An Improved Bat Algorithm based on Novel Initialization Technique for Global Optimization Problem

Waqas Haider Bangyal
Member IEEE SMC
Department of Computer Science,
Iqra University, Islamabad, Pakistan

Jamil Ahmad
Senior Member IEEE, Professor Computer Science
Department of Computer Science,
Kohat University of Science and Technology (KUST),
Kohat, Pakistan

Hafiz Tayyab Rauf
Department of Computer Science,
University of Gujrat, Gujrat, Pakistan

Sobia Pervaiz
Department of Software Engineering,
University of Gujrat, Gujrat, Pakistan

Abstract—**Bat algorithm (BA) is a nature-inspired metaheuristic algorithm which is widely used to solve the real world global optimization problem. BA is a population-based intelligent stochastic search technique that emerged from the echolocation features of bats and created from the mimics of bats foraging behavior. One of the major issue faced by the BA is frequently captured in local optima while handling the complex real-world problems. In this study, a new variant of BA named as improved bat algorithm (I-BAT) is proposed. Improved bat algorithm modifies the standard BA by enhancing its exploitation capabilities, and secondly for initialization of swarm, a quasi-random sequence Torus has been applied to overcome the issue of convergence and diversity. Population initialization is a vital factor in BA, which considerably influences the diversity and convergence of swarm. In order to improve the diversity and convergence, quasi-random sequences are more useful to initialize the population rather than the random distribution. The proposed strategy is applied to standard benchmark functions that are extensively used in the literature. The experimental results illustrate the superiority of the proposed technique. The simulation results verify the efficiency of proposed technique for swarm over the benchmark algorithm that is implemented for the function optimization.**

Keywords—Bat algorithm; local optima; exploration and exploitation; quasi-random sequence

I. Introduction

Optimization of the process that involves searching a vector from a function creates an optimum solution. All possible values are considered as available solutions, while the exceptional value referred as the optimum solution. Generally, optimization algorithms are used to resolve the local and global search optimization issues. Optimization algorithms have two categories: stochastic algorithms and deterministic algorithm [1]. Deterministic algorithms use gradient and generate same solutions for all iterations, which are initiated with the same starting point. Thus, stochastic algorithms generate distinct solutions even if the starting points are same and never uses gradient. Although, the final values which are slightly different are supposed to give the same optimum solutions within a given precision [2]. Stochastic and population-based algorithms have two further parts: Heuristics and Meta-Heuristics [3]. Swarm Intelligence (SI) is one of the nature-inspired meta-heuristic algorithm that is frequently used solve the complex optimization problems. Some traditional neural networks and evolutionary algorithms [4] also use for data classification and optimization.

To handle complex real-world optimization problems, SI nature stimulated technique has been used for many years. Beni [5] was the first who introduced SI, which is inspired with the behavior of birds, fishes, and insects, and their exclusive capability to handle a complex nature of problems in the fashion of swarms. Thus, the same condition would seem complicated if they work individually instead of swarms. Individual bees, ants [6], fishes and birds have limited intelligence, however, when they cooperate with each other for social interaction and interact with the environment, they are capable to accomplish tough tasks, e.g. to get a food source from shortest path and organization of their nests [7].

BA is one of the most famous SI based algorithm was introduced by Xin-She Yang [8], which is inspired by echolocation of micro-bats. Bats produce some echo in the environment, during hunting or flying. By producing an echo, they get an accurate image of the environment and exact location of their prey, due to this reason bats can find their prey in complete darkness [9]. There is a wide range enhanced and improved versions of BA that have been introduced recently. In addition to this, BAT algorithm diversely implemented in various applications of different fields like image processing, engineering design, feature selection and many more [10].

Choosing an initial configuration to initialize the population is one of the primary tasks in evolutionary computing. The performance of evolutionary algorithm may vary due to the different fashion of firing the individuals into the search space

[11]. The swarm covers the more search space; the more there is the fair chance of reaching an optimal solution, Random Initialization of population is usually employed when there is no candidate solution available. The final solution can be improved by selecting the most suitable distribution for population initialization.

Swarm convergence has been considered as the dominant issue for the researchers. So far many researchers have focused to determine whether bat concentrates to the same curve or not. They are assumed to point out the feature that performs the significant role in swarm convergence. Performance of BAT algorithm is intensely concerned by the premature convergence of bats [12]. BA may stuck in local minima due to premature convergence before a global optimum found. To overcome local optima problem, many researchers have proposed various improve methods [13] in which disparate mutations are performed on relevant parameters such as velocity, pulse rate, frequency, loudness and swarm size [14], that help bats to move into a new area of search space. Mutation in BA helps the bats to avoid premature convergence around local optimal [15].

Premature convergence is the main problem with BAT algorithm and other optimization techniques including Evolutionary Algorithms (EAs) such as Gene expression programming (GEP), Genetic Algorithm (GA), Differential Evolution (DE) and Genetic Programming (GP) [16]. Diversification (exploration) and intensification (exploitation) plays an important role in heuristics. Exploitation referred to the local search ability while exploration represents the global search ability of any population-based algorithm [17]. The performance of swarm-based algorithms is highly dependent on the balance between exploitation and exploration. Premature convergence is caused by excessive exploitation and less exploration, while greater exploration and less exploitation may provoke difficulties to reach the optimal solution [18].

The local search ability (exploration) of standard BA is better than the global search ability (exploration). Therefore, to improve exploration capability, we have carried out an improved version of BA called improved bat algorithm (I-BAT). For ensuring the integrity of proposed technique I-BAT is compared with the original BAT on nine well-known benchmark test functions. Experimental result shows that proposed variant has performed well as compared to original BA on specific test functions.

The rest of the paper is structured as: Section 2 presents related work, while working of original Bat algorithm is described in Section 3. Methodology is presented in Section 4 and Section 5 contains discussion and final results for the proposed method. Section 6 presents the conclusion and future work.

II. Related Work

In the field of medical science, P. Kora et al. [19] implemented a new modified BA to extract main features of each cardiac beat. After the extraction of these best features, they are embedded as an input in neural architecture classifier. According to the exhaustive analysis results, it is illustrated that by using optimization on main features, the execution of classifier is significantly improved. Moreover, a novel method of BA described by authors in [20], for parameter estimation in the nonlinear dynamic biological system. The authors included the impact of both Levy Fight and Chaotic dynamics. The optimization is performed on the parameters of secondary system with the use of introduced Chaotic Levy Fight BA to follow the dynamics related to the primary system. Statistical results illustrate the efficiency and stability of proposed algorithm in biological systems. On the other hand, for image recognition J. Zhang in [21] tried to solve image recognition problem that is the reason he proposed a new method Bat Algorithm with Mutation (BAM). In BAM, a modification was embedded during the process of updating the BAs in which BA mutate for the optimal solution.

Paiva *et al.* [22] proposed a new version of BA algorithm with name Modified Bat having the Cauchy mutation and elite opposition Based learning. The objective of this proposed version is to expand convergence velocity and produce the diversity of algorithm. A comparison is conducted with all recent research of BA, and four standard benchmark functions are implemented in the proposed version for the sake of comparison. After the exhaustive analysis, the excellence of proposed version is proved. An Accelerated Bat Algorithm (ABATA) proposed by the authors in [23], where the author used Nelder-Mead approach for local search, to refine optimal best solution in all iterations. Nelder-Mead approach performs a well-defined local search, and able to improve the exploitation abilities in ABATA. The working of ABATA is verified through seven integer programming problems, as well as, compared with four standard algorithms. The results illustrated that ABATA could obtain the optimal global solution in less computational time.

Enhanced method of BAT introduced by authors in [20], for parameter estimation in the nonlinear dynamic biological system. The authors included the impact of both Levy Fight and Chaotic dynamics. The optimization is performed on the parameters of secondary system with the use of introduced Chaotic Levy Fight BAT, to follow the dynamics related to the primary system. Statistical results illustrate the efficiency and stability of introduced algorithm in biological systems.

To sort out the global numerical optimization problem, a robust hybrid metaheuristic optimization method (HS/BA) provided by the authors in [24] that is considered as an improved version of the traditional BA. In the proposed method, a mutation operator was adjusted for maximizing convergence speed. The defined method was verified through fourteen test functions, which described that proposed method outperforms. In [25], authors proposed a new approach of BAT Algorithm with named Hybrid BA Algorithm (HBA). In this approach, the authors merged traditional BAT with Differential Evolution (DE). Standard functions are used for implementation. According to the experimental results, HBA provides improved results than a traditional BAT.

An advanced version of Bat Algorithm called Modified Bat Algorithm (MBA) was proposed in [26]. They introduced modified Bat with enhancement of exploration methodology, in which loudness and pulse emission rate of BATs were changed. Experimental analysis was performed on 15 standard functions, which showed that MBA gives a quality solution related to optimization problems. The authors in [27] introduced a new approach Hybrid Self-Adaptive Bat Algorithm (HSABA). The new approach was carried out by combining Self-Adaptive Bat Algorithm (SABA) and various Differential Evolution (DE) strategies. This new approach was implemented as a heuristic for local search (a modified operator). A comparison was performed with well-defined comparative studies as well as with other standard algorithms. The comparison results certified that HSABA works adequately for enhancing the impact of Population-Based Algorithms.

In [28], authors proposed a new method in the computational intelligence field, which was an improved version of BAT named as IBACH, and used to solve the problems of integer programming. The introduced algorithm implements chaotic behavior to produce BAT solution behavior like the acoustic monopoly. A numerical analysis was conducted with the comparison of other algorithms: PSO, traditional BA, and various harmony search algorithm. Although, the ability of this introduced algorithm was obtained when it computes optimal solution in fewer computations. To avoid from pre mature convergence, directional echolocation was proposed with respect to traditional BAT in [29], which improves the exploitation and exploration abilities of the traditional BA. For improving the performance of BAT, three other enhancements have been included in the BAT. The proposed approach Directional BA Algorithm (dBA), has been verified through various functions belongs to a CEC’2005 standard suite. Similarly, the algorithm was compared with various BAT versions and additional ten algorithms. The results concluded that dBA is better than others.

III. BAT Algorithm

BAT algorithm is a nature-inspired algorithm belongs to SI family, proposed by Xin-She Yang [8]. Bat algorithm works on the echolocation of micro bats and used echo of bats for seeking of food. Yang focused on three rules for the implementation of the bat: Firstly, to measure the distance to the specific point, all bats use echolocation. Secondly, bats fly randomly with fixed frequency towards specified location with specific velocity, however, the loudness and wavelength can vary. Thus, bats automatically adjust their wavelengths according to their target. Thirdly, the author considered that loudness is varied from maximum to minimum rather than any other way.

In bat algorithm each bat of population reveals a candidate solution. Each candidate solution is illustrated with the help of vector $x_i = (x_1, \dots x_i)^t$ with real value elements x_{ij} ,for $i = 1 \dots N_p$and the interval for each element is taken from $x_{ij} \in [x_{lb} \dots x_{ub}]$. While, x_{lb} and x_{ub} determines the upper and lower bounds, however, N_p represents the size of population [27]. The major components of algorithms are initialization, variation operation, local search, evaluation of a solution, and replacement.

Step 1: In initialization, the parameters of an algorithm are initialized, after that it generates an initial population using random distribution, and at last, the best solution is illustrated from that initial population.

Step 2: By using natural rules of bat echolocation, the variation operator is used to move and represent virtual bats in search space and generate new solutions by following equation:

$$f_i^{(t)} = f_{min} + (f_{max} - f_{min})U(0,1) \tag{1}$$

The value of f_{max} and f_{min} depends upon the problem nature, where U $(0,1)$ is uniform random number generator.

Where the updated velocity of particles can be represented by the following equation:

$$v_i^{(t+1)} = v_i^t + (x_i^t - best)f_i^t \tag{2}$$

$best$ is the current globally best location and x_i^t represents current bat position at iteration t .

Bats are moved towards the bats new position in Dimension D with the following equation:

$$x_i^{(t+1)} = x_i^t + v_i^{t+1} \tag{3}$$

Step 3: In local search, current best solution is modified by using random walk with direct exploitation, where following equation is used:

$$x_{new} = best + \epsilon A_i^t \tag{4}$$

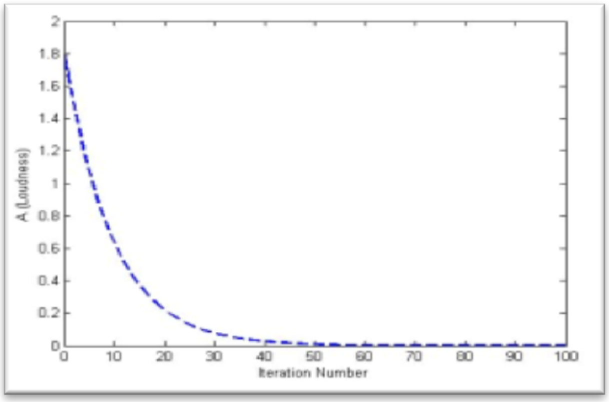

Fig. 1. Changing loudness with respect to iterations.

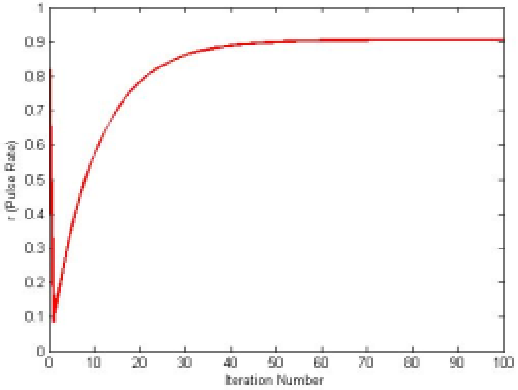

Fig. 2. Changing loudness with respect to iterations.

Here, $^{(t)}_{i}$ is used for loudness and ϵ is a random number between $[-1,1]$ used for scaling factor.

Step 4: Probability of pulse rate r_i is carried out to launch pulse rate. To accept a new best solution, the probability depends upon the loudness A_i^t. Basically, the loudness A_i^t and the pulse rate r_i two parameters are used to control the standard BAT algorithm. Normally, when the population reaches closer the local optimum, then the loudness A_i^treduces (decrease) and pulse rate $r_i$enlarge (increase) (Fig. 1 and 2). When the bat finds its prey, and the loudness increases and pulse rate decreases these both features simulate the natural bats. The equations for decreasing loudness and increasing pulse rate are as follows:

$$^{(t+1)}_{i} = \alpha A_i^{(t)} \tag{5}$$

$$r_i^{(t)} = r_i^0\ [1 - \exp(-\gamma^t)] \tag{6}$$

In above equations, α and γ are constants where α is that parameter which handles the convergence, it is same as simulated annealing algorithm's cooling factor. Algorithm 1 presents the pseudo code for the original Bat algorithm (Fig. 3).

```
_Algorithm_1:Standard_Bat_Algorithm
Input: $x_i = (x_1, ... x_i)^t$ →Bat population
Output: $x^{best}$ & $\min(f(x))$ →optimal solution and minimal objective function value
$x_i$ =Rand_Bat_Init($x_i$); population initialization
Evaluate($x_i$) ;evaluate newly generated population
$Compute(x^{best})$;find current global best
While ($t \leq t_{max}$ ) do
    for $i = 1 ... N_p$ do
        Update frequency using eq.1
        Update velocity using eq.2
        Update position using q.3
          if rand(0,1) > $r_i^{(t)}$ then
          $x_j^t$ = improve solution using eq.4
          end
  if $f_{new}$ = Compute $x_j^t$; generate new solution
  evaluate = evaluate +1;
  if $f_{min}^{new} < f_{old}$ and Rand(0,1) < $A_i^{(t)}$ then
  $x_i = x_j^t$; $f_{old} = f_{min}^{new}$;
  end if
  $f_{min} = \min(f(x^{best})))$;
 end for
end while
```

IV. Methodology

As it has been mentioned above, we have made two major contributions to this study. First, we have introduced one novel methods of initialization of population using low discrepancies sequence that uses the torus quasi-random sequence to create the initialization of the swarm rather random distribution. Second, we have proposed a new strategy of searching for standard bat algorithm by improving the exploitation and convergence capability with controlled parameters.

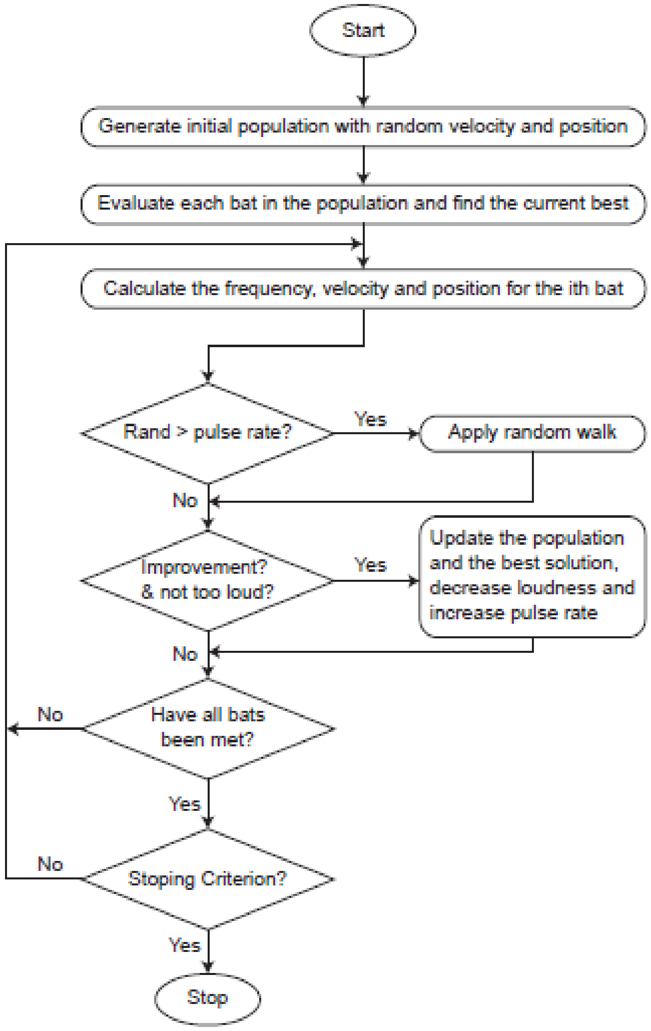
Start
Generate initial population with random velocity and position
Evaluate each bat in the population and find the current best
Calculate the frequency, velocity and position for the ith bat
Rand > pulse rate?
Yes
Apply random walk
No
Improvement? & not too loud?
Yes
Update the population and the best solution, decrease loudness and increase pulse rate
No
Have all bats been met?
No
Yes
Stoping Criterion?
No
Yes
Stop

Fig. 3. Flow chart for standard bat algorithm.

A. Random Number Generator

The function, $Random(a_{min,}a_{max})$ of the built in library is used to generate uniform numbers at random points [30]. Influence of regularity on any sequence is recognized by the probability density function of a constant uniform distribution. Given below is the equation of probability density function:

$$f(p) = \begin{cases} \frac{1}{q-r}\ for\ q < p < r \\ 0 \quad for\ t < q\ or\ p > r \end{cases} \tag{7}$$

Where q and r denotes the features of maximum likelihood. The importance of $f(p)$ is worthless at the edge of q and r, because of 0 impact at the integrals of $f(p)dp$ across any interval. Fig. 4 contains the graphical representation of random number generation following uniform distribution. The probability function of evaluation estimates the evaluation of the parameter of maximum likelihood using the equation below:

$$P(q,r|p) = nlog(r-q) \tag{8}$$

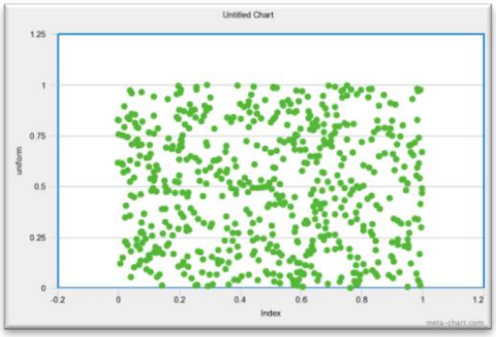

Fig. 4. Random Data Generation using Uniform Distribution [30].

B. The Torus Sequence

The authors in [31] first time, introduced the geometric term Torus for generating a torus mesh that is needed for the geometric correlative system. Torus mesh is generally utilized in the game development community and can be produced using the left-hand or right-hand correlative system. The torus can be represented at 1d, 2d, and 3d by the circle, donut, and 2d quadrilateral sequentially. The equations for torus 3d representation are given below:

$$a(\alpha,\rho) = (d + r_c cos\theta)cos\rho \qquad (9)$$

$$b(\alpha,\rho) = (d + r_c cos\theta)sin\rho \qquad (10)$$

$$c(\alpha,\rho) = r_c sin\rho \qquad (11)$$

α, ρ are their angles of circles and d representing the distance to torus center from tube center, r_c indicates to the circle radius. The author used, R studio has been carried out with the latest version of 3.4.3 using the package "Rand toolbox" to produce torus distribution based random data series. The mathematical representation of torus distribution is as follows:

$$x_z = \left(f\left(z\sqrt{p_1}\right), \dots, f\left(z\sqrt{p_d}\right)\right) \qquad (12)$$

f is a fraction computed by $f = x - floor(x)$ where p_1 indicates the sequences of ith prime number. Prime parameters allow only 100,000 dimensions, for more than 100,000 dimensions a manual configuration will be needed. Fig. 5 contains the graphical representation of random number generation following uniform distribution.

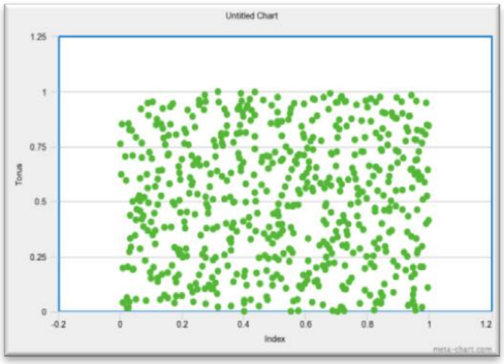

Fig. 5. Random data generation using torus distribution [31].

C. Improved Bat Algorithm

To enhance the local search capability and to sustain the divergence of population so that the algorithm will avoid to trap in local optima, there is no bound to variate the BA through mutation adoption, parameter selection or hybridized the bat algorithm with other algorithms. Although, the process of modification in BA remained with the same issue as it never ensures a modified BA each time achieves a global optimum solution. To reduce this issue, in this paper, proposed improved bat algorithm (I-BA), which enhance the exploitation ability of bats and sustain the divergence of population to acquire consistent results. In this study is presented two modifications with objective to improve the exploration and exploitation abilities of bat algorithm for the improvement of its performance.

- *Novel Initialization Approach Torus*

In this study, we have proposed following one novel method of population initialization approach using low discrepancies sequence Torus named as (TO-BA) that uses the torus quasi-random sequence to create the initialization of the swarm. It can be seen that selection of robust distribution for population initialization may enhance the convergence rate. For population creation, random uniform distribution usually used to initialize the swarm. Generating Torus random population at random location as $\text{Torus}(\text{P}_\text{n}, \text{P}_\text{d})$.

Where P_n and P_d population size and population are dimension respectively. The addition in the standard bat version is as:

$$x_i = \text{Torus\_Bat\_Init}\ (x_i); \qquad (13)$$

- *Enhanced Local Search Method*

We introduced next modification by focusing on local search method. In conventional BA, swarm's bats are granted to travel from their present position to new random position by applying local random walk. Local search pattern plays an important role to determine the local optimum. In (4), the local search will only focus on the neighbors of best position obtained by the entire swarm. However, if this position is far from the global optimal solution, then some local search capabilities will be useless due to the inefficient exploitation. To overcome this, we modify random walk by the following equation:

$$x_{new} = best + \varepsilon A_i^t * (0.1 * N(0,1) * (x_i^{(t)} - Iter * best)) \qquad (14)$$

Where A_i^t representing average loudness and ε is random number over the interval of [-1, 1], 0.1 is controlled factor of exploitation. $N(0,1)$ represents random number generated by Gaussian distribution over the interval of $[0,1]$. $Iter$ is described as current iteration in Jth dimension and $best$ is global best position of bat.

Our enhanced local search approach is proposed for overcoming this exploitation problem, i.e., improves exploitation ability around the best solution of bats and preserves population diversity and obtains a steady result. The algorithm for I-BAT can be found in Algorithm 2.

Algorithm_2: Improved_Bat_Algorithm

```
Input: $x_i = (x_1, \dots x_i)^t$ →Bat population
Output: $x^{best}$ & $\min(f(x))$ →optimal solution and minimal objective function value
$x_i$ =Torus_Bat_Init($x_i$); population initialization
Evaluate($x_i$) ;evaluate newly generated population
$Compute(x^{best})$;find current global best
While ($t \leq t_{max}$ ) do
    for $i = 1 \dots N_p$ do
        Update  frequency using eq.1
        Update velocity  using eq.2
        Update  position using q.3
        if rand(0,1) > $r_i^{(t)}$ then
            $\boldsymbol{x_j^t}$ = im
prove solution using eq.14
        end
    if $f_{new}$ = Compute $\boldsymbol{x_j^t}$; generate new solution
    evaluate = evaluate +1;
    if $f_{min}^{new} < f_{old}$ and Rand(0,1) < $A_i^{(t)}$ then
    $x_i = \boldsymbol{x_j^t}$; $f_{old} = f_{min}^{new}$;
    end if
    $f_{min} = \min(f(x^{best}))$;
  end for
end while
```

V. Results and Discussion

The proposed improved Bat algorithm (I-BAT) is implemented on the machine with the specification of 2.3 GHz Core (M) 2 Duo CPU processor. To assure the robustness and integrity of proposed algorithms, a collection of nine benchmark test functions has been employed to perform the comparison of proposed I-BAT with Standard Bat algorithm. These are the standard nonlinear benchmark functions usually carried out to investigate the performance of any population-based algorithm in terms of convergence speed, exploitation, and exploration capability. Table I represents the definition of benchmark test function and the properties of these functions. In Table I, $\boldsymbol{f}^*$ shows global minimum of the objective functions$\boldsymbol{f}$, where $\boldsymbol{x}^*$ represents the possible minimum values. The experimental results for proposed techniques are presented in Table II.

A. Parameter Setting

Simulation parameters are fixed as: Population size is 40 where the dimensions for all functions are set to 10, 20 and 30. A number of iterations for 10, 20 and 30 dim are 1000, 2000 and 3000 respectively. For relatively fair and effective results, all techniques have been implemented to similar parameters. All techniques were examined for 30 runs to compare the performances.

B. Analysis

The objective of this research work is to improve the collective performance of bat algorithm by applying two basic modifications in standard bat algorithm. We have modified the method of population initialization and also improved the exploration capability of bat algorithm. The goal of the research is to find how the nature of simulation results relies on dimensions of the benchmark functions for the optimization. For this, dimensions taken for different functions presented in Table I are D=10, D=20, and D=30.

TABLE I. Definitions of Bench mark Test Functions and Properties of Bench-mark Functions

$\boldsymbol{f}$	**Function name**	**Definition**	$\boldsymbol{f}^*$	$\boldsymbol{x}^*$	**Domain**
f_1	Sphere	$Min\, f(x) = \sum_{i=1}^{n} x_i^2$	0.00	(0,0,0…,0)	$-5.12 \leq x_i \leq 5.12$
f_2	Axis parallel hyper-Ellipsoid	$Min\, f(x) = \sum_{i=1}^{n} i.x_i^2$	0.00	(0,0,0…,0)	$-5.12 \leq x_i \leq 5.12$
f_3	Schumer Steiglitz	$Min\, f(x) = \sum_{i=1}^{n} x_i^4$	0.00	(0,0,0…,0)	$-5.12 \leq x_i \leq 5.12$
f_4	Schwefel 1.2	$Min\, f(x) = \sum_{i}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	0.00	(0,0,0…,0)	$-100 \leq x_i \leq 100$
f_5	Rotated hyper-Ellipsoid	$Min\, f(x) = \sum_{i}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	0.00	(0,0,0…,0)	$-65.536 \leq x_i \leq 65.536$
f_6	Moved axis parallel hyper-Ellipsoid	$Min\, f(x) = \sum_{i=1}^{n} 5i.x_i^2$	0.00	(5*i)	$-5.12 \leq x_i \leq 5.12$
f_7	Sum of different power	$Min\, f(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	0.00	(0,0,0…,0)	$-1 \leq x_i \leq 1$
f_8	Sum Squares Function	$Min\, f(x) = \sum_{i=1}^{n} i.x_i^2$	0.00	(0,0,0…,0)	$-10 \leq x_i \leq 10$
f_9	Zakharov Function	$Min\, f(x) = \sum_{i=1}^{n} x_i^2 + (1/2 \sum_{i=1}^{n} i.x_i)^2 + (1/2 \sum_{i=1}^{n} i.x_i)^4$	0.00	(0,0,0…,0)	$-5 \leq x_i \leq 10$

TABLE II. COMPERISON OF STANDARD BAT AND IMPROVED BAT

			Standard-BAT					**I-BAT**				
Functions	***Iterations***	***DIM***	***Best***	***Worst***	***Mean***	***Median***	***Std. Dev.***	***Best***	***Worst***	***Mean***	***Median***	***Std. Dev.***
F1	1000	10	7.70E-08	2.16E-07	1.52E-07	1.49E-07	4.08E-08	3.26E-11	8.46E-11	**5.76E-11**	5.47E-11	**1.56E-11**
	2000	20	8.22E-08	2.40E-07	1.38E-07	1.29E-07	4.22E-08	8.30E-10	3.14E-09	**2.19E-09**	2.28E-09	**5.85E-10**
	3000	30	1.29E-07	2.37E-07	1.60E-07	3.25E-08	1.49E-07	3.33E-09	6.77E-09	**5.34E-09**	5.38E-09	**8.78E-10**
F2	1000	10	8.22E-08	1.18E-06	5.93E-07	4.65E-07	2.52E-07	1.64E-09	4.79E-09	**3.03E-09**	2.78E-09	**9.44E-10**
	2000	20	9.14E-07	2.78E-06	1.57E-06	1.41E-06	2.56E-07	8.74E-09	2.74E-08	**1.37E-08**	9.96E-09	**5.61E-09**
	3000	30	2.25E-06	3.23E-06	**3.53E-06**	5.26E-06	**1.18E-06**	6.56E-08	4.26E-05	8.75E-06	9.49E-08	1.62E-05
F3	1000	10	2.84E-15	1.25E-14	7.84E-15	6.97E-15	3.43E-15	1.99E-20	1.11E-18	**3.68E-19**	3.71E-19	**3.03E-19**
	2000	20	1.47E-15	4.51E-15	2.68E-15	2.48E-15	7.72E-16	2.49E-19	1.22E-18	**6.53E-19**	5.12E-19	**2.88E-19**
	3000	30	1.35E-15	3.54E-15	2.31E-15	1.99E-15	7.38E-16	1.85E-18	9.07E-18	**4.41E-18**	3.10E-18	**2.38E-18**
F4	1000	10	1.34E+03	7.31E+03	3.57E+03	2.49E+03	2.06E+03	1.31E-02	9.37E-02	**6.00E-02**	6.62E-02	**2.46E-02**
	2000	20	1.13E+04	2.37E+05	8.69E+04	4.53E+04	7.99E+04	7.28E+02	1.75E+03	**1.38E+03**	1.39E+03	**2.89E+02**
	3000	30	4.50E+04	5.43E+05	2.48E+05	1.45E+05	1.80E+05	4.42E+03	9.64E+03	**7.32E+03**	7.58E+03	**1.53E+03**
F5	1000	10	3.49E+03	7.30E+06	1.49E+06	1.90E+05	2.32E+06	8.59E-19	8.89E-08	**2.37E-08**	5.70E-18	**3.66E-08**
	2000	20	3.94E+07	2.43E+08	1.44E+08	1.43E+08	5.67E+07	5.81E-17	2.51E-07	**8.17E-08**	5.94E-08	**7.79E-08**
	3000	30	3.00E+08	1.06E+09	7.18E+08	7.04E+08	2.06E+08	7.91E-07	1.56E-06	**1.27E-06**	1.26E-06	**2.21E-07**
F6	1000	10	4.11E-07	5.90E-06	3.21E-06	2.32E-06	1.88E-06	8.21E-09	2.39E-08	**1.52E-08**	1.39E-08	**4.72E-09**
	2000	20	4.57E-06	1.39E-05	7.94E-06	7.06E-06	2.67E-06	4.37E-08	1.37E-07	**6.83E-08**	4.98E-08	**2.80E-08**
	3000	30	1.13E-05	2.63E-05	1.72E-05	1.62E-05	4.69E-06	7.05E-07	1.92E-06	**1.22E-06**	9.46E-07	**4.27E-07**
F7	1000	10	1.93E-04	7.91E-04	5.00E-04	4.77E-04	1.70E-04	4.77E-07	3.54E-06	**1.47E-06**	1.06E-06	**1.07E-06**
	2000	20	5.68E-04	1.32E-03	9.24E-04	8.23E-04	2.05E-04	1.57E-07	2.65E-06	**1.36E-06**	1.36E-06	**7.13E-07**
	3000	30	7.14E-04	1.81E-03	1.24E-03	1.11E-03	3.52E-04	4.33E-07	2.86E-06	**1.59E-06**	1.31E-06	**7.53E-07**
F8	1000	10	8.99E-08	1.02E-06	6.77E-07	6.10E-07	2.73E-07	1.64E-09	6.19E-09	**4.01E-09**	4.09E-09	**1.58E-09**
	2000	20	7.98E-07	1.75E-06	1.44E-06	1.43E-06	2.81E-07	2.57E-08	1.03E-07	**5.28E-08**	4.63E-08	**2.10E-08**
	3000	30	2.17E-06	5.87E-06	3.85E-06	3.39E-06	1.05E-06	6.84E-07	1.29E-06	**9.39E-07**	9.27E-07	**1.58E-07**
F9	1000	10	1.11E-07	4.36E-07	3.27E-07	3.35E-07	1.06E-07	3.57E-09	9.61E-09	**5.31E-09**	4.98E-09	**1.56E-09**
	2000	20	1.91E-07	2.62E-02	2.62E-03	3.06E-07	7.87E-03	9.34E-06	1.34E-05	**1.22E-05**	1.21E-05	**1.14E-06**
	3000	30	1.51E+00	1.86E+02	3.46E+01	1.23E+01	5.14E+01	2.08E-05	1.05E-04	**5.89E-05**	5.31E-05	**2.61E-05**

To compare an algorithm with other algorithm for the same nature of problem, its true value will be seen. Hence, I-BAT is compared with standard BA. Both algorithms were used to solve same standard test functions presented for Table I and utilized the same parameter settings explained in Section V. The best simulation results are highlighted as bold. For the fair comparison, the proposed technique improved BA (I-BAT), compared with the standard bat algorithm on nine well-known benchmark test functions. The comparative results are given in Table II. From Table II, we can see that the performance of I-BAT is better as compared to standard Bat algorithm in terms of exploration capability of bats. After a brief analysis on the basis of results, we can conclude that the proposed methods are enough robust in nature to be used for the purpose of numerical optimization problems. The experimental results show that I-BAT outperforms over standard BA in all nine bench mark test functions. Below is the graphical representation of proposed methods on all benchmark test functions. From Fig. 6 to 14 the graphical representation of detailed comparison for $\boldsymbol{f_1}$ to $\boldsymbol{f_9}$ has been shown.

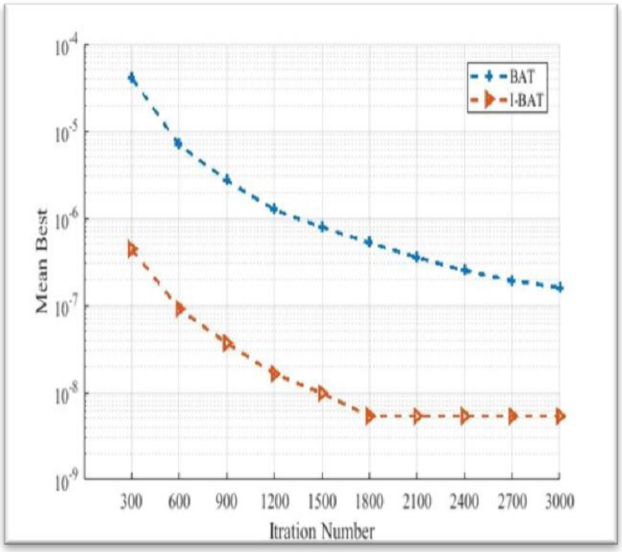

Fig. 6. I-BAT Curve on $\boldsymbol{f_1}$.

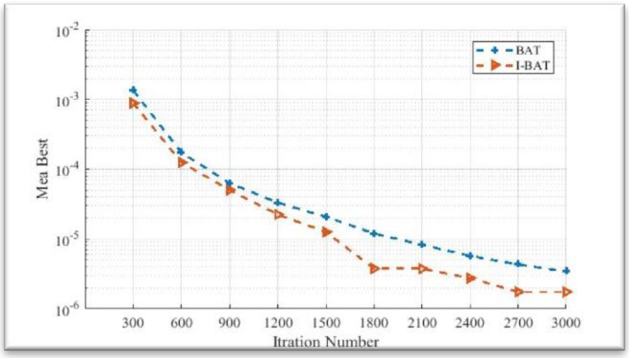

Fig. 7. I- BAT Curve on f_2.

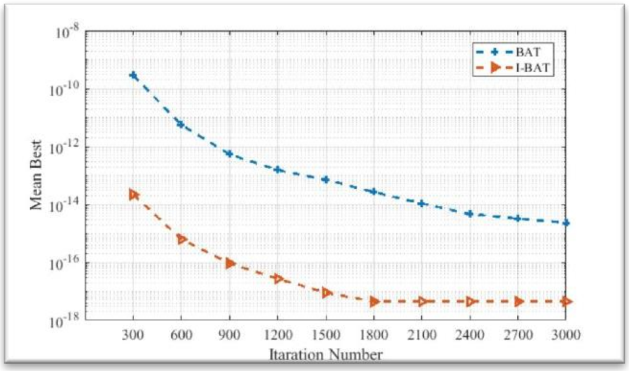

Fig. 8. I- BAT Curve on f_3.

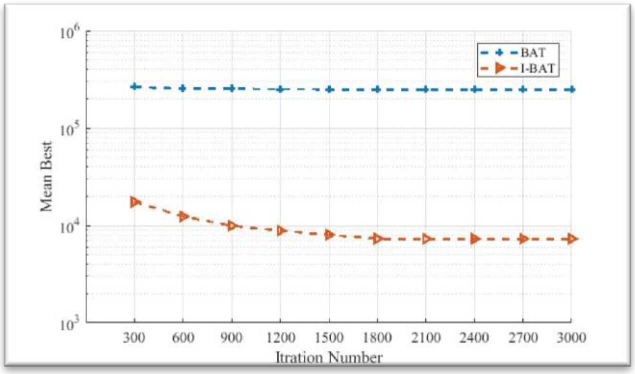

Fig. 9. I- BAT Curve on f_4.

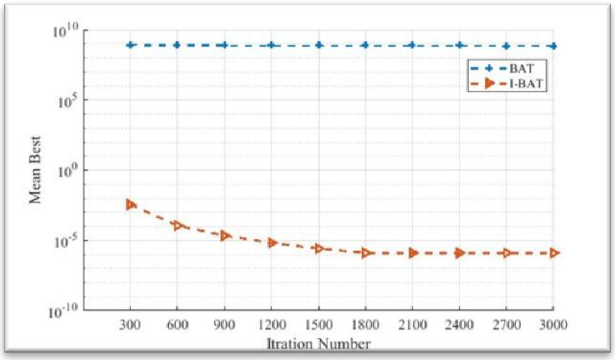

Fig. 10. I- BAT Curve on f_5.

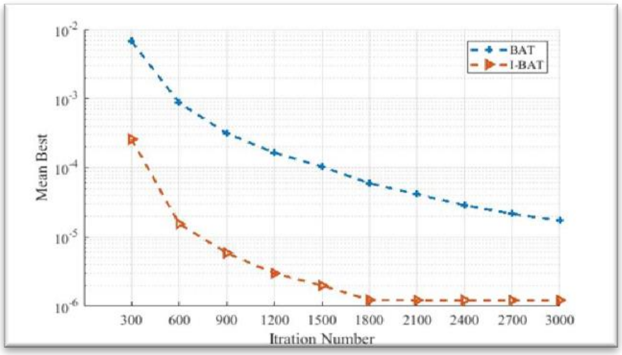

Fig. 11. I- BAT Curve on f_6.

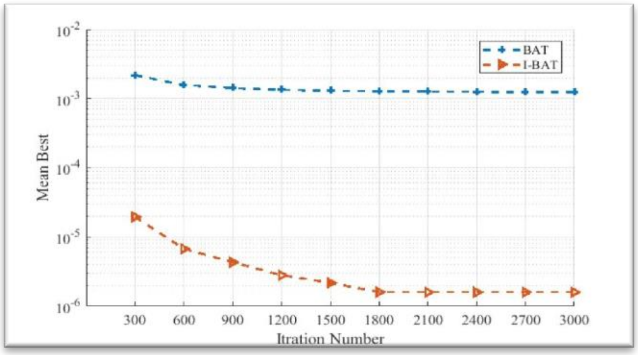

Fig. 12. I- BAT Curve on f_7.

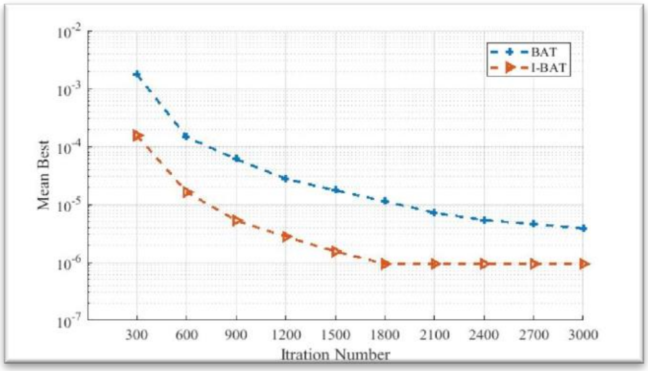

Fig. 13. I- BAT Curve on f_8.

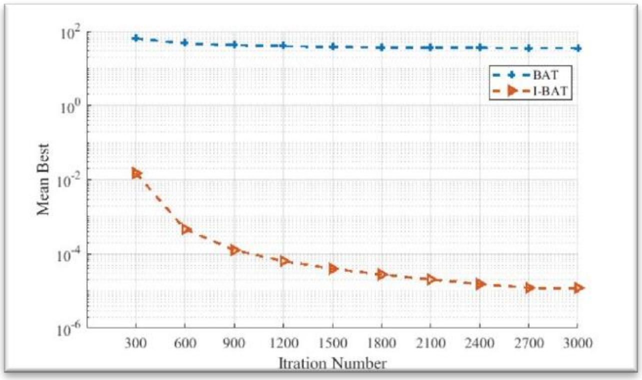

Fig. 14. I- BAT Curve on f_9.

VI. Conclusion

To overcome the issues of exploitation and exploration capabilities of conventional BA, a novel variant of Bat algorithm I-BAT is proposed that would also enhance the searching ability to avoid the local optimum. The novel variant comprises standard BA with strong searching capability joined with novel quasi-random sequence Torus for initialization of swarm and employed on function optimization problems. The proposed strategy maintains the diversity of the swarm and improves the local searching capability. The simulation result shows that the proposed technique has better convergence accuracy and can escape from premature convergence successfully. It also depicts that our designed technique is much better, when it is compared with the standard BA. For future consideration it is appealing to check the performance of new initialization proposed approaches for higher dimensional problems.

References

[1] Fogel, "An introduction to simulated evolutionary optimization," IEEE transactions on neural networks, vol. 5, no. 1, pp. 3-14, 1994.

[2] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," IEEE Transactions on evolutionary computation, vol. 4, no. 3, pp. 284-294, 2000.

[3] X. S. Yang, "Review of meta-heuristics and generalised evolutionary walk algorithm," International Journal of Bio-Inspired Computation, vol. 3, no. 2, pp. 77-84, 2011.

[4] W. H. Bangyal, J. Ahmad, I. Shafi, and Q. Abbas, "A forward only counter propagation network-based approach for contraceptive method choice classification task," Journal of Experimental & Theoretical Artificial Intelligence, vol. 24, no. 2, pp. 211-218, 2012.

[5] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in Robots and Biological Systems: Towards a New Bionics?, Springer, Berlin, Heidelberg, 1993, pp. 703-712.

[6] Christian Blum and Xiaodong Li, "Swarm Intelligence in Optimization," in Swarm Intelligence., 2008, pp. 43-85.

[7] Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," SciVersa ScienceDirect, vol. 30, pp. 625-642, 2011.

[8] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010) , Springer, Berlin, Heidelberg, 2010, pp. 65-74.

[9] Xin-She Yang and Amir Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," Engineering Computations: International Journal for Computer-Aided Engineering and Software, vol. 29, pp. 464-483, 2012.

[10] Xin-She Yang, "Bat Algorithm for Multi-objective Optimisation," Int. J. Bio-Inspired Computation, Vol. 3, No. 5, pp.267-27, vol. 3, pp. 267-274, 2012.

[11] S.M. Rahnamayan, H. R. Tizhoosh, and M. Salama, "A novel population initialization method for accelerating evolutionary algorithms," Computers & Mathematics with Applications, vol. 53, no. 10, pp. 1605-1614, 2007.

[12] Nazir Mohd Nawi, M. Z. Rehma, Abdullah Khan, Haruna Chiroma, and Tutut Herawan, "A Modified Bat Algorithm Based on Gaussian Distribution for Solving Optimization Problem," Journal of Computational and Theoretical Nanoscience, vol. 13, pp. 706-714, 2016.

[13] G. G. Wang, B. Chang, and Z. Zhang, "A multi-swarm bat algorithm for global optimization," in Evolutionary Computation (CEC), 2015 IEEE Congress on IEEE, 2015, pp. 480-485.

[14] X. S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," Engineering Computations, vol. 29, no. 5, pp. 464-483, 2012.

[15] M. W. U. Kabir and M. S. Alam, "Bat algorithm with self-adaptive mutation: a comparative study on numerical optimization problems," International Journal of Computer Applications, vol. 100, no. 10, 2014.

[16] Russell, Eberhart, and Y. H. Shi, "Comparison between genetic algorithms and particle swarm optimization," in International conference on evolutionary programming. Berlin, 1998, pp. 611-616.

[17] S. Yilmaz and E. U. Kucuksille, "Improved bat algorithm (IBA) on continuous optimization problems," Lecture Notes on Software Engineering, vol. 1, no. 3, p. 279, 2013.

[18] G. Rudolph, "Self-adaptive mutations may lead to premature convergence," IEEE Transactions on Evolutionary Computation, vol. 5, no. 4, pp. 410-414, 2001.

[19] Padmavathi Kora and Sri Ramakrishna Kalva, "Improved Bat algorithm for the detection of myocardial infarction," SpringerPlus, vol. 4, p. 666, 2015.

[20] Jiann-Horng Lin, Chao-Wei Chou, Chorng-Horng Yang, and Hsien-Leing Tsai, "A Chaotic Levy Flight Bat Algorithm for Parameter Estimation in Nonlinear Dynamic Biological Systems," Journal of Computer and Information Technology, vol. 2, pp. 56-63, 2011.

[21] Jiawei Zhang and Gaige Wang, "Image Matching Using a Bat Algorithm with Mutation," Applied Mechanics and Materials, vol. 203, pp. 88-93, 2012.

[22] Fábio A. P. Paiva, Marcos H. F. Marcone, Cláudio R. M. Silva, and Izabele V. O. Leite, "Modified Bat Algorithm With Cauchy Mutation and Elite Opposition-Based Learning," in Computational Intelligence (LA-CCI), 2017 IEEE Latin American Conference on., 2017, pp. 1-6.

[23] Ahmed Fouad Ali, "Accelerated Bat Algorithm for Solving Integer Programming Problems," Egyptian Computer Science Journal, vol. 39, pp. 25-40, 2015.

[24] Gaige Wang and Lihong Guo, "A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization," Journal of Applied Mathematics, vol. 2013, 2013.

[25] Iztok Fister Jr., Dusan Fister, and Xin-She Yang, "A Hybrid Bat Algorithm," arXiv preprint arXiv:1303.6310, 2013.

[26] S. Yılmaz, E. Ugur Kucuksille, and Y. Cengiz, "Modified Bat Algorithm," vol. 20, pp. 71-78, 2014.

[27] Iztok Fister Jr., Simon Fong, Janez Brest, and Iztok Fister, "A Novel Hybrid Self-Adaptive Bat Algorithm," The Scientific World Journal, 2014.

[28] Osama Abdel-Raouf, Mohamed Abdel-Baset, and Ibrahim El-Henawy, "An Improved Chaotic Bat Algorithm for Solving Integer Programming Problems," IJCEM International Journal of Computational Engineering & Management, vol. 17, pp. 2230-7893, 2014.

[29] Asma CHAKRI, Rabia KHELIF, Mohamed BENOUARET, and Xin-She YANG, "New directional bat algorithm for continuous optimization problems," Expert Systems with Applications, vol. 69, pp. 159-175, 2017.

[30] S. Kotz, N. Balakrishnan, and N. L. Johnson, Continuous multivariate distributions, Models and applications ed.: John wiley & sons, 2004, vol. 1.

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