PSO with new initialization approach for solving global optimization problem

Waqas Haider Bangyal
Member IEE SMC
Department of Computer Science
University of Gujrat, Gujrat, Pakistan.
waqas_bangyal@hotmail.com

Hafiz Tayyab Rauf
Department of Computer Science
University of Gujrat, Gujrat, Pakistan
hafiztayyabrauf093@gmail.com

Jamil Ahmad
Senior Member IEEE, Professor Computer Science
Department of Computer Science
Kohat University of Science and Technology (KUST)
Kohat, Pakistan
jamil@ieee.org

Saad Abdullah Bangyal
Department of Computer Science
Abasyn University, Islamabad Campus, Pakistan
saadbangyal@gmail.com

***Abstract*—Particle Swarm optimization (PSO) is a nature-inspired metaheuristic algorithm, which is widely used to solve the real world global optimization problem. PSO has been mostly used to resolve diverse kind of optimization problems. Major issues faced by the PSO are lack of diversity and frequently captured in local optima while handling the complex real-world problems. Initialization of population plays a significant job in metaheuristic algorithm since they can influence on convergence, diversity and find the better final solution. In this study, to improve the convergence, rather applying random distribution for initialization, a new distribution is proposed for initialization of swarm. This paper presents a new initialization population approach using Log-logistic named as (LOG-PSO) that uses the Log-logistic to create the initialization of the swarm. Initializing PSO using Log-logistic is examined on 8 well-known non-linear benchmark test problems extensively used in the literature and its promising performance is analyzed and compared with basic PSO, PSO initialized with Sobol sequence (SO-PSO) and PSO initialized with Halton sequences (HA-PSO). The promising experimental result suggests the superiority of the proposed technique. The results present foresight that how the proposed initialization technique influences on divergence and convergence speed.**

Keywords— Particle swarm optimization, Quasi random sequence, Log-logistic distribution, Swarm Convergence.

I. INTRODUCTION

Optimization is a process that implies to sort out the problems from given condition. The critical task of an optimization process is to increase advantages of engineering systems and to decrease the wastage of time. Every system, which is considered for optimization must have objective function and many variables for the decision that is used for an influence of function. An optimization technique is a procedure to obtain an optimum solution, by satisfying the objective functions [1].

A stochastic algorithm, one type of algorithm that belongs to optimization technique. It is always preferred to solve the multi-modal benchmark functions, because, it can easily tackle with local minima [2]. Furthermore, stochastic algorithms also called meta-heuristic algorithms [3] are used for solving deferent types of classification and clustering problems. Swarm algorithm that is a subtype of meta-heuristic algorithms, implemented to find global optimum inspired by the collective behavior of swarm that they show during any task. [4]. A variety of literature is based on such meta-heuristics algorithms that are broadly used to solve optimization problems.

Particle swarm optimization (PSO) is population based evolutionary algorithm firstly proposed by Kenndy and Eberhart in 1995 [5]. PSO is used to solve diverse kind of optimization problems as well as real-world classification problems. In PSO the swarm moves towards food in multidimensional entire search space. Initialization of swarm into the search space plays important role in PSO to find the vector solution using predefined ornament. Selection of robust distribution for population initialization is difficult task. If the search space is completely covered by the initial particles then there is greater chance to found the global best position in less number of iterations. The standard PSO used random pattern that follows uniform distribution to choose the starting location of particles that are not much reasonable and robust.

Many researchers are still confused about the convergence and diversity of PSO for the same curve, which becomes a major issue for them. They are supposed to be pointing out those parameters that cause convergence and diversity in the swarm. The premature convergence of PSO intensely affects the performance of the algorithm [6]. Due to premature convergence, the algorithm gets stuck into the local minima before finding the global optimum solution. Population initialization is one of the major parameters that affect PSO performance. The convergence rate and the diversity of the swarm can be enhanced by selecting the most appropriate distribution for population initialization.

In this paper, we proposed a novel distribution called log-logistic distribution for generating random numbers in population initialization process. The proposed Log-logistic based PSO (LL-PSO) based on new method of population initialization is compared with the simple PSO that based on uniformly distributed population initialization, the Sobol based population initialization (SO-PSO) and the Halton based population initialization HA-PSO. Sobol and Halton sequences are quasi random number generator having low discrepancy of density function. The experimental results reveal that the proposed LL-PSO outperforms the Simple PSO, SO-PSO and

H-PSO in terms of enhancing PSO diversity and convergence rate.

The rest of the paper is structured like this: Related work is presented in section II. Section III contains the general description of Standard PSO. Proposed methodology is described in Section IV. Results are presented in section V.Conclusion, and future work described in the section VI.

II. Related Work

Researchers are used to employing the different random number generator to initialize the swarm into the search area of multi-dimensions. To sharpen the performance of population-based stochastic evolutionary algorithms pseudo-random, quasi-random and probability sequences are one of the most famous distributions used in the process of choosing the initial configuration. Low discrepancy sequences have been compared with simple uniform distribution based PSO for distributing the particles into the random positions at the search area [7].Their investigations include only benchmark minimization function to compare the performance of various versions of low-discrepancy sequences.

An improved PSO based on Halton series was carried out by the authors for optimizing a genetic algorithm. They investigate the proposed PSO on functions optimization problems and proved their technique superior [8].A comparative analysis of Sobol, Halton, and Faure sequences has been carried out by the authors in [9].The numerical results declared the Sobol sequences winner for the initialization of the swarm. Vander Corput sequence was proposed by Vander Corput [10].it is also belongs to low-discrepancy sequences. Van der Corput sequences produced with the initial dimension $d = 1$ and base $b = 2$.The preliminary results have confirmed that Van der Corput outperforms the other quasi-random sequences, Faure sequence, Sobol sequence and Halton sequence. The family of probability sequences i.e. Beta distribution, Gaussian distribution, Exponential distribution and Cauchy distribution has been widely used for the same purpose, exponential distribution based PSO were used in [11] for the purpose of parameter tuning of PSO.

In [12] the extensive observation of probability distributions have been carried out to minimize several multi-modal functions. The numerical results show the better performance of exponential distribution as compared to Gaussian distribution and Beta distribution. In [13] the authors used NSM method (Non-Liner Simplex Method) for population initialization. The NSM incorporates the primary particles of a maximum of $(D + 1)$ where D represents the dimensions of the search space. Richards Ventura introduced the novel PSO initialization method based on Centroidal Voronoi tessellations (CVT) [14].In CVT method, it divides the whole search space into multiple blocks. Each particle occupies a position in the first section of blocks. Opposition-based initialization (O-PSO) was proposed by the authors in [15].The concept of opposition based-learning was incorporated in the PSO to initialize the swarm. O-PSO tends to increase the chances of reaching an optimal solution at the very early stage of the algorithm, because of some particles gets an initial place at opposite direction of the entire search space. To investigate a new search space O-PSO improves the heterogeneity of particles in reverse direction that is identical to the same direction.

Gutiérrez et al. [16], conducted the comparison of three different methods to initialize the swarm,he compared the orthogonal array initialization, the chaotic initialization, and the opposition-based initialization. In orthogonal array initialization a normal perturbation that is based on the collection of the variable, is estimated to select the initial location for particles. The working flow of chaotic initialization can be described by the following equation:

$$x_t = \omega . x_{t-1}(1 - x_{t-1})$$

Where x_{t-1} is current particle generated by chaotic distribution.

III. Particle Swarm Optimization

In the field of advanced technology, PSO is a global optimizing technique that rivals a significant role and has been broadly employed in numerous real-life application of engineering and technology such as communication networks, machine learning, Information Security, machine learning, power allocation of cooperative systems, data mining, heating system planning and pattern recognition.PSO works at the application of candidates solution. In order to reach the optimal solution, each candidate depicted the optimal solution that is termed as the particle. The current position of each particle is represented by a vector solution x depicts by an n-dimensional search area. The individual solution of each particle is translated in the form of the fitness value given by the particles at each iteration t on dimensions d. Each particle p provokes a new position vector x in n -dimensional search space. In addition position p, velocity v can be determined by the step measurement of an entire swarm and the action of particles in the search space.

For seeking the optimal solution, the population consists of n particles that travel in the iteration t in n-dimensional search space. The objective function of the candidate solution may be turned by the mutation of the swarm. Eq.1 and eq.2 are employed for updating the position vector and velocity vector of the particles.

$$\mathrm{v}_{z+1}^{t} = \mathrm{v}_{z}^{t} + \alpha_1 \times (\mathrm{p}_{\mathrm{best}} - \mathrm{x}_{z}^{t}) + \alpha_2 \times (\mathrm{g}_{\mathrm{best}} - \mathrm{x}_{z}^{t}) \quad (1)$$

$$\mathrm{x}_{z+1}^{t} = \mathrm{x}_{z}^{t} + \mathrm{v}_{z+1}^{t} \quad (2)$$

v_{z}^{t} and x_{z}^{t} denotes the position vector and velocity vector respectively, where $\mathrm{p}_{\mathrm{best}}$ designating the local best fitness solution of intact swarm enlarged by utilizing its own preceding knowledge and $\mathrm{g}_{\mathrm{best}}$ describes the global best fitness solution of intact swarm obtained by utilizing the experience of its neighbor. In eq.1 α_1 and α_2 are described as $c1r1$ and $c2r2$ respectively.PSO need acceleration factors to adjust its weight that acceleration factors are represented by $c1$and $c2$,$r1$ and $r2$ are two random number that follows simple uniform distribution. x_{z+1}^{t} denotes as a new position vector, where v_{z+1}^{t} is an updated velocity directing the new velocity for the current particle at iteration t. We can observe three basic factors from Eq.1. v_{z}^{t} is called momentum factor which is the old velocity of particles, where $\alpha_1 \times (\mathrm{p}_{\mathrm{best}} - \mathrm{x}_{z}^{t})$ is termed as cognitive which is the current best fitness

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

chosen from all previous best fatnesses, $\alpha_2 \times (g_{best} - x_z^t)$ is the social factor that exhibits global best fitness, raised from the entire neighbour particles. In Algorithm 1 the pseudo code for basic PSO is presented.

Algorithm_1: Basic_PSO_Algorithm

Require: $p_z = (p_1, \ldots p_n)^t$
Ensure: x_z^{best} ; optimal solution
(1) **for** $i = 1 \ldots p_n$ **do**
(2) **for** $j = 1 \ldots d_n$ **do**
(3) $x_z^t = Rand(0,1)$
(4) $v_z^t = Rand(0,1)$
(5) **if** x_z^t reaches to p_{best} **then**
(6) Copy p_{best} into x_z^t
(7) **end**
(8) g_{best}=Optimum of all p_{best}
(9) Repeat until k_z
(10) **end for**
(11) **end for**
(12) **for** $i = 1 \ldots p_n$ **do**
(13) Compute v_{z+1}^t using eq.1
(14) Compute x_{z+1}^t using eq.2
(15) **if** $x_{z+1}^t > p_{best}$ **then**; generate new local solution
(16) $p_{best} = x_{z+1}^t$
(17) **end if**
(18) **if** $x_{z+1}^t > g_{best}$; **then** generate new global solution
(19) $g_{best} = x_{z+1}^t$
(20) **end**
(21) **end fo**r

IV. METHDOLOGY

The goal of this paper is to investigate the robustness of the suggested distribution Log-logistic based PSO (LL-PSO). Basic PSO used a simple uniform distribution that is pattern less and much random in nature. In basic PSO there is no any standard pattern that ensures the optimal solution. By taking the advantage of this random nature we have proposed a novel distribution Log-logistic based PSO that is employed to assigning initial locations to the particles. For improving the superiority of proposed technique we compare LL-PSO with basic PSO that follows the uniform distribution, HA-PSO based on Halton distribution and SO-PSO that follows Sobol distribution.

A. Random Number Generator

Inbuilt library of C++ is the most famous way of generating the random numbers [17] over the interval of $Rand(0,1)$.The impact of randomness on uniform distribution can be identified by the density function of uniform distribution which is given as:

$$f(d) = \begin{cases} \frac{1}{i-j} \; for \; i < d < j \\ 0 \quad for \; d < i \; or \; d > i \end{cases} \tag{3}$$

In eq.3 i and j are the two parameters of maximum likelihoods.

B. The Sobol Sequence

The Sobol distribution is most widely used distribution of quasi-random family, it was introduced by the Russian mathematician named Sobol in 1967 [18] in their work of reconstructing the coordinates. The liner recurrence for non negative instance to generate Sobol sequence can be found in the following equation.

$$n = n_1 2^0 + n_2 2^1 + n_3 2^2 + \cdots + n_x 2^{x-1} \tag{4}$$

Where n is random number in Kth itration.

C. The Halton Sequence

Halton sequence is similar to the Van Dar Corput sequence having different probability density functions. It was introduced by J.Halton [19] in 1964. The pseudo code of Halton sequence is given below:

Algorithm_2. Halton Sequence

Halton Sequences:
Initialize the sequence index i and base of $coprime \rightarrow$ b
// output: Halton points = HA
Configure the Parameter and Compute as:

I. $h_{min} = \min\_interval$
II. $h_{max} = \max\_interval$
III. For each iteration $k = 1 \ldots N_k$ do
IV. For each particle $= 1 \ldots p_n$ do
- $h_{max} = h_{max}/cp$
- $h_{min} = h_{min} + h_{max} * i \; mod \; b$
- $i = i/b$

Return HA

D. The Log-logistic Sequence

Log-logistic distribution also called Fisk distribution was proposed by the authors in [20] for non-negative instance referred to as continuous probability sequence. As a parametric paradigm, Log-logistic distribution is usually applied in survival analysis for functions whose frequency rises initially and clips at the end of the function. Log-logistic distribution belongs to the family of the probability distribution having the logarithm that is distributed logistically. It has larger tails as compared to the log-normal distribution. The probability density function for Log-logistic distribution can be determined by the following equation.

$$f(y, \alpha_1, \alpha_2) = \frac{\left(\frac{\alpha_2}{\alpha_1}\right)\left(\frac{y}{\alpha_1}\right)^{\alpha_2 - 1}}{\left(1 + \left(\frac{y}{\alpha_1}\right)^{\alpha_2}\right)^2} \tag{5}$$

α_1, α_2 represents the shape parameter and scale parameter respectively.

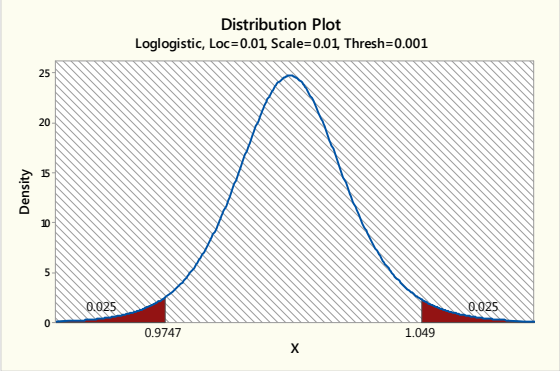

Fig. 1.Random data generation using Log-logistic distribution

Algorithm_3: Proposed_PSO_Algorithm

Require: $p_z = (p_1, \ldots p_n)^t$

Ensure: x_z^{best} ; optimal solution

(1) **for** $i = 1 \ldots p_n$ **do**
(2) **for** $j = 1 \ldots d_n$ **do**
(3) $x_z^t = Log - logistic(0,1)$
(4) $v_z^t = Rand(0,1)$
(5) **if** x_z^t reaches to p_{best} **then**
(6) Copy p_{best} into x_z^t
(7) **end**
(8) g_{best}=Optimum of all p_{best}
(9) Repeat until k_z
(10) **end for**
(11) **end for**
(12) **for** $i = 1 \ldots p_n$ **do**
(13) Compute v_{z+1}^t using eq.1
(14) Compute x_{z+1}^t using eq.2
(15) **if** $x_{z+1}^t > p_{best}$ **then**; generate new local solution
(16) $p_{best} = x_{z+1}^t$
(17) **end if**
(18) **if** $x_{z+1}^t > g_{best}$; **then** generate new global solution
(19) $g_{best} = x_{z+1}^t$
(20) **end**
(21) **end for**

V. Results

The suggested technique Log-PSO is executed on a machine with the specification of 1.7 GHz Core (TM) i3-4010U CPU processor and simulated in C++ programming language. For performance evaluation, 8 well known non-linear benchmark function has been taken for the comparison of LL-PSO with basic PSO, SO-PSO, and HA-PSO. These functions are considered as worldwide famous function and commonly carried out to check the performance of any evolutionary technique. Table I contains the description of standard benchmark functions. The parameter of PSO is fixed as $c1 = c2 = 1.45$ while the interval of inertia weight w is [0.9,0.4] and the population size is 40. In Table I, Domain is a search space size where f_{min} describing the global optimum value. All functions are implemented for 10 Runs while the maximum number of iterations is 3000.

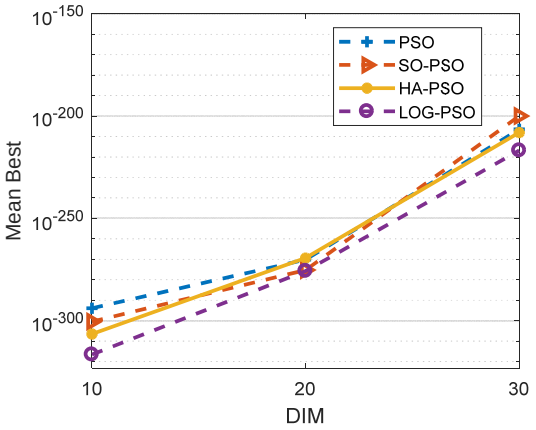

Fig. 2. Convergence Curves on f_1

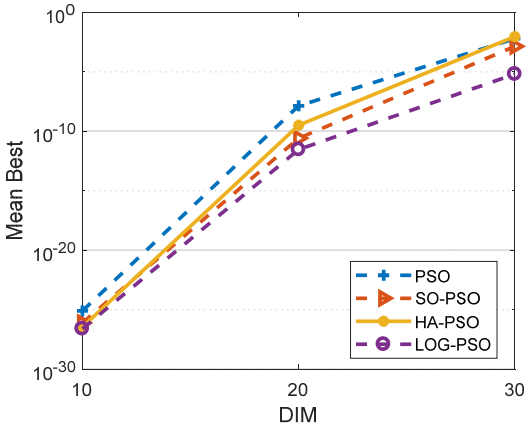

Fig. 3. Convergence Curves on f_2

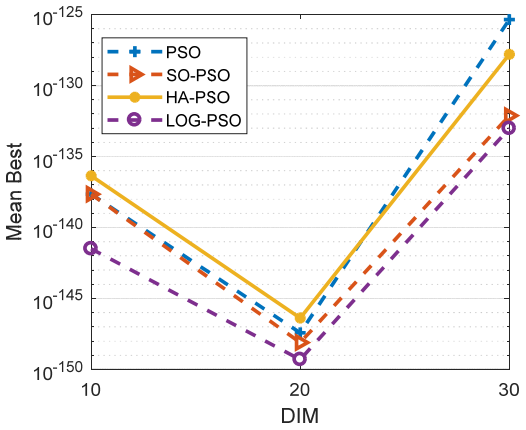

Fig. 4. Convergence Curves on f_3

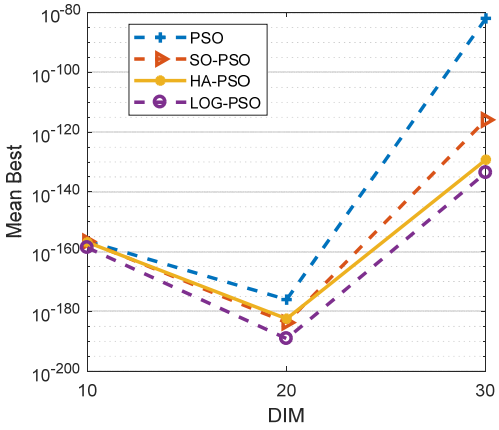

Fig. 5. Convergence Curves on f_4

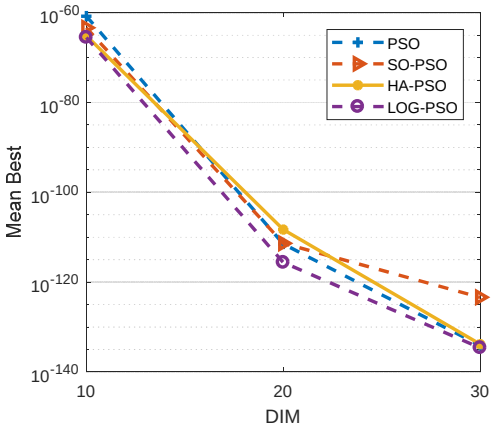

Fig. 6. Convergence Curves on f_5

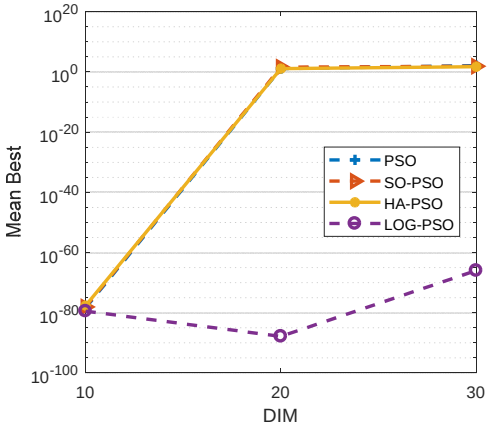

Fig. 7. Convergence Curves on f_6

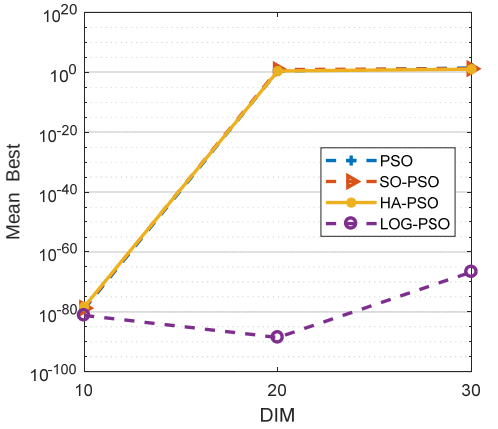

Fig. 8. Convergence Curves on f_7

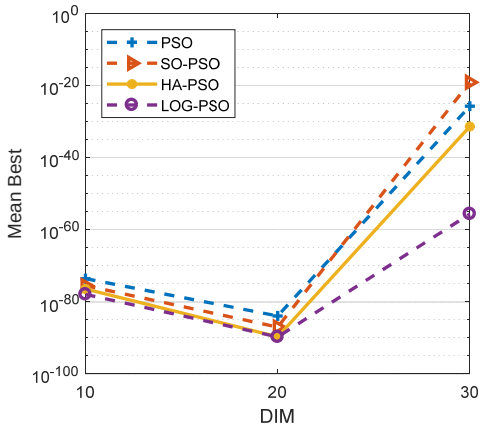

Fig. 9. Convergence Curves on f_8

A. *Discussion*

The goal of this study is to investigate the performance effect of proposed technique LL-PSO. For this, a comparison of proposed technique has been made with basic PSO that is based on the uniform distribution, SO-PSO based Sobol distribution and HA-PSO based on Halton distribution. Table II shows the experimental results. The result for each function is spotlighted in bold. From Table II we can see that the convergence rate of proposed technique LL-PSO is better than other PSO variant PSO, SO-PSO and HA-PSO respectively. Experimental results proved the superiority of the proposed technique.

VI. Conclusion

In This paper presents a novel technique for the initialization of population in the swarm. A set of non-linear benchmark test functions are utilized for experimental validation. The results reveal that using the proposed technique maintains the diversity of the swarm, improves the convergence speed, avoid from local minima and find the better region of the swarm. The results reveals that LL-PSO outperforms the standard PSO, PSO initialized with Sobol (SO-PSO) and PSO initialized with Halton sequences (HA-PSO) The core objective of this research is generic but applicable to other stochastic based metaheuristic algorithm

that develops future direction of our work and also to use this initialization technique with any mutation operator.

TABLE I. DESCRIPTION OF 8 BENCH MARK FUNCTIONS

SR#	Function Name	Objective Function	Domain	f_{min}	x^*
1	Schwefel 2.23	$Min\ f(x) = \sum_{i=1}^{n} x_i^{10}$	$-10 \leq x_i \leq 10$	0	(0,0,0...,0)
2	Schwefel 2.21	$Min\ f(x) = \underset{1<i<D}{max\lvert x_i \rvert}$	$-100 \leq x_i \leq 100$	0	(0,0,0...,0)
3	Schumer Steiglitz	$Min\ f(x) = \sum_{i=1}^{n} x_i^4$	$-5.12 \leq x_i \leq 5.12$	0	(0,0,0...,0)
4	Chung Reynolds	$Min\ f(x) = \left(\sum_{i=1}^{n} x_i^2\right)^2$	$-100 \leq x_i \leq 100$	0	(0,0,0...,0)
5	Sum of different power	$Min\ f(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	$-1 \leq x_i \leq 1$	0	(0,0,0...,0)
6	Moved axis parallel hyper-ellipsoid	$Min\ f(x) = \sum_{i=1}^{n} 5i.x_i^2$	$-5.12 \leq x_i \leq 5.12$	0	(5*i)
7	Axis parallel hyper-ellipsoid	$Min\ f(x) = \sum_{i=1}^{n} i.x_i^2$	$-5.12 \leq x_i \leq 5.12$	0	(0,0,0...,0)
8	Sphere	$Min\ f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \leq x_i \leq 5.12$	0	(0,0,0...,0)

TABLE II. COMPERISON OF LOG-PSO WITH PSO,SO-PSO AND HA-PSO

Functions	*Iter*	*DIM*	PSO		SO-PSO		HA-PSO		LOG-PSO	
			Mean	*Std. Dev*	*Mean*	*Std. Dev*	*Mean*	*Std. Dev*	*Mean*	*Std. Dev*
F1	**1000**	**10**	1.10E-294	0.00E+00	3.19E-301	0.00E+00	2.78E-307	0.00E+00	**2.57E-317**	0.00E+00
	2000	**20**	6.16E-271	0.00E+00	5.09E-276	0.00E+00	3.74E-270	0.00E+00	**1.71E-276**	0.00E+00
	3000	**30**	3.08E-207	0.00E+00	1.04E-200	0.00E+00	8.12E-209	0.00E+00	**1.46E-217**	0.00E+00
F2	**1000**	**10**	8.04E-26	2.41E-25	8.01E-27	2.40E-26	3.59E-27	1.08E-26	**2.54E-27**	**7.63E-27**
	2000	**20**	1.42E-08	4.26E-08	2.64E-11	7.93E-11	3.29E-10	9.86E-10	**2.99E-12**	**8.97E-12**
	3000	**30**	6.20E-03	1.86E-02	1.41E-03	4.23E-03	9.36E-03	2.81E-02	**6.55E-06**	**1.97E-05**
F3	**1000**	**10**	2.23E-138	2.23E-138	2.23E-138	3.15E-137	4.35E-137	1.31E-136	**3.16E-142**	**9.48E-142**
	2000	**20**	3.79E-148	1.14E-147	7.87E-149	2.36E-148	4.19E-147	1.26E-146	**4.97E-150**	**1.49E-149**
	3000	**30**	4.43E-126	1.33E-125	7.52E-133	2.26E-132	1.57E-128	4.71E-128	**9.42E-134**	**2.83E-134**
F4	**1000**	**10**	2.96E-157	8.87E-157	2.39E-157	7.18E-157	1.28E-157	3.84E-157	**2.07E-159**	**6.21E-159**
	2000	**20**	8.79E-177	0.00E+00	1.77E-184	0.00E+00	3.49E-183	0.00E+00	**7.33E-190**	**0.00E+00**
	3000	**30**	1.23E-82	3.68E-82	1.25E-116	3.74E-116	5.99E-130	5.99E-130	**2.45E-134**	**7.35E-134**
F5	**1000**	**10**	1.70E-61	5.11E-61	4.45E-64	1.33E-63	7.29E-66	2.19E-65	**3.07E-66**	**9.21E-66**
	2000	**20**	3.25E-112	9.74E-112	4.39E-112	1.32E-111	5.01E-109	1.50E-108	**2.41E-116**	**7.23E-116**
	3000	**30**	7.21E-135	2.16E-134	4.10E-124	1.23E-123	1.51E-134	4.54E-134	**2.36E-135**	**7.07E-135**
F6	**1000**	**10**	4.35E-79	1.30E-78	8.95E-79	2.69E-78	2.43E-78	7.30E-78	**3.38E-80**	**1.02E-79**
	2000	**20**	1.31E+01	3.93E+01	3.93E+01	1.18E+02	1.31E+01	3.93E+01	**1.43E-88**	**4.30E-88**
	3000	**30**	1.31E+02	3.93E+02	7.86E+01	2.36E+02	5.24E+01	1.57E+02	**1.06E-66**	**3.18E-66**
F7	**1000**	**10**	8.70E-80	2.61E-79	1.79E-79	5.37E-79	4.87E-79	1.46E-78	**6.77E-82**	**2.03E-81**
	2000	**20**	2.62E+00	7.86E+00	7.86E+00	2.36E+01	2.62E+00	7.86E+00	**2.86E-89**	**8.59E-89**
	3000	**30**	2.62E+01	7.86E+01	1.57E+01	4.72E+01	1.05E+01	3.15E+01	**2.12E-67**	**6.36E-67**
F8	**1000**	**10**	2.33E-74	7.36E-74	2.74E-76	8.66E-76	3.10E-77	9.79E-77	**1.02E-78**	**3.05E-77**
	2000	**20**	1.02E-84	3.22E-84	8.20E-88	2.59E-87	1.76E-90	5.58E-90	**1.55E-90**	**4.66E-90**
	3000	**30**	1.77E-26	5.32E-26	7.67E-20	2.30E-19	4.13E-32	1.24E-31	**2.13E-56**	**6.38E-56**

REFERENCES

[1] Selim Yılmaz and Ecir U. Kücüksille, "A new modification approach on bat algorithm for solvingoptimization problems," *Applied Soft Computing*, vol. 28, pp. 259-275, 2015.

[2] Xin-She Yang, *Nature-inspired metaheuristic algorithms*.: Luniver press, 2010.

[3] Mathew M Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum," *Applied Soft Computing*, vol. 12, pp. 353-359, 2012.

[4] Amir Hossein Gandomi, Xin-She Yang, Siamak Talatahari, and Amir Hossein Alavi, "Metaheuristic algorithms in modeling and optimization," in *Metaheuristic applications in structures and infrastructures*., 2014, pp. 1-24.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proccedings of the fourth IEEE international conference on neural networks*, Perth, Australia. IEEE Service Center, Piscataway, NJ., 1995.

[6] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information processing letters*, vol. 85, no. 6, pp. 317-325, 2003.

[7] N. Q. Uy, N. X. Hoai, R. I. McKay, and P. M. Tuan, "Initialising PSO with randomised low-discrepancy sequences: the comparative results," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 1985-1992.

[8] S. Kimura and K. Matsumura, "Genetic Algorithms using low discrepancy sequences," *in proc of GEECO 2005*, 2005, pp. 1341-1346.

[9] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "A niching Particle Swarm Optimizater," *In proceedings of the fourth Asia Pacific Conference on Simulated Evolution and learning*, 2002, pp. 692-696.

[10] J. G. van der Corput, "Verteilungsfunktionen," in *Proc.Ned. Akad. v. Wet., 38*, 1935, pp. 813-821.

[11] A. krohling Renoto and L. S. Coelho, "PSO-E: Particle Swarm with Exponential Distribution," *In Proc. of IEEE Congress on Evolutionary Computation*, Canda, 2006, pp. 1428 – 1433.

[12] R. Thangaraj, M. Pant, and K. Deep, "Initializing pso with probability distributions and low-discrepancy sequences: The comparative results,"*In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pp. 1121-1126.

[13] K. E. Parsopoulos and M. N. Vrahatis, "Initializing the particle swarm optimizer using the nonlinear simplex method," in *Advances in intelligent systems, fuzzy systems, evolutionary computation,216*, 2002, pp. 1-6.

[14] M. Richards and D. Ventura, "Choosing a starting configuration for particle swarm optimization," *In IEEE Int. Joint. Conf. Neural*, 2004, pp. 2309-2312.

[15] H. Jabeen, Z. Jalil, and A. R. Baig, "Opposition based initialization in particle swarm optimization (O-PSO)," *In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, 2009, pp. 2047-2052.

[16] A. L. Gutiérrez et al., "Comparison of Different PSO Initialization Techniques for High Dimensional Search Space Problems: A Test with FSS and Antenna Arrays," *In Antennas and Propagation (EUCAP), Proceedings of the 5th European Conference on IEEE*, 2011, pp. 965-969.

[17] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3-30, 1995.

[18] I. Y. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784-802, 1967.

[19] J. H. Halton, "Algorithm 247: Radical-inverse quasi-random point sequence," *Communications of the ACM*, vol. 7, no. 12, pp. 701-702, 1964.

[20] P. R. Tadikamalla and N. L. Johnson, "Systems of frequency curves generated by transformations of logistic variables," *Biometrika*, vol. 69, no. 2, pp. 461-465, 1982.