Improved PSO Performance using LSTM based Inertia Weight Estimation

Y.V.R.Naga Pawan
Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Green Fields, Vaddeswaram, Guntur District
Andhra Pradesh. India

Kolla Bhanu Prakash
Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Green Fields, Vaddeswaram, Guntur District
Andhra Pradesh. India

Abstract—Particle Swarm Optimization (PSO) is first introduced in the year 1995. It is mostly an applied population-based meta-heuristic optimization algorithm. PSO is diversely used in the areas of sciences, engineering, technology, medicine, and humanities. Particle Swarm Optimization (PSO) is improved its performance by tuning the inertia weight, topology, velocity clamping. Researchers proposed different Inertia Weight based PSO (IWPSO). Every Inertia Weight based PSO in excelling the existing PSOs. A Long Short Term Memory (LSTM) predicting inertia weight based PSO (LSTMIWPSO) is proposed and its performance is compared with constant, random, and linearly decreasing Inertia Weight PSO. Tests are conducted on swarm sizes 50, 75, and 100 with dimensions 10, 15, and 25. The experimental results show that LSTM based IWPSO supersedes the CIWPSO, RIWPSO, and LDIWPSO.

Keywords—Particle swarm optimization; inertia weight; long short term memory; benchmark functions; convergence

I. INTRODUCTION

Kennedy and Eberhart [1] [2] developed a stochastic population-based optimization algorithm based on the social-behaviour metaphor of a flock of birds or a swarm of bees searching for food. It solves global optimization numerical problems. PSO is applied in every discipline of Science, Engineering, and Technology [3-8]. It is widely applied as an optimization technique in areas like communications, electronics, electrical, manufacturing, grids, cloud computing, algorithms, numerical optimization, etc. [28-40]. PSO can be extended to non-differentiable, non-linear, large search space issues, and provides better performance with decent quality [9].

Since 1995, each year, new PSO variants have been created based on initialization parameters, constriction factor, mutation operator, inertia weight, topologies, parallel processing, fuzzy logic, neural networks, ensemble, etc. The new variants mostly supersede established PSO variants. A comprehensive review of PSO variants is discussed in [10] [11].

Many researchers focused their attention on computing inertia weight for faster convergence of the swarm. Different Inertia Weight Particle Swarm Optimizations (IWPSO) are discussed in [12]. It is observed that every inertia weight computing strategy supersedes the other.

In this work, a new inertia weight computing strategy is proposed. It uses a trained Long Short Term Memory (LSTM) to predict the inertia weight in every iteration, till stopping criteria is met. The predicted IW is used for computation of fitness function. Its performance is compared with Constant Inertia Weight PSO (CIWPSO) [13], Random Inertia Weight PSO (RIWPSO) [14], and Linear Decreasing Inertia Weight PSO (LDIWPSO) [15] using benchmark functions [12].

The remainder of the paper is organized as follows: Particle Swarm Optimization (PSO) and Inertia Weight based PSO is summarized in Section II, the Recurrent Neural Network, LSTM and LSTMIWPSO is briefed in Section III, Experimental Results are discussed in Section IV, and in Section V Conclusion and Future Work is briefed.

II. PARTICLE SWARM OPTIMIZATION

The formulation of PSO [16] [17] [18] is done based on the objective function given in equation (1). The objective function measures the closeness of the corresponding solution to the optimum.

\[ f(x) : \mathbb{R}^d \rightarrow \mathbb{R} \]  (1)

where \( d \) is the number of dimensions of search space, \( S \) is a subset of \( \mathbb{R}^d \), shown in equation (2) and defined by equation (3). The global optimization problem is shown in equation (4) and equation (5).

\[ S \subseteq \mathbb{R}^D \]  \hspace{1cm} (2)

\[ S = \{ px_1 | px_{min} \leq px_i \leq px_{max} \} \]  \hspace{1cm} (3)

\[ \min_{x \in S} f(x) \]  \hspace{1cm} (4)

\[ \forall p y_i \in S : f(p x_i) \leq f(p y_i) \]  \hspace{1cm} (5)

In the Basic PSO (BPSO), a Swarm, \( SW \), consists of \( n \) particles represented as \( SW = \{ P_1, P_2, P_3, \ldots, P_n \} \). Each Particle \( P_i \) has a position in the search space represented by \( PX_i = \{ px_{i1}, px_{i2}, px_{i3}, \ldots, px_{iD} \} \) where \( D \) is D-dimensional search space. In the search space, each particle \( P_i \) moves with a velocity \( V_i \), represented as \( PV_i = \{ pv_{i1}, pv_{i2}, pv_{i3}, \ldots, pv_{iD} \} \). Each particle, \( P_i \) maintains its best position, \( Pb_i \), represented as \( Pb_i = \{ pb_{i1}, pb_{i2}, pb_{i3}, \ldots, pb_{iD} \} \). Among the population of all particles, the best particle is determined and represented as
(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 11, 2020

$p_g = \{p_{g1}, p_{g2}, p_{g3}, \ldots, p_{gD}\}$. The basic equations with the functioning of BPSO are given by (6) and (7).

\[ pv_{id} = pv_{id} + c_1 \times \text{random}() \times (pb_i - px_{id}) + c_2 \times \text{Random}() \times (pg_i - px_{id}) \]  
\[ px_{id} = px_{id} + pv_{id} \]  
(6)  
(7)

where $c_1$ and $c_2$ are two positive acceleration coefficients, random() and Random() are two random functions in the [0,1]. $pv_{i}$ is then clamped to a maximum velocity $pv_{\text{max}}$, the parameter given by the user. The first part of the (6) represents the previous velocity, the second part is the cognition part of the particle, and the third part represents the cooperation among the particles [1][17][19].

As particles tend to explore the search space hugely, the velocities of the particles are limited to the constant $pv_{\text{max}}$ [16]. The particle velocity is adjusted using:

\[ pv_{id} = \begin{cases} pv_{id} \text{ if } -pv_{\text{max}} \leq pv_{id} \leq pv_{\text{max}} \\ pv_{\text{max}} \text{ if } pv_{id} > pv_{\text{max}} \\ -pv_{\text{max}} \text{ if } pv_{id} < -pv_{\text{max}} \end{cases} \]  
(8)

The velocity $pv_{\text{max}}$ is typically chosen as a fraction of the search space dimension shown as (4) [20][21], where $\delta$ is the velocity clamping factor.

\[ pv_{\text{max}} = \delta \times (px_{\text{max}} - px_{\text{min}}) \text{ where } \delta \in (0,1) \]  
(9)

As the search space, $S$, is bounded by the interval $[px_{\text{min}}, px_{\text{max}}]$, the velocity clamping [22] of the particle is in the interval $[-pv_{\text{max}}, pv_{\text{max}}]$. The interval $[-pv_{\text{max}}, pv_{\text{max}}]$ is shown through equation (17) – equation (22).

### A. Inertia Weight based PSO

Shi and Russell Eberhart [13], developed inertia weight based PSO (IWPSO). In IWPSO, exploration and exploitation of swarm particles are controlled. The equation (6) with inertia weight is given by equation (10).

\[ pv_{id} = \omega \times pv_{id} + c_1 \times \text{random}() \times (pb_i - px_{id}) + c_2 \times \text{Random}() \times (pg_i - px_{id}) \]  
(10)

### III. RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNN) are time-dynamic discrete systems dealing with input vector sequences [23][24]. RNNs traditionally propagate information forward in time, forming predictions using only past and present inputs. The basic Recurrent Neural network is shown in Fig. 1. The traditional RNN, for each time step $t$, the output is computed using equation (11), and the activation function $a^{<t>}$ is computed using equation (12).

\[ y^{<t>} = h(W_{ya} a^{<t-1>} + b_y) \]  
(11)

\[ a^{<t>} = g(W_{aa} a^{<t-1>} + W_{ax} x^{<t-1>} + b_a) \]  
(12)

where $t$ represents time, $y^{<t>}$ is the predicted value, $W_{ya}, W_{ax}, b_y$, and $b_a$ are the coefficients, and $h$ and $g$ are the activation functions. Generally, activation functions are given in equations (13), (14), and (15).

\[ \text{Sigmoid function, } g(a) = \frac{1}{1+e^{-a}} \]  
(13)

\[ \tanh, g(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \]  
(14)

\[ \text{RELU, } g(a) = \max(0,a) \]  
(15)

RNN is observed with vanishing [25] and exploding gradient [26] phenomenon. It is due to multiplicative gradient and resulting in its inability to catch dependencies that can be exponentially decreasing/increasing with respect to the number of layers.

In RNN, the loss function, $L$, for all time steps is defined based on the loss obtained at every time step.

\[ \text{Loss Function, } L(y, \hat{y}) = \sum_{t=1}^{T} L(\hat{y}^{<t>}, y^{<t>}) \]  
(16)

where $\hat{y}^{<t>}, y^{<t>}$ are predicted and expected outputs.

### A. Long Short Term Memory

Long Short Term Memory is special kind of RNN architecture capable in learning long term dependencies. Hochreiter and Schmidhuber [27] introduced the efficient and effective, gradient based the Long Short Term Memory (LSTM). Fig. 2 depicts the dependencies of the memory cell of an LSTM depicting dependencies. In order to deal with vanishing gradient problem, The LSTM has the power to delete or add information to a cell state that is carefully controlled by mechanisms called gates [27]. LSTM uses three gates called update gate ($\Gamma_u$), forget gate ($\Gamma_f$) and output gate ($\Gamma_o$). The computation of $\hat{c}^{<t>}, c^{<t>}, a^{<t>}, \Gamma_u, \Gamma_f, \Gamma_o$ are shown through equation (17) – equation (22).

\[ \hat{c}^{<t>} = \tanh(w_c [a^{<t-1>}, x^{<t>}] + b_c) > \text{Fункциония} \]  
(17)

\[ \Gamma_u = \sigma(w_u [a^{<t-1>}, x^{<t>}] + b_u) \]  
(18)

\[ \Gamma_f = \sigma(w_f [a^{<t-1>}, x^{<t>}] + b_f) \]  
(19)

\[ \Gamma_o = \sigma(w_o [a^{<t-1>}, x^{<t>}] + b_o) \]  
(20)

\[ c^{<t>} = \Gamma_u \times \hat{c}^{<t>} + \Gamma_f \times c^{<t-1>} \]  
(21)

\[ a^{<t>} = \Gamma_o \times \tanh c^{<t>} \]  
(22)

Let $\hat{y}^{<t>}$ be the predicted output at each time step and $y^{<t>}$ be the actual output at each time step. Then the error at each time step is given by:

\[ E^{<t>} = -y^{<t>} \log(\hat{y}^{<t>}) \]  
(23)

\[ E_{\text{total}} = \sum_{t} E^{<t>} \]  
(24)

\[ E_{\text{total}} = \sum_{t} -y^{<t>} \log(\hat{y}^{<t>}) \]  
(25)

The value of $\frac{\partial E}{\partial w}$ can be calculated as the summation of the gradients at each step

\[ \frac{\partial E}{\partial w} = \frac{\partial E^{<t>}}{\partial \hat{y}^{<t>}} \frac{\partial \hat{y}^{<t>}}{\partial a^{<t>}} \frac{\partial a^{<t>}}{\partial c^{<t>}} \frac{\partial c^{<t>} \partial c^{<t-1>} \ldots \partial c^{<0>}}{\partial \hat{c}^{<t>}} \]  
(26)
Thus the total error gradient is given by equations (27) and (28):

$$\frac{\partial E}{\partial W} = \sum_{t} \frac{\partial E^{<t>}}{\partial W}$$ (27)

$$\frac{\partial E}{\partial W} = \sum_{t} \frac{\partial E^{<t>}}{\partial y^{<t>}} \frac{\partial y^{<t>}}{\partial a^{<t>}} \frac{\partial a^{<t>}}{\partial c^{<t>}} \frac{\partial c^{<t-1>}}{\partial c^{<t>}} \ldots \frac{\partial c^{<0>}}{\partial W}$$ (28)

It is to note the gradient equation involves a chain of $\frac{\partial c^{<t>}}{\partial W}$ for an LSTM Back-Propagation while the gradient equation involves a chain of $\frac{\partial a^{<t>}}{\partial W}$ for a basic Recurrent Neural Network.

B. LSTM Inertia Weight based PSO

In LSTMIWPSO, the new inertia weight is computed using LSTM. Initially, LSTM is trained with different inertia weights from 0.05 to 1.00. In every iteration, a new IW is predicted using trained LSTM. The predicted IW is used to move the swarm using equations (10) and (7). The process is terminated when the stopping criterion is reached. The pseudocode for LSTMIWPSO is shown in Fig. 3.
The pseudocode for LSTMIWPSO is given below:

**Step 1:**
- **Initialization**
  - For each particle, \( P_i \), in the population
  - Initialize \( p_{xi} \) with uniform distribution
  - Initialize \( p_{vi} \) randomly.
  - Build and Train LSTM network for Inertia Weight Prediction.
  - Predict the new Inertia Weight.
  - Evaluate the objective function of \( p_{xi} \) and assigned the value to \( \text{fitness}[i] \).
  - Initialize \( p_{best}[i] \) with a copy of \( p_{xi} \).
  - Initialize \( p_{best\_fitness}[i] \) with a copy of \( \text{fitness}[i] \).
  - Initialize \( p_{gbest} \) with index of the particle with the least fitness.

**Step 2:**
- Repeat until stopping criterion is reached
  - For each particle, \( P_i \):
    - Update \( p_{vi} \) and \( p_{xi} \) according to the equations (10) and (7)
    - Evaluate \( \text{fitness}[i] \)
    - If \( \text{fitness}[i] < p_{best\_fitness}[i] \), then
      - \( p_{best}[i] = p_{xi} \)
      - \( p_{best\_fitness}[i] = \text{fitness}[i] \)
    - Update \( p_{gbest} \) by the particle with current least fitness among the population
    - Predict the new Inertia Weight using trained LSTM

Fig. 3. Pseudocode of LSTMIWPSO.

**IV. EXPERIMENTAL RESULTS**

Experiments are conducted with different Inertia Weight based PSOs namely, CIWPSO, RIWPSO, LDIWPSO, and LSTMIWPSO over different optimization test problems tabulated in Table I.

Swarm sizes of 50, 75 and 100 particles of different dimensions, 10, 15 and 25, are considered for experiments. A total of 15 simulations are performed to reduce the occurrence of randomness. Along with LSTMIWPSO, LDIWPSO, RIWPSO and CIWPSO are implemented. The results are collected in terms of the best error, mean error, variance, standard deviation, mean square error, root mean square error, mean iteration and mean time taken (in seconds) to evaluate the performance of LSTMIWPSO with CIWPSO, RIWPSO and LDIWPSO.

From Table II and Fig. 4, the performance of LSTMIWPSO, for benchmark functions \( f_1, f_3, f_4, \) and \( f_5 \) as fitness functions, swarm size with the dimension 10, the best error is nearer to CIWPSO, RIWPSO, and LDIWPSO. The best error is moderately higher, in the case of dimensions 15 and 25. For \( f_2 \) function, the best error for LSTMIWPSO is the same as CIWPSO, RIWPSO, and LDIWPSO.

For swarm sizes 50, 75, and 100 with dimensions 10, 15, and 25 and \( f_1-f_5 \) as fitness functions, the mean error is computed using the CIWPSO, RIWPSO, and LDIWPSO, and LSTMIWPSO. The processed results are collected and tabulated in Table III and graphically shown in Fig. 5. The mean error, except for swarm size 100 and dimension 10, when compared to CIWPSO, RIWPSO and LDIWPSO, for LSTMIWPSO, is smaller.

The variance and standard deviation are computed to access the performance of CIWPSO, RIWPSO, LDIWPSO and LSTMIWPSO. The computed results are tabled in Table IV and V. The same are shown graphically in Fig. 6 and Fig. 7. From Table IV, Table V, Fig. 6, and Fig. 7, it is evident that the performance of LSTMIWPSO in terms variance and standard deviation is flair with swarm sizes 50, 75, and 100, with dimensions 10, 15 and 25 on the benchmark functions \( f_1-f_5 \).

To access the CIWPSO, RIWPSO, LDIWPSO and LSTMIWPSO performance, the MSE and RMSE are computed. Tables VI and VII show the computed results. The same is seen in Fig. 8 and Fig. 9 graphically. It is evident from Table VI, Table VII, Fig. 8, and Fig. 9 that LSTMIWPSO's output in terms of MSE and RMSE is substantially better for swarm sizes 50, 75, and 100, and for benchmark functions \( f_1-f_5 \), with dimensions 10, 15 and 25, except for the swarm size 100 and dimension 10.

From Table VIII and Fig. 10, the meantime for LSTMIWPSO is transcending for the swarm sizes 75 and100 with dimension 10. In other scenarios, it is non-paying when compared with other methods for the benchmarks considered.

From Table IX and Fig. 11, the mean iterations for LSTMIWPSO are decent when compared with CIWPSO, RIWPSO, and LDIWPSO, with Swarm size 100 and dimension 10. Similarly, LSTMIWPSO has achieved adequate performance with \( f_2, f_3, \) and \( f_4 \) benchmark functions.

LSTMIWPSO delivered adequate results over CIWPSO, RIWPSO, and LDIWPSO from the perspective of mean error, variance & standard deviation and, MSE & RMSE. It is good in limited scenarios in terms of best error, Mean Time and Mean Iterations.
### TABLE I. BENCHMARK FUNCTIONS (BMF)

| Benchmark Function name | Properties | Benchmark Function | Search Space | Best fitness value at |
|-------------------------|------------|--------------------|--------------|-----------------------|
| Ackley (f1)            | n-dimensional, continuous, multimodal, non-convex, differentiable | $-20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{d=1}^{n} \text{pos}_d^2} - \exp(-\frac{1}{n} \sum_{d=1}^{n} \cos(2\pi \text{pos}_d)))$ | $[-32, +32]$ | $f(0) = 0$ |
| Alpine (f2)            | n-dimensional, non-separable, multimodal, non-convex, differentiable | $\sum_{d=1}^{n} |\text{pos}_d - \sin(\text{pos}_d)| + 0.1 \text{pos}_d|$ | $[0, 10]$ | $f(0) = 0$ |
| Rastrigin (f3)         | n-dimensional, continuous, differentiable, separable, multimodal, convex | $10.n + \sum_{d=1}^{n} (\text{pos}_d^2 - 10 \cos(2\pi \text{pos}_d))$ | $[-5.12, +5.12]$ | $f(0) = 0$ |
| Rosenbrock (f4)        | n-dimensional, continuous, differentiable, non-separable, multimodal, non-convex | $\sum_{d=1}^{n} [100.(\text{pos}_{d+1} - \text{pos}_d^2)^2 + (1 - \text{pos}_d)^2]$ | $[-5, 10]$ | $f(1) = 0$ |
| Sphere (f5)            | n-dimensional, continuous, convex, differentiable, unimodal, separable | $\sum_{d=1}^{n} \text{pos}_d^2$ | $[-5.12, +5.12]$ | $f(0) = 0$ |

### TABLE II. COMPUTED BEST ERROR FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS (FIG. 4)

| Swarm Size | Dimension | BMF | PSOs          | CIWPSO | RIWPSO | LDWIWPSO | LSTMWIWPSO |
|------------|-----------|-----|---------------|--------|--------|----------|------------|
|            |           |     |               | 0.000010 | 0.000009 | 0.000009 | 0.000059   |
|            |           |     |               | 0.000000 | 0.000000 | 0.000000 | 0.000000   |
|            |           |     |               | 0.000009 | 0.000010 | 0.000007 | 0.000009   |
|            |           |     |               | 0.000008 | 0.000007 | 0.000008 | 0.000006   |
|            |           |     |               | 0.000007 | 0.000009 | 0.000007 | 0.000009   |
| 10         | 10        | f1  | 0.002602      | 0.003873 | 0.005756 | 0.015133 |
|            |           |     |               | 0.000000 | 0.000000 | 0.000000 | 0.000000   |
|            |           |     |               | 0.000009 | 0.000009 | 0.000004 | 0.000008   |
|            |           |     |               | 0.000008 | 0.000007 | 0.000007 | 0.000008   |
|            |           |     |               | 0.000010 | 0.000010 | 0.000010 | 0.000114   |
| 50         | 15        | f1  | 0.031975      | 0.056392 | 0.429402 | 1.262133 |
|            |           |     |               | 0.000000 | 0.000000 | 0.000000 | 0.000000   |
|            |           |     |               | 0.000052 | 0.000040 | 0.000093 | 0.000209   |
|            |           |     |               | 0.000012 | 0.000010 | 0.000032 | 0.000155   |
|            |           |     |               | 0.000010 | 0.000028 | 0.000460 | 0.001303   |
| 25         | 25        | f1  | 0.027271      | 0.044731 | 0.124196 | 0.486989 |
|            |           |     |               | 0.000008 | 0.000009 | 0.000008 | 0.000009   |
|            |           |     |               | 0.000000 | 0.000000 | 0.000000 | 0.000000   |
|            |           |     |               | 0.000009 | 0.000009 | 0.000004 | 0.000008   |
|            |           |     |               | 0.000006 | 0.000007 | 0.000007 | 0.000008   |
|            |           |     |               | 0.000009 | 0.000009 | 0.000007 | 0.000008   |
| 75         | 10        | f1  | 0.000519      | 0.000813 | 0.000330 | 0.002932 |
|            |           |     |               | 0.000000 | 0.000000 | 0.000000 | 0.000000   |
|            |           |     |               | 0.000009 | 0.000009 | 0.000008 | 0.000010   |
|            |           |     |               | 0.000010 | 0.000010 | 0.000010 | 0.000114   |
| 25         | 15        | f1  | 0.027271      | 0.044731 | 0.124196 | 0.486989 |
|            |           |     |               | 0.000008 | 0.000009 | 0.000008 | 0.000009   |
|            |           |     |               | 0.000000 | 0.000000 | 0.000000 | 0.000000   |
|            |           |     |               | 0.000009 | 0.000009 | 0.000008 | 0.000010   |
|            |           |     |               | 0.000010 | 0.000010 | 0.000010 | 0.000114   |
|            |           |     |               | 0.000010 | 0.000010 | 0.000010 | 0.000114   |
|            |           |     |               | 0.000010 | 0.000010 | 0.000010 | 0.000114   |
|            |           |     |               | 0.000010 | 0.000010 | 0.000010 | 0.000114   |
| Swarm Size | 50 | 75 | 100 |
|------------|----|----|-----|
| Dim | 10 | 15 | 25 |
| Error | Error | Error | Error |
| Error | Error | Error | Error |

Comparison of PSOs with Best Error

Fig. 4. Best Error Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.
| Swarm Size | Dimension | BMF | PSOs | CIWPSO | RIWPSO | LDIWPSO | LSTMIWPSO |
|------------|-----------|-----|------|--------|--------|---------|------------|
| 10         | f1        | 3.829570 | 5.472203 | 8.170220 | 2.726947 |
|            | f2        | 54.932726 | 54.983211 | 62.611015 | 41.404392 |
|            | f3        | 0.127753 | 0.150475 | 0.328519 | 0.098769 |
|            | f4        | 0.052547 | 0.028840 | 0.061509 | 0.030330 |
|            | f5        | 0.225241 | 0.148772 | 0.319764 | 0.102929 |
| 50         | f1        | 10.028167 | 14.396357 | 19.456177 | 6.658518 |
|            | f2        | 85.250766 | 90.343807 | 91.788437 | 76.181956 |
|            | f3        | 0.244939 | 0.261647 | 0.445211 | 0.110673 |
|            | f4        | 0.039496 | 0.053941 | 0.073551 | 0.027963 |
|            | f5        | 0.268360 | 0.376879 | 0.456456 | 0.178421 |
| 25         | f1        | 33.171864 | 39.722826 | 52.601215 | 25.140189 |
|            | f2        | 105.036487 | 104.440098 | 150.532868 | 102.950521 |
|            | f3        | 0.779991 | 1.220720 | 1.224990 | 0.636701 |
|            | f4        | 0.039496 | 0.053941 | 0.073551 | 0.027963 |
|            | f5        | 0.268360 | 0.376879 | 0.456456 | 0.178421 |
| 75         | f1        | 4.463292 | 4.308925 | 10.524850 | 3.454499 |
|            | f2        | 57.500383 | 58.948269 | 64.581158 | 44.469230 |
|            | f3        | 0.217088 | 0.141495 | 0.297972 | 0.219178 |
|            | f4        | 0.050142 | 0.030269 | 0.057245 | 0.055129 |
|            | f5        | 0.210561 | 0.142058 | 0.341972 | 0.294266 |
| 100        | f1        | 8.007375 | 10.336619 | 14.119257 | 5.558153 |
|            | f2        | 87.973719 | 85.348446 | 101.576149 | 69.612380 |
|            | f3        | 0.204710 | 0.265032 | 1.080641 | 0.460648 |
|            | f4        | 0.045569 | 0.045599 | 0.065754 | 0.021016 |
|            | f5        | 0.199728 | 0.283209 | 0.368008 | 0.138094 |
| 15         | f1        | 23.154072 | 27.528130 | 41.54209 | 17.278593 |
|            | f2        | 158.914098 | 154.805922 | 159.278378 | 105.304169 |
|            | f3        | 0.602166 | 0.707678 | 1.080641 | 0.460648 |
|            | f4        | 0.097639 | 0.141208 | 0.176523 | 0.074067 |
|            | f5        | 0.557023 | 0.795652 | 1.061555 | 0.464447 |
| 25         | f1        | 4.888757 | 5.141855 | 9.926288 | 8.860705 |
|            | f2        | 53.176830 | 50.619999 | 59.964870 | 34.532425 |
|            | f3        | 0.214487 | 0.131158 | 0.275377 | 0.304044 |
|            | f4        | 0.051496 | 0.030658 | 0.050414 | 0.070342 |
|            | f5        | 0.232056 | 0.15269 | 0.306427 | 0.364919 |
| 100        | f1        | 6.210106 | 6.730709 | 12.998519 | 4.498284 |
|            | f2        | 94.473432 | 82.665049 | 92.99693 | 69.658835 |
|            | f3        | 0.213848 | 0.267667 | 0.361411 | 0.103306 |
|            | f4        | 0.049426 | 0.041846 | 0.079216 | 0.024861 |
|            | f5        | 0.197348 | 0.222021 | 0.349822 | 0.104669 |
| 15         | f1        | 20.104113 | 22.065833 | 35.406524 | 14.607688 |
|            | f2        | 158.586419 | 104.118385 | 180.423557 | 102.180907 |
|            | f3        | 0.491150 | 0.705747 | 0.947726 | 0.383817 |
|            | f4        | 0.078310 | 0.113074 | 0.150317 | 0.059783 |
|            | f5        | 0.505790 | 0.651178 | 0.972611 | 0.386508 |
**Comparison of PSOs with Mean Error**

**TABLE IV.**  **COMPUTED VARIANCE FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS. (FIG. 6)**

| Swarm Size | Dimension | BMF | PSOs CIWPSO | RlWPSO | LDIWPSO | LSTMIWPSO |
|------------|-----------|-----|-------------|--------|---------|------------|
| 50         | 10        | $f_1$ | 2.24626E+03 | 3.38851E+03 | 4.67975E+03 | 9.63606E+02 |
|            |           | $f_2$ | 3.70666E+03 | 3.69348E+03 | 4.13007E+03 | 9.41880E+01 |
|            |           | $f_3$ | 1.62064E+00 | 2.07616E+00 | 5.02929E+00 | 2.80457E+03 |
|            |           | $f_4$ | 3.80843E+00 | 2.25555E+00 | 5.23459E+00 | 5.96364E+01 |
|            | 15        | $f_1$ | 1.03989E+04 | 1.26799E+04 | 1.72068E+04 | 3.34864E+03 |
|            |           | $f_2$ | 1.07078E+04 | 1.22065E+04 | 1.20338E+04 | 8.28701E+03 |
|            |           | $f_3$ | 6.12127E+00 | 5.95316E+00 | 1.12496E+01 | 2.54517E+00 |
|            |           | $f_4$ | 1.53739E-01 | 1.85184E-01 | 2.84031E-01 | 5.52154E-02 |
|            |           | $f_5$ | 6.13349E+00 | 7.48563E+00 | 9.61024E+00 | 2.78997E+00 |
|            | 25        | $f_1$ | 5.01816E+04 | 4.97107E+04 | 7.48699E+04 | 1.94431E+04 |
|            |           | $f_2$ | 3.92425E+03 | 4.12993E+03 | 3.72531E+03 | 2.44186E+03 |
|            |           | $f_3$ | 2.79965E+01 | 2.78540E+01 | 4.39840E+01 | 1.18317E+01 |
|            |           | $f_4$ | 1.61978E+00 | 6.27931E+00 | 1.18338E+00 | 3.70447E+01 |
|            |           | $f_5$ | 3.28743E+01 | 3.26392E+01 | 4.72444E+01 | 1.44918E+01 |
|            | 75        | $f_1$ | 2.09251E+03 | 2.13522E+03 | 5.23097E+03 | 1.20472E+03 |
|            |           | $f_2$ | 3.88104E+03 | 3.57230E+03 | 4.06253E+03 | 2.88180E+03 |
|            |           | $f_3$ | 3.10357E+00 | 1.88301E+00 | 3.61849E+00 | 1.57710E+00 |
|            |           | $f_4$ | 1.11167E+01 | 6.39797E+00 | 1.36452E+01 | 6.08377E+02 |
|            |           | $f_5$ | 3.25187E+00 | 1.78081E+00 | 4.39770E+00 | 2.39049E+00 |
|            | 15        | $f_1$ | 7.28370E+03 | 8.33270E+03 | 1.39262E+04 | 3.36059E+03 |

Fig. 5. Mean Error Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.
Fig. 6. Variance Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.
| Swarm Size | Dimension | BMF | PSOs |
|------------|-----------|-----|------|
|            |           |     | CIWPSO | RIWPSO | LDIWPSO | LSTMIWPSO |
| 10         | f1        | 47.394690 | 58.210892 | 68.408725 | 31.042005 |
|            | f2        | 60.882317 | 60.774044 | 64.265634 | 52.958210 |
|            | f3        | 1.273044  | 1.440890  | 2.242608  | 0.970505  |
|            | f4        | 0.347028  | 0.245135  | 0.374315  | 0.216737  |
|            | f5        | 1.951520  | 1.501847  | 2.287923  | 0.952031  |
| 50         | f1        | 101.974922 | 112.604830 | 131.174709 | 57.867444 |
|            | f2        | 224.012548 | 222.959056 | 273.607132 | 139.438498 |
|            | f3        | 2.476588  | 2.735988  | 3.100039  | 1.667625  |
|            | f4        | 0.392095  | 0.430330  | 0.532945  | 0.234980  |
|            | f5        | 5.733607  | 5.713069  | 6.873457  | 3.806805  |
| 25         | f1        | 45.743919 | 46.208445 | 72.325443 | 34.709071 |
|            | f2        | 62.43794 | 64.260301 | 193.016188 | 49.415159 |
|            | f3        | 1.761696  | 1.372227  | 1.902232  | 1.255827  |
|            | f4        | 0.333417  | 0.263400  | 0.369394  | 0.246653  |
|            | f5        | 1.803295  | 1.334471  | 2.097070  | 1.546120  |
| 75         | f1        | 85.346604 | 91.283633 | 118.009469 | 57.970561 |
|            | f2        | 101.389839 | 99.364403 | 110.43280 | 92.279921 |
|            | f3        | 2.285847  | 2.451669  | 2.910414  | 1.326765  |
|            | f4        | 0.431915  | 0.397092  | 0.499696  | 0.213786  |
|            | f5        | 2.100994  | 2.395941  | 2.864034  | 1.422026  |
| 25         | f1        | 195.105057 | 192.012109 | 239.600461 | 118.145195 |
|            | f2        | 196.025427 | 195.014739 | 191.355322 | 65.883501 |
|            | f3        | 4.832010  | 4.396835  | 6.065558  | 3.138851  |
|            | f4        | 0.810568  | 0.833175  | 1.000340  | 0.515799  |
|            | f5        | 4.645820  | 4.761545  | 6.132645  | 3.181614  |
| 100        | f1        | 46.527046 | 49.671001 | 70.998898 | 51.879999 |
|            | f2        | 53.180790 | 56.009131 | 56.279433 | 47.363728 |
|            | f3        | 1.642218  | 1.220329  | 1.865850  | 1.524173  |
|            | f4        | 0.331908  | 0.243425  | 0.340103  | 0.295787  |
|            | f5        | 1.853558  | 1.449718  | 1.998035  | 1.730258  |
| 15         | f1        | 76.755788 | 78.944668 | 111.393549 | 49.040620 |
|            | f2        | 102.665394 | 94.551437 | 101.867981 | 89.722486 |
|            | f3        | 2.294524  | 2.471681  | 2.879141  | 1.189510  |
|            | f4        | 0.441106  | 0.368965  | 0.557601  | 0.225771  |
|            | f5        | 2.247819  | 2.221878  | 2.704631  | 1.191100  |
| 25         | f1        | 172.330480 | 179.237775 | 218.763244 | 111.983985 |
|            | f2        | 198.980864 | 56.086860 | 204.318741 | 39.473677 |
|            | f3        | 4.417719  | 4.837791  | 5.983300  | 2.757501  |
|            | f4        | 0.711278  | 0.834077  | 0.938832  | 0.473201  |
|            | f5        | 4.587458  | 4.706557  | 5.513798  | 2.875684  |
Fig. 7. Standard Deviation Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.

TABLE VI. COMPUTED MEAN SQUARED ERROR (MSE) FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS. (Fig. 8)

| Swarm Size | Dimension | BMF | PSOs  | CIWPSO | RIWPSO | LDIWPSO | LSTMIWPSO |
|------------|-----------|-----|-------|--------|--------|---------|-----------|
| 10         |           | f1  | 2.26077E+03 | 3.41820E+03 | 4.74615E+03 | 9.70978E+02 |
|            |           | f2  | 6.67734E+03 | 6.66534E+03 | 7.98667E+03 | 4.49193E+03 |
|            |           | f3  | 1.63680E+00 | 2.09866E+00 | 5.13660E+00 | 9.51544E-01 |
|            |           | f4  | 1.23163E-01 | 6.09165E-02 | 1.43872E-01 | 4.78862E-02 |
|            |           | f5  | 3.85862E+00 | 2.27750E+00 | 5.33624E+00 | 9.16860E+01 |
| 50         | 10        | f1  | 1.04988E+04 | 1.28863E+04 | 1.75842E+04 | 3.39275E+03 |
|            |           | f2  | 1.78673E+04 | 2.02623E+04 | 2.03206E+04 | 1.40315E+04 |
|            |           | f3  | 6.18086E+00 | 6.02122E+00 | 1.14470E+01 | 2.57817E+00 |
|            |           | f4  | 1.55289E-01 | 1.88081E-01 | 2.89422E-01 | 5.59373E-02 |
|            |           | f5  | 6.20509E+00 | 7.62717E+00 | 9.81795E+00 | 2.81262E+00 |
| 15         | 25        | f1  | 5.12787E+04 | 5.12853E+04 | 7.62288E+04 | 2.00738E+04 |
|            |           | f2  | 1.49534E+04 | 1.50335E+04 | 5.96243E+04 | 1.30396E+04 |
|            |           | f3  | 2.86030E+01 | 2.93423E+01 | 4.54817E+01 | 1.22363E+01 |
|            |           | f4  | 7.09540E-01 | 9.05991E-01 | 1.22516E+00 | 3.82671E-01 |
|            |           | f5  | 3.35760E+01 | 3.36012E+01 | 4.90412E+01 | 1.49388E+01 |
| 25         | 10        | f1  | 2.11220E+03 | 2.15361E+03 | 5.34109E+03 | 1.21651E+03 |
|            |           | f2  | 7.12371E+03 | 6.99224E+03 | 8.16321E+03 | 4.82459E+03 |
|            |           | f3  | 3.15013E+00 | 1.90286E+00 | 3.70678E+00 | 1.62461E+00 |
|            |           | f4  | 1.13649E-01 | 7.02846E-02 | 1.39706E-01 | 6.38427E-02 |
|            |           | f5  | 3.29564E+00 | 1.80078E+00 | 4.51394E+00 | 2.47614E+00 |
| Swarm Size | MSE | Sphere | Rastrigin | Rosenbrock | Ackley | Alpine |
|------------|-----|--------|-----------|-------------|--------|--------|
| 10         |     |    7.34733E+03 | 8.43899E+03 | 1.41247E+04 | 3.39126E+03 |        |
| 15         |     | 1.78875E+04 | 1.62664E+04 | 8.60735E+04 | 1.77677E+04 |        |
| 25         |     | 4.45377E+00 | 6.31172E+00 | 6.16264E+00 | 1.54260E+04 |        |
| 100        |     | 2.18834E+03 | 2.49337E+03 | 5.18688E+03 | 2.76904E+03 |        |
| 25         |     | 1.78875E+04 | 1.62664E+04 | 8.60735E+04 | 1.77677E+04 |        |
| 15         |     | 4.45377E+00 | 6.31172E+00 | 6.16264E+00 | 1.54260E+04 |        |
| 25         |     | 2.18834E+03 | 2.49337E+03 | 5.18688E+03 | 2.76904E+03 |        |

Fig. 8. MSE Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.
| Swarm Size | Dimension | BMF | CIWPSO | RIWPSO | LDIWPSO | LSTMIWPSO |
|------------|-----------|-----|--------|--------|---------|-----------|
| 10         | 10        | f₁  | 47.547507 | 58.465408 | 68.892273 | 31.160521 |
|            |           | f₂  | 81.715000  | 81.641530  | 89.368178  | 67.021852  |
|            |           | f₃  | 1.279373   | 1.448675   | 2.266406   | 0.975471   |
|            |           | f₄  | 0.350946   | 0.246813   | 0.379305   | 0.218829   |
|            |           | f₅  | 1.964338   | 1.509139   | 2.310031   | 0.957528   |
| 50         | 15        | f₁  | 102.463436 | 113.517652 | 132.605429 | 58.247349  |
|            |           | f₂  | 133.668738 | 142.345836 | 142.550222 | 118.454673 |
|            |           | f₃  | 2.486133   | 2.453817   | 3.383343   | 1.605668   |
|            |           | f₄  | 0.394067   | 0.433683   | 0.537979   | 0.236630   |
|            |           | f₅  | 2.491002   | 2.761733   | 3.133362   | 1.677088   |
| 25         | 10        | f₁  | 226.447895 | 226.462645 | 278.608614 | 141.682136 |
|            |           | f₂  | 122.283946 | 122.611075 | 244.180950 | 114.190940 |
|            |           | f₃  | 5.348177   | 5.416850   | 6.744009   | 3.498047   |
|            |           | f₄  | 0.394067   | 0.433683   | 0.537979   | 0.236630   |
|            |           | f₅  | 2.491002   | 2.761733   | 3.133362   | 1.677088   |
| 75         | 15        | f₁  | 45.958712  | 46.406975  | 73.082779  | 34.878568  |
|            |           | f₂  | 84.402064  | 83.619612  | 90.350480  | 69.459280  |
|            |           | f₃  | 1.774860   | 1.379441   | 1.925301   | 1.274603   |
|            |           | f₄  | 0.337119   | 0.265112   | 0.373773   | 0.252671   |
|            |           | f₅  | 1.815388   | 1.341929   | 2.124604   | 1.573574   |
| 25         | 10        | f₁  | 85.716591  | 91.863986  | 118.847212 | 58.234483  |
|            |           | f₂  | 133.740488 | 130.526763 | 149.444504 | 115.296759 |
|            |           | f₃  | 2.294917   | 2.465871   | 2.933828   | 1.332957   |
|            |           | f₄  | 0.337119   | 0.265112   | 0.373773   | 0.252671   |
|            |           | f₅  | 2.110395   | 2.412541   | 2.887486   | 1.428668   |
| 100        | 15        | f₁  | 196.467698 | 193.969044 | 243.167928 | 119.398100 |
|            |           | f₂  | 251.628992 | 248.246634 | 248.241652 | 124.201480 |
|            |           | f₃  | 4.869227   | 4.532777   | 6.160870   | 3.172369   |
|            |           | f₄  | 0.816401   | 0.845029   | 1.015762   | 0.521073   |
|            |           | f₅  | 4.678940   | 4.827408   | 6.223642   | 3.215230   |
| 100        | 10        | f₁  | 46.779695  | 49.933663  | 71.684569  | 52.621656  |
|            |           | f₂  | 74.916360  | 75.174099  | 81.887555  | 58.311259  |
|            |           | f₃  | 1.655961   | 1.227283   | 1.885909   | 1.553789   |
|            |           | f₄  | 0.335829   | 0.245325   | 0.343790   | 0.303934   |
|            |           | f₅  | 1.867821   | 1.457922   | 2.021234   | 1.767802   |
| 100        | 15        | f₁  | 77.004050  | 79.228452  | 111.947053 | 49.244864  |
|            |           | f₂  | 139.013944 | 125.46853  | 136.841670 | 113.219355 |
|            |           | f₃  | 2.304377   | 2.486044   | 2.901624   | 1.193948   |
|            |           | f₄  | 0.443839   | 0.371318   | 0.562281   | 0.227125   |
|            |           | f₅  | 2.256382   | 2.232855   | 2.727056   | 1.195651   |
| 100        | 25        | f₁  | 173.493486 | 180.584992 | 221.602772 | 112.929011 |
|            |           | f₂  | 253.690037 | 218.251843 | 271.642611 | 109.469148 |
|            |           | f₃  | 4.444791   | 4.888839   | 6.057695   | 2.783994   |
|            |           | f₄  | 0.715552   | 0.841015   | 0.950759   | 0.476947   |
|            |           | f₅  | 4.615105   | 4.751235   | 5.598742   | 2.901447   |
Comparison of PSOs with RMSE

![Graphs showing comparison of PSOs with RMSE for different swarm sizes and dimensions.]

Fig. 9. RMSE Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.

| Swarm Size | Dimension | BMF | PSOs | CIWPSO | RIWPSO | LDIWPSO | LSTMIWPSO |
|------------|-----------|-----|------|--------|--------|---------|-----------|
| 50         | 10        | f1  | 3.312812 | 3.183287 | 3.589538 | 0.020334 | 7.459693 |
|            |           | f2  | 0.021386 | 0.025384 | 0.020334 | 0.053375 | 0.053375 |
|            |           | f3  | 2.257866 | 3.293823 | 1.886251 | 5.159045 | 5.159045 |
|            |           | f4  | 1.138227 | 2.282051 | 1.398866 | 2.658974 | 2.658974 |
|            |           | f5  | 1.624259 | 3.076825 | 2.082976 | 4.621895 | 4.621895 |
| 75         | 10        | f1  | 3.273303 | 4.050883 | 3.842036 | 5.982670 | 5.982670 |
|            |           | f2  | 0.026450 | 0.026450 | 0.022253 | 0.059991 | 0.059991 |
|            |           | f3  | 1.855718 | 3.775724 | 2.897390 | 2.032672 | 2.032672 |
|            |           | f4  | 1.208450 | 2.102096 | 2.027942 | 1.292319 | 1.292319 |
|            |           | f5  | 1.956389 | 2.852364 | 2.153064 | 1.722074 | 1.722074 |
### Table 1: Comparison of PSOs with Mean Time (in Secs)

| Swarm Size | Dim | PSO Type | Time (Secs) |
|------------|-----|----------|-------------|
| 50         | 10  | PSO1     | 6.526218    | 6.518706    | 6.866313    | 13.257469  |
|            |     | PSO2     | 0.041508    | 0.038976    | **0.030914**| 0.117080   |
|            |     | PSO3     | 6.276307    | 6.611565    | 6.669365    | 13.423912  |
|            |     | PSO4     | 4.794559    | 6.383440    | 5.537492    | 12.628298  |
|            |     | PSO5     | 6.380909    | 6.439272    | 6.652887    | 13.195447  |
| 50         | 15  | PSO1     | 8.987696    | 9.227687    | 9.441670    | 19.331790  |
|            |     | PSO2     | 0.073887    | 0.077486    | **0.063161**| 1.549432   |
|            |     | PSO3     | 9.008012    | 9.178240    | 9.021073    | 19.068662  |
|            |     | PSO4     | 9.058448    | 9.075037    | **8.982564**| 19.274107  |
|            |     | PSO5     | 9.163049    | 9.168713    | 9.048472    | 19.067931  |
| 50         | 25  | PSO1     | 3.016330    | 4.171748    | 3.530148    | **2.414135**|
|            |     | PSO2     | 0.032180    | 0.033646    | 0.028782    | 0.058631   |
|            |     | PSO3     | 1.819471    | 3.790044    | 2.733838    | **1.586756**|
|            |     | PSO4     | 1.574290    | 2.531497    | 2.621043    | **1.241292**|
|            |     | PSO5     | 2.020080    | 3.923700    | 2.951757    | **1.423464**|
| 50         | 10  | PSO1     | 9.002749    | 8.977202    | 9.125075    | **17.114905**|
|            |     | PSO2     | 0.046437    | 0.049836    | 0.046704    | 0.112377   |
|            |     | PSO3     | 7.197202    | 8.082389    | 7.425860    | 18.320117  |
|            |     | PSO4     | 4.608342    | 8.081904    | 5.849772    | **13.59670**|
|            |     | PSO5     | 7.656653    | **7.275687**| 8.531872    | 17.034604  |
| 50         | 15  | PSO1     | 12.429958   | **12.134209**| 12.552006   | 24.936847  |
|            |     | PSO2     | 0.091344    | 0.841345    | **0.070490**| 3.426473   |
|            |     | PSO3     | 12.091566   | 12.138803   | **12.043132**| 25.225323  |
|            |     | PSO4     | 12.34614    | 13.969420   | **12.167084**| 27.915432  |
|            |     | PSO5     | **11.938530**| 12.319092   | 12.264439   | 24.890808  |

---

**Fig. 10.** Mean Time (In Secs) Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.
| Swarm Size | Dimension | BMF | PSOs CIWPSO | RIWPSO | LDlWPSO | LSTMIWPSO |
|------------|-----------|-----|-------------|--------|---------|------------|
| 50         | 10        | f1  | 954.67      | 907.13 | 864.67  | 1000.00    |
|            |           | f2  | 5.27        | 4.80   | 4.33    | 6.93       |
|            |           | f3  | 652.13      | 942.40 | 543.67  | 694.47     |
|            |           | f4  | 306.33      | 617.27 | 411.20  | 353.20     |
|            |           | f5  | 468.20      | 842.73 | 584.67  | 615.20     |
| 50         | 15        | f1  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f2  | 6.60        | 4.80   | 5.80    | 9.33       |
|            |           | f3  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f4  | 1000.00     | 989.87 | 1000.00 | 1000.00    |
|            |           | f5  | 983.93      | 1000.00| 1000.00 | 1000.00    |
| 50         | 25        | f1  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f2  | 73.80       | 75.53  | 8.53    | 148.40     |
|            |           | f3  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f4  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f5  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
| 75         | 10        | f1  | 623.13      | 791.27 | 536.87  | 579.20     |
|            |           | f2  | 4.07        | 4.33   | 3.87    | 5.53       |
|            |           | f3  | 363.33      | 731.40 | 491.60  | 198.40     |
|            |           | f4  | 231.93      | 409.73 | 395.33  | 118.47     |
|            |           | f5  | 378.07      | 545.87 | 417.47  | 168.73     |
| 75         | 15        | f1  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f2  | 5.20        | 5.47   | 4.53    | 8.33       |
|            |           | f3  | 974.93      | 991.07 | 1000.00 | 1000.00    |
|            |           | f4  | 729.13      | 982.33 | 851.73  | 957.67     |
|            |           | f5  | 981.60      | 992.33 | 1000.00 | 1000.00    |
| 75         | 25        | f1  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f2  | 7.07        | 6.87   | 6.73    | 80.47      |
|            |           | f3  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f4  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f5  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
| 100        | 10        | f1  | 442.47      | 595.20 | 481.87  | 178.07     |
|            |           | f2  | 4.33        | 4.33   | 3.67    | 4.20       |
|            |           | f3  | 265.33      | 545.87 | 402.27  | 120.67     |
|            |           | f4  | 217.93      | 360.47 | 379.67  | 94.00      |
|            |           | f5  | 296.67      | 551.93 | 406.20  | 108.67     |
| 100        | 15        | f1  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f2  | 5.00        | 5.33   | 4.93    | 6.40       |
|            |           | f3  | 835.07      | 932.47 | 854.53  | 1000.00    |
|            |           | f4  | 533.40      | 937.13 | 667.60  | 698.20     |
|            |           | f5  | 892.40      | 836.73 | 856.60  | 1000.00    |
| 100        | 25        | f1  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f2  | 6.87        | 7.23   | 5.47    | 140.33     |
|            |           | f3  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f4  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
|            |           | f5  | 1000.00     | 1000.00| 1000.00 | 1000.00    |
V. CONCLUSION AND FUTURE WORK

In this paper, a new inertia weight based PSO using LSTM (LSTMIWPSO) is presented. A set of 5 most common optimization test problems and eight criteria are considered to assess the performance of LSTMIWPSO against CIWPSO, RIWPSO, and LDIWPSO. The overall outcome shows that LSTMIWPSO is progressive with CIWPSO, RIWPSO, and LDIWPSO. In the future, the parameters of LSTM are tuned to enhance efficiency. Also, more experiments with larger swarm sizes and dimensions are conducted to evaluate LSTMIWPSO performance with other existing inertia weight based PSO. There is a scope for the use of LSTMIWPSO in the optimization of the different optimization applications without any restriction of the domains specified.

REFERENCES

[1] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in International Conference on Neural Networks, vol. 4. IEEE, 1995, pp. 1942-1948.
[2] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in International Symposium on Micro Machine and Human Science IEEE, 1995, pp. 39–43.
[3] C. M. Huang, C. J. Huang and M. L. Wang, A Particle Swarm Optimization to Identifying the ARMAX Model for Short-Term Load Forecasting. IEEE Tran. on Power Systems, 20(2): 1126-1133, May 2005.
[4] X. H. Hu, R. C. Eberhart, Y. H. Shi, Engineering optimization. with particle swarm. Proc. of the IEEE Swarm Intelligence Symposium, pages 53-57, 2003.
[5] X. H. Hu, Y. H. Shi, R. Eberhart, Recent advances in particle swarm. Congress on Evolutionary Computation, pages 90-97, 2004.
[6] M. Clerc, Particle Swarm Optimization, ISTE Publishing Company, 2006.
[7] N. Nedjah, L. D. M. Mourelle, Systems Engineering Using Particle Swarm Optimization, Nova Science Publishers, 2007.
[8] Yudong Zhang, Shuihua Wang, and Genlin Ji, A Comprehensive Particle Swarm Optimization Algorithm and Its Applications, Mathematical Problems in Engineering, Volume-2015, pp 1-38.
[9] Ashok Kumar, Brajesh Kumar Singh, B.D.K.Patro, Particle Swarm Optimization: A Study of Variants and their applications, International Journal of Computer Applications, Volume 135 (5), November 2016, PP 24-30.
[10] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, Math. Probl. Eng. (2015).
[11] M. R. Bonyadi and Z. Michalewicz, “Particle swarm optimization for single-objective continuous space problems: a review,” Evolutionary computation, 2016.
[12] J.C. Bansal, P.K. Singh, M. Saraswat, A. Verma, S.S. Jadon, A. Abraham, Inertia weight strategies in particle swarm optimization, in: 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC), 2011, pp.633–640.
[13] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceeding of the IEEE Congress on Evolutionary Computation (CEC), IEEE, Piscataway, NJ, USA, 1998, pp. 69–73. Anchorage, AC, USA.
[14] R.C. Eberhart and Y. Shi., “Tracking and optimizing dynamic systems with particle swarms”, In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, volume 1, pages 94–100. IEEE, 2002.

[15] J. Xin, G. Chen, and Y. Hai., “A Particle Swarm Optimizer with Multistage Linearly-Decreasing Inertia Weight”, In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, volume 1, pages 505–508. IEEE, 2009.

[16] Serani, Andrea & Leotardi, Cecilia & Jemna, Umberto & Campana, Emilio & Fasano, Giovanni & Diez, Matteo. (2016). Particle selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems. Applied Soft Computing. 49. 313-334. 10.1016/j.asoc.2016.08.028.

[17] M.R. Bonyadi, Z. Michalewicz, Impacts of coefficients on movement patterns in the Particle Swarm Optimization algorithm, IEEE Trans. Evol. Comput. 21 (3) (2017) 378–390.

[18] Andries P. Engelbrecht, Computational Intelligence: An Introduction, 2nd Edition, Wiley Publications, West Sussex, England, 2007.

[19] Riccardo Poli, "Analysis of the Publications on the Applications of Particle Swarm Optimisation", Journal of Artificial Evolution and Applications, Vol. 2008, January, 2018, 10 Pages.

[20] E. T. Oldewage, A. P. Engelbrecht and C. W. Cleghorn, The merits of velocity clamping particle swarm optimisation in high dimensional spaces, 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1-8, doi: 10.1109/SSCI.2017.8280887.

[21] Frans van den Bergh. An Analysis of Particle Swarm Optimizers. PhD thesis, University of Pretoria, 2001.

[22] Federico Marin, Beata Waleczak, Particle swarm optimization (PSO). A tutorial, Chemometrics and Intelligent Laboratory Systems, Volume 149, Part B, 2015, Pages 153-165, ISSN 0169-7439, https://doi.org/10.1016/j.chemolab.2015.08.020.

[23] Barbara Hammer, On the approximation capability of recurrent neural networks, Neurocomputing, Volume 31, Issues 1–4, 2000, Pages 107-123, ISSN 0925-2312, https://doi.org/10.1016/S0925-2312(99)00174-5.

[24] Ashfane Amid, Shervine Amidi, Recurrent Neural Network Cheatsheet, https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture, Last accessed on 17.10.2020.

[25] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181.

[26] Mikolov, Tomas & Joulin, Armand & Chopra, Sumit & Mathieu, Michael & Ranzato, Marc'Aurelio. (2014). Learning Longer Memory in Recurrent Neural Networks.

[27] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[28] Bewoor, L.A., Chandra Prakash, V., Sapkal, S.U., Evolutionary hybrid particle swarm optimization algorithm for solving NP-hard no-wait flow shop scheduling problems, Algorithms, Vol: 10, Issue: 4, pp: -, 2017, Doi :10.3390/a10040121.

[29] Lakshmi Prasad, M., Sastry, J.K.R., Building test cases by particle swarm optimization (PSO) for multi output domain embedded systems using combinatorial techniques, Journal of Advanced Research in Dynamical and Control Systems, Vol: 10, Issue: 6, pp: 1221 - 1229, 2018, DoI.

[30] Bewoor, L.A., Prakash, V.C., Sapkal, S.U., Production scheduling optimization in foundry using hybrid Particle Swarm Optimization algorithm, Procedia Manufacturing, Vol: 22, Issue: , pp: 57 - 64, 2018, DoI :10.1016/j.promfg.2018.03.010.

[31] Bewoor, L.A., Chandra Prakash, V., Sapkal, S.U., An improved evolutionary hybrid particle swarm optimization algorithm to minimize makespan for no wait flow shop scheduling, Journal of Theoretical and Applied Information Technology, Vol: 96, Issue: 9, pp: 2534 - 2545, 2018, DoI :

[32] Prasanna Kumar, K., Kishore, M.G.V., Hemanth, K.V., Sreekar, L., Synthesis of antenna array using modified particle swarm optimization technique, International Journal of Innovative Technology and Exploring Engineering, Vol: 8, Issue: 5, pp: 1 - 5, 2019, DoI :

[33] Ahilan, A., Manogaran, G., Raja, C., Kadry, S., Kumar, S.N., Ages Kumar, C., Jarin, T., Krishnamoorthy, S., Malavizhi Kumar, P., Chandra Babu, G., Senthil Murugan, N., Parthasarathy, Segmentation by Fractional Order Darwinian Particle Swarm Optimization Based Multilevel Thresholding and Improved Lossless Prediction Based Compression Algorithm for Medical Images, IEEE Access, Vol: 7, Issue: , pp: 89570 - 89580, 2019, DoI :10.1109/ACCESS.2019.2891632.

[34] Senthil Kumar, A.M., Parthiban, K., Siva Shankar, S., An efficient task scheduling in a cloud computing environment using hybrid Genetic Algorithm - Particle Swarm Optimization (GA-PSO) algorithm, Proceedings of the International Conference on Intelligent Sustainable Systems, ICISIS 2019, Vol: , Issue: , pp: 29 - 34, 2019, DoI :10.1109/ISSI.2019.8908041.

[35] Bewoor, L.A., Chandraprakash, V., Sapkal, S.U., Evolutionary hybrid particle swarm optimization algorithm to minimize makespan to schedule a flow shop with no wait, Journal of Engineering Science and Technology, Vol: 14, Issue: 2, pp: 609 - 628, 2019, DoI :

[36] Kumar, V.H., Varma, P.S., Kumar, T.B., Sreelatha, E., Economic and emission dispatch problem using particle swarm optimization, International Journal of Innovative Technology and Exploring Engineering, Vol: 8, Issue: 6, pp: 939 - 944, 2019, DoI :

[37] Patro, P., Kumar, K., Suresh Kumar, G., Neuro fuzzy system with hybrid ant colony particle swarm optimization (Haso) and robust activation, Journal of Advanced Research in Dynamical and Control Systems, Vol: 12, Issue: 3 Special Issue, pp: 741 - 750, 2020, DoI :10.5373/JARDCS/V12SP3/2020/1312.v

[38] Mehrotra, S., Sharan, A., Comparative Analysis of K-Means Algorithm and Particle Swarm Optimization for Search Result Clustering, Smart Innovation, Systems and Technologies, Vol: 165, Issue: , pp: 109 - 114, 2020, DoI :10.1007/978-981-15-0077-0_12.

[39] Nagesh, P., Srinivasu, N., Ranganadh, N.S.R., Jayanth, J., Naveen, A., Optimal vm placement using particle swarm optimization, International Journal of Scientific and Technology Research, Vol: 9, Issue: 4, pp: 611 - 615, 2020, DoI :

[40] Syamala Rao, P., Parthasaradhi Varma, G., Durga Prasad, Ch., Identification of linear and non linear curve fitting models using particle swarm optimization algorithm, AIP Conference Proceedings, Vol: 2269, Issue: , pp: -, 2020, DoI :10.1063/5.0019657.