions (i.e., 50, 60, 70, 80, and 100) are considered for each test function. In each trial, HQPSO was run 30 times independently on each dimension of the test functions. Each boxplot in Figure C4 shows the statistical results of the best fitness values achieved by HQPSO.

Figure C4 shows that HQPSO successfully achieves the acceptance value listed in Table C1 on all selected test functions, except f_{10}, f_{14} and f_{16}. The most interesting result is that the statistical results of each test function worsen with increasing dimensions, exceptf_8. For the maximum number of iteration $T = 5000$, the statistical results of f_{10}, f_{14}, and f_{16} are shown in Figure C5. Figure C5 shows that HQPSO successfully achieves the acceptance value on f_{10}, f_{14}, and f_{16} when $T = 5000$. Therefore, we can conclude that it will become increasingly difficult for HQPSO to obtain the optimal solution with increasing dimensions, and the problem can be solved by increasing the number of iterations.

From the comparison results of all test algorithms and the adaptability of the HQPO algorithm on high-dimensional functions, we can conclude that the proposed HQPO performed better than the other algorithms. It had better convergence precision, speed and stability, and could complete the optimization of complex high-dimensional functions.



(a) f_{10} (b) f_{14} (c) f_{16}

Figure C5 Statistical results of f_{10}, f_{14}, and $f_{16}(T = 5000)$.

Appendix D Parameters settings of HQPSO, Gupta function and its parameters for Au cluster.

The parameters of the HQPSO algorithm were set as follows: search range $[-5, 5]$, population size $M = 100$, max iteration number $T = 10^5$, $\lambda = 0.001$, and $L = 30$.

The equation of the Gupta potential function is as follows:

$$V = \sum_i \left(\sum_{j(\neq i)} A \exp(-p(\frac{r_{ij}}{r_0} - 1)) - \sqrt{\sum_{j(\neq i)} \xi^2 \exp(-2q(\frac{r_{ij}}{r_0} - 1)} \right) \tag{D1}$$

where, r_{ij} is the distance of the ith and jth atoms; For Au cluster, $A = 0.1156eV$, $p = 16.980$, $q = 2.691$, $\xi = 2.289eV$ and $r_0 = 2.71\mathring{A}$[8].

Appendix E Energy comparison results achieved using HQPSO and the current best value [8].

Table E1 Energy comparisons of Au $(n = 12 - 30)$ clusters

No.	n	Ea	Eb
1	12	-38.92256917465254	-38.92256917
2	13	-42.55493083681036	-42.55493084
3	14	-45.95827566086828	-45.95827566
4	15	-49.50545131022577	-49.50545129
5	16	-53.05765637989333	-53.05765637
6	17	-56.54939854621972	-56.54939854
7	18	-60.01051500664349	-60.01051500
8	19	-63.43510905636050	-63.43510905
9	20	-66.86962079634721	-66.86962080
10	21	-70.35745012379466	-70.35745012
11	22	-73.92076803655492	-73.92076803
12	23	-77.47804456077746	-77.47804456
13	24	-81.00593616084861	-81.00593615
14	25	-84.43636253003858	-84.43636252
15	26	-87.93389515032121	-87.93389515
16	27	-91.50188836846107	-91.50188836
17	28	-95.03354850404784	-95.03354849
18	29	-98.62039883108864	-98.62039882
19	30	-102.2182980311356	-102.21829802

Ea, achieved by HQPSO. Eb, the current best value listed in literature 8.

References

1 Wang G G, Gandomi A H, Alavi A H, et al. A hybrid method based on krill herd and quantum-behaved particle swarm optimization. Neural Comput Appl, 2016, 27: 989-1006

2 Wu T, Yan Y S, Chen X. Improved dual-group interaction QPSO algorithm based on random evaluation. Control Decision, 2015, 3: 526-530

3 Rehman O U, Yang J, Zhou Q, et al. A modified QPSO algorithm applied to an engineering inverse problems in electromagnetics. Int J Appl Electrom, 2017, 54: 1-15

4 Luo Q, Gong Y, Jia C. Stability of gene regulatory networks with Lvy noise. Sci China Ser F-Inf Sci, 2017, 60: 072204

5 Turgut O E. Hybrid Chaotic Quantum behaved Particle Swarm Optimization algorithm for thermal design of plate fin heat exchangers. Appl Math Model, 2016, 40: 50-69

6 Zhao J, Fu Y, Mei J. An improved cooperative QPSO algorithm with adaptive mutation based on entire search history. Acta Electronica Sini, 2016, 44: 2900-2907

7 Zhang J, Dolg M. ABCluster: The artificial bee colony algorithm for cluster global optimization. Phys Chem Chem Phys, 2015, 17: 24173

8 Wang Q WLi X QChen H J, et al. The study of structures of gold clusters by artifical bee colony algorithm. J At Mo Phys, 2017, 34: 1040-1048

9 Ren Z, Zhang A, Wen C, et al. A scatter learning particle swarm optimization algorithm for multimodal problems. IEEE Trans Cybern, 2017, 44: 1127-1140

10 Heidari A A, Pahlavani P. An efficient modified grey wolf optimizer with *Lévy* flight for optimization tasks. Appl Soft Comput, 2017, 60: 115-134

11 Ghanem W A H M, Jantan A. An enhanced Bat algorithm with mutation operator for numerical optimization problems. Neural Comput Appl, 2017, 1: 1-35

12 Cheung N J, Ding X M, Shen H B. A nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization. IEEE Trans Cybern, 2017, 47: 391-402