Improved PSO Performance using LSTM based Inertia Weight Estimation

Y.V.R.Naga Pawan[1]
Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Green Fields, Vaddeswaram, Guntur District
Andhra Pradesh. India

Kolla Bhanu Prakash[2]
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 weigh 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.

$$f(x): \mathbb{R}^{d} \rightarrow \mathbb{R} \quad (1)$$

where d is the number of dimensions of *search space*, *S*. 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).

$$S \subseteq \mathbb{R}^D \quad (2)$$

$$S = \{px_i \mid px_{min} \leq px_i \leq px_{max}\} \quad (3)$$

$$\min_{x \in S} f(x) \quad (4)$$

The objective function, $f(x)$, needs $px_i \in$ S such that:

$$\forall\, py_i \in S: f(px_i) \leq f(py_i) \quad (5)$$

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

$P_g = \{pg_{i1}, pg_{i2}, pg_{i3}, \dots, pg_{iD}\}$. The basic equations with the functioning of BPSO are given by (6) and (7).

$$pv_{id} = pv_{id} + c_1 * random(\ ) * (pb_i - px_{id}) + c_2 * Random(\ ) * (pg_i - px_{id}) \quad (6)$$

$$px_{id} = px_{id} + pv_{id} \quad (7)$$

where c_1 and c_2 are two positive acceleration coefficients, random() and Random() are two random functions in the [0,1]. pv_i s then clamped to a maximum velocity pv_{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 tends to explore the search space hugely, the velocities of the particles are limited to the constant pv_{max} [16]. The particle velocity is adjusted using.

$$pv_{id} = \begin{cases} pv_{id} \ if \ -pv_{max} \le pv_{id} \le pv_{max} \\ pv_{max} \ if \ pv_{id} > pv_{max} \\ -pv_{max} \ if \ pv_{id} < -pv_{max} \end{cases} \quad (8)$$

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

$$pv_{max} = \delta * (px_{max} - px_{min}) \text{ where } \delta \in (0, 1) \quad (9)$$

As the search space, S, is bounded by the interval $[px_{min}, px_{max}]$, the velocity clamping [22] of the particle is in the interval [-pv_{max}, pv_{max}] $[pv_{min}, pv_{max}]$,

where $pv_{max} = \delta * {(px_{max} - px_{min})}/{2}$.

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).

$$pv_{id} = \omega * pv_{id} + c_1 * random(\ ) * (pb_i - px_{id}) + c_2 * Random(\ ) * (pg_i - px_{id}) \quad (10)$$

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).

$$y^{<t>} = h\left(W_{ya} a^{<t>} + b_y\right) \quad (11)$$

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

where t represents time, $y^{<t>}$ is the predicted value, W_{ya}, W_{aa}, 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).

$$Sigmoid\ function, g(a) = \frac{1}{1 + e^{-a}} \quad (13)$$

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

$$RELU, g(a) = \max(0, a) \quad (15)$$

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, $\mathcal{L}$, for all time steps is defined based on the loss obtained at every time step.

$$\text{Loss Function}, \mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>}) \quad (16)$$

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 (Γ_u), forget gate (Γ_f) and output gate (Γ_o). The computation of $\tilde{c}^{<t>}$, $c^{<t>}$, $a^{<t>}$, Γ_u, Γ_f, Γ_o are shown through equation (17) – equation (22).

$$\tilde{c}^{<t>} = \tanh(w_c[a^{<t-1>}, x^{<t>}] + b_c) > FUnctiontion \quad (17)$$

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

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

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

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \quad (21)$$

$$a^{<t>} = \Gamma_o * \tanh c^{<t>} \quad (22)$$

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:-

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

$$E_{total} = \sum_t E^{<t>} \quad (24)$$

$$E_{total} = \sum_t - y^{<t>} \log(\hat{y}^{<t>}) \quad (25)$$

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

$$\frac{\partial E}{\partial W} = \frac{\partial E^{<t>}}{\partial \hat{y}^{<t>}} \frac{\partial \hat{y}^{<t>}}{\partial a^{<t>}} \frac{\partial a^{<t>}}{\partial c^{<t>}} \frac{\partial c^{<t>}}{\partial c^{<t-1>}} \dots \frac{\partial c^{<0>}}{\partial W} \quad (26)$$

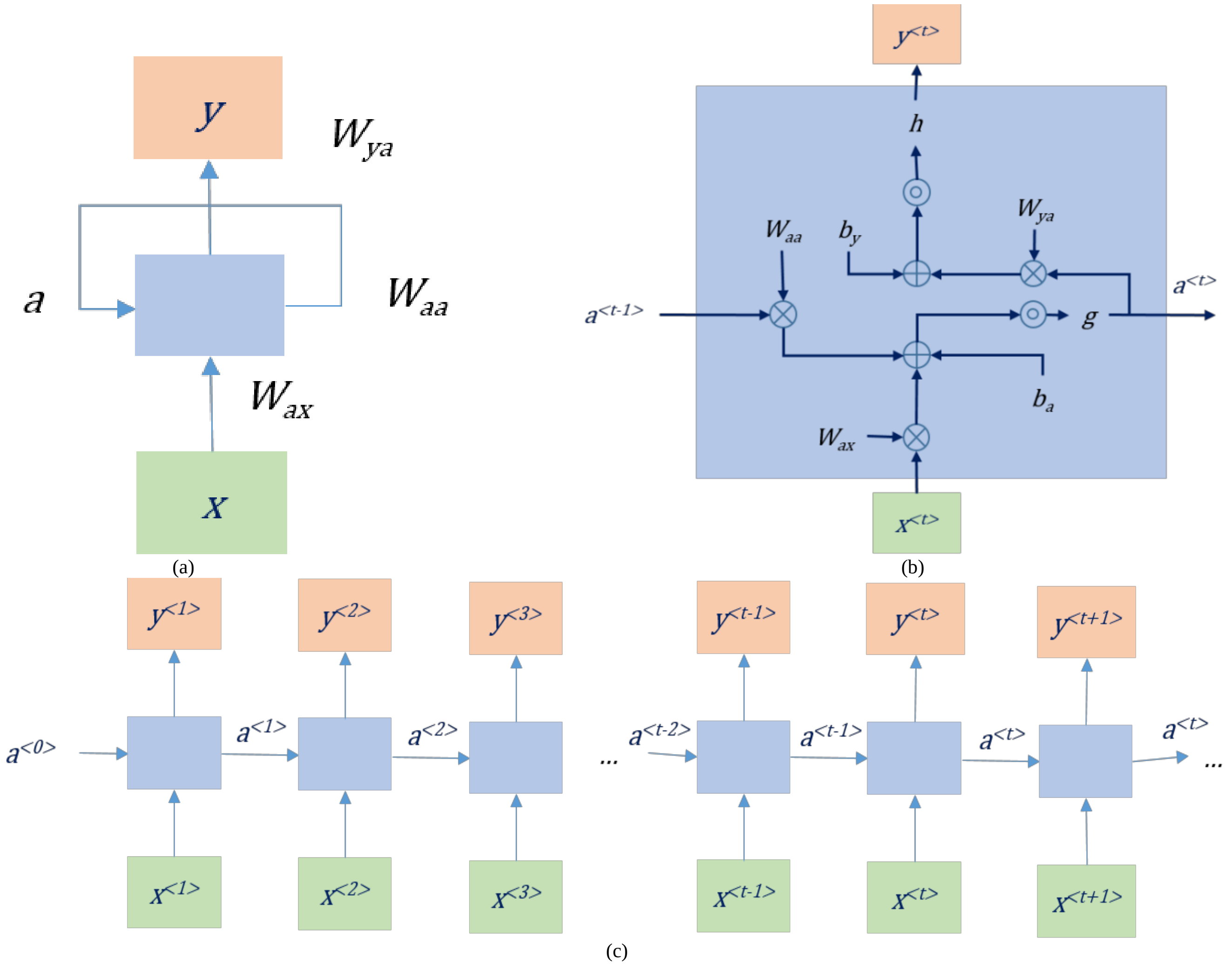

Fig. 1. (a): A Recurrent Neural Network. (b): RNN Cell Handling Dependencies. (c): Unrolled Recurrent Neural Network.

Thus the total error gradient is given by equations (27) and (28):-

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E^{<t>}}{\partial W} \tag{27}$$

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E^{<t>}}{\partial \hat{y}^{<t>}} \frac{\partial \hat{y}^{<t>}}{\partial a^{<t>}} \frac{\partial a^{<t>}}{\partial c^{<t>}} \frac{\partial c^{<t>}}{\partial c^{<t-1>}} \dots \frac{\partial c^{<0>}}{\partial W} \tag{28}$$

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

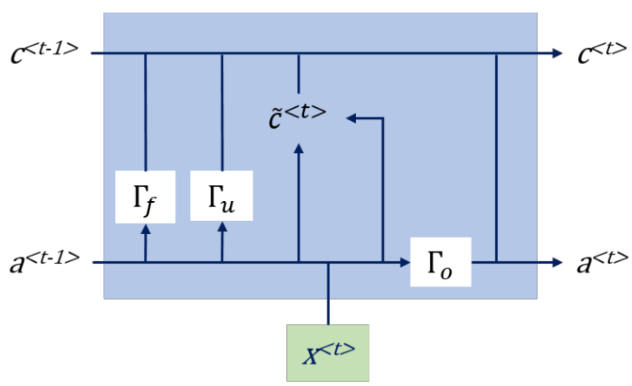

Fig. 2. Memory Cell of an LSTM Showing Dependencies.

The pseudocode for LSTMIWPSO is given below:

Step 1:
 Initialization
  For each particle, P_i, in the population
 Initialize px_i with uniform distribution
 Initialize pv_i randomly.
 Build and Train LSTM network for Inertia Weight Prediction.
 Predict the new Inertia Weight.
 Evaluate the objective function of px_i and assigned the value to fitness[i].
 Initialize $pbest_i$ with a copy of px_i.
 Initialize $pbest\_ftness_i$ with a copy of $fitness_i$.
 Initialize pgbest with index of the particle with the least fitness.
Step 2:
 Repeat until stopping criterion is reached
  For each particle, P_i,:
 Update pv_i and px_i according to the equations (10) and (7)
 Evaluate $fitness_i$
 If $fitness_i$ < $pbest\_fitness_i$ then
  $Pbest_i = px_i$
  $Pbest\_fitness_i = fitness_i$
 Update pgbest 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 f1, f3, f4, and f5 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 f2 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 f1-f5 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 f1 - f5.

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 f1 - f5, 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 f2, f3, and f4 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	$-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{d=1}^{n} pos_d^2} - \exp(\frac{1}{n}\sum_{d=1}^{n}\cos(2\pi\, pos_d)) + 20 + \exp(1)$	[-32, +32]	f(0) = 0
Alpine (f2)	n-dimensional, non-separable, multimodal, non-convex, differentiable	$\sum_{d=1}^{n} \lvert pos_d \,.\sin(pos_d) + 0.1\, pos_d \rvert$	[0, 10]	f(0) = 0
Rastrigin (f3)	n-dimensional, continuous, differentiable, separable, multimodal, convex	$10\,.\,n + \sum_{d=1}^{n}(pos_d^2 - 10\,.\cos(2\pi pos_d))$	[-5.12, +5.12]	f(0) = 0
Rosenbrock (f4)	n-dimensional, continuous, differentiable, non-separable, multimodal, non-convex	$\sum_{d=1}^{n}[100\,.(pos_{d+1} - pos_d^2)^2 + (1 - pos_d)^2]$	[-5, 10]	f(1) = 0
Sphere (f5)	n-dimensional, continuous, convex, differentiable, unimodal, separable	$\sum_{d=1}^{n} pos_d^2$	[-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**	**LDIWPSO**	**LSTMIWPSO**
50	10	f1	0.000010	**0.000009**	**0.000009**	0.000059
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	0.000009	0.000010	**0.000007**	0.000009
		f4	0.000008	0.000007	0.000008	**0.000006**
		f5	**0.000007**	0.000009	**0.000007**	0.000009
	15	f1	**0.002602**	0.003873	0.005756	0.015133
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	0.000052	**0.000040**	0.000093	0.000209
		f4	0.000012	**0.000010**	0.000032	0.000155
		f5	0.000010	0.000283	0.000460	0.001303
	25	f1	**0.031975**	0.056392	0.429402	1.262133
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	**0.001684**	0.001990	0.007345	0.021180
		f4	**0.000319**	0.000401	0.001488	0.003145
		f5	**0.001131**	0.004371	0.017108	0.035360
75	10	f1	**0.000008**	0.000009	**0.000008**	0.000009
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	0.000009	0.000009	**0.000004**	0.000008
		f4	**0.000006**	0.000007	0.000007	0.000008
		f5	0.000009	**0.000006**	0.000007	0.000008
	15	f1	0.000519	0.000813	**0.000330**	0.002932
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	**0.000010**	**0.000010**	0.000022	0.000158
		f4	0.000009	**0.000008**	0.000010	0.000010
		f5	**0.000010**	**0.000010**	**0.000010**	0.000114
	25	f1	**0.027271**	0.044731	0.124196	0.486989

		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	**0.000640**	0.001245	0.006856	0.014690
		f4	0.000125	**0.000074**	0.000165	0.001404
		f5	**0.000529**	0.001319	0.004397	0.008516
100	10	f1	0.000007	0.000007	0.000008	**0.000006**
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	0.000007	0.000006	**0.000005**	0.000007
		f4	0.000008	0.000007	**0.000006**	0.000007
		f5	**0.000006**	0.000007	0.000007	0.000008
	15	f1	0.000087	0.000278	**0.000035**	0.001478
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	0.000010	0.000010	**0.000009**	0.000013
		f4	0.000009	**0.000007**	0.000008	0.000010
		f5	0.000010	**0.000009**	**0.000009**	0.000012
	25	f1	**0.014225**	0.030603	0.083952	0.133495
		f2	**0.000000**	**0.000000**	**0.000000**	**0.000000**
		f3	**0.000387**	0.001223	0.000928	0.002193
		f4	0.000076	**0.000030**	0.000340	0.000644
		f5	0.000279	**0.000026**	0.001968	0.004294

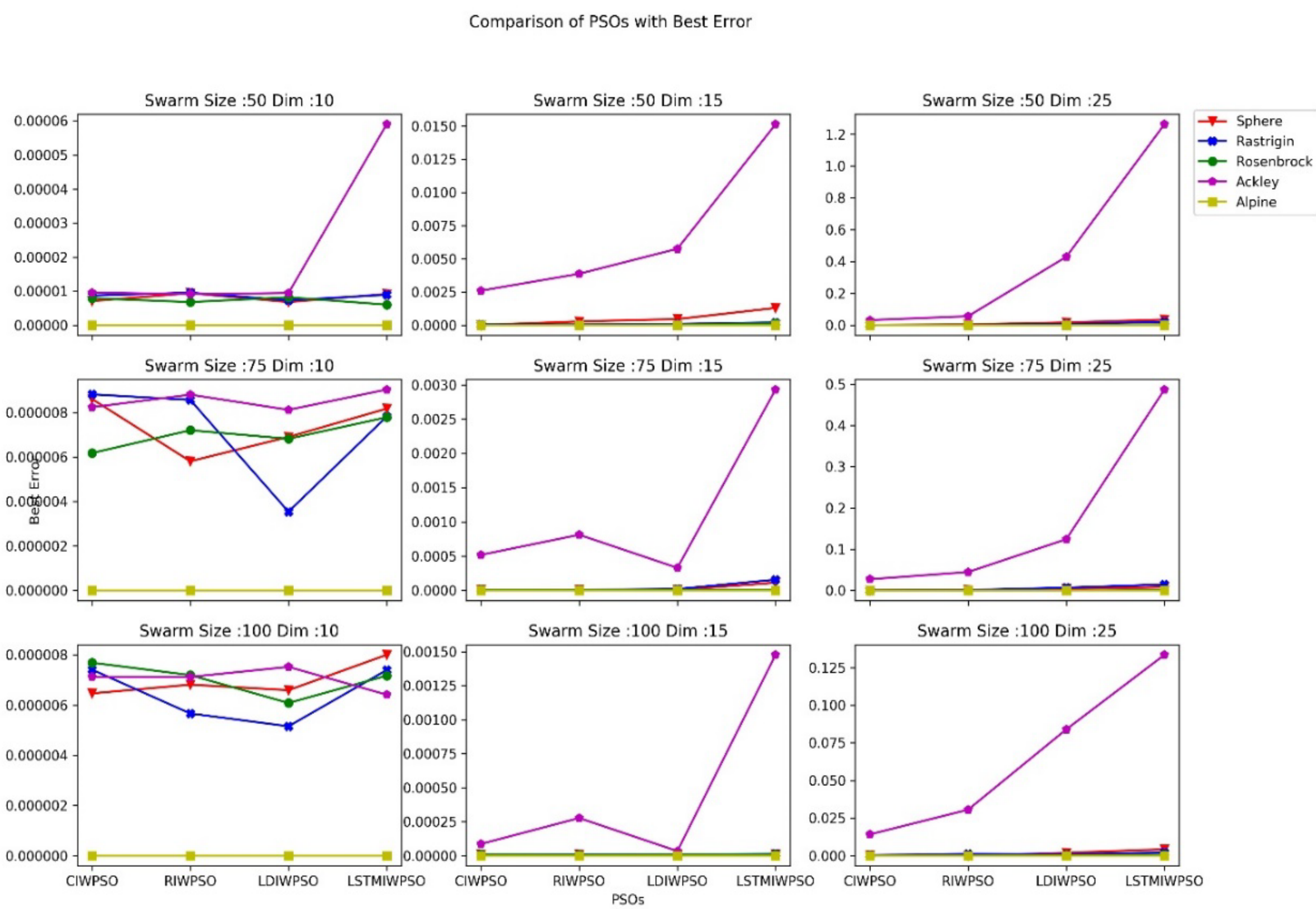

Fig. 4. Best Error Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.

TABLE III. COMPUTED MEAN ERROR FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS. (FIG 5)

Swarm Size	**Dimension**	**BMF**	**PSOs**			
			CIWPSO	**RIWPSO**	**LDIWPSO**	**LSTMIWPSO**
50	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**
	15	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	**0.182114**
		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	0.117500	0.181210	0.204600	**0.110673**
		f5	0.839036	0.981923	1.341617	**0.669297**
75	10	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
	15	f1	8.007375	10.336619	14.119257	**5.558153**
		f2	87.973719	85.348446	101.576149	**69.612380**
		f3	0.204710	0.265032	0.370679	**0.128782**
		f4	0.045569	0.045599	0.065754	**0.021016**
		f5	0.199728	0.283209	0.368008	**0.138094**
	25	f1	23.154072	27.528130	41.546209	**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**
100	10	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	**0.155269**	0.306427	0.364919
	15	f1	6.210106	6.730709	12.998519	**4.498284**
		f2	94.473432	82.665049	92.999693	**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**
	25	f1	20.104113	22.065833	35.406524	**14.607688**
		f2	158.586419	104.118385	180.423557	**102.108097**
		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**

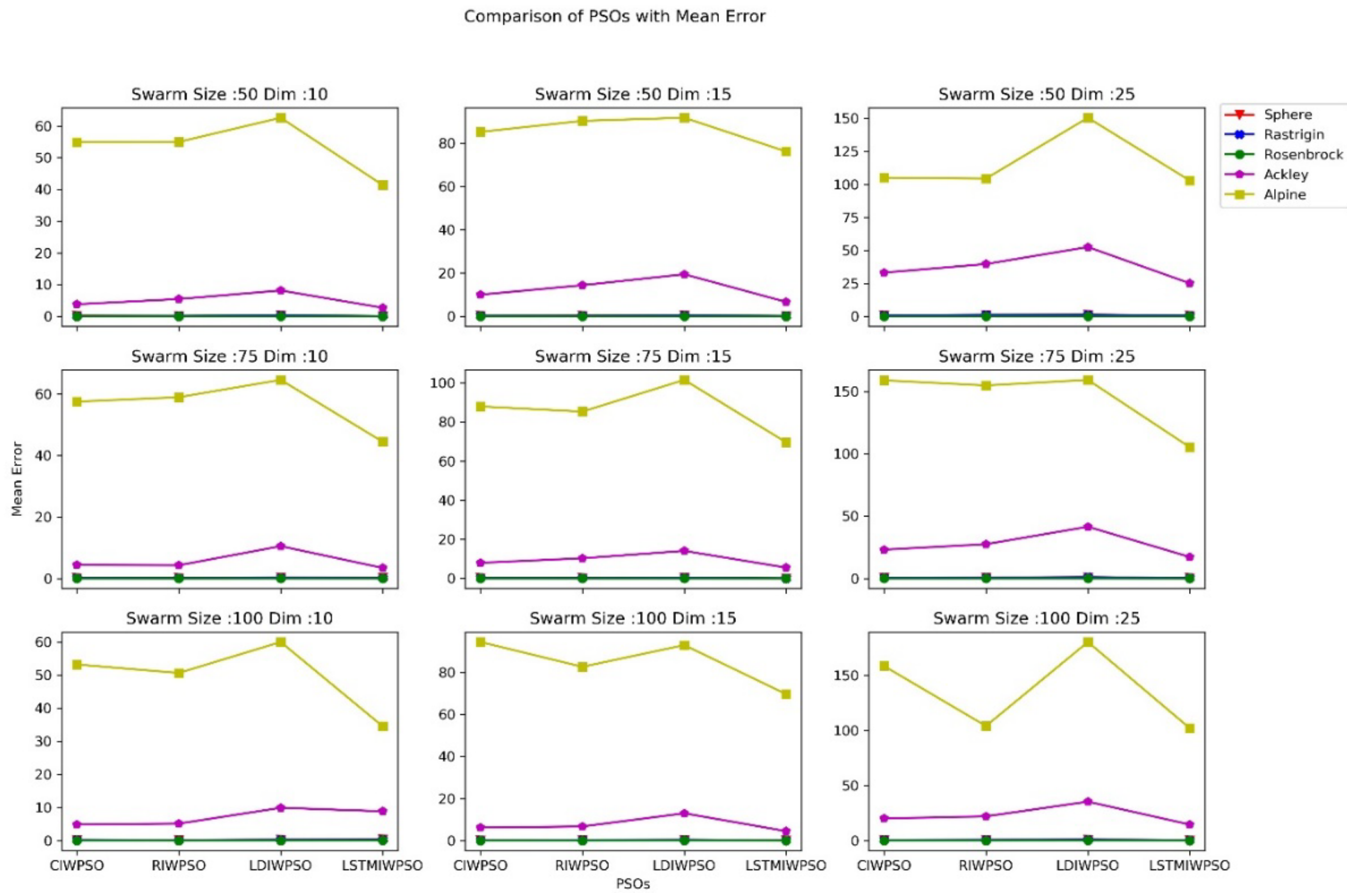

Fig. 5. Mean Error Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.

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

Swarm Size	**Dimension**	**BMF**	**PSOs**			
			CIWPSO	**RIWPSO**	**LDIWPSO**	**LSTMIWPSO**
50	10	f1	2.24626E+03	3.38851E+03	4.67975E+03	**9.63606E+02**
		f2	3.70666E+03	3.69348E+03	4.13007E+03	**2.80457E+03**
		f3	1.62064E+00	2.07616E+00	5.02929E+00	**9.41880E-01**
		f4	1.20428E-01	6.00913E-02	1.40112E-01	**4.69751E-02**
		f5	3.80843E+00	2.25555E+00	5.23459E+00	**9.06364E-01**
	15	f1	1.03989E+04	1.26799E+04	1.72068E+04	**3.34864E+03**
		f2	1.07078E+04	1.22065E+04	1.20338E+04	**8.28701E+03**
		f3	6.12127E+00	5.95316E+00	1.12496E+01	**2.54517E+00**
		f4	1.53739E-01	1.85184E-01	2.84031E-01	**5.52154E-02**
		f5	6.13349E+00	7.48563E+00	9.61024E+00	**2.78097E+00**
	25	f1	5.01816E+04	4.97107E+04	7.48609E+04	**1.94431E+04**
		f2	3.92425E+03	4.12939E+03	3.72553E+04	**2.44186E+03**
		f3	2.79965E+01	2.78540E+01	4.39840E+01	**1.18317E+01**
		f4	6.95780E-01	8.72313E-01	1.18338E+00	**3.70447E-01**
		f5	3.28743E+01	3.26392E+01	4.72444E+01	**1.44918E+01**
75	10	f1	2.09251E+03	2.13522E+03	5.23097E+03	**1.20472E+03**
		f2	3.88104E+03	3.57230E+03	4.06253E+03	**2.88180E+03**
		f3	3.10357E+00	1.88301E+00	3.61849E+00	**1.57710E+00**
		f4	1.11167E-01	6.93797E-02	1.36452E-01	**6.08377E-02**
		f5	3.25187E+00	**1.78081E+00**	4.39770E+00	2.39049E+00
	15	f1	7.28370E+03	8.33270E+03	1.39262E+04	**3.36059E+03**

		f2	1.02799E+04	9.87329E+03	1.21953E+04	**8.51558E+03**
		f3	5.22510E+00	6.01068E+00	8.47051E+00	**1.76031E+00**
		f4	1.86551E-01	1.57682E-01	2.49966E-01	**4.57045E-02**
		f5	4.41418E+00	5.74053E+00	8.20269E+00	**2.02216E+00**
	25	f1	3.80660E+04	3.68687E+04	5.74084E+04	**1.39583E+04**
		f2	3.84260E+04	3.80308E+04	3.66169E+04	**4.34064E+03**
		f3	2.33483E+01	1.93322E+01	3.67910E+01	**9.85238E+00**
		f4	6.57021E-01	6.94181E-01	1.00068E+00	**2.66049E-01**
		f5	2.15837E+01	2.26723E+01	3.76093E+01	**1.01227E+01**
100	10	f1	**2.16477E+03**	2.46721E+03	5.04084E+03	2.69153E+03
		f2	2.82820E+03	3.13702E+03	3.16738E+03	**2.24332E+03**
		f3	2.69688E+00	**1.48920E+00**	3.48140E+00	2.32310E+00
		f4	1.10163E-01	**5.92556E-02**	1.15670E-01	8.74897E-02
		f5	3.43568E+00	**2.10168E+00**	3.99214E+00	2.99379E+00
	15	f1	5.89145E+03	6.23226E+03	1.23640E+04	**2.40498E+03**
		f2	1.05402E+04	8.93997E+03	1.02147E+04	**8.05012E+03**
		f3	5.26484E+00	6.10921E+00	8.28945E+00	**1.41493E+00**
		f4	1.94574E-01	1.36136E-01	3.09916E-01	**5.09724E-02**
		f5	5.05269E+00	4.93674E+00	7.31503E+00	**1.41872E+00**
	25	f1	2.96978E+04	3.21262E+04	4.78574E+04	**1.25404E+04**
		f2	3.95934E+04	3.14574E+03	4.17462E+04	**1.55817E+03**
		f3	1.95162E+01	2.34042E+01	3.57999E+01	**7.60381E+00**
		f4	5.05917E-01	6.94567E-01	8.81406E-01	**2.23919E-01**
		f5	2.10448E+01	2.21517E+01	3.04020E+01	**8.26956E+00**

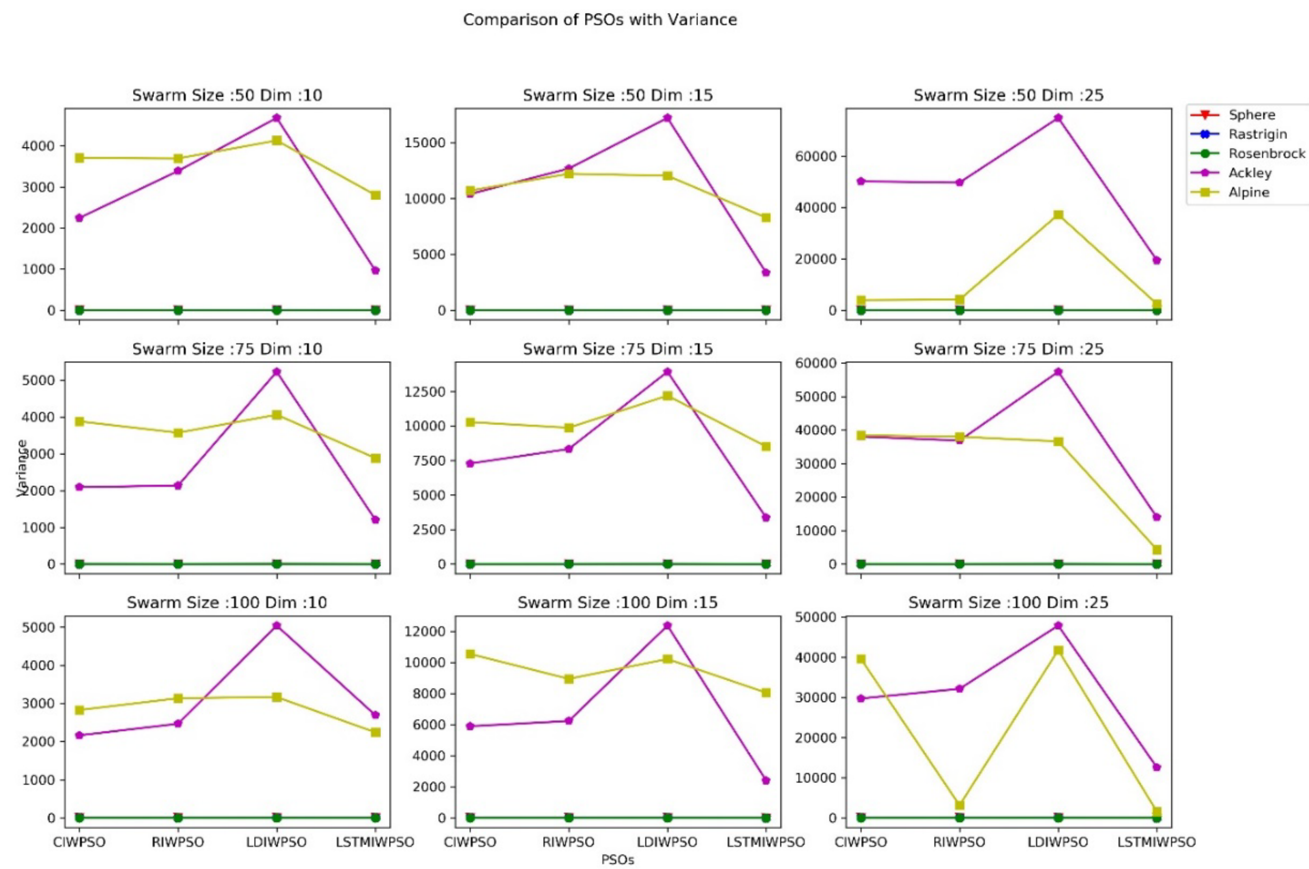

Fig. 6. Variance Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.

TABLE V. COMPUTED STANDARD DEVIATION FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS. (FIG. 7)

Swarm Size	**Dimension**	**BMF**	**PSOs**			
			CIWPSO	**RIWPSO**	**LDIWPSO**	**LSTMIWPSO**
50	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**
	15	f1	101.974922	112.604830	131.174709	**57.867444**
		f2	103.478490	110.482926	109.698532	**91.033028**
		f3	2.474120	2.439909	3.354034	**1.595360**
		f4	0.392095	0.430330	0.532945	**0.234980**
		f5	2.476588	2.735988	3.100039	**1.667625**
	25	f1	224.012548	222.959056	273.607132	**139.438498**
		f2	62.643794	64.260301	193.016188	**49.415159**
		f3	5.291170	5.277686	6.632043	**3.439728**
		f4	0.834135	0.933977	1.087831	**0.608644**
		f5	5.733607	5.713069	6.873457	**3.806805**
75	10	f1	45.743919	46.208445	72.325443	**34.709071**
		f2	62.297977	59.768716	63.737955	**53.682396**
		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
	15	f1	85.344604	91.283633	118.009469	**57.970561**
		f2	101.389839	99.364403	110.432280	**92.279921**
		f3	2.285847	2.451669	2.910414	**1.326765**
		f4	0.431915	0.397092	0.499966	**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	10	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	111.193549	**49.040620**
		f2	102.665394	94.551437	101.067981	**89.722486**
		f3	2.294524	2.471681	2.879141	**1.189510**
		f4	0.441106	0.368965	0.556701	**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	0.833407	0.938832	**0.473201**
		f5	4.587458	4.706557	5.513798	**2.875684**

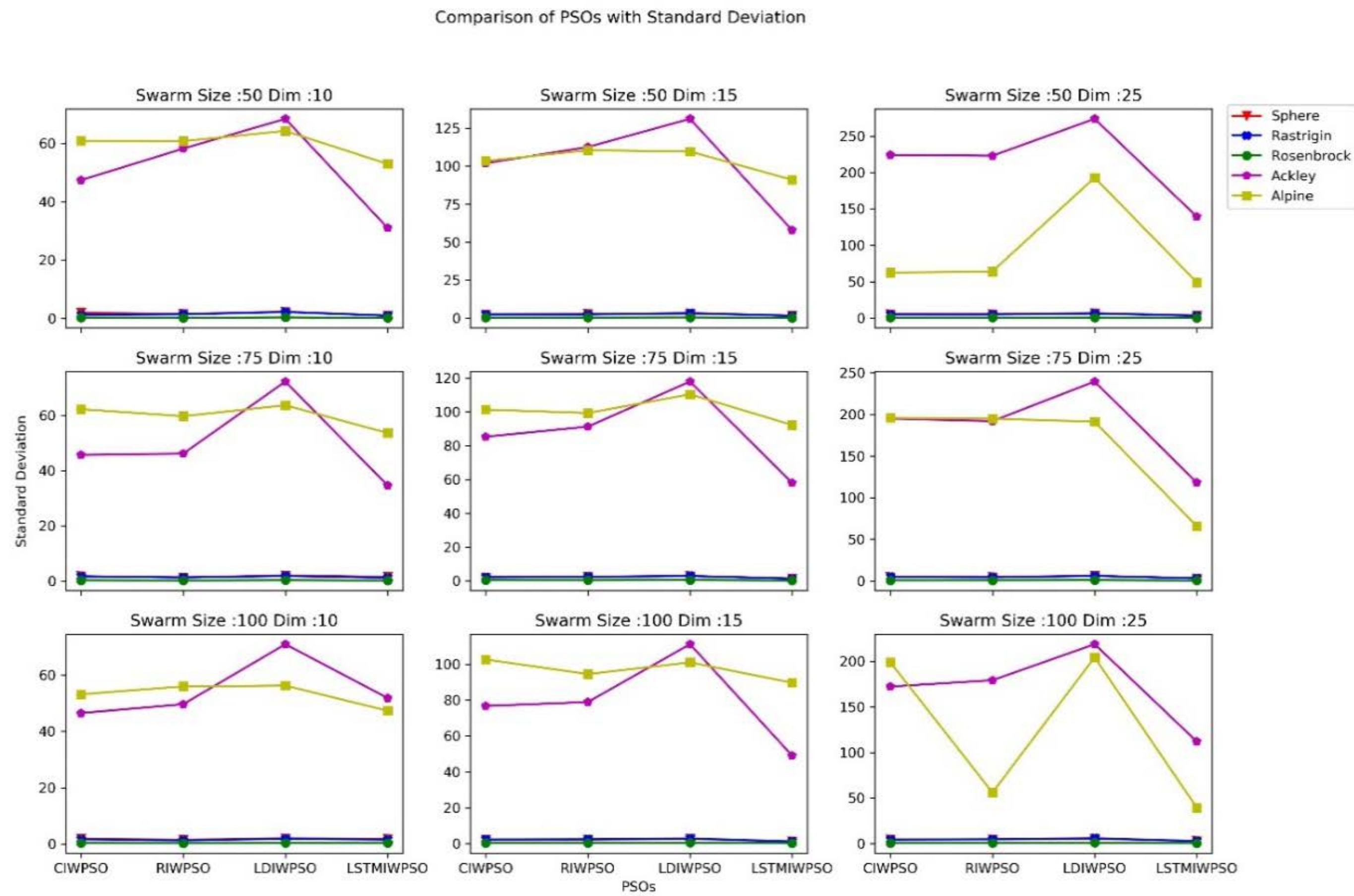

Fig. 7. Standard Deviation Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.

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**
50	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**
	15	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	**5.59937E-02**
		f5	6.20509E+00	7.62717E+00	9.81795E+00	**2.81262E+00**
	25	f1	5.12787E+04	5.12853E+04	7.76228E+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	9.05091E-01	1.22516E+00	**3.82671E-01**
		f5	3.35760E+01	3.36012E+01	4.90412E+01	**1.49388E+01**
75	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**

	15	f1	7.34733E+03	8.43899E+03	1.41247E+04	**3.39126E+03**
		f2	1.78875E+04	1.70372E+04	2.23337E+04	**1.32933E+04**
		f3	5.26664E+00	6.08052E+00	8.60735E+00	**1.77677E+00**
		f4	1.88610E-01	1.59751E-01	2.54270E-01	**4.61430E-02**
		f5	4.45377E+00	5.82035E+00	8.33757E+00	**2.04109E+00**
	25	f1	3.85996E+04	3.76240E+04	5.91306E+04	**1.42559E+04**
		f2	6.33172E+04	6.16264E+04	6.16239E+04	**1.54260E+04**
		f3	2.37094E+01	1.98317E+01	3.79563E+01	**1.00639E+01**
		f4	6.66511E-01	7.14074E-01	1.03177E+00	**2.71517E-01**
		f5	2.18925E+01	2.33039E+01	3.87337E+01	**1.03377E+01**
100	10	f1	**2.18834E+03**	2.49337E+03	5.13868E+03	2.76904E+03
		f2	5.61246E+03	5.65115E+03	6.70557E+03	**3.40020E+03**
		f3	2.74221E+00	**1.50622E+00**	3.55665E+00	2.41426E+00
		f4	1.12781E-01	**6.01845E-02**	1.18192E-01	9.23756E-02
		f5	3.48875E+00	**2.12554E+00**	4.08539E+00	3.12512E+00
	15	f1	5.92962E+03	6.27715E+03	1.25321E+04	**2.42506E+03**
		f2	1.93249E+04	1.56617E+04	1.87256E+04	**1.28186E+04**
		f3	5.31015E+00	6.18042E+00	8.41942E+00	**1.42551E+00**
		f4	1.96993E-01	1.37877E-01	3.16160E-01	**5.15856E-02**
		f5	5.09126E+00	4.98564E+00	7.43683E+00	**1.42958E+00**
	25	f1	3.01000E+04	3.26109E+04	4.91078E+04	**1.27530E+04**
		f2	6.43586E+04	1.39835E+04	7.37897E+04	**1.19835E+04**
		f3	1.97562E+01	2.39007E+01	3.66957E+01	**7.75062E+00**
		f4	5.12015E-01	7.07306E-01	9.03942E-01	**2.27478E-01**
		f5	2.12992E+01	2.25742E+01	3.13459E+01	**8.41840E+00**

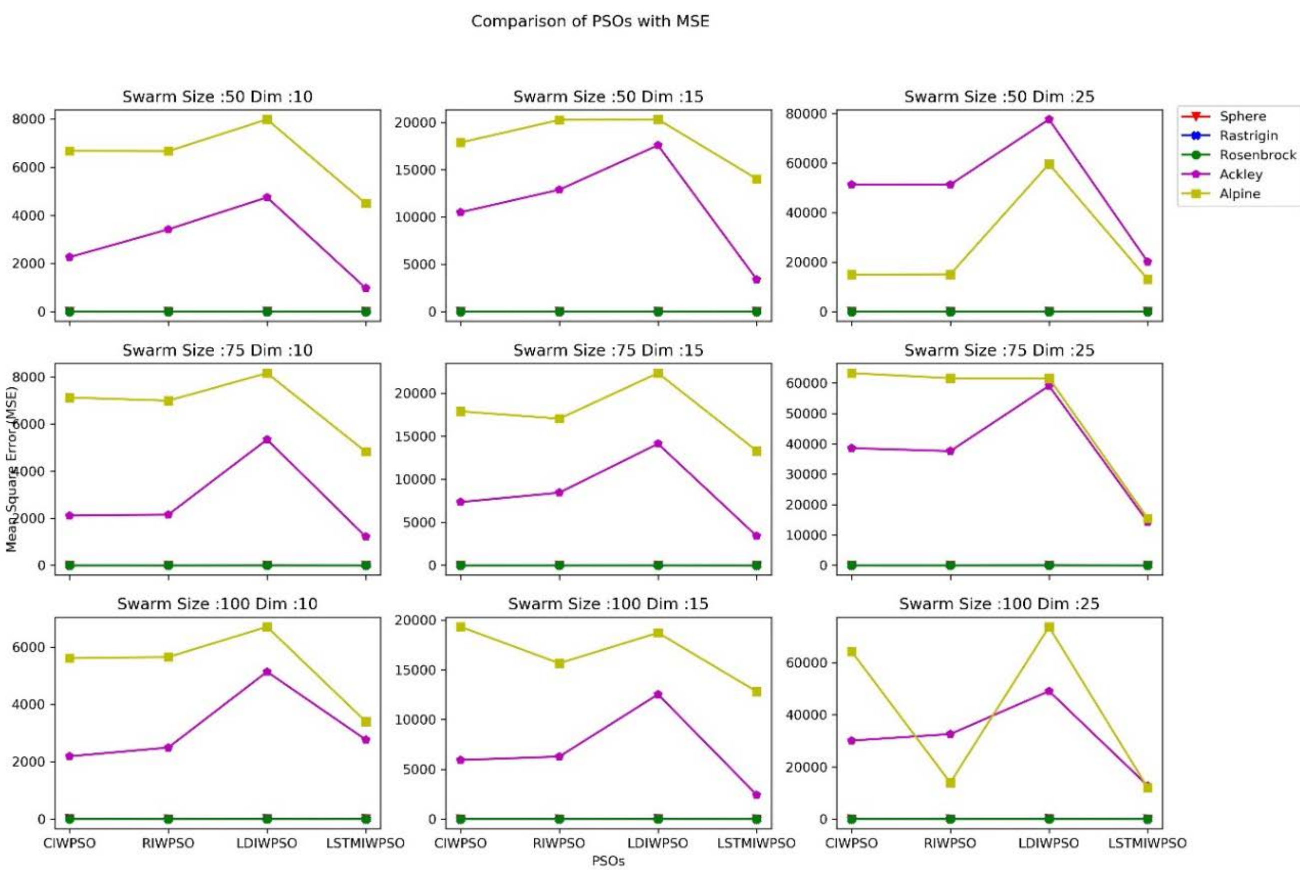

Fig. 8. MSE Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.

TABLE VII. COMPUTED ROOT MEAN SQUARED ERROR (RMSE) FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS. (FIG. 9)

Swarm Size	**Dimension**	**BMF**	**PSOs**			
			CIWPSO	**RIWPSO**	**LDIWPSO**	**LSTMIWPSO**
50	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**
	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	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	0.842342	0.951363	1.106869	**0.618604**
		f5	5.794484	5.796650	7.002942	**3.865068**
75	10	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
	15	f1	85.716591	91.863986	118.847212	**58.234483**
		f2	133.744088	130.526763	149.444504	**115.296759**
		f3	2.294917	2.465871	2.933828	**1.332957**
		f4	0.434293	0.399688	0.504252	**0.214809**
		f5	2.110395	2.412541	2.887486	**1.428668**
	25	f1	196.467698	193.969044	243.167928	**119.398100**
		f2	251.628992	248.246634	248.241652	**124.201480**
		f3	4.869227	4.453277	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
	15	f1	77.004050	79.228452	111.947053	**49.244864**
		f2	139.013944	125.146853	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**
	25	f1	173.493486	180.584992	221.602772	**112.929011**
		f2	253.690037	118.251843	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**

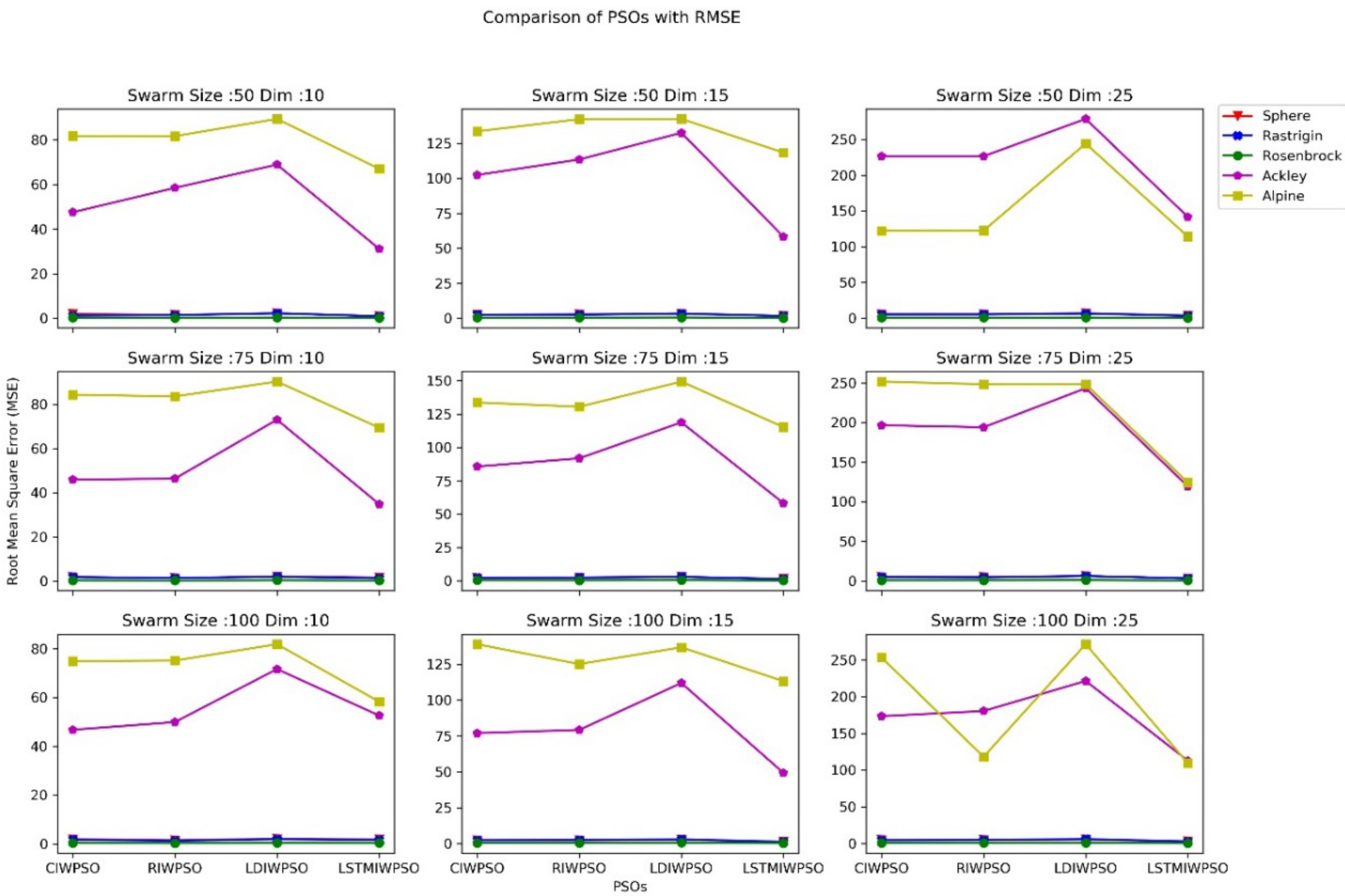

Fig. 9. RMSE Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.

TABLE VIII. Computed Mean Time (in Seconds) for PSOs with Respect to different Swarm Sizes and Dimensions. (Fig. 10)

Swarm Size	**Dimension**	**BMF**	**PSOs**			
			CIWPSO	**RIWPSO**	**LDIWPSO**	**LSTMIWPSO**
50	10	f1	3.312812	**3.183287**	3.589538	7.459693
		f2	0.021386	0.025384	**0.020334**	0.053375
		f3	2.257866	3.293823	1.**886251**	5.159045
		f4	1.**138227**	2.282051	1.398866	2.658974
		f5	1.**624259**	3.076825	2.082976	4.621895
	15	f1	4.306329	**4.294003**	4.869144	9.409708
		f2	0.032513	0.035312	**0.026717**	0.089734
		f3	4.**308127**	4.373820	5.000313	9.357706
		f4	4.**368624**	4.424788	4.412930	9.501312
		f5	4.**314457**	4.375219	4.511282	9.428390
	25	f1	6.178701	**6.145023**	6.274644	13.369280
		f2	0.461714	0.442326	**0.055366**	1.930295
		f3	6.135661	6.**082227**	6.220433	13.182448
		f4	6.**006940**	6.069369	6.115273	13.603162
		f5	6.129612	6.086691	6.**078426**	13.568970
75	10	f1	3.273303	4.050883	2.**842036**	5.982660
		f2	0.026450	0.026450	0.**022253**	0.059991
		f3	1.**855718**	3.775724	2.897390	2.032672
		f4	1.**208450**	2.102096	2.027942	1.292319
		f5	1.956389	2.852364	2.153064	**1.722074**

	15	f1	6.526218	6.**518706**	6.866313	13.257469
		f2	0.041508	0.038976	**0.030914**	0.117080
		f3	6.**276307**	6.611565	6.669305	13.423912
		f4	4.**794559**	6.383440	5.537492	12.628298
		f5	6.**380909**	6.439272	6.652887	13.195447
	25	f1	8.**987696**	9.227687	9.441670	19.331790
		f2	0.073887	0.077486	**0.063161**	1.549432
		f3	9.**008012**	9.178240	9.021073	19.068662
		f4	9.058448	9.075037	8.**982564**	19.274107
		f5	9.163049	9.168713	9.**048472**	19.067931
100	10	f1	3.016330	4.171748	3.530148	2.**414135**
		f2	0.032180	0.033646	0.**028782**	0.058631
		f3	1.819471	3.799044	2.733838	1.**586756**
		f4	1.574290	2.531497	2.621043	1.**241292**
		f5	2.020080	3.923700	2.951757	1.**423464**
	15	f1	9.002749	8.**977202**	9.125075	17.114905
		f2	0.**046437**	0.049836	0.046704	0.112377
		f3	7.**197202**	8.082389	7.425860	18.320117
		f4	4.**608342**	8.081904	5.849772	13.539670
		f5	7.656653	**7.275687**	8.531872	17.034604
	25	f1	12.429958	**12.134209**	12.552006	24.936847
		f2	0.091344	0.841345	**0.070490**	3.426473
		f3	12.091566	12.138803	**12.043132**	25.225323
		f4	12.340614	13.969420	**12.167084**	27.915432
		f5	**11.938530**	12.319092	12.264439	24.890808

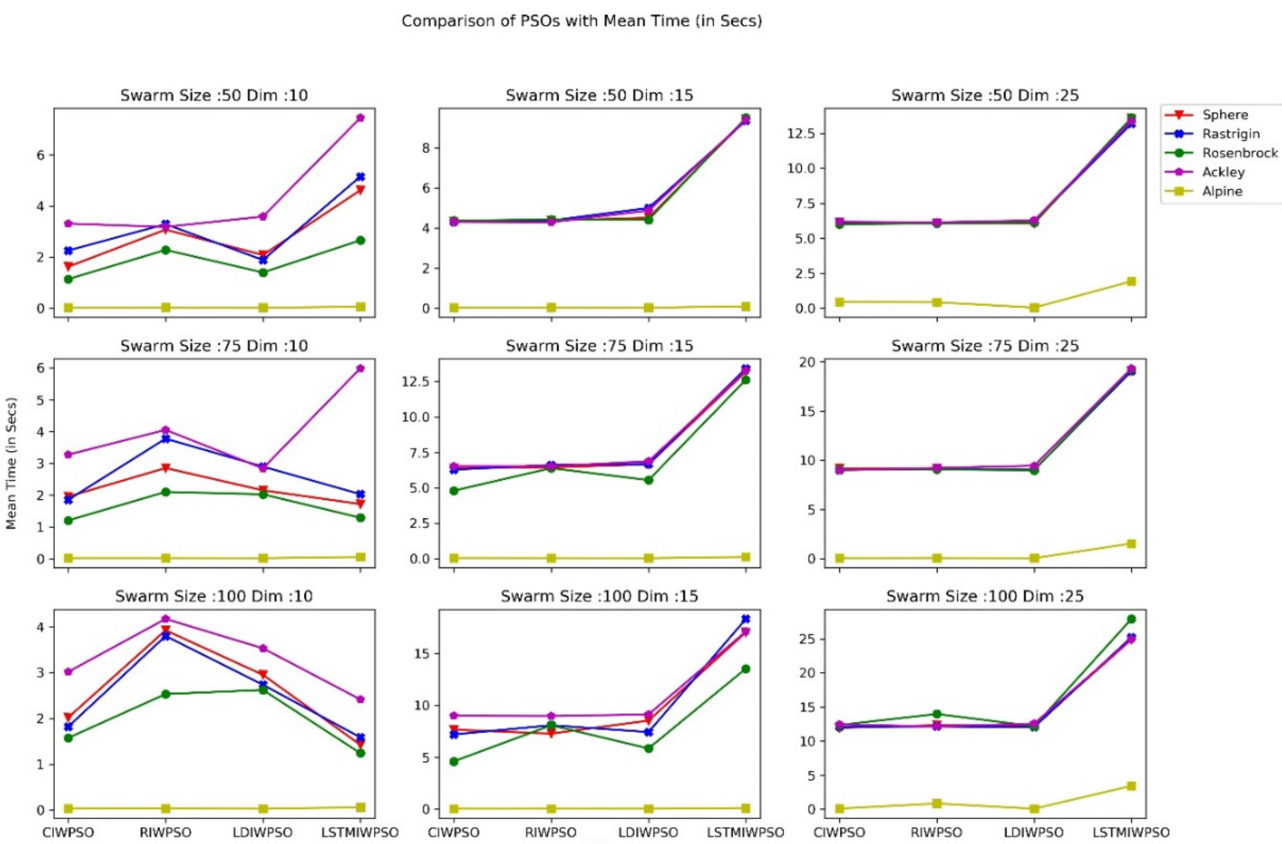

Fig. 10. Mean Time (In Secs) Computed for the Swarm Size of 50, 75, and 100 with Dimensions 10, 15 and 25.

TABLE IX. MEAN ITERATIONS FOR PSOS WITH RESPECT TO DIFFERENT SWARM SIZES AND DIMENSIONS. (FIG. 11)

Swarm Size	Dimension	BMF	PSOs			
			CIWPSO	**RIWPSO**	**LDIWPSO**	**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
	15	f1	**1000.00**	**1000.00**	**1000.00**	**1000.00**
		f2	6.60	7.67	**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**
	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**
	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
	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**
	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
	25	f1	**1000.00**	**1000.00**	**1000.00**	**1000.00**
		f2	**6.87**	72.93	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**

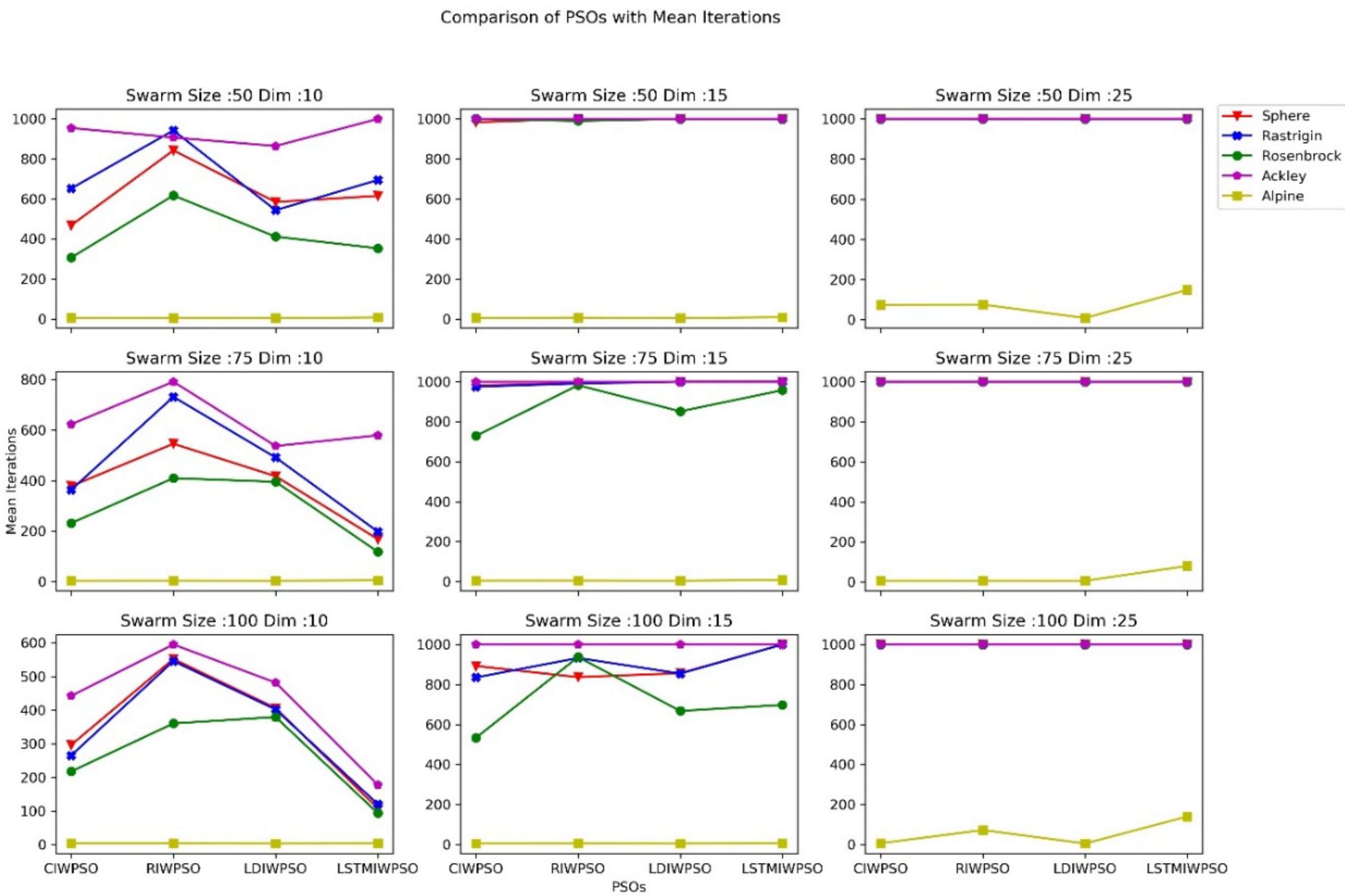

Fig. 11. Mean Iterations Computed for the Swarm size of 50, 75, and 100 with Dimensions 10, 15 and 25.

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. Anchorange, 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 & Iemma, Umberto & Campana, Emilio & Fasano, Giovanni & Diez, Matteo. (2016). Parameter 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 Marini, Beata Walczak, 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] Afshine Amidi, 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., Agees Kumar, C., Jarin, T., Krishnamoorthy, S., Malarvizhi 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, ICISS 2019, Vol: , Issue: , pp: 29 - 34, 2019, DoI :10.1109/ISS1.2019.8908041.

[35] Bewoor, L.A., Chandraprakash, V., Sapkal, S.U., Evolutionary hybrid particle swarm optimization algorithm to minimize makespan toschedule 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/V12SP3/20201312.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.