Particle swarm optimisation with multi-strategy learning

Guohan Lin*

Hunan Province Cooperative Innovation Center for Wind Power Equipment and Energy Conversion,
Hunan Institute of Engineering,
Xiangtan 411104, Hunan, China
Email: lgh@hnu.edu.cn
*Corresponding author

Jing Sun

College of Electrical and Information,
Hunan Institute of Engineering,
Xiangtan 411104, Hunan, China
Email: 357477191@qq.com

Abstract: To ease the conflict between diversity and convergence rate encountered by Particle Swarm Optimisation (PSO), a multi-strategy learning PSO Algorithm (Multi-strategy Learning PSO, MSLPSO) is proposed. The proposed method can effectively preserve the heuristic information; a modified differential mutation is combined with PSO to expand search range and to increase the diversity of the population. The inferior particle adopts opposition-based learning when the population was trapped into local optimum. This mechanism can improve the diversity and can help the particles' flight away from the local optimum. Gaussian disturbance is applied to elite particles to further improve the diversity of particles. Twelve benchmark function tests from CEC2005 are used to evaluate the performance of the proposed algorithm. The results show that the proposed multi-strategy learning has performed consistently well compared to other state-of-art PSO algorithms.

Keywords: PSO; particle swarm optimisation; learning strategy; differential mutation; perturbation strategy; numerical optimisation.

Reference to this paper should be made as follows: Lin, G. and Sun, J. (2020) 'Particle swarm optimisation with multi-strategy learning', *Int. J. Wireless and Mobile Computing*, Vol. 18, No. 1, pp.22–30.

Biographical notes: Guohan Lin received his BS and MS degrees from Hunan University in 1996 and 2005, respectively. He is currently an Associate Professor in College of Electrical and Information Engineering, Hunan Institute of Engineering. His current research interests include evolutionary computation techniques, electric machine drives, and intelligent control theory.

Jing Sun received the PhD degree in Materials Science and Engineering in 2012 from Xiangtan University, Xiangtan, China. She is currently an Associate Professor in Hunan Institute of Engineering. Her research interests include semiconductor device physics and processing, device modelling, and fabrication of the ferroelectric random access memory. Her current research is focused on device modelling and application.

1 Introduction

Particle Swarm Optimisation (PSO) was proposed by Eberhart and Kennedy (1995), which was inspired by the bird flocking social behaviour. Owing its structural simplicity and few parameters, PSO has been widely applied in many fields such as function optimisation (Qin et al., 2015), image processing (Liu et al., 2015), neural networks (Xiao et al., 2018), electric power systems, parameter identification (William et al., 2018).

However, it is verified by theoretical analysis and experiments that the basic PSO algorithm lacks the ability to jump out from local optima, which means the PSO cannot search through the solution space of the solving problem (Zhan et al., 2009). To ease this drawback, scholars proposed many strategies to improve the performance of basic PSO on diverse aspects. Shi and Eberhart (1999) introduced the inertia weight to PSO, adjusted the inertia weight linear (or non-linear) according to particle flying position. Alireza (2011) presented the dynamic inertia weight adjustment strategies to strengthen the global search ability and search precision. Adaptive mutation strategy is introduced into the PSO to improve its performance (Wang et al., 2013). Unlike basic PSO algorithm where the particles are updated based on historical information, each particle in social learning PSO learns from any better particles in the current swarm (Cheng and Jin, 2015). To increase the diversity of the population, the researchers dedicated to design different types of topologies to change the particle learning mode. Mendes et al. (2004) proposed several basic neighbourhood structure, such as ring and star structure, and analysis the effects of different neighbourhood structure on optimisation results. Particles move toward nearby particles of higher fitness (Euclidean distance), instead of attracting each particle towards just the best position.

Combined with co-evolutionary thought, Van den Bergh and Engelbrecht (2004) proposed Cooperative Particle Swarm Optimiser (CPSO) algorithm, the CPSO divided particles into several sub-swarms, each sub-swarm optimises different components of the solution vector cooperatively.

In CLPSO (Liang et al., 2006), each particle learned from select particles in different dimensions, this strategy enables the diversity of the particles to be preserved and avoiding of premature convergence.

Combined PSO with another optimiser to form hybrid is another improvement strategy. In Aydilek (2018), local search in PSO is combined with global search in artificial bee colony to solve high-dimensional optimisation problems, the proposed algorithm adopts different search strategies according to the ageing degree of pbest for each individual. Wang et al. (2018) developed a hybrid PSO algorithm which employs an adaptive learning strategy to balance the exploration ability and the exploitation ability well. Xia et al. (2014) integrated tabu detecting, shrinking and local learning into PSO to overcome the problems of premature convergence and slow convergence speed, search space of each dimension is divided into many equal sub-regions, through statistical analysis of the historical evolution information, which sub-region should be the focus is decided. In Smairi et al. (2016), a multi-objective algorithm of hybrid particle swarm optimisation with tabu search is presented, tabu search is applied on the particles of the archive, the best particle of each tribe and each particle of the swarm. Dynamic multi-swarm particle swarm optimiser hybrid with cooperative learning strategy is developed to achieve a good balance between the exploration and exploitation abilities, using tournament selection strategy, each dimension of the two worst particles in each sub-swarm learn from two better particles randomly selected from two sub-swarms (Xu et al., 2015).

Multi-swarm PSO with sharing of information and elitist perturbation learning is proposed in Zhao et al. (2014), via information communicating and sharing among different sub-swarms with different evolution mechanisms the evolving efficiency can improve. Elitist perturbation learning strategy can enlarge the exploration domain and diversify the flying tracks of particles.

To ease the conflict between diversity and convergence rate encountered by PSO, particle swarm algorithm with multi-strategy learning (Multi-strategy Learning PSO, MSLPSO) is presented. In each iteration, local search is enforced via disturbing learning of elite particles, when the premature convergence is detected, a new opposition-based learning strategy is applied to the worst particle, this strategy can helps the particle to search more effective areas while improving the diversity of population and enhancing the global exploration capability of the algorithm, DE operator is incorporated to update the previous best positions of particles to force population jump out of stagnation. The performance of the proposed MSLPSO is evaluated by solving twelve of the CEC2005 contest functions. The results show that the proposed algorithm speeds up the convergence rate and improves the algorithm's performance.

2 Multi-strategy learning PSO

2.1 Basic PSO

In the basic PSO, a swarm with N individuals are generated randomly in D-dimensional search space. Each individual in the swarm represents a candidate solution, the i-th individual's position and velocity is represented by $X_i = \{x_{i1}, x_{i2}, ..., x_{iD}\}$ and $V_i = \{v_{i1}, v_{i2}, ..., v_{iD}\}$, respectively. The particles fly through the D-dimensional problem space by learning from the best flying experience of itself and all others particles. The velocity and position of the d-th of the i-th particle are updated as follows:

$$\begin{aligned} v_{id}(t+1) &= wv_{id}(t) + c_1 r_1 \left(p_{id}(t) - x_{id}(t)\right) + c_2 r_2 \left(p_{gd}(t) - x_{id}(t)\right) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \quad (1)$$

where t is the current iteration number, w is the inertia weight, c_1, and c_2 are the acceleration coefficients, r_1 and r_2 are two random functions between the range of [0, 1]; p_{id} represents the best location in the problem space ever found by i-th particle, and p_{gd} is the best location visited by all individuals so far.

2.2 Multi-strategy learning PSO

Similar to other swarm intelligence algorithms, contradiction between population diversity and the rate of convergence exists in basic PSO. The main purpose of the improvements for basic PSO, whether the suitable parameters selection, niche technology, or hybrid strategy, is to strengthen the local search capabilities while maintaining the diversity of the population, and to prevent premature convergence. MSLPSO algorithm is also based on this idea that relying on the opposition-based learning to maintain the diversity of the population while strengthening the local search capacity of the best individual in history.

1) *Improved differential mutation:* Considering that in basic PSO, particle adjusts its flight direction according

to its history best position and the global best position, however, this learning strategy cannot provide adequate diversified information for particle to search in the solution space, and particles are prone to trap into local optimum. To deal with this drawback, an improved differential mutation considering the mean fitness was applied to basic PSO.

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) + c_3 r_3 (x_u - x_l) \quad (2)$$

where x_u and x_l are randomly select particles which fitness value is superior and inferior to the particle with mean fitness value respectively. c_3 is the weight factor that control the particle flying direction, r_3 is random function in the range [0, 1]. The differential mutation vectors make the particles not only follow the individual optimum, but also learn from the global optimum. Each individual adjust space flight direction and search behaviour by dynamically changing the direction and step size of the difference term. This strategy ensures that particles fly in a better direction at each iteration.

2) *Elite disturbance learning strategy*: In basic PSO, at the late stage of iteration, the particles will gather near the optimal position, the personal history best position is very close to the global best position, in this case, the search efficiency decreases rapidly. When the global optimum particle located in or neighbourhood by the local optima location, the entire population convergence fast around the local optimum, which result in rapid decrease of population diversity, and the algorithm is likely to stagnate and trap into local optimum. To overcome this drawback, elite disturbance was applied to proposed PSO to make the stagnate particles regain some momentum for prompting particle ongoing exploration in the search space.

$$p^*_{gd} = N(p_{gd}, \sigma) \quad (3)$$

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p^*_{gd}(t) - x_{id}(t)) \quad (4)$$

$$\sigma = \sigma_{\max} (1 - random(0,1))^{(1-t/T)^c} \quad (5)$$

during the iteration, σ decrease non-linear, σ max=0.5, c =6, t is current iteration number, T is the maximum number of iterations. In the early stage, σ is larger, particle focused on global search, and then σ decrease nonlinear and the particle focus on fine tune local search. This strategy can effectively maintain the diversity of the population; the particles can search more potential area.

3) *Opposition-based learning strategy*: Opposition-Based Learning (OBL) was proposed first by Tizhoosh (2005), and has been proven to be an effective strategy to enhance various optimisation algorithm performances (Ma et al., 2014). When evaluating a solution x to a given problem, simultaneously computing its opposite solution may find a candidate solution which is closer to the global optimum (Wang et al., 2011).

Opposite point: Let $X = (x_1, x_2, ..., x_D)$ be an D-dimensional point, where $x_1, x_2, ...x_D \in R$, $x_j \in [a_j, b_j]$, $j \in 1, 2, ..., D$, the opposite point $X^* = (x^*_1, x^*_2, ..., x^*_D)$ is defined as:

$$x^*_j = a_j + b_j - x_j \quad (6)$$

when stagnation is detected, opposition-based learning is applied to select randomly n particles from current population, the learning object for the i-th particle are the particles' worst position in its history and m particles in bad position select from initial population. The i-th particle updates its velocity according to

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (x_{id}(t) - w_{id}(t)) + c_2 r_2 (x_{id}(t) - w^0_{id}(t)) \quad (7)$$

where $w_{id}(t)$ is the particles' worst position in its history, $w^0_{id}(t)$ is the randomly selected particles with poor position in initial population.

To ensure that the worst initial particle can help the opposition-based learning individual escape from the local optimal area and distribution more extensive on the search area, the m particles should be with larger Euclidean distance. The pseudo code for selection the poor particles algorithm is shown in Algorithm 1.

Algorithm 1: Pseudo code for generation of poor particle algorithm

```
Begin
 Initial the swarm with N individuals and sort the
 individuals by fitness {X_1^0, X_2^0, ..., X_N^0},
   W_1^0 = X_1^0, count=1, W = W ∪ {W_count^0}
     for i=2 to N
         if count ≥ m ,break
         Else if
           ∀j(|| W_j^0 − X_i^0 ||> R), 1 ≤ j ≤ count
           count++, W_count^0 = X_i^0, W = W ∪ {W_count^0}
    End if
      End for
      While count<m
      if ∀j(|| ind − W_J^0 ||> R), 1 ≤ j ≤ count
      count++; W_count^0 = X_i^0, W = W ∪ {W_count^0}
      End if
      End while
End
```

During opposition-based learning, the maximum flight speed is dynamically adjusted. The purpose of opposition-based learning is to utilise the traction effect of the initial worst position of population and the worst position of individual in history. This strategy can help the particle search to be more effective in a wide range and to improve the success rate of finding the global optimum.

4) *Stagnation detection*: Fitness variance or average distance amongst points is usually used to detect stagnation,

$$\delta^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i - f_{avg}}{f}\right) \quad (8)$$

$$D(t) = \frac{1}{NL}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{D}\left(p_{ij} - p_j\right)^2} \quad (9)$$

where N is the population size, L is the diagonal length of the search space, f_i is the fitness of the i-th individual, f_{avg} is the mean fitness of population, p_{ij} is the value of j-th dimensional of the i-th particle, $\overline{p_j}$ is the mean value of j-th dimensional of all individual. $f = \max\left[1, \max|f_i - f_{avg}|\right]$ is the normalisation factor.

From equations (8) and (9), we can see that the fitness variance reflected the particle distribution with function value, and the average distance amongst points reflected the particle dispersion distribution from the Euclidean distance among individuals. When the swarm shows premature convergence or global convergence, the particles will gather in one or several specific locations within search space, when these specific particles are very close to each other, the fitness variance will be small while the average particle distance is large, therefore, the average-distance-amongst-points is not perfect to reflect the diversity of swarm.

The pseudo code of MSLPSO is shown in Algorithm 2.

Algorithm 2: Pseudo code of the MSLPSO algorithm

Begin
Random initialisation of particle $X^0 = \{X_1^0, X_2^0, ..., X_N^0\}$ and $V^0 = \{V_1^0, V_2^0, ..., V_N^0\}$
Calculate the fitness values of all particles, P_{g1} = the optimal particle, P_{g2} = the sub-optimal particle
construct the worst subpopulation according to Algorithm 1
While(stop condition is not satisfied)
update V^t and X^t according to (1) and (2)
Calculate the fitness values of all particles , update P_{g1} and P_{g2}
If (stagnation detection)
opposition-based learning strategy is applied according (6) and (7)
End if
$t=t+1$
End while
End

3 Experimental studies on MSLPSO

3.1 Test problems

Selected 12 benchmark problems proposed for CEC2005 special session on real-parameter optimisation were chosen for testing the MSLPSO performance. Brief descriptions of these benchmark problems are listed in Table 1. The acceptable tolerance of each function is also defined in Table 1. If the result is obtained within the acceptable tolerance of the global optimum, it is defined that the run is successful.

Table 1 Benchmark test functions

Function	*Search range*	*Global optimum*	*z*
$F_1(x) = \sum_{i=1}^{D} z_i^2 + f\_bias1$	$[-100,100]^D$	–450	$z=x-o$
$F_2(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2 + f\_bias2$	$[-100,100]^D$	–450	$z=x-o$
$F_3(x) = \sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} z_i^2 + f\_bias3$	$[-100,100]^D$	–450	$z=(x-o)*M$
$F_4(x) = \left(\sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2\right)*\left(1+0.4\,|N(0,1)|\right) + f\_bias4$	$[-100,100]^D$	–450	$z=x-o$
$F_5(x) = \max\{|\mathbf{A}_i x - \mathbf{B}_i|\} + f\_bias5$	$[-100,100]^D$	–310	
$F_6(x) = \sum_{i=1}^{D-1}\left(100\left(z_i^2 - z_{i+1}\right)^2\right) + f\_bias6$	$[-100,100]^D$	390	$z=x-o+1$
$F_7(x) = \sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias7$	$[0,600]^D$	–180	$z=(x-o)*M$

Table 1 Benchmark test functions (continued)

Function	*Search range*	*Global optimum*	*z*
$F_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{D} z_i}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{D}\cos(2\pi z_i)\right) + 20 + e + f\_bias8$	$[-32,32]^D$	–140	*z*=(*x*–*o*)**M*
$F_9(x) = \sum_{i=1}^{D}\left(z_i^2 - 10\cos(2\pi z_i) + 10\right) + f\_bias9$	$[-5,5]^D$	–330	*z*=*x*–*o*
$F_{10}(x) = \sum_{i=1}^{D}\left(z_i^2 - 10\cos(2\pi z_i) + 10\right) + f\_bias10$	$[-5,5]^D$	–330	*z*=(*x*–*o*)**M*
$F_{11}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\left(z_i + 0.5\right)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\left(z_i + 0.5\right)\right)\right] + f\_bias11$	$[-0.5,0.5]^D$	90	*z*=(*x*–*o*)**M*
$F_{12}(x) = \sum_{i=1}^{D}\left(\mathbf{A}_i - \mathbf{B}_i(x)\right)^2 + f\_bias12$	$[-\pi,\pi]^D$	–460	

3.2 Parameter settings for the involved PSO algorithms

The experiments are conducted to compare the proposed MSLPSO with state-of-art PSO algorithms such as conventional PSO (CPSO) (Shi and Eberhart, 1998), Comprehensive Learning PSO (CLPSO) and Orthogonal Learning PSO (OLPSO). Parameters set up for the involved PSO variants are as follow. CPSO: $c_1 = c_2 = 2.05$, $c_3 = 2$, w: 0.9 – 0.6, number of sub-swarm: 4, population size: 20. CLPSO: $c_1 = c_2 = 1$, w: 0.95 – 0.4, OLPSO, w: 0.9 – 0.4, G=5.

We perform the evaluation with 30 variables. For the purpose of reducing statistical errors, each algorithm was independently executed 25 times for each function with $D *$ 10,000 function evaluations (FES) as the termination criterion. The average and standard deviation (Std. Dev) of error obtained at the end of each run were computed in the comparison. The error was defined as $f(x) - f(x^*)$, where x is the global optimum of the benchmark function and x^* is the best solution found by the algorithm after $D *$ 10,000 function evaluations.

To evaluate the statistical significance of the performance differences between MSLPSO and its peers, a two-sided Wilcoxon rank sum test was conducted at 5% significance level, and the value of h was recorded. When the null hypothesis is rejected at the 5% significance level, we mark with '+' the cases where the MSLPSO exhibits superior performance to its peer and with '–' when it exhibits inferior performance. When the performance difference is not statistically significant, we mark with '≈'.

Note that all optimisers involved in the experiment were carried out on the same machine with 3.00 GHz AMD Athlon (tm) II X2 250 processor, and 3.25 GB internal memory with Windows XP3 operating system.

3.3 Experimental results and discussions

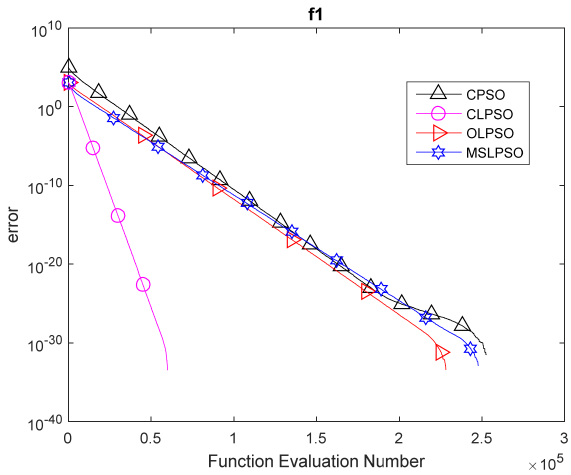
Table 2 depicts the results of the mean and standard deviation of the function error of the 25 runs for all the test functions.

The best results among those obtained by all algorithms are highlighted in bold. Figure 1 shows us the convergence graphs for 12 out of the 30-dimensional CEC 2005 benchmark functions.

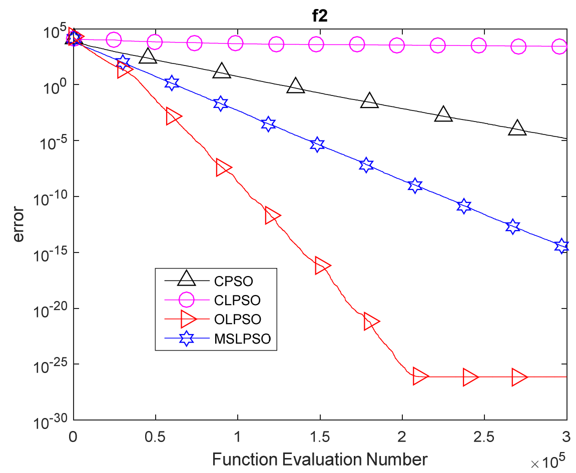
Table 2 Summary of the Wilcoxon rank sum test results

function		*CPSO mean(Std. Dev)*	*CLPSO mean(Std. Dev)*	*OLPSO mean(Std. Dev)*	*MSLPSO mean(Std. Dev)*
F_1		0.00E+00(0.00E+00)=	0.00E+00(0.00E+00)=	0.00E+00(0.00E+00)=	0.00E+00(0.00E+00)
F_2		4.61E–11(1.11E–10)+	5.35E–23(3.90E–23)+	1.05E–27(6.26E–28) –	1.15E–27(2.67E–27)
F_3		9.86E+02(4.35E+02)+	1.22E+04(6.14E+03)+	5.38E+05(2.06E+05)+	1.21E+01(3.47E+01)
F_4		9.40 E–12(2.35 E–12)-	6.65E+03(1.43E+03)+	3.38E+05(1.08E+06)+	2.77E–10(1.16 E–10)
F_5		3.89E+04(4.79 E+04)+	1.34E+03(5.86 E+02)+	3.28E+04(5.98 E+03)+	2.58 E+01(4.24 E+01)
F_6		5.23 E+03(4.43 E+03)+	4.78 E–01(1.32 E+00)=	4.78 E+04(4.78 E+03)+	1.59 E–01(7.97 E–0)
F_7		3.10E+01(4.37 E–02)=	2.09 E+00(1.90 E–01)=	2.01 E+00(1.32 E–01)=	1.03 E+00(2.31 E–01)
F_8		4.12E+01(4.35E+01)+	6.53 E+01(3.26 E+01)+	1.53 E–01(1.26 E-01)=	2.06 E–02(2.01 E–03)
F_9		7.53 E+1(1.23 E+01)+	2.53 E+01(4.83 E+00)=	6.75 E+01(1.06 E+01)+	2.22 E+01(5.16 E+00)
F_{10}		8.51 E+01(5.83 E+01)+	6.53 E+00(2.53 E+00) –	4.68 E+01(2.60 E+00)=	3.54 E+01(2.74 E+00)
F_{11}		6.74E+03(1.09E+04)+	4.07E+03(3.05E+03)+	2.15E+03(2.98E+03)+	1.19E-03(1.85E–03)
F_{12}		1.49 E+00(3.06 E–01)=	3.69 E+00(1.09 E+00)+	1.53 E+00(3.26 E–01)=	1.64 E+00(6.07 E–01)
	≈	3	4	5	
h	+	8	7	6	
	–	1	1	1	

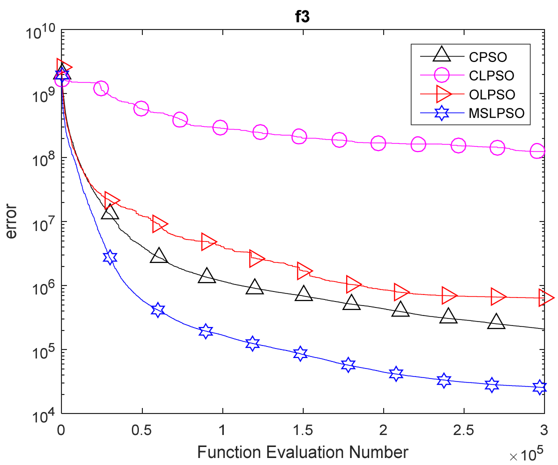
Figure 1 Convergence graph (median curves) for all PSO variants and their corresponding best performing on the benchmark functions



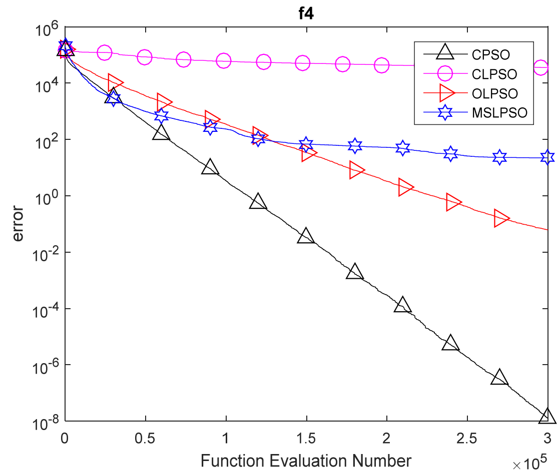
(a) *f*1



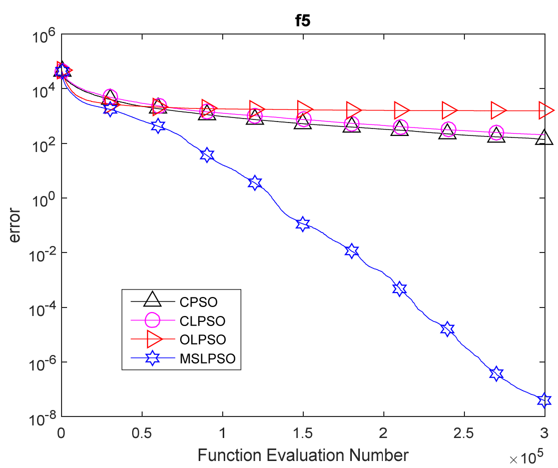
(b) *f*2



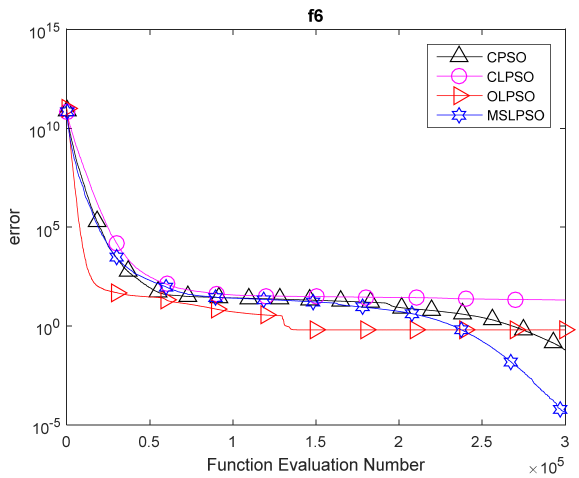
(c) *f*3



(d) *f*4

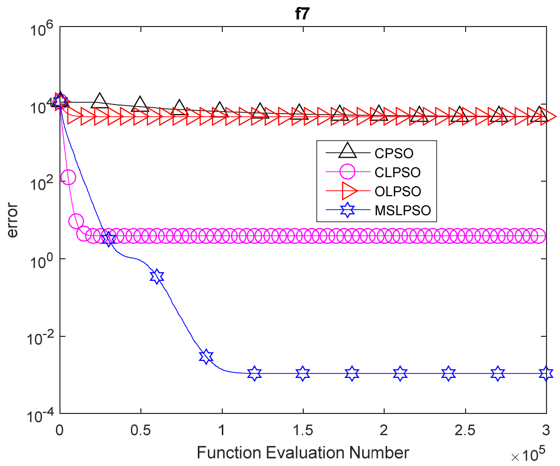


(e) *f*5

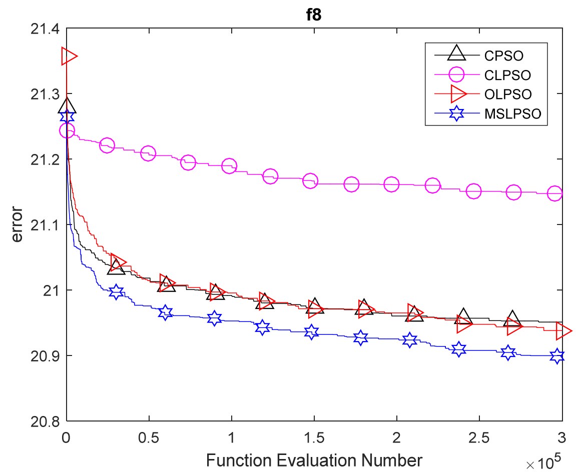


(f) *f*6

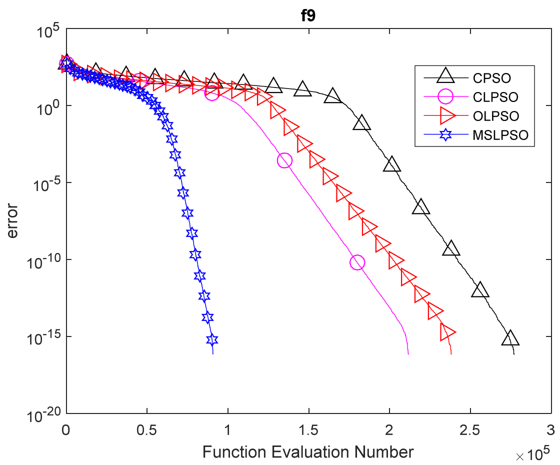
Figure 1 Convergence graph (median curves) for all PSO variants and their corresponding best performing on the benchmark functions (continued)



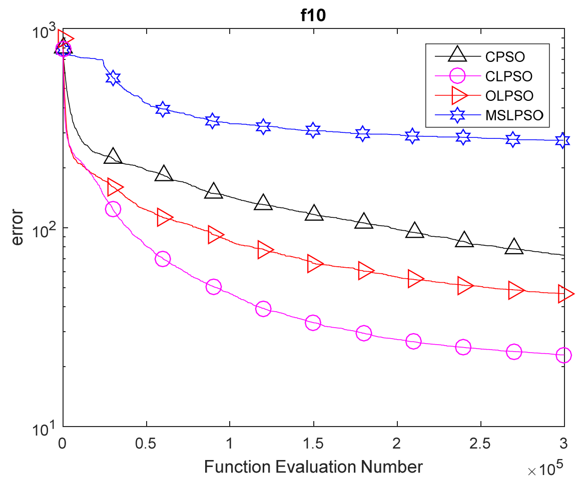
(g) *f*7



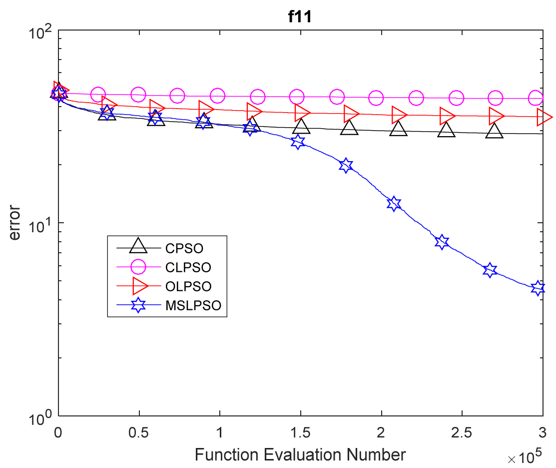
(h) *f*8



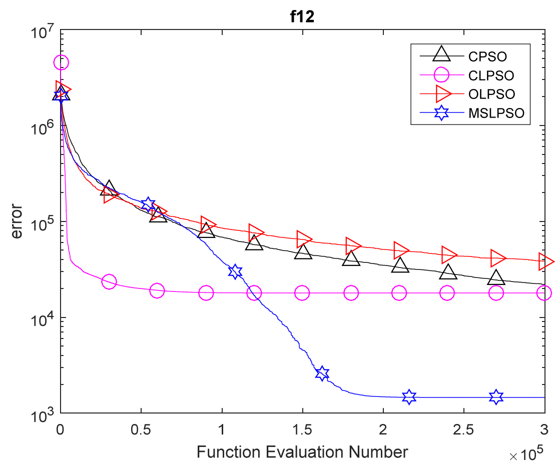
(i) *f*9



(j) *f*10



(k) *f*11



(l) *f*12

From Table 2, as well as the convergence curves in Figure 1, we observe that the MSLPSO dominates most of its peers in terms of the solution accuracy and convergence speed. It is ranked first among the four involved methods on solving eight functions (*f*3, *f*5, *f*6, *f*7, *f*8, *f*9, *f*11, *f*12). CLPSO performs better than MSLPSO on functions *f*1 and *f*10. OLPSO outperforms the MSLPSO on functions *f*2, CPSO performs better than other algorithms on functions *f*4. The reasons why MSLPSO has excellent performance in solution accuracy and convergence speed are: (1) opposition-based learning help to explore more high-quality search areas and improve the diversity of the population; (2) improved differential mutation helps to enhance the exploitation ability of the algorithm.

The reason why MSLPSO does not perform well on some functions can be explained by No Free Lunch Theorem. According to No Free Lunch Theorem, a general-purpose universal optimisation algorithm is theoretically impossible. Therefore, no strategy can be expected to outperform another on all types of optimisation problems.

As to Wilcoxon rank sum test, the numbers of benchmark tests where MSLPSO's performance is significantly better than ('+'), significantly worse than ('–'), and statistically equivalent ('≈') to the performance of its peers were recorded. The Wilcoxon rank sum test results are also listed in Table 2.

It can see from Table 2 that MSLPSO outperforms its peers in statistical sense, as the number of problems where MSLPSO performs significantly better than its peers is larger than the number of problems where MSLPSO is significantly worse than its peers. For example, Wilcoxon rank sum test results between the MSLPSO and CPSO are '+'=8, '–'=1, '≈'=3, that means the MSLPSO achieves significantly better results than, significantly worse result than, and statistical equivalent results to the CPSO on 8, 1, and 3 benchmark tests, respectively. When solving the functions with 30 dimensions, MSLPSO is significantly better than CPSO, CLPSO and OLPSO on 8, 7 and 6 test functions, respectively.

5 Conclusions

In this paper, a PSO variant named MSLPSO was proposed for unconstrained numerical optimisation. In the MSLPSO algorithm, improved differential mutation was introduced to update the particle velocity to expand the search scope and improve particle diversity. Elite disturbance was applied to the proposed method at the late stage of iteration to increase the diversity and to avoid trapping into local optimum. When stagnation is detected, opposition-based learning was integrated to select randomly inferior particles from current population to improve the diversity of the population and help the particles flight away from the local optimum. The results obtained from twelve benchmark functions demonstrate the superiority of the proposed algorithm to three other algorithms in terms of solution accuracy and convergence speed.

Based on the significantly encouraging results obtained from the experiments, it can be concluded that MSLPSO significantly improves the PSO's performance and gives better performance on most optimisation problems when compared with other PSO versions

Acknowledgements

This work is supported by the Scientific Research Funds of Hunan Provincial Education Department (No. 19A117, No. 18B385), NNSF of China (No. 51502087, No. 61841103) and the Nature Science Foundation of Hunan Province (No. 2019JJ50106).

References

Alireza, A.L.F.I. (2011) 'PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems', *Acta Automatica Sinica*, Vol. 37, No. 5, pp.541–549.

Aydilek, I.B. (2018) 'A hybrid firefly and particle swarm optimisation algorithm for computationally expensive numerical problems', *Applied Soft Computing*, Vol. 66, pp.232–249.

Cheng, R. and Jin, Y. (2015) 'A social learning particle swarm optimisation algorithm for scalable optimisation', *Information Sciences*, No. 291, pp.43–60.

Eberhart, R. and Kennedy, J.A. (1995) 'A new optimizer using particle swarm theory', *Proceeding of International Symposium on Micro machine and Human Science*, IEEE, Nagoya, Japan, pp.39–43.

Liang, J.J., Qin, A.K. and Suganthan, P.N. et al. (2006) 'Comprehensive learning particle swarm optimizer for global optimisation of multimodal functions', *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp.281–295.

Liu, Y., Mu, C. and Kou, W. et al.(2015) ' Modified particle swarm optimisation-based multilevel thresholding for image segmentation', *Soft Computing*, Vol. 19, No. 5, pp.1311–1327.

Ma, X., Liu, F. and Qi, Y. et al. (2014) 'MOEA/D with opposition-based learning for multiobjective optimisation problem', *Neurocomputing*, No. 146, pp.48–64.

Mendes, R., Kennedy, J. and Neves, J. (2004) 'The fully informed particle swarm: simpler, maybe better', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.204–210.

Qin, Q., Cheng, S. and Zhang, Q. et al. (2015) 'Multiple strategies based orthogonal design particle swarm optimizer for numerical optimisation', *Computers and Operations Research*, No. 60, pp.91–110.

Shi, Y. and Eberhart, R. (1998) 'A modified particle swarm optimizer', *Proceedings of the IEEE International Conference on Evolutionary Computation*, IEEE, pp.69–73.

Shi, Y. and Eberhart, R.C. (1999) 'Empirical study of particle swarm optimisation', *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE, pp.320–324.

Smairi, N., Siarry, P. and Ghedira K. (2016) 'A hybrid particle swarm approach based on Tribes and tabu search for multi-objective optimisation', *Optimisation Methods and Software*, Vol. 36, No. 1, pp.204–231.

Suganthan, P.N., Hansen, N. and Liang, J.J. et al.(2005) *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimisation*, Technical Report; Nanyang Technological University AND KanGAL Report #2005005, IIT Kanpur, India.

Tizhoosh, H.R. (2005) 'Opposition-based learning: a new scheme for machine intelligence', *Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation*, pp.695–701.

Van den Bergh, F. and Engelbrecht, A.P. (2004) 'A cooperative approach to particle swarm optimisation', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.225–239.

Wang, F., Zhang, H., Li, K., Lin, Z. et al. (2018) 'A hybrid particle swarm optimisation algorithm using adaptive learning strategy', *Information Sciences*, Vol. 436, pp.162–177.

Wang, H., Wang, W. and Wu, Z. (2013) 'Particle swarm optimisation with adaptive mutation for multimodal optimisation', *Applied Mathematics and Computation*, No. 221, pp.296–305.

Wang, H., Wu, Z. and Rahnamayan, S. et al.(2011) 'Enhancing particle swarm optimisation using generalized opposition-based learning', *Information Sciences*, Vol. 181, No. 20, pp.4699–4714.

William, A., Schmitt, K.R.B. and Venstrom, L. (2018) 'A parameter estimation method for stiff ordinary differential equations using particle swarm optimisation', *Journal of Wireless and Mobile Computing Neurocomputing*, Vol. 9, No. 5, pp.419–432.

Xia, X., Liu, J. and Hu, Z. (2014) 'An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space', *Applied Soft Computing*, No. 23, pp.76–90.

Xiao, G., Lin, H. and Guo, W. (2018) 'A hybrid training method of convolution neural networks using adaptive cooperative particle swarm optimiser', *International Journal of Wireless and Mobile Computing*, Vol. 16, No. 1, pp.18–26.

Xu, X., Tang, Y. and Li, J. et al. (2015) 'Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy', *Applied Soft Computing*, No. 29, pp.169–183.

Zhan, Z.H., Zhang, J. and Li, Y. et al. (2009) 'Adaptive particle swarm optimisation', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 39, No. 6, pp.1362–1381.

Zhao, X., Liu, Z. and Yang, X. (2014) 'A multi-swarm cooperative multistage perturbation guiding particle swarm optimizer', *Applied Soft Computing*, No. 22, pp.77–93.