Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.015810
Article

Tech Science Press

Particle Swarm Optimization with New Initializing Technique to Solve Global Optimization Problems

Adnan Ashraf [1], Abdulwahab Ali Almazroi[2], Waqas Haider Bangyal[3,*] and Mohammed A. Alqarni[4]

[1]Govt. College Women University Sialkot, Sialkot, 51310, Pakistan
[2]University of Jeddah, College of Computing and Information Technology at khulais, Dept. of information Technology, Jeddah, Saudi Arabia
[3]University of Gujrat, Gujrat, 50700, Pakistan
[4]University of Jeddah, College of Computer Science and Engineering, Dept. of Software Engineering, Jeddah, Saudi Arabia
*Corresponding Author: Waqas Haider Bangyal. Email: waqas_bangyal@hotmail.com
Received: 08 December 2020; Accepted: 24 May 2021

Abstract: Particle Swarm Optimization (PSO) is a well-known extensively utilized algorithm for a distinct type of optimization problem. In meta-heuristic algorithms, population initialization plays a vital role in solving the classical problems of optimization. The population's initialization in meta-heuristic algorithms urges the convergence rate and diversity, besides this, it is remarkably beneficial for finding the efficient and effective optimal solution. In this study, we proposed an enhanced variation of the PSO algorithm by using a quasi-random sequence (QRS) for population initialization to improve the convergence rate and diversity. Furthermore, this study represents a new approach for population initialization by incorporating the torus sequence with PSO known as TO-PSO. The torus sequence belongs to the family of low discrepancy sequence and it is utilized in the proposed variant of PSO for the initialization of swarm. The proposed strategy of population's initialization has been observed with the fifteen most famous unimodal and multimodal benchmark test problems. The outcomes of our proposed technique display outstanding performance as compared with the traditional PSO, PSO initialized with Sobol Sequence (SO-PSO) and Halton sequence (HO-PSO). The exhaustive experimental results conclude that the proposed algorithm remarkably superior to the other classical approaches. Additionally, the outcomes produced from our proposed work exhibits anticipation that how immensely the proposed approach highly influences the value of cost function, convergence rate, and diversity.

Keywords: Particle swarm optimization; swarm intelligence; TO-PSO; quasi-random sequence

1 Introduction

Optimization is currently an effective paradigm of research and development. In this era, unbeatable optimization algorithms like PSO are needed to efficiently deal with complex real-world optimization problems [1]. A fundamental subdivision of artificial intelligence is Swarm intelligence (SI) that is responsible to process the multi-agent system and its morphological design; additionally, SI provides

satisfactory results on complex optimization problems. SI is influenced by the community actions of social insects like wasps, termites, ants, bees, as well as, by other social animal colonies i.e., (bird flocking or fish schooling) [2]. The title SI was firstly defined by Beni et al. [3] during the process of the cellular robotic system. For several years, researchers have indicated their curiosity towards the colonies of social insects, however, their community activities and behavior have remained hidden. While a single citizen is perceived to be an un-cultured single citizen in such communities, although, they have a strong capacity to solve complicated problems on a community basis. The transfer of knowledge between each participant and their comparatively uncomplicated acts improves their quality to accomplish the complex undertaking. SI techniques have been utilized successively in the domain of optimization, which becomes a major fact for science and industry.

Particle swarm optimization (PSO) is accepted as the SI-based stochastic algorithm, introduced by Kenndy et al. [4], is undoubtedly acknowledged as a beneficial approach to solving global optimization problems. PSO has been recognized as the most famous technique to solve complex optimization problems due to robustness and non-complexity, which is illustrated in the heterogeneous fields of engineering and science. In PSO, the target solution population becomes the swarm and is used to investigate the new search space; social movements of "flocks, birds" during their search for the food will be recorded in this new search area. From the observations of the remaining swarm, the transmission of information between all entities is considered as particles, occurs and all singles are accommodated. Each individual has two rules to endure, first looking to return to its previous best point and the other pursuing its swarm's sub sequential best position.

In the standard PSO, Velocity (v_i) and Positions (x_i), for $i = 1$ to n, are upgraded as given in Eqs. (1) and (2):

$$v_i \leftarrow v_i + b_1 r_1 \otimes (p_i - x_i) + b_2 r_2 \otimes \left(p_g - x_i\right) \tag{1}$$

$$x_i \leftarrow x_i + v_i \tag{2}$$

where r_1 *and* r_2 are two different function search that gives a vector of irregular values, which is uniformly generated in an interval of [0,1]. b_1 and b_2 are acceleration co-efficient. Inertia co-efficient ω is being used for better scaling. So updated version is given in Eq. (3).

$$v_i \leftarrow w v_i + b_1 r_1 \otimes (p_i - x_i) + b_2 r_2 \otimes \left(p_g - x_i\right) \tag{3}$$

The EAs and PSO techniques are implemented with the generation of random numbers, so the generation of random numbers works very well on these techniques for the small area of problems i.e., (low dimensional search areas). High dimensional search regions, though, appear to decrease the performance due to the premature particle convergence, as this causes sub-optimal outcomes [5]. For multimodal problems, this pattern is more persevering and intolerable, since it includes many local and global optimums. The inadequate distribution of the population in the search region can be the most important factor for this deprived performance, i.e., to assume that if the initial population does not properly scan the entire search space, which makes it difficult to recognize the robust global solution [6]. By adding the most formal and coordinated random distribution for population initialization, this issue can be resolve. Sequences of random numbers differ in the essence of their morphological design i.e., (quasi-random sequences, pseudo-random sequences, and probability distribution).

In the study of EAs, genetic algorithms, and PSO, it is observed that not sufficient research was performed to consider the QRS when initializing the swarm population in the search space [7]. Due to this fact, we have proposed a new low discrepancy sequence called Torus sequence for initializing the particles in the search space at an early stage. The totus algorithm is used to engage the particles with determined prime iterations using the Torus sequence. In comparison with the nine real-time classification

problems, the proposed algorithm TO-PSO has been observed with several uni-modal, multimodal, and rotated benchmark functions. The analogy with many other variants of PSO base on low discrepancy sequences has clarified that TO-PSO offers better performance, primarily for the complex optimization problems of higher dimensions. Comparison studies have also observed that TO-PSO outperforms in real-time classification problems than others.

The majority of the paper is organized as follows: Section II. Overviews the previous literature. The initialization methodologies and inertia weight strategies are discussed in Section III. Section IV addresses the implementation and comparison of initialization techniques using inertia weight on fifteen functions of benchmark measures, as well as the findings of our proposed TO-PSO in terms of the research accuracy. Finally, Section V will conclude the paper.

2 Literature Review

Researchers were interested to adopt the probability sequence and QRS to increase the reliability of population-based EAs [8], contrasting the generator of pseudo-random sequences with the generator of low discrepancy sequences showed that the generator of low discrepancy sequences performs best on the Halton sequence incorporated by EAs [9], which is one of the sets of low discrepancy sequences. The Halton sequence has been used to increase the efficiency of genetic algorithms.

For population initialization, the Sobol sequence and Faure sequence are incorporated in Brits et al. [10]. Furthermore, a comprehensive comparison of the sequences of Halton, Sobol, and Faure has been formed by [10].

Van der Corput sequence [11] is compatible with many of the associated low discrepancy sequences and can be initialized for the dimension and foundation.

In Krohling et al. [12], for the fine-tuning of PSO parameters, they utilize Exponential distribution. Comparative probability distribution analysis, including Gaussian, Beta, Gamma, and Exponential, was carried out in Thangaraj et al. [13].

The Non-linear simplex method (NSM) approach was implemented by Parsopoulos et al. [14]. The first particles of vertices (D + 1) are included in NSM, where D shows the dimensional search space. The first S(D + 1) particles are introduced to initiate them in a random position based on their initial vertices, wherein S is the size of the targeted population.

To initiate the initial point of the swarm population, Mark Richards and Dan Ventura conducted a centroidal Voronoi tessellations CVT generator [15].

Chengwei Yang [16] introduced a novel initialization approach QRS and time-varying inertia weight, titled LHNPSO, for non-linear high-order functions. The successful initialization of PSO using the Quasi-random Sobol and Halton sequences was validated by Shubham et al. [17].

A novel O-PSO algorithm belongs to the opposition-based particle learning implemented by the researchers [18], which increases the likelihood of capturing the global minimum at a very early stage. Three independent variants of the PSO algorithm focused on the following initialization techniques were contrasted by authors in Gutiérrez et al. [19]: the chaotic initialization, the orthogonal array initialization, and the opposition-based initialization.

In 2017 [20], researchers presented two modifications to the standard PSO as it is termed PSO-IM. These two modifications are, first, the heuristic inertia weight equation for accurate and rapid convergence of PSO, and two different types of mutations implemented on swarm particles, and selection of fuzzy controller. Large inertia weight is initially used to explore search space, but it continues to decrease with the growth in the number of iterations so local search in the final iteration will be essential. At the initial point, few

randomly selected particles are used for each tournament, and particles with the best-fitted value are chosen. Repeat this process until the mutation is chosen for a given group of particles and then the uniform mutation and non-uniform mutation are applied, which improves the searchability of PSO particles and avoids premature convergence. Finally, relative to the standard PSO, the PSO-IM variant exhibits optimal performance.

Kessentini & Barchiesi in 2015 [21] highlighted the drawbacks of the standard PSO algorithm, which is not sufficient to solve engineering problems related to optimization because it slows down and converges to local optima in large dimensional scenarios. Numerous PSO variants that maintain an intact harmony between local and global discovery have been suggested to deal with these limitations. Initially, Y. Shi and R. Eberhart [22], intruded inertia weight in PSO, later on, many types of research were denoted on this and considered as an essential part of PSO to balance search ability at local and global levels. An adaptive inertia weight (w-PSO) technique was presented in this study. For dynamic inertia weight setting, the feedback response from the best-positioned particle in the swarm is used. In w-PSO, the dynamic environment ranges varies from 0.4 to 0.9, which drastically increases the searchability of PSO. Results of the significant improvement of w-PSO on the set of CEF 2014 benchmark functions compared to other variants of PSO [23].

In 2016, Taherkhani et al. [24] explored the key and critical parameter in EA's algorithm (PSO) to limit particles in the search space, termed as inertia weight. This research presented an adaptive approach, based on the individual executions of particles and their distance from the best location,to remove inertia weight from all the dimensions and for all particles.

3 Methodologies

3.1 Initialization

On the traditional PSO algorithm, four various population initialization methodologies have been implemented, which are Uniform Pseudo-Random initialization technique [8]: Halton sequence for initialization [25], Sobol sequence for initialization [26], and finally a proposed technique is based on Quasi-Random Sequences (QRS) termed as Torus Sequence (low discrepancy sequence) [27], opposed to pseudo-random sequence when applied on traditional benchmark functions, the Torus technique enhances the performance of standard PSO and achieve optimal results.

3.1.1 Uniform

By following a uniform-distribution [8], the pseudo-random sequence produces random numbers and can be characterized by probability-density-function (PDF) of const. Uniform-distribution. As is given in Eq. (4) as:

$$f(a) = \begin{cases} \frac{1}{x-y} for\, x < a < y \\ 0\, for\, a < x\, or\, a > y \end{cases}, \tag{4}$$

whereas, the maximum likelihood parameters are variables x and y. Due to 0, the effect of adjusting the integral $f(a)da$ over any other interval. Therefore, at the boundary of x and y, the value of $f(a)$ value is not important. Function for probability is given in Eq. (5) that can be used to determine the parameter with the highest probability.

$$l(x, y|a) = nlog(y - x) \tag{5}$$

3.1.2 Halton

An improved version of the Van Dar Corput [11] sequence is presented by J. Halton [25] is known as the Halton sequence. Halton sequence emphasizes on co-prime base to generate the random sequence pattern. The pseudo-code for the Haltom sequence generation is given as follows:

Halton Sequence:

//input: Initial index = ***s*** & base = ***co-prime***

//output: instances = h

Interval lower and upper limits

a) *minimum* = 0

b) *maximum* =1

For each epoch $k_1, k_2, k_3 \ldots k_n$:do

a. For each particle $\{p_1, p_2, p_3, \ldots..p_n\}$

- $max = max/coprime$
- $min = min + max * smodb$
- $s = s/b$

Return h

3.1.3 Sobol

Ilya M. Sobol, who was a Russian mathematician, primarily proposed Sobol [26] sequence. Sobol sequence works by re-constructing the coordinates. For each dimension, all of these co-ordinates possess linear recurrence relations. Let, a non-negative instance s contains an expression in binary form, where s is as follows in (6):

$$s = s_1 2^0 + s_2 2^1 + s_3 2^2 + \ldots + s_w 2^{w-1}, \tag{6}$$

Then, the i^{th} instance for dimension D is generated using the (7) as follows:

$$X_i^D = i_1 v_1^D + i_2 v_2^D + \ldots + i_w v_w^D, \tag{7}$$

However, v_1^D represents the binary function followed by i^{th} direction instance and dimension D. The direction instance is generated by using (8) and c_q is a polynomial coefficient where $i > q$.

$$V_i^D = c_1 v_{i-1}^D + c_2 v_{i-2}^D + \ldots + c_q v_{q-1}^D + \left(\frac{v_{i-q}^D}{2^q}\right), \tag{8}$$

3.1.4 Torus

Torus is a geometric term introduced in 1987 [27], primarily focused on the development of geometric co-ordinate structure Torus-mesh. The foundation of game development is also concentrated on the mesh structure that can be achieved with torus sequence. With torus-mesh, the coordinated structure of the left and right hands can be easily accomplished. Torus has various shapes at different dimensions. For example, for 1D-shapes is the line, 2D-shapes are circle, rectangle and 3D shapes is a donut, mainly 3D shape is generated by the given set of Eqs. (9)–(11) given as:

$$a(\theta, \delta) = (D + rcos\theta)cos\delta, \tag{9}$$

$$b(\theta, \delta) = (D + rcos\theta)sin\delta, \tag{10}$$

$$c(\theta, \delta) = rsin\delta, \tag{11}$$

whereas, θ, δ and D are defined as the angles of circles, r is proportional to the circle radius, distant from the middle of the tube to the center of Torus. Low discrepancy sequences are driven by the Torus-mesh, as well as, its initialization with prime series results as a Torus effect. The mathematical depiction of the Torus sequence in (12) is given below:

$$\alpha_k = (f(k\sqrt{s_1}), \ldots, f(k\sqrt{s_d})), \tag{12}$$

Here, s_1 represents the set of i^{th} prime-numbers and f represents fraction which is evaluated by $f = a - floor(a)$. There are key constraints on the Torus function for the use of prime parameters that limit its dimension to 100,000. The provision of number is handled manually as dimensions exceed 100,000.

3.2 Inertia Weight Strategies

Inertia weight was firstly presented by Y. shi and R. Eberhart [22] and recently, the most important PSO parameter is inertia weight. High velocity in the standard PSO allows particles to exist from the search space limits. To deal with this limitation, the velocity-clamping or maximum velocity limit is set. The introduction of a new inertia weight parameter increases the standard PSO performance because particles do not diverge and the inertia weight range is set from 0.4 to 0.9. These inertia weight values are an integral part of balancing local and global search. Furthermore, two inertia weight strategies are applied in this research work [28] on standard PSO, which are Linearly Decreasing inertia-weight and the Chaotic inertia-weight (with and without Torus initialization) to enhance the standard PSO efficiency.

3.2.1 Linear Decreasing Inertia-Weight

Linearly-decreasing inertia-weight has also been implemented on standard PSO [29]. Using linear-decreasing inertia weight, the speed of the exploration process increases inside the search space, which took more time to complete with the standard PSO. Therefore for each repetition, iterative evaluation of inertia weight is conducted using (13) as:

$$\omega(t) = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})}{iter_{\max}} \times k, \tag{13}$$

Here, $w_{max} = 0.9$, $w_{min} = 0.4$, k = current iteration and $iter_{max}$ = max. iterations. The initialization of this strategy will linearly decrease inertia-weight from 0.9 to 0.4. $\omega(t) > 0.4$ and $\omega(t) < 0.78540$, shows better convergence behavior. Linear-decreasing inertia weight exhibits better performance for both the PSO with Torus and the standard PSO version.

3.2.2 Chaotic Inertia Weight

Chaotic inertia-weight (**ω**) was implemented on standard PSO [30] and designed to improve the searchability on both local and global search space by decreasing inertia weight. This approach is a non-linear inertia weight strategy, which calculates chaotic term Z by randomly selecting any number between 0 and 1 intervals and evaluates logistic mapping by using (14), given as:

$$z = 4 \times z \times (1 - z), \tag{14}$$

Then in an iterative process, the value of inertia weight is also measured from the (15) as:

$$\omega(t) = (\omega_{max} - \omega_{min}) - \frac{(iter_{max} - iter_{current})}{iter_{max}} + \omega_{min} \times z, \tag{15}$$

Whereas, $\omega_{max} = 0.9$, $\omega_{min} = 0.4$, $iter_{current}$ = current iteration and $iter_{max}$ = max. iterations. Due to the non-linear strategy, it decreases the inertia-weight from $\omega_{max} = 0.9$ to $\omega_{min} = 0.4$. It shows better convergence speed and converges when $\omega(t) > 0.4$ and $\omega(t) < 0.78540$. The implementation of chaotic inertia-weight on the *TO-PSO* and the standard PSO, both gives better performance.

4 Results and Discussion

4.1 Experimental Settings of Particle Swarm Optimization

4.1.1 Parameters

Upper and lower boundary of the search space is set according to the applied function, population size will be 30, three dimensional levels will be used for PSO that are 10, 20 and 30 for three sets of iterations 1000, 2000 and 3000 used for each size represented in Tab. 1. PSO was run 10 times and the average value was calculated.

Table 1: Experimental setting of parameters

Parameters	Values
Search Space	[–100, 100]
Dimensions	(10, 20, 30)
Iterations	(1000, 2000, 3000)
Population Size	30
PSO Runs	10

4.1.2 Functions

Standard benchmark functions [31] are given in Tab. 2 over the continues range of values available in the literature (used by different researchers) used to implement the evolutionary computing algorithm/s.

Table 2: Objective functions and their optimal values

No.	Function Name	Objective Function	Search Space	Optimum Value
F1	Sphere	$Minf(\mathrm{x}) = \sum_{i=1}^{D} x_i^2$	$-5.12 \leq \mathrm{x_i} \leq 5.12$	0
F2	Michalewiczs	$Minf(\mathrm{x}) = \sum_{i=1}^{D} \sin\left(\mathrm{x_i}\sin^{2\mathrm{m}}\left(\frac{\mathrm{ix_i^2}}{\pi}\right)\right)$	$-0 \leq \mathrm{x_i} \leq \mathrm{pi}$	0
F3	Alpine-1	$Minf(\mathrm{x}) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	$-10 \leq \mathrm{x_i} \leq 10$	0
F4	XinShe-1 (Separable)	$Minf(\mathrm{x}) = \sum_{i=1}^{D} \in_i \lvert x_i \rvert^i$	$-5 \leq \mathrm{x_i} \leq 5$	0

(Continued)

Table 2 (continued).

No.	Function Name	Objective Function	Search Space	Optimum Value
F5	XinShe-4 (non-Separable)	$Minf(\mathrm{x}) = \left[\sum_{i=1}^{D} \sin^2(x_i) - e^{-\sum_{i=1}^{D}\sqrt{\lvert x_i\rvert}}\right]$	$-10 \le \mathrm{x_i} \le 10$	0
F6	Deb-1	$Minf(\mathrm{x}) = -\frac{1}{D}\sum_{i=1}^{D} \sin^6(5\pi x_i)$	$-1 \le \mathrm{x_i} \le 1$	0
F7	Giunta	$Minf(\mathrm{x}) = 0.6\sum_{i=1}^{2}\left[\sin\left(\frac{16}{15}x_i - 1\right) + \sin^2\left(\frac{16}{15}x_i - 1\right) + \frac{1}{50}\sin\left(4\left(\frac{16}{15}x_i - 1\right)\right)\right]$	$-1 \le \mathrm{x_i} \le 1$	0
F8	Powell Sum	$Minf(\mathrm{x}) = \sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	$-1 \le \mathrm{x_i} \le 1$	0
F9	Qing	$Minf(\mathrm{x}) = \sum_{i=1}^{D} \left(x_i^2 - i\right)^2$	$-500 \le \mathrm{x_i} \le 500$	0
F10	Quartic	$Minf(\mathrm{x}) = \sum_{i=1}^{D} ix_i^4 + random[0.1]$	$-1.28 \le \mathrm{x_i} \le 1.28$	0
F11	Zakharov	$Minf(\mathrm{x}) = \sum_{i=1}^{D} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^4$	$-5 \le \mathrm{x_i} \le 10$	0
F12	Csendes	$Minf(\mathrm{x}) = \sum_{i=1}^{D} x_i^6\left(2 + \sin\frac{1}{x_i}\right)$	$-1 \le \mathrm{x_i} \le 1$	0
F13	Shubert-4	$Minf(\mathrm{x}) = \sum_{i=1}^{D}\sum_{j=1}^{5} j.\cos((j+1)x_i + j)$	$-10 \le \mathrm{x_i} \le 10$	0
F14	step	$Minf(\mathrm{x}) = \sum_{i=1}^{D} (\lfloor \lvert x_i\rvert \rfloor)$	$-100 \le \mathrm{x_i} \le 100$	0
F15	Sum Squares	$Minf(\mathrm{x}) = \sum_{i=1}^{D} ix_i^2$	$-10 \le \mathrm{x_i} \le 10$	0

4.2 Implementation Results of Initialization Techniques

The comparative results of four PSO initialization techniques on 23 benchmark test functions are given in Tab. 3, where the column "Mean" shows the calculated mean value. Bold values are describing the winner function among the four PSO algorithms.

Table 3: Comparative results among the four pso variants on 15 benchmark test functions

Functions	Iterations	Dimensions	PSO	H-PSO	SO-PSO	PSO Torus
			Mean	Mean	Mean	Mean
F1	1000	10	1.78E-23	2.59E-23	4.38E-23	**5.81E-27**
	2000	20	1.27E-06	5.45E-07	**2.09E-07**	3.45E-07
	3000	30	2.07E-03	3.61E-03	2.42E-03	**2.01E-03**
F2	1000	10	2.15E-28	**4.07E-30**	1.51E-28	1.08E-27
	2000	20	2.10E-06	3.83E-10	2.91E-07	**2.98E-12**
	3000	30	1.10E-02	2.37E-03	1.37E-03	**5.35E-15**

Table 3 (continued).

Functions	Iterations	Dimensions	PSO	H-PSO	SO-PSO	PSO Torus
			Mean	Mean	Mean	Mean
F3	1000	10	7.66E-05	1.77E-04	3.94E-04	**6.11E-05**
	2000	20	**2.37E-02**	1.24E-01	5.03E-01	2.70E-02
	3000	30	8.72E-01	1.12E+00	7.13E-01	**3.02E-01**
F4	1000	10	2.45E-03	4.04E-01	3.79E-03	**2.29E-03**
	2000	20	2.12E-01	2.35E+00	1.03E+00	**1.18E-01**
	3000	30	9.08E+00	2.54E+00	3.49E+01	**5.15E-01**
F5	1000	10	–2.20E+04	–2.20E+04	–2.20E+04	**–2.20E+04**
	2000	20	–4.85E+08	–4.85E+08	–3.57E+08	**–4.85E+08**
	3000	30	–9.03E+12	–8.76E+12	–6.48E+12	**–9.53E+12**
F6	1000	10	–1.00E+00	–1.00E+00	–1.00E+00	**–1.00E+00**
	2000	20	–8.00E-01	–9.95E-01	**–1.00E+00**	–9.90E–01
	3000	30	–9.67E-01	–9.66E-01	**–9.79E-01**	–9.76E-01
F7	1000	10	–2.61E+00	–2.49E+00	**–2.66E+00**	–2.62E+00
	2000	20	–5.17E+00	–5.11E+00	**–5.18E+00**	–5.14E+00
	3000	30	**–7.72E+00**	–7.62E+00	–7.62E+00	–7.50E+00
F8	1000	10	6.27E-45	9.07E-47	**7.91E-49**	2.76E-44
	2000	20	2.75E-24	4.55E-23	4.08E-23	**2.36E-24**
	3000	30	1.97E-17	7.17E-17	8.03E-16	**7.40E-19**
F9	1000	10	9.26E-17	6.97E-19	**5.73E-20**	5.30E-15
	2000	20	5.02E-01	1.86E+00	1.02E+00	**1.94E-01**
	3000	30	1.26E+02	2.20E+02	3.21E+02	**1.14E+02**
F10	1000	10	5.06E-06	9.72E-05	1.02E-04	**2.70E-06**
	2000	20	**1.07E-01**	2.67E-01	1.41E+00	4.71E-01
	3000	30	6.01E+00	7.70E+00	7.77E+00	**3.57E+00**
F11	1000	10	**2.12E-06**	2.23E-05	1.39E-05	2.97E-05
	2000	20	1.07E+01	2.43E+01	4.90E+01	**9.68E+00**
	3000	30	1.64E+02	8.47E+01	1.95E+02	**6.39E+01**
F12	1000	10	5.99E-26	6.18E-28	4.34E-23	**3.46E-30**
	2000	20	5.72E-08	6.47E-08	4.49E-08	**3.89E-08**
	3000	30	6.43E-07	**4.41E-07**	6.20E-07	5.60E-07
F13	1000	10	–5.84E+02	–5.64E+02	–5.71E+02	**–5.98E+02**
	2000	20	**–1.87E+03**	–1.79E+03	–1.77E+03	–1.79E+03
	3000	30	**–3.67E+03**	–3.44E+03	–3.53E+03	–3.42E+03

(Continued)

Table 3 (continued).

Functions	Iterations	Dimensions	PSO	H-PSO	SO-PSO	PSO Torus
			Mean	Mean	Mean	Mean
F14	1000	10	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**
	2000	20	8.60E+00	3.41E+01	9.00E+00	**3.70E+00**
	3000	30	4.57E+01	4.74E+01	4.25E+01	**2.68E+01**
F15	1000	10	7.25E-22	**2.23E-23**	4.16E-22	5.61E-21
	2000	20	9.64E-06	2.20E-05	2.87E-04	**2.22E-06**
	3000	30	3.52E-01	1.94E-01	1.13E-01	**1.01E-01**

The comparative results of four different population initialization techniques are generated by using the standard PSO. All these four PSO initialization variants are implemented on fifteen benchmark test functions, where "Mean" shows mean value. The best results are presented in bold among the four PSO algorithms. In contrast with the traditional PSO, SO-PSO, and H-PSO, TO-PSO technique performs remarkably well. In terms of improved convergence, speed, consistency and durability, the TO-PSO sequence based on QRS showed better performance on implemented benchmark functions for minimization optimization than other initialization techniques.

4.3 Graphs of PSO Initialization Techniques

All the graphs shown in Fig. 1 are generated as per the results of Tab. 3 with 10 dimensions and 1000 iterations, 20 dimensions and 2000 iteration similarly 30 dimensions and 3000 iterations.

4.4 Implementation Results of Initialization Techniques using Linear Decreasing and Chaotic Inertia Weights

Inertia weight is an essential parameter of the PSO algorithm, which plays a vital role to balance the global exploration and local exploitation of the search space. Due to the importance of inertia weight, we have used the linear decreasing inertia weight (LDIW) with PSO, SO-PSO, and H-PSO and proposed initialization approach TO-PSO. The impact of linear decreasing inertia weight is remarkably good on the performance of all variants that can observe from the aforementioned Tab. 4. Our proposed technique beats the other three PSO initialization approaches on fifteen benchmark functions. The winner results among all the four PSO approaches are differentiated as bold.

The use of chaotic mapping to adjust the coefficient is the fundamental concept of chaotic inertia weight. In this work logistic mapping is being used. The logistic mapping is:

$$z = \mu \times z \times (1\text{-}z)$$

when $3.57 < \mu \leq 4$, chaotic phenomena arise during logistic mapping.

Result from Tab. 5, of chaotic inertia weight with PSO strategies has a positive effect on the optimization of functions relative to the fixed inertia weight of PSO. PSO initialization strategies with chaotic inertia weight initially perform search covering a wide range called *"rough search"*, while the search obtains a relatively small range called "*minute search"* during the optimization process.

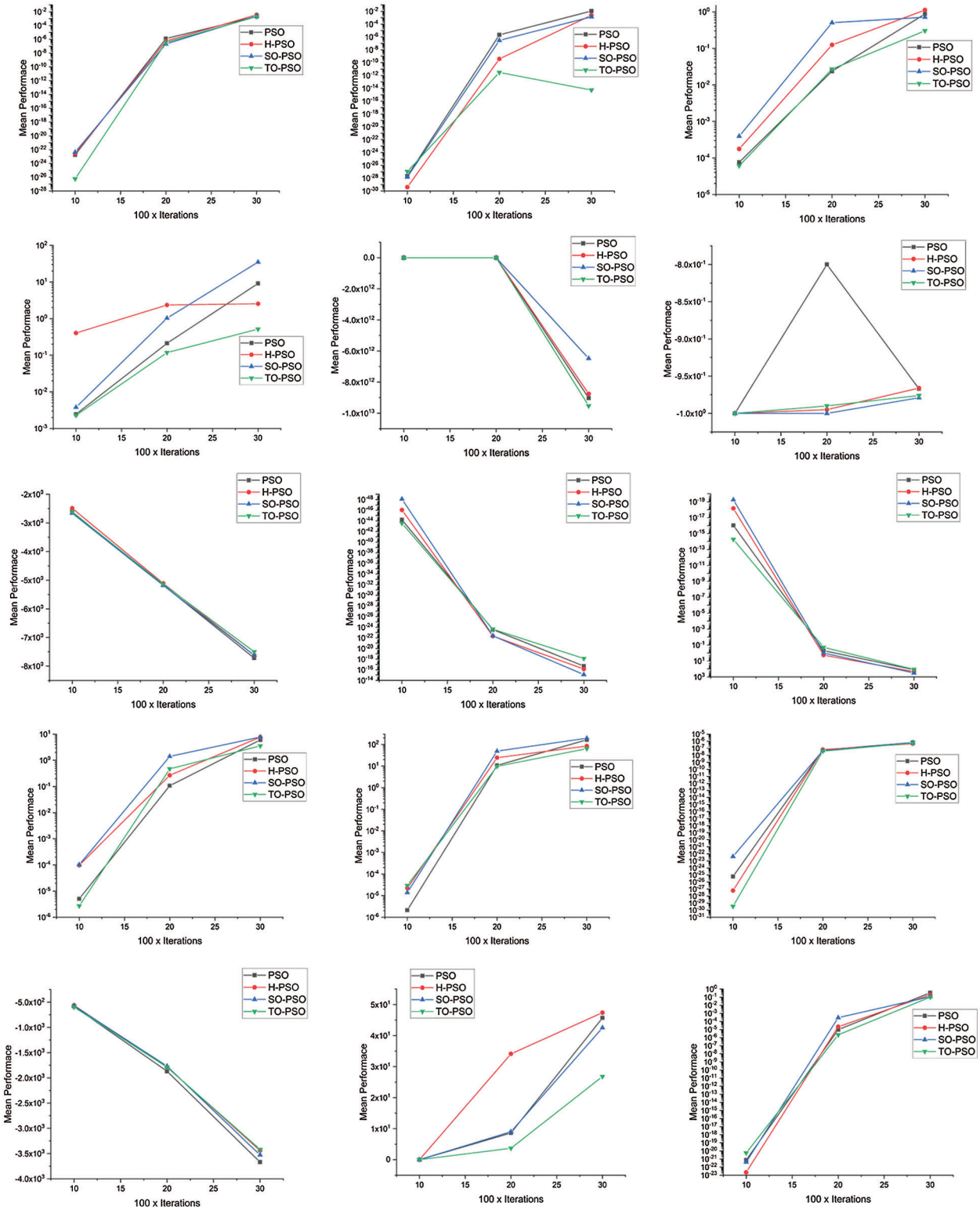

Figure 1: Graphs of PSO initialization techniques

Table 4: Average fitness result of all techniques with linear decreasing inertia weight

Functions	Iter	DIM	PSO	SO-PSO	H-PSO	TO-PSO
			Mean	Mean	Mean	Mean
F1	1000	10	3.95E-83	5.11E-84	7.25E-82	3.34E-82
	2000	20	9.19E-80	1.82E-72	3.47E-40	1.19E-80
	3000	30	2.12E-55	2.97E-54	5.92E-53	3.28E-50
F2	1000	10	2.50E-30	4.54E-30	4.12E-31	5.92E-31
	2000	20	4.62E-30	1.13E-43	2.83E-30	4.36E-32
	3000	30	2.77E-29	1.19E-32	3.30E-29	3.16E-29
F3	1000	10	9.94E-16	4.13E-50	4.83E-15	1.09E-15
	2000	20	1.31E-10	5.20E-14	3.30E-11	5.02E-09
	3000	30	4.44E-01	1.96E-08	2.53E-08	2.01E-07
F4	1000	10	3.40E-03	1.14E-07	9.07E-09	1.64E-07
	2000	20	5.02E-03	1.51E-03	3.71E-02	9.50E-03
	3000	30	1.96E+00	7.68E-06	1.22E-02	3.58E-02
F5	1000	10	–2.20E+04	–2.20E+04	–2.20E+04	–2.20E+04
	2000	20	–4.36E+08	–3.50E+08	–4.12E+08	–4.36E+08
	3000	30	–3.76E+12	–4.80E+12	–6.85E+12	–7.45E+12
F6	1000	10	–1.00E+00	–9.90E-01	–1.00E+00	–1.00E+00
	2000	20	–1.00E+00	–9.70E-01	–9.90E–01	–9.95E-01
	3000	30	–9.80E-01	–9.67E-01	–1.93E+04	–9.73E-01
F7	1000	10	–2.63E+00	–2.02E+00	–1.91E+00	–2.62E+00
	2000	20	–5.11E+00	–4.58E+00	–4.43E+00	–5.15E+00
	3000	30	–7.42E+00	–6.97E+00	–7.00E+00	–7.50E+00
F8	1000	10	1.25E-127	1.85E-128	1.21E-129	5.04E-128
	2000	20	1.30E-182	8.21E-183	1.13E-181	1.25E-184
	3000	30	2.12E-189	5.91E-194	6.16E-202	3.86E-191
F9	1000	10	9.07E-30	9.07E-30	9.07E-30	9.07E-30
	2000	20	6.23E-29	6.40E-29	5.64E-29	5.64E-29
	3000	30	3.54E-27	6.05E-25	2.44E-26	1.66E-25
F10	1000	10	8.88E-17	0.00E+00	1.78E-16	0.00E+00
	2000	20	6.87E-13	9.40E-12	9.87E-12	5.69E-14
	3000	30	2.93E+00	3.00E-08	5.14E-09	4.88E+00
F11	1000	10	5.38E-44	2.44E-44	1.51E-42	4.06E-45
	2000	20	1.15E+01	5.35E+00	4.88E+01	2.16E+01
	3000	30	1.13E+02	4.41E+01	5.74E+01	1.09E+01
F12	1000	10	7.94E-170	1.15E-206	5.62E-209	2.90E-186
	2000	20	5.46E-34	1.15E-37	8.59E-34	6.66E-37
	3000	30	1.70E-11	1.94E-13	1.77E-14	3.84E-16

Table 4 (continued).

Functions	Iter	DIM	PSO	SO-PSO	H-PSO	TO-PSO
			Mean	Mean	Mean	Mean
F13	1000	10	–6.65E+02	–6.08E+02	–6.21E+02	–6.49E+02
	2000	20	–2.36E+03	–2.22E+03	–2.23E+03	–2.32E+03
	3000	30	–4.79E+03	–4.57E+03	–4.52E+03	–4.62E+03
F14	1000	10	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	2000	20	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	3000	30	0.00E+00	3.01E+01	0.00E+00	0.00E+00
F15	1000	10	5.51E-81	2.10E-82	4.56E-82	6.37E-81
	2000	20	1.06E-79	1.11E-78	9.64E-77	3.63E-72
	3000	30	1.49E-52	4.04E-46	5.01E-51	3.31E-53

Table 5: Avg fitness result of all techniques with chaotic inertia weight

Functions	Iter	DIM	PSO	SO-PSO	H-PSO	TO-PSO
			Mean	Mean	Mean	Mean
F1	1000	10	3.95E-83	2.99E-83	3.89E-84	3.34E-82
	2000	20	9.19E-80	3.85E-35	1.67E-49	1.19E-80
	3000	30	2.12E-55	5.34E-30	2.41E-33	3.28E-50
F2	1000	10	2.50E-30	3.92E-30	9.85E-30	5.92E-31
	2000	20	4.62E-30	7.91E-31	1.76E-29	4.36E-32
	3000	30	2.77E-29	5.98E-29	3.81E-28	3.16E-29
F3	1000	10	9.94E-16	2.19E-15	2.69E-13	1.09E-15
	2000	20	1.31E-10	6.48E-08	1.14E-05	5.02E-09
	3000	30	4.44E-01	4.22E-05	1.09E-02	2.01E-07
F4	1000	10	3.40E-03	3.25E-03	5.47E-03	1.64E-07
	2000	20	5.02E-03	4.74E-02	4.29E-03	9.50E-03
	3000	30	1.96E+00	7.15E-02	1.80E-01	3.58E-02
F5	1000	10	–2.20E+04	–2.20E+04	–2.20E+04	–2.20E+04
	2000	20	–4.36E+08	–4.12E+08	–4.36E+08	–4.36E+08
	3000	30	–3.76E+12	–5.44E+12	–8.41E+12	–7.45E+12
F6	1000	10	–1.00E+00	–1.00E+00	–1.00E+00	–1.00E+00
	2000	20	–1.00E+00	–9.80E–01	–1.00E+00	–9.95E-01
	3000	30	–9.80E-01	–9.80E-01	–9.87E-01	–9.73E-01
F7	1000	10	–2.63E+00	–2.01E+00	–1.92E+00	–2.62E+00
	2000	20	–5.11E+00	–4.38E+00	–4.51E+00	–5.15E+00
	3000	30	–7.42E+00	–6.95E+00	–6.97E+00	–7.50E+00

(Continued)

Table 5 (continued).

Functions	Iter	DIM	PSO	SO-PSO	H-PSO	TO-PSO
			Mean	Mean	Mean	Mean
F8	1000	10	1.25E–127	1.25E-132	2.52E-133	5.04E-128
	2000	20	1.30E-182	1.53E-221	3.12E-188	1.25E-184
	3000	30	2.12E-189	6.89E-209	4.56E-187	3.86E-191
F9	1000	10	9.07E-30	9.07E-30	9.07E-30	9.07E-30
	2000	20	6.23E-29	2.67E-27	4.99E-25	5.64E-29
	3000	30	3.54E-27	1.60E-26	6.22E-15	1.66E-25
F10	1000	10	8.88E-17	4.21E-12	2.59E-05	0.00E+00
	2000	20	6.87E-13	3.53E+00	9.85E-02	5.69E-14
	3000	30	2.93E+00	8.61E+00	1.30E-01	4.88E+00
F11	1000	10	5.38E-44	1.31E-34	1.79E-45	4.06E-45
	2000	20	1.15E+01	4.19E-03	4.88E-03	2.16E+01
	3000	30	1.13E+02	2.85E+01	2.42E+01	1.09E+01
F12	1000	10	7.94E-170	8.00E-181	1.49E-217	2.90E-186
	2000	20	5.46E-34	5.12E-12	3.66E-09	6.66E-37
	3000	30	1.70E-11	2.41E-07	6.21E-08	3.83E-16
F13	1000	10	–6.65E+02	–6.41E+02	–6.35E+02	–6.49E+02
	2000	20	–2.36E+03	–2.08E+03	–2.11E+03	–2.32E+03
	3000	30	–4.78E+03	–4.17E+03	–3.94E+03	–4.62E+03
F14	1000	10	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	2000	20	0.00E+00	5.00E-01	9.00E-01	0.00E+00
	3000	30	0.00E+00	1.90E+00	6.60E+00	0.00E+00
F15	1000	10	5.51E-81	3.31E-114	8.00E-106	6.37E-81
	2000	20	1.06E-79	5.25E-73	2.75E-40	3.63E-72
	3000	30	1.49E-52	4.64E-37	4.01E-34	3.31E-53

5 Conclusion

In this study, a novel initialization technique is proposed for the population's initialization of PSO. For experimental validation, a set of twenty-three unimodal and multi-modal benchmark test functions are utilized. The analysis of experimental results shows that the proposed technique enhances the convergence speed and improves the search ability to find a better region for the swarm, as well as, maintains the diversity of the swarm. The presented technique also emphasizes that if there is no information on the existence of candidate solutions, then it is possible to proceed with the most suitable candidate solution first. Additionally, the evaluation of results illustrates that the proposed TO-PSO initialization technique outperforms the standard PSO, and other variants such as PSO with Sobol and Halton sequences as well. To measure the performance efficiency of PSO, the inertia weight factor in PSO is also embedded using TO-PSO initialization, similarly, it is compared with the other inertia

weights presented in the literature. The primary objective of this research work is generic but applicable to other stochastic based metaheuristic algorithms that develops the future direction of our work.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors have no conflicts of interest to report regarding the present study.

References

[1] V. Bhaskar, S. K. Gupta and A. K. Ray, "Applications of multiobjective optimization in chemical engineering," Vol. 16, no. 1, pp. 1–54, 2000.

[2] C. Blum and X. Li, "Swarm intelligence in optimization," in *Swarm intelligence: Springer*, pp. 43–85, 2008.

[3] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and biological systems: Towards a new bionics?: Springer*, pp. 703–712, 1993.

[4] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. of the IEEE int. conf. on neural networks: Citeseer*, vol. 4, pp. 1942–1948, 1995.

[5] H. Liu and A. Abraham, "Fuzzy adaptive turbulent particle swarm optimization," in *Fifth Int. Conf. on Hybrid Intelligent Systems (HIS'05): IEEE*, pp. 6, 2005.

[6] Crina Grosan, A. Abraham and M. Nicoara, "Search optimization using hybrid particle sub-swarms and evolutionary algorithms," in *International Journal of Simulation Systems, Science & Technology*, vol. 6, no. 10, pp. 60–79, 2005.

[7] M. Pant, R. Thangaraj, C. Grosan and A. Abraham, "Improved particle swarm optimization with low-discrepancy sequences," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence): IEEE*, pp. 3011–3018, 2008.

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

[9] S. Kimura and K. Matsumura, "Genetic algorithms using low-discrepancy sequences," in *Proc. of the 7th annual conf. on Genetic and evolutionary computation: ACM*, pp. 1341–1346, 2005.

[10] R. Brits, A. P. Engelbrecht and F. Van den Bergh, "A niching particle swarm optimizer," *Proc. of the 4th Asia-Pacific conf. on simulated evolution and learning: Singapore: Orchid Country Club*, vol. 2, pp. 692–696, 2002.

[11] J. Van der Corput, *Verteilungsfunktionen: Mitteilg 7*. NV Noord-Hollandsche Uitgevers Maatschappij, 1936.

[12] R. A. Krohling and L. dos Santos Coelho , "PSO-E: Particle swarm with exponential distribution," in *2006 IEEE Int. Conf. on Evolutionary Computation: IEEE*, pp. 1428–1433, 2006.

[13] R. Thangaraj, M. Pant and K. Deep, "Initializing PSO with probability distributions and low-discrepancy sequences: The comparative results," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC): IEEE*, pp. 1121–1126, 2009.

[14] 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*, vol. 216, pp. 1–6, 2002.

[15] M. Richards and D. Ventura, "Choosing a starting configuration for particle swarm optimization," *IEEE Int. Joint. Conf. Neural*, vol. 3, pp. 2309–2312, 2004.

[16] C. Yang, W. Gao, N. Liu and C. Song, "Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight," vol. 29, pp. 386–394, 2015.

[17] S. K. Gupta, H. Gupta, S. Arora, P. Nayak and T. Shrivastava, "Efficient Initialization of Particle Swarm Optimization Using Low Discrepancy Sequence," in *Int. Conf. on Soft Computing and Pattern Recognition: Springer*, pp. 440–449, 2016.

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

[19] A. L. Gutiérrez, M. Lanza, I. Barriuso, L. Valle, M. Domingo *et al.,* "Comparison of different pso initialization techniques for high dimensional search space problems: A test with fss and antenna arrays," in *Proc. of the 5th European Conf. on Antennas and Propagation (EUCAP): IEEE*, pp. 965–969, 2011.

[20] S. Sheikhpour and A. Mahani, "Particle swarm optimization with intelligent mutation for nonlinear mixed-integer reliability-redundancy allocation," vol. 16, no. 01, pp. 1750003, 2017.

[21] S. Kessentini and D. Barchiesi, "Particle swarm optimization with adaptive inertia weight," vol. 5, no. 5, pp. 368, 2015.

[22] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE int. conf. on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360): IEEE*, pp. 69–73, 1998.

[23] A. R. Lynn, "Instrumentation Set Points," *Los Alamos National Lab.(LANL), Los Alamos, NM (United States)*, vol. 11, no. 61, pp. 723A–734A, 2015.

[24] M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," vol. 38, pp. 281–295, 2016.

[25] X. Wang and F. J. Hickernell, "Randomized halton sequences," vol. 32, no. 7-8, pp. 887–899, 2000.

[26] M. Pant, R. Thangaraj, V. P. Singh and A. Abraham, "Particle swarm optimization using Sobol mutation," in *2008 First Int. Conf. on Emerging Trends in Engineering and Technology: IEEE*, pp. 367–372, 2008.

[27] V. V. Nikulin and I. R. Shafarevich, *Geometries and groups*. Springer Science & Business Media, 2012.

[28] J. C. Bansal, P. Singh, M. Saraswat, A. Verma, S. S. Jadon *et al.,* "Inertia weight strategies in particle swarm optimization," in *2011 Third world congress on nature and biologically inspired computing*, IEEE, pp. 633–640, 2011.

[29] M. Pluhacek, R. Senkerik, D. Davendra, Z. K. Oplatkova, I. J. C. Zelinka *et al.,* "On the behavior and performance of chaos driven PSO algorithm with inertia weight," vol. 66, no. 2, pp. 122–134, 2013.

[30] Y. Feng, G. F. Teng, A. X. Wang and Y. M. Yao, "Chaotic inertia weight in particle swarm optimization," in *Second Int. Conf. on Innovative Computing, Information and Control (ICICIC 2007): IEEE*, pp. 475, 2007.

[31] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimization problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.