24th International Symposium on Automation & Robotics in Construction (ISARC 2007)
Construction Automation Group, I.I.T. Madras

THE COMBINED EFFECT COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZER

Samrat L. Sabat
School of Physics
University of Hyderabad
Hyderabad, India 500046
slssp@uohyd.ernet.in

Layak Ali
School of Physics
University of Hyderabad
Hyderabad, India 500046
ph06pe02@uohyd.ernet.in

ABSTRACT

This paper introduces a novel and efficient optimization method, the Combined Effect Comprehensive Learning Particle Swarm Optimizer (CECLPSO) to handle the problems of premature and slow convergence with inferior solution prevailing in PSO and its variants. These weaknesses are resolved by introducing the combined effect of two consecutive global best particles contribution on the learning strategies of particles with the integration of Comprehensive Learning. This is in contrast to the original Comprehensive Learning PSO (CLPSO) technique, in which, the particles learning strategy is based on the knowledge of only one global best gbest. The performance of the CECLPSO is compared with basic PSO (BPSO) and CLPSO algorithms, on search efficiency, with the set of benchmark functions of dimension 50. The simulation result clearly indicates that the proposed CECLPSO algorithm prevents premature convergence and obtains better solution over basic PSO and CLCPSO in optimizing higher dimensional multimodal functions.

KEYWORDS

Particle swarm optimization, Learning strategy, CLPSO

1. INTRODUCTION

Evolutionary algorithms are stochastic optimization techniques, which learns and adapts from social behavior of species. To replicate and to use the techniques followed by the species, various researchers have developed natural computing algorithms for optimization problems. The first natural algorithm was Genetic algorithm (GA)[1]. GA efficiently handles for optimizing multi-modal and multi objective function. In spite of its efficiency in producing better solution, it suffers with heavy computational burden to solve complex real world problems. The Particle Swarm Optimization(PSO) is similar to GA, was introduced in 1995 by Kennedy and Eberhart [2-3]. PSO is a population based stochastic optimization technique. This algorithm has been an increasingly popular tool for optimizing realworld problems [4-8]. PSO simulate [4] the social behavior of swarm such as bird flocking, fish schooling to solve optimization problems. To understand PSO, social life of the species need to be analyzed closely. Imagine a group of birds searching food in a bounded area. Let their goal is to find maximum density food in the search space, where food is being scattered with varied density. The maximum density food location corresponds to the global solution in PSO world. No bird knows where the high density of food is, but they know how far the goal is from their present position. Without prior knowledge, all the birds start searching food randomly. Each can memorize two facts i.e it can remember its own history where it found maximum density of food and also it knows which bird has found the maximum density of food among the whole swarm. The flying trajectory of the birds will be altered with these historical facts which helps in converging towards the global solution. This scenario can be used to solve the optimization problems. In PSO, each single solution is a bird called particle in the search space. All the particles have fitness values, which are evaluated by the fitness function to be optimized. Every particle is associated with two memories i.e. self previous best experience (personal best or pbest) and best experience of the swarm (global best or gbest), these factors decides the trajectory of particle. In the past few years different research have been conducted to improve the performance of the original PSO algorithm [8-14] and are reported to give better solution.

Although different variants of original PSO give better solutions, PSO still suffers with premature convergence and inferior solutions for complex multimodal objective functions.This paper introduces a novel and efficient optimization technique, Combined Effect Comprehensive Particle Swarm Optimization (CECLPSO)to handle the problems of slow and premature convergence with improved solution. The weaknesses are reduced by introducing the combined effect of second consecutive global best particles on learning strategies with the integration of Comprehensive Learning algorithm. This is in contrast to Comprehensive Learning PSO (CLPSO) technique, in which, the particles learning strategies is based on only one global best.The performance of the CECLPSO is compared with the original PSO i.e. Basic PSO (BPSO) [1][2] and Comprehensive Learning PSO (CLPSO) [5][8], on different benchmark functions of higher dimensions. Experimental result shows that CECLPSO algorithm overcomes the premature convergence with better solutions.

The paper is organized as follows: Section 2 briefly describes the BPSO and CLPSO. Section 3 describesproposed CECLPSO algorithm. Section 4 presents the benchmark functions and experimental setup adopted for performance comparison. Section 5 describes the simulation results, and Section 6 concludes the paper.

2. PSO ALGORITHM VARIANTS

2.1. Basic PSO algorithm overview

PSO is essentially an evolutionary algorithm for solving optimization problems, which mimics the artificial life of the swarm of birds or school of fish [4]. Each particles position in a swarm represents the potential solution and is evaluated based on the fitness function. The value of fitness function infers the quality of solution. As the particle fly randomly in D-dimensional search space, the position and velocity of i^{th} particle is represented as $X_i = (x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,D})$ and $V_i = (v_{i,1}, v_{i,2}, v_{i,3}, \cdots, v_{i,D})$ respectively. With increased iteration, the swarm will move towards the global best position by keeping track of their personal best. In a D dimensional search space The $pbest$ of the i^{th} particle is represented as $pbest = (p_{i,1}, p_{i,2}, p_{i,3}, \cdots, p_{i,D})$ and the $gbest$ of the whole swarm is represented as $gbest = (g_1, g_2, g_3, \cdots, g_D)$. The PSO algorithm updates the velocity and position by the following equations .

$$\begin{aligned} V_{i,d}^{t+1} &= V_{i,d}^{t} + c_1 * rand_1 * (pbest_{i,d}^{t} - X_{i,d}^{t}) \\ &+ c_2 * rand_2 * (gbest_{d}^{t} - X_{i,d}^{t}) \quad (1) \\ X_{i,d}^{t+1} &= X_{i,d}^{t} + V_{i,d}^{t+1} \quad (2) \end{aligned}$$

Where c_1=2 and c_2=2 are the learning factors which determines the relative influence of cognitive and social component to update the position and velocity respectively. $rand_1$ and $rand_2$ are two random numbers in the range of [0,1]. $V_{i,d}^{t}$ and $X_{i,d}^{t}$ are the velocity and position of i^{th} particle in d^{th} dimension till t^{th} iteration respectively. The $gbest_d^t$ is the global best in d^{th} dimension till t^{th} iteration and $pbest_{i,d}^t$ is the personal best of i^{th}

particle in d^{th} dimension till t^{th} iteration.

2.2. CLPSO Algorithm overview

The CLPSO [5][8] was developed to overcome the problem of premature convergence of BPSO for complex multimodal functions. The learning strategy CLPSO is different from BPSO. Instead of learning from the two best factors i.e. $gbest$ and $pbest$ of the particle simultaneously, in CLPSO, the particles learn either from the $gbest$ of the swarm or particle's $pbest$, or the $pbest$ of other particle of different dimensions. In this, if D is the total number of dimension of optimizing problem then, $k-$dimensions will be randomly chosen to learn from the $gbest$, some of the $(D-k)$ dimensions from some randomly chosen particle's $pbest$ and the remaining dimensions learn from its own $pbest$[5][8].The CLPSO algorithm updates the velocity and position as in [5][8].

3. CECLPSO ALGORITHM

CECLPSO is proposed to overcome the premature convergence with improved solutions especially for multi-modal functions. CECLPSO discussed in three parts (a) Combined Effect of first and second consecutive global best on learning strategy, (b) capturing of the weak particles and then direct these in the direction of better particle, (c) integration (a) and (b) with Comprehensive Learning strategy.

3.1. Combined Effect

The combined effect concept is based on the fact of human life. The human beings often be suspicion about the decision made by single person and hence always take the guidance of the second person for the final decision. Thus the influence of more than one person's opinion. The same situation can be simulated in PSO for optimization. The searching decision will be efficient if second best particle's experience is also considered along with the first. The velocity and position update equations now becomes as

$$\begin{aligned} V_{i,d}^{t+1} &= w^t * V_{i,d}^t + rand_1 * (gbest1_d^t - X_{i,d}^t) \\ &+ rand_2 * (gbest2_d^t - X_{i,d}^t) \qquad (3) \\ X_{i,d}^{t+1} &= X_{i,d}^t + V_{i,d}^{t+1} \qquad (4) \end{aligned}$$

3.2. Identification of weak particles

For multi-modal objective functions some of the particles will be trapped in the deep local minima and may not be available for searching the good solutions. These trapped particles are called weak particles and are major source of poor solutions and premature convergence. CECLPSO identifies such particles which gives poor solutions, and then accelerates those particles in the direction of better particle of the swarm. Algorithm 1 shows the piece of pseudo code to identify and accelerate the trapped particles.

Algorithm 1 Identification of weak particles

$Select \leftarrow S_p$ Selection factor for trapped particles
$m \leftarrow S_p * NP$ Number of particles to selected ($S_p = 0.5$ assumed here for simulation)
$q \leftarrow NP - m$
$sortedfitness \leftarrow sort(f^{t+1})$ 'sort' Arranges the f^{t+1} of the particles in ascending order and stores in *sortedfitness*
for $k \leftarrow 1, m$ **do**
  for $l \leftarrow 1, NP$ **do**
    Δ=$sortedfitness(q+k) - f^{t+1}(l)$
    if $\Delta == 0$ **then**
      $X(l)$ =$rand * gbest$
    end if
  end for l
end for k
$i \leftarrow i + 1$

3.3. Integration with Comprehensive Learning

In comprehensive learning the particles learn from different exemplary at different dimension and time. The comprehensive learning concept is integrated with stated proposal with the refreshing rate of 10. The particle's velocity and position are updated by selecting either of the following velocity and position update equation.

$$V_{i,d}^{t+1} = w^t * V_{i,d}^t + rand_1 * (gbest1_d^t - X_{i,d}^t)$$

$$
\begin{aligned}
& + \quad rand_2 * (gbest2_d^t - X_{i,d}^t) && (5)\\
V_{i,d}^{t+1} &= w^t * V_{i,d}^t + rand_1 * (pbest_{f,d}^t - X_{i,d}^t) &&\\
& + \quad rand_2 * (gbest2_d^t - X_{i,d}^t) && (6)\\
V_{i,d}^{t+1} &= w^t * V_{i,d}^t + rand_1 * (pbest_{i,d}^t - X_{i,d}^t) &&\\
& + \quad rand_2 * (gbest2_d^t - X_{i,d}^t) && (7)\\
X_{i,d}^{t+1} &= X_{i,d}^t + V_{i,d}^{t+1} && (8)
\end{aligned}
$$

Algorithm 2 presents the detailed CECLPSO algorithm.

4. SIMULATION

4.1. Experimental setup

The simulations were conducted on Windows XP with MatLab and PIV 2.6GHz with 512MB of RAM. The population size of 25, for 1000 iteration. The algorithms were tested on set of ten benchmark functions with dimension 50. The results obtained are the average of 30 trials. The performance comparisons of CECLPSO are done with BPSO and CLPSO algorithms.

4.2. Benchmark functions

PSO algorithms presents the difficulty on multimodal functions with multiple minima, therefore we focus on well-known standard benchmark functions shown in Table 2. The benchmark function are numbered as f_1 to f_{10}. Among ten-mentioned benchmark functions; Dixon and Price f_2, Sphere f_8, Sum Square f_9 and Zakharov f_{10}functions are simple, strongly convex and can be considered as unimodal or multimodal. The Ackley function f_1 at a low resolution the landscape of this is unimodal; however, the second exponential term covers the landscape with many small peaks and valleys. Griewank function f_3 is also multimodal with multiple minima. It has a product term, introducing interdependency between the variables. The Levy function f_4 is highly multimodal with several local minima. The Powel function f_5 is also highly multimodal and has several minima but they are non-symmetrical and are randomly distributed. The characteristic of Rastrigin function f_6 is the existence of many suboptimal peaks whose values increase as the distance from the global optimum point increases. The Rosenbrock function f_7 is characterized by an extremely deep valley along the parabola that leads to the global minimum. Due to the nonlinearity of the valley, many algorithms converge slowly because they change the direction of the search repeatedly. The function has a long gully with very steep walls and almost flat bottom.

5. RESULTS

The simulations are conducted on the set of ten standard benchmark functions with dimension 50. The simulation results are categorized in two a) Convergence by the algorithms over iteration,shown in Fig.$1-10$ and b) the average of 30 trials shown in Table 1. The Table 1 gives the complete information about the best and worst value ever achieved by the algorithms over 30 trial, where each algorithm is run for 1000 iteration in a trial.The mean, median and the standard deviation of each algorithm are also presented. The best mean result and best standard deviations achieved by the algorithms are shown in bold. The Fig.$1-10$ shows the convergence graphs in terms of best fitness value achieved by each of the algorithms over 1000 iterations. From Table 1 it can be observed that the mean results of CECLPSO over 30 trials surpasses BPSO and CLPSO on the $f_1(x)$ to $f_{10}(x)$ except $f_4(x)$. CECLPSO especially dominates the$f_1(x), f_3(x), f_5(x), f_8(x)$ and $f_9(x)$ where the standard deviation is almost nil. The nil standard deviation show the robustness of the CECLPSO algorithm. The CLPSO outstands on $f_4(x)$ i.e. on Levy function and obtains comparable solution on $f_6(x)$ too. From the convergence graphs in Fig.$1-10$ it can be concluded that CECLPSO obtains good quality solution and avoids premature converges on $f_1(x)$ to $f_3(x)$ and $f_5(x)$ to $f_{10}(x)$. Fig 4 shows the outstanding performance by the CLPSO on $f_4(x)$.

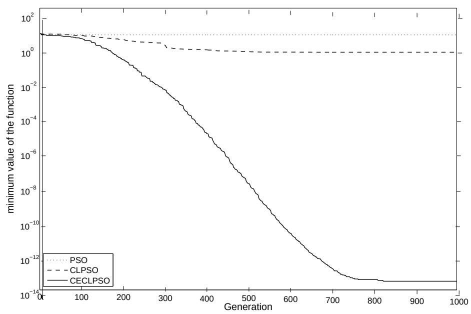

Figure 1. Convergence graph of function $f_1(x)$

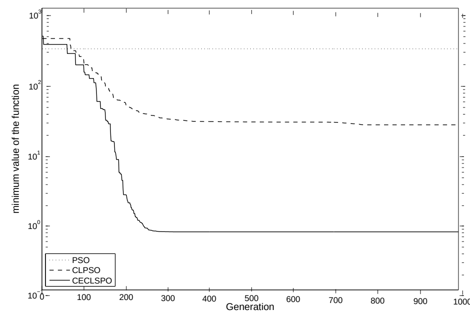

Figure 2. Convergence graph of function $f_2(x)$

Algorithm 2 AEPSO algorithm

Initialize

1: $Set \leftarrow X_{max}, X_{min}, D, NP$
2: $V_{max} \leftarrow 0.20 * (X_{max} - X_{min})$
3: $t \leftarrow 0, i \leftarrow 0$ ▷ *t* for iterations, *i* for particles
4: Randomly initialize position $X_i^0 \in D$ in R;
5: Randomly initialize velocity $v_i^0 \leq V_{max}$
6: Evaluate f_i^0
7: $pbest_i^0 \leftarrow f_i^0$,
8: $gbest^0 \leftarrow f_{best}^0$

Optimize

9: **while** $i \leq NP$ **do**
10: Update velocity using either of (5),(6) and (7) equations
11: Update position using equation (8).
12: Evaluate f_i^{t+1}
13:     **if** $f_i^{t+1} < pbest_i^t$ **then**
14:         $pbest_i^{t+1} \leftarrow f_i^{t+1}$
15:     **end if**
16: Find $gbest_d^{t+1}$
17:     **if** $gbest_d^{t+1} < gbest_d^t$ **then**
18:         $gbest_d^{t+1} \leftarrow gbest_d^t$
19:     **end if**
20:     $i \leftarrow i + 1$
21: **end while**
22: If stop criteria not met increment t.
23: Go to step 9.

Report results

Terminate

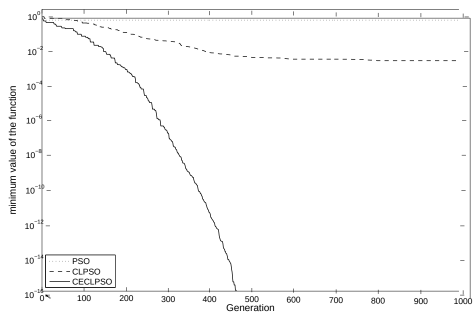

Figure 3. Convergence graph of function $f_3(x)$

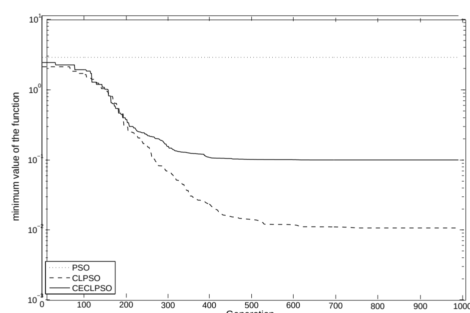

Figure 4. Convergence graph of function $f_4(x)$

6. CONCLUSION

In this paper, BPSO and CLPSO algorithms are compared against CECLPSO. Ten standard

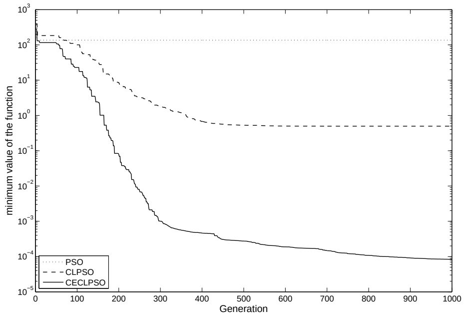

Figure 5. Convergence graph of function $f_5(x)$

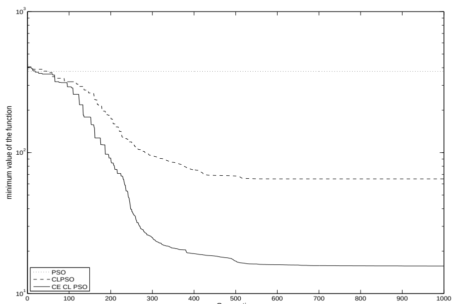

Figure 6. Convergence graph of function $f_6(x)$

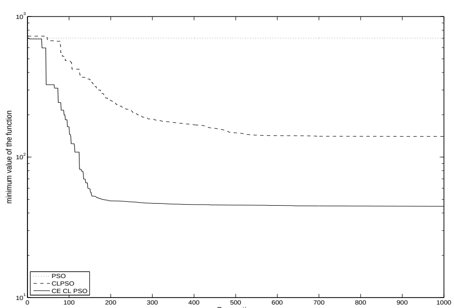

Figure 7. Convergence graph of function $f_7(x)$

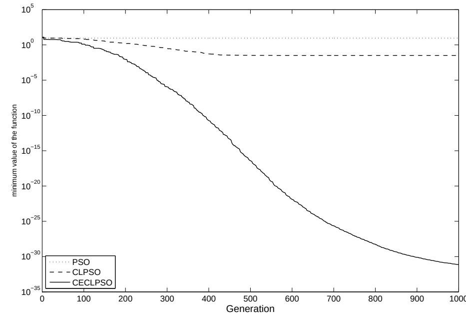

Figure 8. Convergence graph of function $f_8(x)$

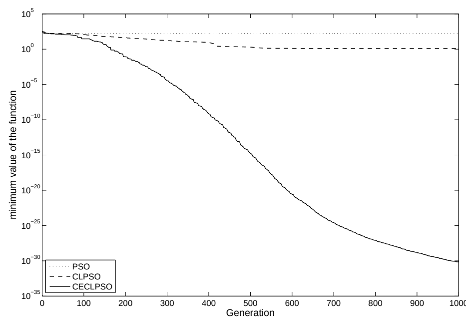

Figure 9. Convergence graph of function $f_9(x)$

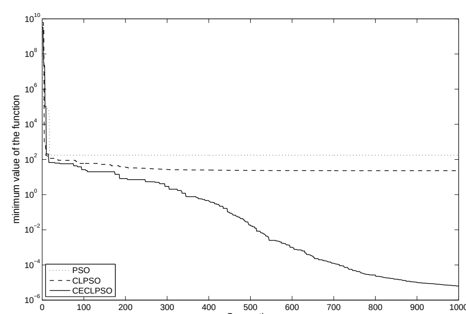

Figure 10. Convergence graph of function $f_{10}(x)$

benchmark functions of 50 dimensions are applied for optimization. Brief descriptions of each algorithm along with CECLPSO are presented. The comparative results of the simulation are also presented. From the simulation, it is found that the CECLPSO outperforms the other said PSO variants in higher dimensional problems. In addition, to obtain better quality of solution, CECLPSO also overcome the premature convergence for higher dimensions. The strength of the proposed scheme lies in finding the global solution of the objective functions whose minima are regularly distributed. The proposed approach can not be taken as granted for all the types of the objective functions. Further work need to be done to overcome this drawback. This algorithm can be applied for delay optimization, molecular geometry optimization, placement and routing optimization in Very Large Scale Integrated circuit design.

ACKNOWLEDGMENT

The authors are thankful to the ACRHEM for providing financial support to carry out research work.

REFERENCES

1. K F Man, K S Tang, S Kwong, " Genetic algorithms: concepts and applications in engineering design," *in Proc. of IEEE Transactions on, Industrial Electronics* , Vol.43, 1996, pp. 519-534.
2. J. Kennedy, R C Eberhart, " Particle swarm optimization," *in Proc. of IEEE Conference on Neural Networks* , Piscataway, NJ, 1995, pp. 1942-1948.
3. J. Kennedy, R C Eberhart, " A new optimizer using particle swarm theory," *in Proc. of Sixth Int. Symposium on Micro machine and Human Science,* Nagoya, Japan, 1995, pp. 39-43.
4. J. Robinson and Yahya Rahmath Samii, " Particle Swarm Optimization in Electromagnetic," *IEEE Transactions on Antenna and Propagation,* Vol.52(2) , 2004, pp.397-400.
5. S. Bhasker, A. Alphones, P.N.Suganthan and J.J. liang, " Design of Yagi-Uda antenna using comprehensive learning particle swarm optimization," *IEEE Transactions on Microw. Antenna Propapg,* Vol.152(5), 2005,pp.340-345.
6. Jaco, Byung, Jeffrey, Raphael, Alan, Benjamin, " Evaluation of a Particle Swarm Algorithm For Biomechanical Optimization," *Journal of Biomechanical Engineering,* Vol. 127, 2005, pp. 465-473.
7. Xiao-Feng Xie, Wen-Jun Zhang, De-Chun Bi, " Optimizing Semiconductor Devices by Self-organizing Particle Swarm," *in Proc. of Congress on Evolutionary Computation,* Oregon, USA, 2004, pp. 2017-2022.
8. V.L. Huang, P.N. Suganthan, J.J. Liang, " Comprehensive Learning Particle Swarm Optimizer for Solving Multiobjective Optimization Problems," *in Proc. of Int. Journal of Intelligent Systems,* Vol. 21, 2006, pp. 209-211.
9. T. Peram, K. Veeramachaneni, and C.K. Mohan, " Fitness-Distance Ratio Based Particle Swarm Optimization," *in Proc. of Int. IEEE Swarm Intelligence System,* Indianapolis, Indiana, USA, 2003, pp.174-181.
10. P.N. Suganthan, " Particle Swarm Optimizer with neighborhood operator," *in Proc. of IEEE 1999,* pp. 1958-1959.
11. Shi,Y., R C Eberhart, " A modified particle swarm optimizer," *in Proc. of IEEE World Congress on Computational Intelligence,* Piscataway, NJ, IEEE Press (1998) 69-73.
12. Kennedy, J, " Stereotyping: Improving particle swarm performance with cluster analysis," *in Proc. of IEEE Congress on Evolutionary Computation,* Piscataway, NJ, IEEE Press (2000) 1507-1512.
13. Fan, H, " A modification to particle swarm optimization algorithm," *in Proc. of Engineering Computations,* 19(8) (2002) 970-989.
14. Mendes, R., Kennedy, J., Neves, J, " The fully informed particle swarm: Simpler maybe better," *in Proc. of IEEE Transactions on Evolutionary Computation,* 8(3) (2004) 204-210.

Table 1
Summary of results for function with dimension **50**

Functions	Algorithms	Best	Worst	Mean	Median	STD
f_1	PSO	2.85	3.38	3.19	3.20	1.17e-01
	CLPSO	7.80e-02	1.59	5.39e-01	5.06e-01	4.13e-01
	CECLPSO	2.66e-15	2.66e-15	**2.66e-15**	2.66e-15	**0**
f_2	PSO	2.12e+02	5.14e+2	3.40e+02	3.36e+02	6.57e+01
	CLPSO	5.47	2.05e+01	1.24e+01	1.21e+01	4.07
	CECLPSO	6.66e-01	6.66e-01	**6.66e-01**	6.66e-01	**2.82e-06**
f_3	PSO	1.72e-01	3.17e-01	2.41e-01	2.37e-01	3.738e-02
	CLPSO	4.86e-04	1.09e-02	2.76e-03	2.49e-03	2.19e-03
	CECLPSO	0	0	**0**	0	**0**
f_4	PSO	1.88	2.59	2.31	2.34	1.63e-01
	CLPSO	5.19e-03	4.37e-01	**6.86e-02**	3.24e-02	**8.69e-02**
	CECLPSO	2.78e-01	6.76e-01	4.61e-01	4.49e-01	1.17e-01
f_5	PSO	1.01e+02	3.18e+02	2.01e+02	1.88e+02	5.01e+01
	CLPSO	7.55e-02	4.07	9.96e-01	7.39e-01	8.35e-01
	CECLPSO	8.87e-45	1.01e-31	**3.36e-33**	2.24e-43	**1.84e-32**
f_6	PSO	3.47e+02	4.15e+02	3.84e+02	3.85e+02	1.57e+01
	CLPSO	2.73e+01	9.86e+01	5.91e+01	5.76e+01	**1.576e+01**
	CECLPSO	0	1.62e+02	**5.21e+01**	5.43e+01	5.14e+01
f_7	PSO	5.19e+02	7.93e+02	7.03e+02	7.07e+02	6.89e+01
	CLPSO	5.77e+01	2.03e+02	1.49e+02	1.62e+02	4.44e+01
	CECLPSO	4.75e+01	4.87e+01	**4.82e+01**	4.83e+01	**3.46e-01**
f_8	PSO	6.42	1.07e+01	8.34	8.36	1.05
	CLPSO	1.08e-02	3.18e-01	7.79e-02	5.16e-02	7.09e-02
	CECLPSO	8.42e-49	8.66e-47	**1.15e-47**	4.17e-48	**1.79e-47**
f_9	PSO	1.38e+02	2.35e+02	1.87e+02	1.86e+02	2.17e+01
	CLPSO	7.90e-02	5.73	1.30	9.74e-01	1.17
	CECLPSO	8.63e-48	1.00e-44	**8.67e-46**	3.56e-46	**1.85e-45**
f_{10}	PSO	1.18e+01	4.06e+02	1.44e+02	1.08e+02	1.09e+02
	CLPSO	5.31	4.46e+01	1.47e+01	1.19e+01	9.28
	CECLPSO	1.30e-05	1.06e-01	**1.76e-02**	4.17e-03	**2.78e-02**

Table 2
Benchmark Functions Definition

Name	Definition	Range
Ackley	$f_1(x)$=20 $+ e^{-\frac{1}{5}\sqrt{\frac{1}{n}\sum_i (x_i)^2}} - e^{-\frac{1}{n}\sum_i^n cos(2\pi x_i)}$	[32.768,-32.768]
Dixon Price	$f_2(x) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - xi - 1)^2$	[-10, 10]
Griewank	$f_3(x)$=$\sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^N cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-100, 100]
Levy	$f_4(x)$=$sin^2(\pi y_1) + \sum_{i=1}^{n-1}[(y_i - 1)^2$ $(1 + 10sin^2(\pi y_i + 1))]$ $+(y_n - 1)^2\left(1 + 10sin^2(2\pi y_n)\right)$, where, $y_i = 1 + \frac{x_i - 1}{4}, i = 1, 2, \cdots n$	[-10, 10]
Powel	$f_5(x)$=$\sum_{i=1}^{n/4}\left(x_{4i-3} + 10x_{4i-2}\right)^2 + 5(x_{4i-1} - x_{4i})^2$ $+(x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	[-4 5]
Rastrigin	$f_6(x)$=$\sum_{i=1}^D\left(x_i^2 - 10cos(2\pi x_i) + 10\right)$	[-5.12 5.12]
Rosenbrock	$f_7(x)$=$\sum_{i=1}^{D-1}\left[100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right]$	[-100 100]
Sum Square	$f_8(x)$=$\sum_{i=1}^D ix_i^2$	[-10, 10]
Zakharov	$f_9(x)$=$\sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$	[-5, 10]