Research Article

Sanjoy Debnath*, Wasim Arif, and Srimanta Baishya

Buyer Inspired Meta-Heuristic Optimization Algorithm

https://doi.org/10.1515/comp-2020-0101
Received Oct 17, 2019; accepted Mar 16, 2020

Abstract: Nature inspired swarm based meta-heuristic optimization technique is getting considerable attention and established to be very competitive with evolution based and physical based algorithms. This paper proposes a novel Buyer Inspired Meta-heuristic optimization Algorithm (BIMA) inspired form the social behaviour of human being in searching and bargaining for products. In BIMA, exploration and exploitation are achieved through shop to shop hoping and bargaining for products to be purchased based on cost, quality of the product, choice and distance to the shop. Comprehensive simulations are performed on 23 standard mathematical and CEC2017 benchmark functions and 3 engineering problems. An exhaustive comparative analysis with other algorithms is done by performing 30 independent runs and comparing the mean, standard deviation as well as by performing statistical test. The results showed significant improvement in terms of optimum value, convergence speed, and is also statistically more significant in comparison to most of the reported popular algorithms.

Keywords: Nature Inspired; Optimization Algorithm; Swarm Intelligence

1 Introduction

Optimization has become an integral part of scientific and engineering problems in order to maximize the output performance of a given system under a set of defined constraints. As compared to conventional deterministic approach, swarm-based stochastic optimization approach has found significant attention due to simplicity, less complexity, speed and robustness in finding optimal solutions of a given function in various applications [1, 2]. Recently developed swarm-based optimization technique based on the behavior of social agents like ants, bees, fish, birds etc. are gaining attention among the research community. The foundation of popular meta-heuristic optimization algorithms was created in 1995 when Eberhart and Kennedy developed Particle Swarm Optimization (PSO) [3], a novel solution for solving complex problems inspired from the behavior of a flock of birds. PSO was found very effective in solving single-objective as well as multi-objective constraint optimization problems in various engineering fields [4–6] and [7]. Subsequently in 2004, Ant Colony optimization [8] was developed, which was inspired by the natural behavior of ants and was found very effective while searching for optimal solutions in structure-based problems. In 2012, inspired by the social behavior of bee colonies, Akay and Karaboga proposed an optimization technique which is found to be very effective in solving constraint engineering design problems [9]. Accordingly, based on the social behavior of animals, authors have proposed effective, less complex, robust optimization techniques such as Bat algorithm [10], Firefly algorithm [11], Whale Optimization Algorithm (WOA) [12], Grey Wolf optimization [13], Ageist Spider Monkey optimization [14], Moth Search optimization [15], Moth Flame optimization algorithm [16], ant Lion optimizer [17], Salp Swarm algorithm (SSA) [18] and Dragonfly algorithm [19], which are some of the widespread swarm-based meta-heuristic optimization algorithms in literature.

On the other hand, a hybrid optimization algorithm aims to minimize complexity, improve stability, enhance convergence speed and provide better accuracy in comparison to standard algorithms [20]. Any hybrid optimization technique thus developed has contributed to notable improvement in the performance measuring matrices of the parent algorithm. The dynamic, self-adaptive and diverse learning strategy of the swarm agent makes the algorithm more robust to deal with diverse situations. There

***Corresponding Author: Sanjoy Debnath:** Electronics and Communication Engineering, National Institute of Technology Silchar, Assam, India; Email: sanjoydebnath80@gmail.com
Wasim Arif: Electronics and Communication Engineering, National Institute of Technology Silchar, Assam, India; Email: arif.ece.nits@gmail.com
Srimanta Baishya: Electronics and Communication Engineering, National Institute of Technology Silchar, Assam, India; Email: sb@ece.nits.ac.in

is a lot of scope for improving the exploration and exploitation rate with the key tools of newly developed optimization techniques. The PSO, due to its lower complexity and high exploration rate, has become a very popular optimization algorithm among researchers for various applications. In [21], author proposed an improved version of PSO algorithm namely fitness-based multi-role PSO (FMPSO), where the particle velocity parameter of the algorithm is updated through subsocial-learning component. To enhance the diversity in the swarm, in [22], authors used logistic map-based initialization, sigmoid-like inertia weight and wavelet mutation to the worst particle in the swarm in order to avoid premature saturation. The effective performance of any optimization algorithm, including PSO, largely depends on the selection of system parameters. In [23], authors proposed a Fuzzy Self-Tuning based PSO (FST-PSO) algorithm where the algorithm itself finds the best optimal parameter setting for the global best solution. In ALC-PSO [24], an aging leading and comparison parameter was introduced to the parent PSO technique in such a way that when the particles are getting aged, their leading capacity is checked by generating new challenger and comparing. Through improving the inertial weight of PSO, an adaptive weighted PSO is proposed in [25] and found to be best suited for real time engineering problems. Combining the features of an artificial bee colony and PSO, a hybrid technique namely PS-ABC [26] was proposed and the result shows its superiority to PSO in terms of speed and convergence.

In an evolutionary algorithm, Differential Evolution (DE) [27] is mostly preferred for solving problems with real valued parameters and since finding an optimal hyperplane is a hard computing task, this metaheuristic (MH) is chosen to conduct an intelligent search of a near-optimal solution. DE is chosen for its significant features like high exploration rate, fast convergence and lower complexity. DE-based evolutionary algorithm with affinity propagation clustering is proposed in [28], where authors proposed a dual strategy mutation scheme which efficiently balances the exploration and exploitation rates to avoid local optima.

To enhance the performance of the algorithm, the diversification of the population plays an important role. Considering this dual population based framework, a bee colony algorithm (BCA) is proposed in [29], where one set of population is responsible for exploitation and another for exploration through the diversification of the population. In [30], authors described a novel algorithm to improve the exploitation capability of the BCA, by invoking Force model which is inspired by the Gravity model. By incorporating mutation and crossover strategy into the artificial bee colony, algorithm variables are updated self adaptively each time for improving the search mechanism [31]. Artificial BCA is further improvised in [32] with the innovative search mechanism for finding out the global optimal solution in complex problems.

Finding the optimal solution in multi-modal functions where many local optima are present is treated as a considerably difficult job for meta-heuristic optimization algorithms [33]. To avoid getting trapped in the local optima, there should be proper trade-off between exploration and exploitation of the algorithm. High population diversity based PSO algorithm with the inclusion of a learning mechanism and local search strategy is proposed in [34], where each particle is inspired and learning from different neighborhood particles rather than traditional personal best only. Further to improve the population diversity of the swarm, bottleneck objective learning (BOL) strategy is applied to the PSO algorithm for finding high convergence rate in a large set of complex objective functions [35].

Solving real life problems is an important and necessary task for a state-of-the-art optimization algorithm. On this basis, a random walk grey wolf optimizer (RW-GWO) is proposed in [36] where in updating of position, each well fitted wolf of α and β is updated by the most prominent and leading wolves and avoids premature convergence. In [37], authors presented a hybrid meta-heuristic algorithm, where the position of salp swarm is updated using the position equations of sine cosine algorithm (SCA) for improving the global search ability of the SSA. As fewer parameters need to be tuned in GWO algorithm, a PSO-inspired efficient and robust GWO algorithm is proposed in [38], which is capable of solving large scale numerical problems of optimization. In engineering problem solving, the criteria for selecting optimization are based on less complexity and computational cost effectiveness [39]. The recently proposed Dragonfly Algorithm (DA) by Mirjalilii *et al.* [19] which was inspired by the static and dynamic features of dragonflies in nature, provides very competitive results as compared to the state-of-the-art algorithms in literature. In DA, exploration simulates the dynamic swarming behavior of dragonflies upon encountering an enemy as well as their levy flight search, which ensures the diversity of the dragonfly solution. In order to improve exploitation and avoid premature convergence, some features like addition of memory was proposed in [40]. But the proposed model failed to address the issues of computational complexity and convergence at local minima. The authors in [41] deliberated the use of diversity in order to improve the global convergence of PSO.

This motivated us to design an innovative optimization algorithm inspired by the peculiar ability of humans to purchase the best quality product at minimum cost and subsequent bargaining for optimum cost and quality with the location of shops, reviews of products as key parameters. We developed a novel optimization algorithm for linear, non-linear and non-differentiable optimization problems.

In our proposed model, the cost and choice provide exploration, which is otherwise known as dynamic swarm behavior, and quality and user reviews provide exploitation, which is otherwise known as static swarm behavior, as features to the algorithm. Thereafter, the buyer step size is formulated, which provides diversity to the population in finding the global best solution. Therefore, the algorithm guarantees exploration at the early steps and exploitation at the later steps and confirms global optimum with improved accuracy. The proposed algorithm was tested on standard benchmark functions and engineering design problems in order to establish its validity. The detailed comparison is presented which proves the superiority of the algorithm over popular optimization algorithms in literature.

The rest of the paper is organized as follows. Section 2 consists of the description of the proposed algorithm. In Section 3, formulation of the proposed algorithm is presented. Section 4 and Section 5 consist of the performance evaluation and convergence analysis of the proposed algorithm. In Section 6, the performance analysis of the algorithm on benchmark engineering design problems is described and a detailed comparison is deliberated. Finally, a conclusion is drawn in Section 7.

2 Inspiration

The main inspiration for this algorithm is a bargainer or buyer who tries to purchase the best product from a different number of shops available in the market. A group of bargainers/buyers are considered as a *swarm*. Depending on different parameters and the corresponding fitness value, the buyer will move from one shop to the other in order to find the best product and use the result of other buyers to find the optimal shop. The worst scenario for one buyer in buying a product is measured as the position of the shop which results in a bad product in terms of price and quality. The worst position of the buyer is used for prioritizing the choice factor. While updating the process of the position towards the global best position, a buyer has to avoid the worst position of the respected space. As shown in Figure 1, in our proposed algorithm, the total area and the swarm population are initially partitioned and characterized into different sets of zones/clusters. Based on the population density, the cluster radius may be varied. All the assigned swarm particles (buyers) of one zone search their local cluster area which provides the exploitation to the algorithm. Through iteration/time the cluster radius is gradually increased and finally converted into one cluster (similar to buyers' nature to learn from their neighbors and update their choices over time) which provides the exploration rate to the algorithm.

The shops are randomly distributed over a specific search space and hypothetically located near the buyer's location. Generally, every buyer prefers to buy from the nearby shops. But for a higher profit and a desired product, even marginally, buyers explore shops located at a larger distance based on their neighbor buyers' reviews. Here in BIMA the main aim is to find out the best shop (location in the search space) which could provide the best quality product at optimum price.

Zones are basically the region that is within radius r of the current position of buyers in the search space.

Here, initially each buyer forms a zone around themselves of calculated radius ($r_{initial}$) to search locally. Gradually with evolution, the cluster size increases and buyers move towards the global optimal solution. At the end of the iteration all the buyers came in the same cluster of radius r_{inital} ($r_{final} \gg r_{initial}$) as shown in Figure 1.

As shown in the Figure 1(b), all the buyers are randomly distributed initially and through searching of the global best shop the buyers are attracted towards the global best shop. Thus, at the end of the iterations or meetings of the search criteria as shown in Figure 1(e), all the buyers updated their position and converge on the global best shop or the best solution.

3 The Proposed BIMA Algorithm

The behavior of the swarm of buyers depends on the following principle components:

3.1 Cost

In general, one buyer will try to buy the best product at minimum price. Therefore, the buyer hops from shop to shop in order to buy the best product at a minimum cost in the search space.

Mathematically it can be calculated as follows:

$$C_{D,i} = gBestX_{D,1} - X_{D,i} \tag{1}$$

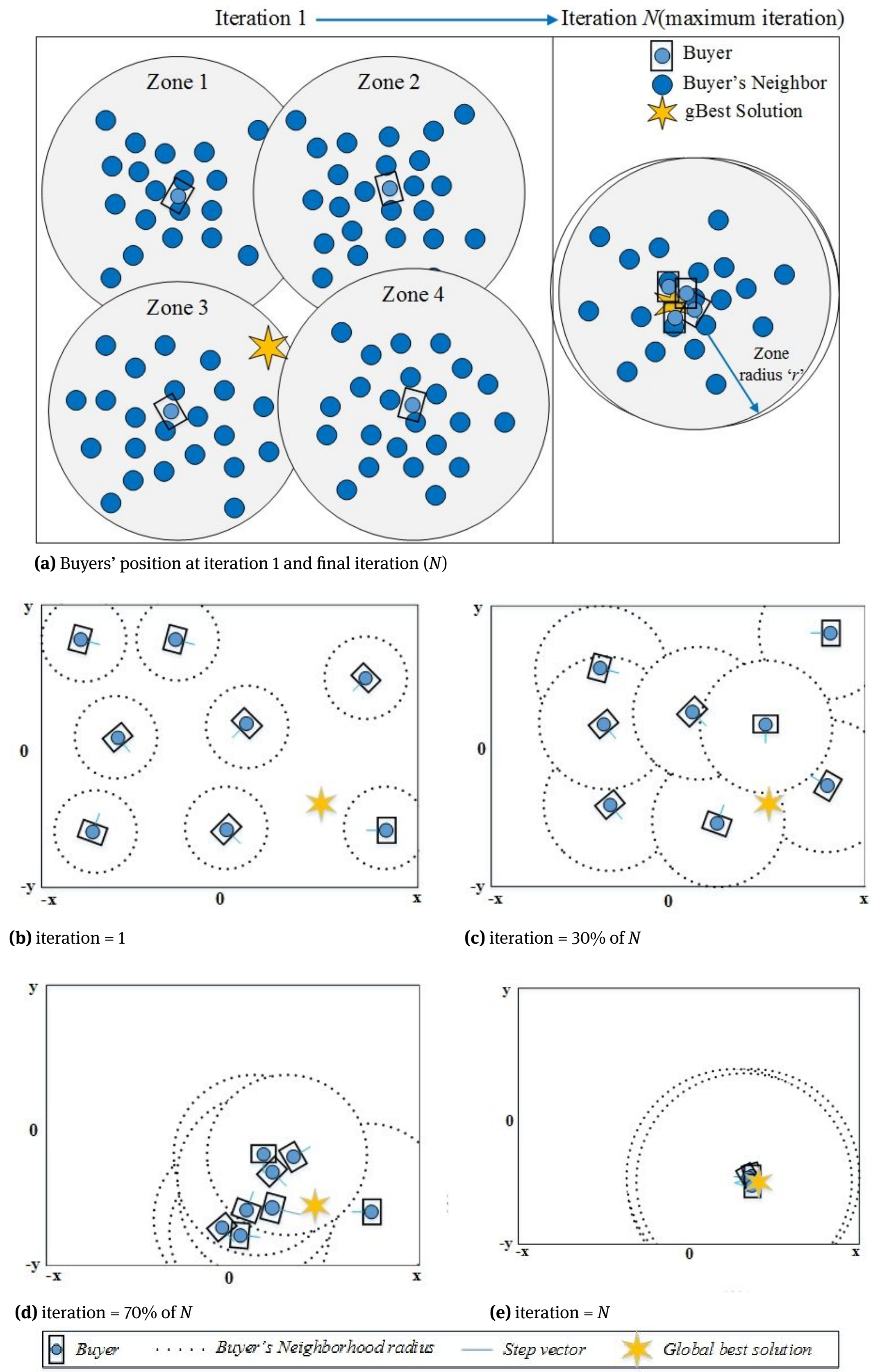

Figure 1: Buyer's zone up-gradation process

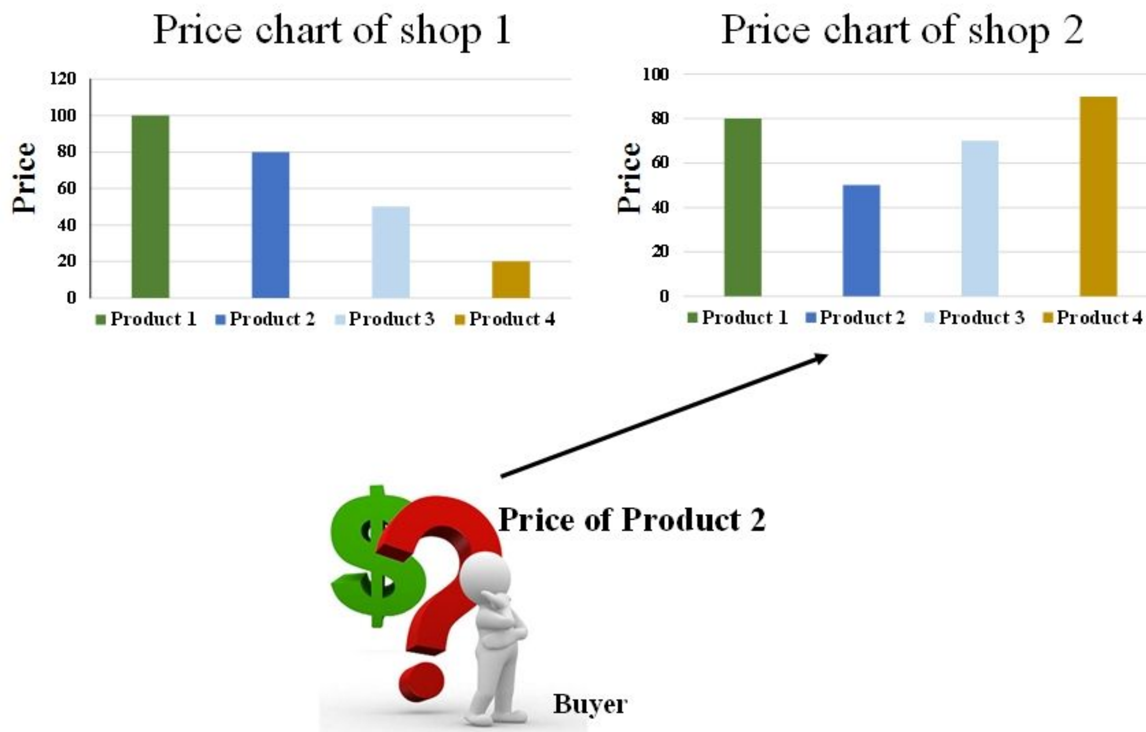

Figure 2: Based on cost of the product buyer tends to move towards the best shop

where *gBestX* is the best position in the swarm, $X_{D,i}$ is the position of the i^{th} individual and D is the dimension of the search space. The global best (gB_{est}) of the swarm represents the best buyer position, thus the distance of the considered buyer from the best buyer is treated as the cost to be afforded by the individual buyer and is represented by Eq. (1).

3.2 Quality

This is an important factor which plays a major role in buying a product. Most buyers are immensely concerned with the quality of the product and perhaps a buyer hops from shop to shop in order to ensure buying the best quality product.

Mathematically, it can be designated as follows:

$$Q_{D,i} = pBestX_{D,i} - X_{D,i} \tag{2}$$

where $pBestX_{D,i}$ is the personal best position of the buyer i and $X_{D,i}$ is the position of the current individual. The physical significance of the term *pBest* is that it stores the pointer addressing to the earlier shop wherein the buyer found the best quality product before their present shop. Thus, for an individual buyer i, the difference between the buyer's current position and *pBest* provides the self-learning strategy to the buyer. The self-learning ability of each individual buyer improves the quality of the product to be searched and depicts the quality parameter of BIMA.

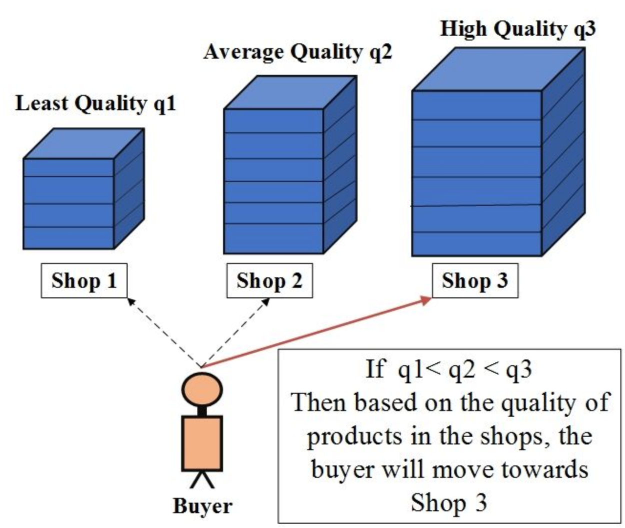

Figure 3: Based on quality of the product the buyer move towards the best shop

3.3 Choice

Individuals will make different choices to buy a particular product. The choice of a product based on some subjective parameters such as color, aesthetic, brand etc. is an important factor for buyers when buying that particular product. In that manner, a buyer is determined to buy the best-choice product by avoiding the worst one.

Mathematically, it can be calculated as follows:

$$H_{D,i} = gWorstX_{D,1} - X_{D,i} \tag{3}$$

where $gWorstX$ is the worst position of the buyer swarm in terms of choice and X_i is the current position of the buyer.

The choice of one buyer basically depends on the current trends of products and the buyer's personal perception. Thus, the global worst (gW_{orst}) solution is used to formulate the choice parameter (H) of the product. The difference between the gW_{worst} solution and the buyer's current position provides an extra weight to divert an individual buyer from the worst product. Therefore, with this parameter an individual buyer is diverted from the worst product and directed towards the best individual.

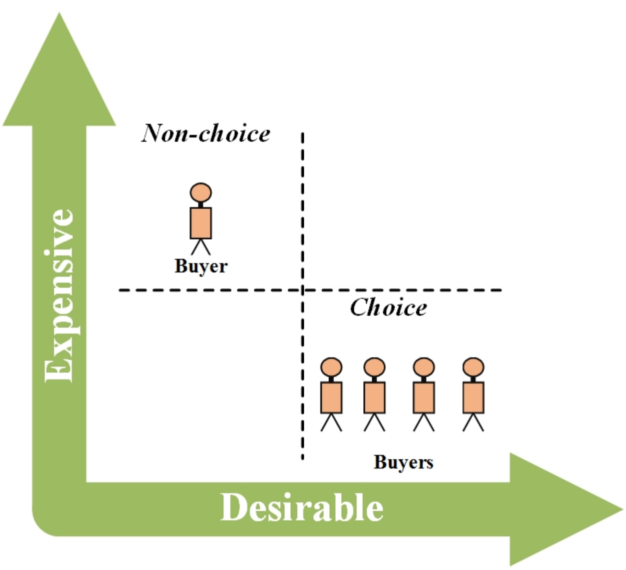

Figure 4: Based on the Choice of the product, the buyer has to move towards the best shop

3.4 User Review Attraction

In any locality, the best shop is considered to be the shop where the probability of getting the best quality of products is higher when compared to other shops. Buyers are attracted towards the shop which their neighbors visited and where they bought products. Buyers rate such shops with higher ratings which subsequently attract other buyers towards the shop to buy products. The information collected from the neighbor buyer provides mutual learning to the algorithm. The term User Review Attraction (R) plays a significant role in the selection of proper direction by the buyer to move towards the position of the best shop.

Mathematically shop attraction can be calculated as:

$$R_{D,i} = -\sum_{j=1}^{L_i} Neighbor\_pBestX_{D,j} - X_{D,i} \tag{4}$$

where L_i is the number of neighbors in the neighbor radius of a buyer i, $X_{D,i}$ is the position of the current individual, and *Neighbor_pBestX* is the neighbor's best reported shop. Thus, the difference between the individual buyer's current position and the neighborhood buyer *Neighbor_pBestX* provides the mutual learning strategy to the algorithm to attract towards the best solution.

The selection of the route towards the best shop satisfying all the desired needs of a buyer is largely described by four parameters: *cost, quality, choice and user review attraction*.

In BIMA, buyers are hopping locally in search of getting a better probability of a desired product which represents the exploitation feature of the algorithm and is primarily governed by the quality of the product (Q), and reviews of the neighbor buyer (R) parameters. BIMA achieves a high exploration rate by exploring the maximum area of the search space through maintaining minimum cost (C) and suitable choice (H). The neighborhood buyer plays an important role in the search of the global optimal product. In the initial phase of the evolution, each individual buyer searches in their own cluster of radius r, that radius gradually increases with the generation as per the proposed mathematical Eq. (5).

$$radius = \frac{ub - lb}{4} + \left[(ub - lb)\left(\frac{iteration}{Maximum\_iteration}\right) 2\right] \tag{5}$$

To update the position of buyers in a search space and corresponding movements, two vectors are considered: *stepsize* (ΔX) and *position* (X). The *step* and *position* vector of the proposed model are analogous to the velocity and current position vector of PSO.

Unlike PSO, the step vector in BIMA provides the direction of movement of the buyers and is developed as follows:

$$\Delta X_{g+1} = w\Delta X_g + c_1 C_{D,i} + qQ_{D,i} + c_2 H_{D,i} + sR_{D,i} \tag{6}$$

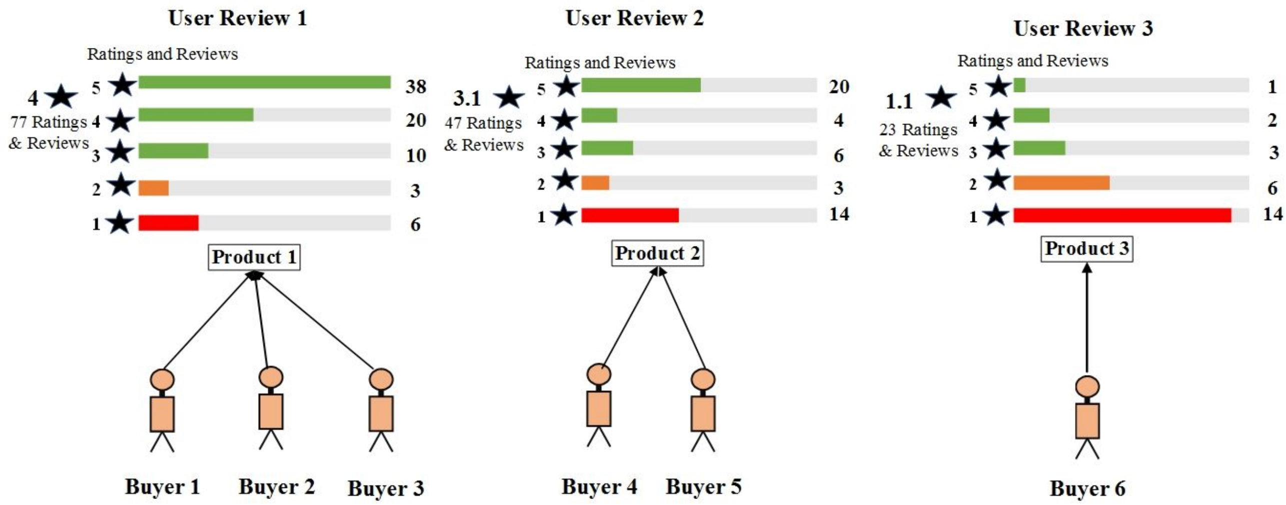

Figure 5: Attraction towards best shop based on user reviews

where, c_1 indicates the cost weight, q is the quality weight, c_2 is the choice factor, s is the neighbor review factor of the i^{th} individual, and g represents the current generation. The inertia weight (w) can be written as:

$$w = 0.9 - iteration * (0.9 - 0.4)/Maximum\_iteration \quad (7)$$

where the parameter *iteration* represents the current iteration/generation (g).

After calculating the step vector, the position vectors are calculated as follows:

$$X_{g+1} = X_g + \Delta X_{g+1} \quad (8)$$

where g is the current iteration/generation.

With cost, quality, choice and review (c_1, q, c_2 and s), different explorative and exploitative behaviors can be achieved during optimization. Neighbors of the buyer are very important, as a neighborhood buyer zone is created around each buyer considering a cluster representing a different group of buyers with a certain finite radius.

A question may arise here as to how the convergence of the buyer is guaranteed during optimization. The bargainers/buyers are required to change their weights adaptively for transiting from exploration to exploitation in the search space. In our proposed BIMA, proper balance between exploration and exploitation is achieved through adaptive tuning of the swarming factors (c_1, q, c_2, s, and w) during optimization. The tuned values of the swarming factors used in our experiment are adaptively evaluated from the value of ΔX which gradually decreases with the number of iterations. Intuitive understanding of the algorithm supported by several rounds of experimentation leads us to formulate the parameter ΔX and accordingly c_1, q, c_2, s, and w are evaluated as per the following equations:

$$c_1 = 1.2 \times rand(1) \times \Delta X \quad (9)$$

$$q = 1.2 \times rand(1) \times \Delta X \quad (10)$$

$$c_2 = rand(1) \quad (11)$$

$$s = rand(1) \times \Delta X \quad (12)$$

where $rand(1)$ is the randomly generated number in [0, 1]

It is obvious that the buyer observed and learned from his neighborhood buyers in order to select the best route to reach a shop. This resembles the general human behavior to follow the best individual and learn from them. Every buyer is bounded in a cluster containing a group of buyers to observe and learn from the best individuals. Through iteration, the cluster area of each buyer is increased, and the number of clusters is reduced. At the final stage of optimization, it forms a single cluster containing the best individual of buyers to reach the globally optimum shop for the product to be purchased.

The cost and quality are evaluated from the global best and worst solution of the whole set of the buyer swarm. The choice of the product is evaluated from the personal best solution of a buyer in the search space during the iteration. In this manner the buyers are able to gradually converge towards the best area in the search space and diverge away from the non-promising areas of the search space.

3.5 Algorithm of proposed BIMA

1: *Initialize* maximum iteration (N), number of search agents (NP), dimensions (*dim or n*), upper bound (ub) and lower bound (lb) of variable, vector constant *F=0.5*, population vector (X) *and* step vectors (ΔX) *as 0.4X*
2: Calculate *fitness* value ***for*** each Bargainer/buyer
3: ***for*** each Bargainer/buyer
4: *Evaluate fitness*(i)
5: *Initialize pBest*(i) = *fitness*(i)
6: *end* ***for***
7: ***While*** maximum iterations not attain
8: Calculate *radius of buyers neighbor zone and 'w'* as in *Eq.* (5) and (7) respectively
9: Evaluate *fitness* of each buyer
10: ***if*** *fitness*(i) < *pBest*(i) this iteration
11: Update *pBest*(i) as *pBest*(i) = *fitness*(i)
12: Update *pBestX*
13: *end* ***if***
14: Evaluate *gBest* and *gBestX* as *gBest = min(pBest), gBestX = argMin(pbest)*
15: Evaluate *gWorst* and *gWorstX* as *gWorst = max(pBest), gWorstX = argMax(pbest)*
16: *Listing neighbor of each buyer, Calculate Cost Parameter C* (according to Eq. (1)), *Calculate Quality Parameter Q* (according to Eq. (2)), *Calculate Choice Parameter H* (according to Eq. (3)), *Calculate Review Parameter R* (according to Eq. (4))
17: ***for*** each buyer
18: Update weight c_1, q, c_2, and s as in *Eq.* (9), *Eq.* (10), *Eq.* (11) *and Eq.* (12)
19: Update buyers zone *radius*
20: Update F as $F = F - F \times r$
21: $\Delta X_{g+1} = w\,\Delta X_g + [c_1 C + qQ + c_2 H + sR]$ as *Eq.* (6)
22: $\delta_i = F\left(X_i + \Delta X_{g+1}\right)$
23: ***if*** buyer has neighbor in the zone
24: ***for*** $j = 1 : dim$
25: ***if*** *rand*(1) < F
26: $T_i^j = \delta_i^j$
27: ***Else***
28: $T_i^j = X_i^j$
29: *end* ***if***
30: *end* ***for***
31: ***if*** $f(T_i) < f(X_i)$
32: $X_i = T_i$
33: *end* ***if***
34: ***Else***
35: Update X vector using *Eq.* (8)
36: *end* ***if***
37: *end* ***for***

3.6 Description of the algorithm

After initialization of all the parameters and constants in *step*-1, the algorithm evaluates the initial fitness value of all the buyers in *step*-2 to *step*-6. Then it updates the radius of each buyer and correspondingly evaluates the current fitness in *step*-9 and updates the *pBest* (*step*-10 to 13), *gBest* (*step*-14) and *gWorst* (*step*-15). Then in *step*-16, the algorithm evaluates the key parameter of the algorithm, namely Cost Parameter C, Quality Parameter Q, Choice Parameter H and Review Parameter R for each buyer of the swarm. Then for each buyer the weight co-efficients c_1, q, c_2 and s corresponding to cost, quality, choice, and review parameters are updated in *step*-19. In *step*-21 the step vector is updated, which directs the buyer position towards the optimal shop. In *step*-22 to *step*-30, the algorithm checks the boundary condition and evaluates the trial vector from the calculated step vector of buyer. In *step*-31 to *step*-36, current positions of the buyers are updated.

If a buyer fails to find neighborhood buyers, it moves in a random direction in the search space using a random walk (*Lévy flight*) algorithm which improve the randomness, stochastic behavior, and exploration of the population.

Under this situation, a buyer updates its position as follows:

$$X_{g+1} = X_g + Levy(D) \times X_g \tag{13}$$

where X is the position of the bargainer, and D is the dimension of the position vectors.

The *Lévy* flight is basically a random walk whose step size is governed by the *Lévy* distribution. According to *Lévy* flight the position X is calculated as follows [42]:

$$\text{Lévy}(X) = 0.01 \times \frac{R_1 \times \sigma}{|R_2|^{1/\lambda}} \tag{14}$$

where R_1, R_2 are two random numbers existing in [0,1], λ is a constant between 0 and 2 ($0 < \lambda \le 2$) (λ is equal to 1.5 in this work), and σ is calculated on the basis of Eq. (15):

$$\sigma = \left(\frac{\Gamma(1+\lambda) \times sin\left(\frac{\pi\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) \times \lambda \times 2^{\left(\frac{\lambda-1}{2}\right)}} \right)^{1/\lambda} \tag{15}$$

where Γ(z) is the gamma function, $\Gamma(k) = (k-1)!$, when z = k is an integer.

4 Performance Evaluation

The performance of BIMA is evaluated on the basis of 23 benchmark functions, as shown in Table 1, where functions F1 to F7 are unimodal functions, functions F8 to F13 are multimodal functions, F14 to F23 are fixed dimensional multi-modal benchmark functions [40].

Table 1: Description of Unimodal and Multimodal Functions

Sl no.	UNIMODAL FUNCTIONS	Dim	Range [lb, ub]	F_{min}
1	$F_1(X) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
2	$F_2(X) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[−100, 100]	0
3	$F_3(X) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	[−100, 100]	0
4	$F_4(X) = \max\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[−100, 100]	0
5	$F_5 = \left[100\left(X_{i+1} - {X_i}^2\right) + (X_i - 1)^2\right]$	30	[−30, 30]	0
6	$F_6(X) = \sum_{i=1}^{n} [(X_i + 0.5)]^2$	30	[−100, 100]	0
7	$F_7(X) = \sum_{i=1}^{n} i{X_i}^4 + random[0, 1)$	30	[−1.28, 1.28]	0
	MULTIMODAL FUNCTIONS			
8	$F_8(X) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[−500, 500]	−418.98D
9	$F_9(X) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	[−5.12, 5.12]	0
10	$F_{10}(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[−32, 32]	0
11	$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600, 600]	0
12	$F_{12}(X) = \frac{\Pi}{\pi}\left\{\begin{matrix}(10\sin(\Pi y_i) + \sum_{i=1}^{n-1}(y_{i-1})^2 \\ [1 + 10\sin^2(\pi y_{i+1})] + (y_{n-1})^2\end{matrix}\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[−50, 50]	0
13	$F_{13}(X) = 0.1\left\{\begin{matrix}\sin^2(3\Pi x_i) + \sum_{i=1}^{n}(x_{i-1})^2 \\ [1 + \sin^2(3\Pi x_{i+1})] \\ +(x_{n-1})^2\left[1 + \sin^2(2\Pi x_n)\right]\end{matrix}\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)]$	30	[−1.28, 1.28]	0
	FIXED DIMENSIONAL MULTI-MODAL FUNCTIONS			
14	$F_{14} = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[−65, 65]	1
15	$F_{15} = \sum_{j=1}^{11}\left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$	4	[−5, 5]	0.00030
16	$F_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
17	$F_{17} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[−2, 2]	3
18	$F_{18} = \left[1 + (x_1 + x_2 + 1)^2\begin{pmatrix}19 - 14x_1 + 3x_1^2 - 14x_2 \\ +6x_1x_2 + 3x_2^2\end{pmatrix}\right]$ $\left[30 + (2x_1 - 3x_2)^2 \times \begin{pmatrix}18 - 32x_1 + 12x_1^2 + 48x_2 \\ -36x_1x_2 + 27x_2^2\end{pmatrix}\right]$	2	[−2, −2]	−3.86
19	$F_{19} = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[1, 3]	−3.86
20	$F_{20} = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0, 1]	−3.32
21	$F_{21} = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + C_i\right]^{-1}$	4	[0, 10]	−10.1532
22	$F_{22} = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + C_i\right]^{-1}$	4	[0, 10]	−10.4028
23	$F_{23} = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + C_i\right]^{-1}$	4	[0, 10]	−10.5363

Table 2: Comparison between BIMA and other algorithms on optimizing benchmark functions for 30 independent runs

Function		I / II	MHDA / FEP	DA / SSA	PSO / WOA	GWO / GSA	DE / Proposed
		Unimodal Benchmark Function					
F_1	I	Mean	4.07E−42	5.15E−07	0.000136	6.59E−28	8.2E−14
		Std	2.22E−41	2.82E−06	0.000202	6.34E−05	5.9E−14
	II	Mean	0.00057	9.8447e−09	1.41E−30	2.53E−16	**0.00E+00**
		Std	0.00013	8.9132e−09	4.91E−30	9.67E−17	**57.282**
F_2	I	Mean	6.62E−15	4.818E−06	0.042144	7.18E−17	1.5E−09
		Std	3.61E−14	2.50E−05	0.045421	2.91E−02	9.9E−10
	II	Mean	0.0081	0.0359	1.06E−21	0.055655	**0.00E+00**
		Std	0.00077	3.1979e−05	2.39E−21	0.194074	**5.4664**
F_3	I	Mean	2.55E−50	5.366E−07	70.12562	3.29E+06	6.8E−11
		Std	1.3E−49	2.939E−06	22.11924	79.14958	7.4E−11
	II	Mean	0.016	31.6049	5.39E−07	896.5347	**0.00E+00**
		Std	0.014	0.0024	2.93E−06	318.9559	**68.0245**
F_4	I	Mean	4.99E−05	1.349E−04	1.086481	5.61E−07	0.0E+00
		Std	2.73E−04	4.57E−04	0.317039	1.32E+00	0.0E+00
	II	Mean	0.3	2.2605	0.072581	7.35487	**0.00E+00**
		Std	0.5	6.0967e−05	0.39747	1.741452	**34.7607**
F_5	I	Mean	3.34E−22	6.71E−01	96.71832	2.65E+01	**0.0E+00**
		Std	5.67E−22	3.66E+00	60.11559	69.90499	**0.0E+00**
	II	Mean	5.06	29.2188	27.86558	67.54309	5.0135
		Std	5.87	1.2303e−04	0.76362	62.22534	7.1234
F_6	I	Mean	**0.0E+00**	9.047E−06	0.000102	8.17E−01	0.0E+00
		Std	**0.0E+00**	3.31E−05	8.28E−05	1.26E−04	0.0E+00
	II	Mean	0.0E+00	6.7786e−09	3.116266	2.5E−16	0.3010
		Std	0.0E+00	1.1389e−08	0.53242	1.74E−16	3.0362
F_7	I	Mean	**5.25E−05**	4.5E−04	0.122854	2.22E−02	0.00463
		Std	**5.02E−05**	5.71E−04	0.044957	1.00E−01	0.0012
	II	Mean	0.1415	0.0826	0.001425	0.089441	0.024337
		Std	0.3522	0.5459	0.00114	0.04339	0.22929
F_8	I	Mean	−2957.34	−3932.76	−4841.29	−6123.1	−11080.1
		Std	3.86E+02	−3932.76	1152.814	4.08E+04	574.7
	II	Mean	−12554.5	−8.282e+03	−5080.76	−2821.07	**−4442.3483**
		Std	52.6	6.7502e−04	695.7968	493.0375	**4521.8888**
F_9	I	Mean	5.901E−07	3.36E−02	46.70423	3.12E−01	69.2
		Std	3.23E−06	1.81E−01	11.62938	4.74E+01	38.8
	II	Mean	0.046	35.8185	0.0000	25.96841	**0.00E+00**
		Std	0.012	4.2856e−09	0.0000	7.470068	**2.3343**
F_{10}	I	Mean	**6.34E−15**	2.66E−04	0.276015	1.06E−13	9.7E−08
		Std	**2.72E−14**	8.59E−04	0.50901	7.78E−02	4.2E−08
	II	Mean	0.018	2.2210	7.4043	0.062087	7.99E−15
		Std	0.0021	2.1773e−09	9.897572	0.23628	21.0599
F_{11}	I	Mean	2.39E−04	3.83E−03	0.009215	4.49E−03	0.0E+00
		Std	2.25E−02	7.15E−02	0.007724	6.66E−03	0.0E+00
	II	Mean	0.016	0.0099	0.000289	27.70154	**0.00E+00**
		Std	0.022	6.4693e−08	0.00158	5.040343	**0.00E+00**
	I	Mean	**2.34E−31**	7.48E−04	0.006917	5.34E−02	7.9E−15

F_{12}		Std	**4.45E−47**	3.75E−04	0.026301	2.07E−02	8E−15
	II	Mean	9.2E−06	4.6283	0.339676	1.799617	8.9E−15
		Std	3.6E−06	3.0276e−05	0.21486	0.95114	25.7025
F_{13}	I	Mean	**1.39E−32**	1.06E−03	0.006675	6.54E−01	5.1E−04
		Std	**5.57E−48**	3.99E−04	0.008907	4.47E−03	4.8E−04
	II	Mean	0.00016	0.0439	1.889015	8.899084	5.1E−14
		Std	0.000073	2.5607e−09	0.26608	7.126241	27.1233
Fixed Dimension Multi-Modal Benchmark Function							
F_{14}	I	Mean	0.9880	0.9980	3.627168	4.042493	0.99800
		Std	0.9880	0.9980	2.560828	4.252799	3.3E−16
	II	Mean	1.22	0.9980	2.111973	5.859838	**0.9980**
		Std	0.56	1.0381E−15	2.498594	3.831299	**39.6545**
F_{15}	I	Mean	0.0013	0.0015	0.000577	**0.000337**	4.5E−14
		Std	0.0011	0.0017	0.000222	**0.000625**	0.00033
	II	Mean	0.0005	0.0012	0.000572	0.003673	2.01E−05
		Std	0.00032	2.8878e−11	0.000324	0.001647	1.2249
F_{16}	I	Mean	−1.0316	−1.0316	−1.03163	−1.03163	−1.03163
		Std	1.23E−11	1.0316	6.25E−16	−1.03163	3.1E−13
	II	Mean	−1.03	−1.0316	−1.03163	−1.03163	**−1.0316**
		Std	4.9E−07	3.5228e−12	4.2E−07	4.88E−16	**2.3604**
F_{17}	I	Mean	0.3979	0.3979	0.397887	0.397889	0.39788
		Std	5.01E−04	0.3979	0.0E+00	0.397887	9.9E−09
	II	Mean	0.398	0.3979	0.397914	0.397887	**0.3978**
		Std	1.5E−07	2.1985e−12	2.7E−05	0.0E+00	**5.9735**
F_{18}	I	Mean	3.0000	3.0000	3.00	3.00003	3.00
		Std	2.9890	3.0000	1.33E−15	3.00	2E−15
	II	Mean	3.02	3.0000	3.00	3.00	**3.0000**
		Std	0.11	5.8085e−11	4.22E−15	4.17E−15	**3.0662**
F_{19}	I	Mean	−3.8628	−3.8628	−3.86278	−3.86263	N/A
		Std	0.8628	3.8628	2.58E−15	−3.86278	N/A
	II	Mean	−3.86	−3.3220	−3.85616	−3.86278	**−3.8619**
		Std	0.000014	1.9490E−12	0.002706	2.29E−15	**4.4195**
F_{20}	I	Mean	−3.1936	−3.1936	−3.26634	−3.28654	N/A
		Std	3.1901	3.1936	0.060516	−3.25056	N/A
	II	Mean	−3.27	−10.1532	−2.98105	−3.31778	**−3.3199**
		Std	0.059	5.9610e−10	0.376653	0.023081	**3.5723**
F_{21}	I	Mean	−5.0552	−5.0552	−6.8651	−10.1514	−10.1532
		Std	5.0553	5.0552	3.019644	−9.14015	3.00E−06
	II	Mean	−5.52	−5.1288	−7.04918	−5.95512	**−10.1532**
		Std	1.59	1.4741e−10	3.629551	3.737079	**14.3918**
F_{22}	I	Mean	−10.3742	−10.3742	−8.45653	−10.4015	−10.4029
		Std	1.21E−08	10.3742	3.087094	−8.58441	3.9E−07
	II	Mean	−5.53	−10.4029	−8.18178	−9.68447	**−10.4028**
		Std	2.12	5.4429e−10	3.829202	2.014088	**15.772**
F_{23}	I	Mean	−10.5364	−10.5364	−9.95291	−10.5343	−10.5364
		Std	1.87E−03	10.5364	1.782786	−8.55899	1.9E−07
	II	Mean	−6.57	−10.5364	−9.34238	−10.5364	**−10.5364**
		Std	3.14	6.1826e−10	2.414737	2.6E−15	**15.846**

4.1 Experimental Results

In this section, we analyze the performance of BIMA on 13 unimodal and multi-modal benchmark functions and 10 fixed dimensional benchmark functions mentioned above with the recently developed and most popular population-based optimization techniques, namely MHDA [40], DA [19], PSO [3], GWO [13], DE [27], Fast Evolutionary Programming (FEP) [43], SSA [18], WOA [12] and Gravitational Search Algorithm (GSA) [44]. Performance is evaluated on the basis of 50 search agents and the dimension of the search agent was set to 30, setting levy flight constant to 1.5 and simulating the evaluation for 30 independent runs having 500 iterations each. Comparison of the results are made on the basis of mean and standard deviation (Std) of the optimal solution. The detailed comparisons of the functional evaluations (FEs) are presented in Table 2, where the optimal results are in bold.

From the experimental results of the algorithms on benchmark functions, the following observations are made:

- **Unimodal functions (F1 -F7)**: Such functions are used to assess the exploitation ability of optimization algorithms. The proposed BIMA algorithm outperforms other algorithms in four benchmark functions (F1-F4) out of seven. In function F7 BIMA becomes the sixth best, in function F5 BIMA becomes the fourth best and in function F6 BIMA becomes the eight best out of the ten state-of-the-art heuristic algorithms compared. The results indicate the convergence capability and accuracy of the algorithm.
- **Multimodal Functions (F8-F13):** These functions are very useful to assess the exploration ability of the heuristic algorithms. DADE accomplishes better results than other algorithms in three functions (F8, F9 and F11) out of six cases and at least secures the second best position in functions F10 and F13, the third best in function F12. Results show the exploration capability of the proposed algorithm. The hybridization with DE provides a better divergence to the swarm as well as convergence span control to reach an optimal solution rapidly comparing to other algorithms.
- **Fixed Dimension Multi-modal Function (F14-F23):** This set of functions shows the ability to achieve the defined fitness value. From observing the results it may be concluded that the Proposed DADE outperforms in nine benchmark functions (F14, F16, F17, F18, F19, F20, F21 F22 and F23) out of ten functions. It is able to secure at least the second best position in function F15. Results shows the superiority of the algorithm in finding the near optimal solution of the function.

Here, we analyzed the performance of BIMA on 30 benchmark functions of CEC2017 [45] with the recently developed and most popular population-based optimization techniques, namely DA [19], GWO [13], PSO [3], DE [27], SSA [18], LSHADE [46], L-SHADE SPACMA [47] L-SHADE-cnEpSin [48] and CMA-ES [49]. Performance is evaluated on the basis of 50 search agents and the dimension of the search agent was set to 30 and simulated the evaluation for 30 independent runs having 500 iterations each. The detailed comparison of the functional evaluation (FEs) is presented in Table 3.

The algorithm was tested for 51 runs, where each run consisted of a total 1000×D evaluations. The objective is represented by the minimization of the error value. The error value is the difference between the optimal desired value and the value obtained through function evaluation. The error value is treated as zero (0) if the difference between the optimal solution and the obtained solution is less than or equal to 10^{-8}. In Table 3, the best optimal value of error and corresponding standard deviation obtained by the algorithms are marked in bold. For D=30, the experimental results of the algorithms on CEC2017 benchmark functions listed in Table 3 show that the proposed BIMA algorithm outperforms other algorithms in twelve benchmark functions (F1, F2, F3, F6, F7, F9, F12, F16, F18, F23, F24 and F28). BIMA becomes the second in four benchmark functions (F4, F15, F19 and F22). In functions F26, F27 and F29, BIMA becomes the third best and in functions F8, F10, F11, F14, F17, F25 and F30, BIMA becomes the fourth best. In functions F8, F10, F11, F14, F17, F25 and F30, BIMA becomes the fifth best, whereas in functions F5, F13, F20 and F21, BIMA scores sixth among ten state-of-the-art heuristic algorithms compared. The results are very competitive and show that the BIMA performs better for eleven and close to the optimum in four benchmark functions in comparison to other standard and well established optimization algorithms presented in the paper. The result indicates the convergence capability and the accuracy of the algorithm. Assumptions taken in the parameter value of all the considered algorithms are listed in Table 4.

Table 3: Comparison between BIMA and other algorithms on CEC2017 optimization benchmark functions for 30 independent runs

Benchmark Function of CEC 2017							
Function		**I**	**DA**	**GWO**	**PSO**	**DE**	**SSA**
		II	**L-SHADE**	**L-SHADE SPACMA**	**L-SHADE-cnEpSin**	**CMA-ES**	**BIMA**
F1	I	Mean	8.4370E+12	1.565E+07	4.759E+03	1.928E+03	2.060E+04
		Std	5.212E+08	2.232E+06	2.019E+02	1.142E+02	8.125E+03
	II	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.029E+05	**0.00E+00**
		Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.068E+10	**0.00E+00**
F2	I	Mean	1.879E+51	6.198E+28	3.233E+06	1.590E+21	4.875E+18
		Std	1.121E+11	1.026E+12	4.587E+04	1.535E+09	3.175E+15
	II	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.345E+26	**0.00E+00**
		Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.434E+53	**0.00E+00**
F3	I	Mean	1.961E+07	5.893E+04	1.214E+04	8.321E+03	1.506E+04
		std	2.112E+02	4.925E+02	1.001E+03	7.259E+02	1.282E+03
	II	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.312E+05	**0.00E+00**
		std	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.519E+06	**0.00E+00**
F4	I	Mean	5.179E+07	1.418E+02	7.976E+01	9.116E+01	9.180E+01
		std	1.205E+01	4.692E+01	6.235E+01	6.276E+01	8.192E+01
	II	Mean	5.961E+01	5.856E+01	**3.4763E+01**	417.8210	5.786E+01
		std	0.00E+00	0.00E+00	**3.0697E+00**	6.321e+03	0.00E+00
F5	I	Mean	2.563E+04	1.114E+02	1.681E+02	1.987E+02	7.462E+01
		std	4.516E+01	1.002E+02	1.358E+01	1.574E+02	7.156E+01
	II	Mean	6.727E+01	**3.2652E+00**	5.5543E+00	664.6020	8.628E+01
		std	1.202E+00	**1.9515E+00**	2.3430E+00	103.5152	1.891E+00
F6	I	Mean	5.571E+02	5.746E+00	3.441E+01	8.009E−01	3.768E+01
		Std	3.120E+01	1.725E+00	3.256+01	7.982E+01	2.359E+01
	II	Mean	1.521E−08	**0.00E+00**	**0.00E+00**	600.0003	**0.00E+00**
		Std	4.105E−08	**0.00E+00**	**0.00E+00**	30.7247	**0.00E+00**
F7	I	Mean	2.043E+05	1.461E+02	1.474E+02	2.432E+02	1.521E+02
		Std	1.025E+02	2.345E+01	2.542E+01	1.325E+02	1.428E+02
	II	Mean	6.351E+01	3.3865E+01	3.8949E+01	905.7383	**3.343E+01**
		Std	1.124E+00	8.753E−01	2.1667E+00	414.9433	**1.835E+00**
F8	I	Mean	2.405E+04	1.162E+02	1.074E+02	2.019E+02	1.751E+02
		std	2.010E+02	1.025E+01	1.015E+02	2.014E+02	1.642E+02
	II	Mean	8.0155E+00	**3.2492E+00**	5.4402E+00	963.7556	9.765E+00
		std	1.421E+00	**1.5171E+00**	2.8641E+00	91.6537	3.759E+00
F9	I	Mean	3.091E+04	1.928E+03	2.461E+03	6.959E+00	2.599E+03
		std	6.107E+01	1.825E+03	2.0156E+02	5.612E+00	1.256E+02
	II	Mean	**0.00E+00**	2.229E−15	**0.00E+00**	900.0000	**0.00E+00**
		std	**0.00E+00**	1.103E−13	**0.00E+00**	6.326E+03	**0.00E+00**
F10	I	Mean	3.551E+04	2.223E+03	**2.985E+00**	7.417E+03	3.816E+03
		std	1.028E+02	2.015E+03	**2.752E+00**	2.472E+02	2.492E+03
	II	Mean	3.114E+03	1.428E+03	8.4381E+02	9.051E+03	1.534E+03
		std	1.278E+02	2.278E+02	2.1047E+02	391.4174	1.638E+02
F11	I	Mean	9.597E+07	2.668E+02	9.730E+01	1.54E+02	2.585E+02
		std	6.214E+03	2.354E+01	8.754E+01	1.146E+02	2.427E+02
	II	Mean	3.458E+01	1.866E+01	**1.7249E+00**	3.685E+03	3.847E+01
		std	3.125E+01	2.513E+01	**1.9384E+01**	1.476E+04	1.326E+01

F12	I	Mean	3.783E+12	8.372E+03	9.333E+04	7.780E+05	3.773E+06
		std	3.018E+09	7.825E+02	6.589E+03	6.291E+04	2.416E+05
	II	Mean	1.434E+03	5.823E+02	1.3150E+02	1.810E+07	**9.462E+01**
		std	3.424E+02	2.573E+02	2.0053E+02	5.115E+09	**5.892E+01**
F13	I	Mean	1.252E+12	1.366E+05	1.189E+03	1.168E+01	1.810E+04
		std	1.000E+07	1.254E+04	1.025E+03	5.246E+02	1.715E+04
	II	Mean	1.634E+01	1.311E+01	**2.0256E+00**	6.400E+06	1.832E+01
		std	7.101E+00	5.737E+00	**1.0207E+01**	1.945E+09	3.261E+00
F14	I	Mean	4.677E+09	5.721E+05	4.012E+04	2.490E+01	8.226E+03
		std	4.158E+06	2.546E+04	3.254E+03	6.164E+01	7.815E+03
	II	Mean	3.063E+01	2.302E+01	**7.6082E+00**	2.601E+06	2.571E+01
		std	1.101E+00	1.512E+00	**2.2600E+00**	3.662E+06	1.250E+00
F15	I	Mean	7.955E+11	3.690E+04	2.198E+03	9.320E+01	1.582E+05
		std	3.254E+09	2.356E+03	1.289E+02	8.421E+01	1.432E+05
	II	Mean	3.8078E+00	4.777E+00	**5.7530E−01**	1.781E+06	8.896E−01
		std	1.301E+00	2.053E+00	**1.9802E+00**	7.906E+08	7.826E−01
F16	I	Mean	2.693E+06	8.384E+02	1.370E+03	1.414E+03	1.026E+03
		Std	2.120E+05	2.452E+02	1.287E+02	2.409E+02	1.025E+01
	II	Mean	4.238E+01	3.059E+01	3.9799E+00	2.998E+03	**4.537E−01**
		Std	3.303E+01	4.525E+01	3.0731E+01	1.682E+03	**6.724E−01**
F17	I	Mean	3.207E+12	3.689E+02	4.201E+02	5.817E+02	6.214E+02
		Std	1.001E+10	3.154E+02	6.248E+01	5.842E+01	2.156E+01
	II	Mean	2.379E+02	2.927E+01	**1.5817E+01**	2.319E+03	2.569E+02
		Std	6.736E+00	1.008E+01	**5.5593E+00**	2.679E+04	6.361E+00
F18	I	Mean	6.047E+09	2.050E+06	3.978E+01	2.39E+01	2.827E+05
		std	5.031E+05	1.568E+04	3.248E+01	2.042E+01	1.624E+02
	II	Mean	4.930E+01	2.328E+01	2.0199E+01	1.097E+06	**2.031E+01**
		std	2.041E+00	2.075E+00	7.5204E−01	6.484E+07	**2.936E+00**
F19	I	Mean	1.542E+09	1.917E+04	1.015E+03	4.180E+00	4.976E+04
		std	1.122E+10	1.652E+04	1.002E+02	3.927E+00	2.618E+02
	II	Mean	6.981E+00	9.564E+00	**2.9633E+00**	2.228E+06	3.793E+00
		std	2.132E+00	2.431E+00	**1.9247E+00**	1.835E+09	5.825E−01
F20	I	Mean	1.976E+04	4.662E+02	**1.00E+00**	1.027E+01	4.125E+02
		std	1.901E+02	2.587E+02	**7.0189E−01**	1.527E+01	3.214E+01
	II	Mean	1.458E+02	7.794E+01	1.3089E+01	2.373E+03	1.408E+02
		std	1.6045E+01	5.289E+01	7.3523E+00	664.4802	1.026E+01
F21	I	Mean	2.326E+04	2.593E+02	3.141E+02	2.133E+02	3.063E+02
		Std	2.021E+03	2.192E+02	2.387E+01	2.817E+02	2.015E+02
	II	Mean	2.143E+02	2.071E+02	2.0719E+02	**2.070E+02**	2.154E+02
		Std	2.325E+00	4.5212E+00	2.5616E+00	**102.2813**	1.157E+00
F 22	I	Mean	3.716E+04	3.318E+03	**1.00E+00**	1.023E+02	1.02E+02
		Std	2.314E+04	2.014E+03	**0.657E−01**	1.016E+01	1.024E+02
	II	Mean	1.647E+02	1.000E+02	1.0000E+02	9.860E+03	1.00E+02
		Std	4.956E+00	0.00E+00	1.0047E−13	925.4715	1.202E−03
F23	I	Mean	1.647E+04	4.216E+02	7.702E+02	5.604E+02	4.331E+02
		std	1.140E+03	3.147E+02	6.146E+01	2.475E+02	2.482E+02
	II	Mean	4.313E+02	3.560E+02	3.4488E+02	2.815E+03	**3.412E+02**
		std	1.905E+00	2.980E+00	3.7319E+00	167.5846	**4.971E+00**
	I	Mean	2.445E+04	6.276E+02	6.505E+02	6.221E+02	5.371E+02

F24		std	1.874E+03	4.572E+02	3.298E+01	5.146E+02	2.461E+02
	II	Mean	4.418E+02	4.286E+02	4.180E+02	2.910E+03	**4.180E+02**
		std	1.951E+00	2.349E+00	2.948E+00	236.9077	**6.802E+00**
F25	I	Mean	3.142E+07	5.036E+02	**3.846E+02**	3.871E+02	4.509E+02
		std	2.598E+05	4.014E+02	**3.713E+02**	2.428E+02	2.152E+02
	II	Mean	4.812E+02	3.867E+02	3.8666E+02	2.878E+03	3.871E+02
		std	2.305E−02	0.008E+00	8.8975E−03	3.109E+03	1.027E+00
F26	I	Mean	2.693E+05	2.412E+03	**3.00E+02**	3.238E+03	2.231E+03
		std	2.016E+03	2.414E+02	**8.426E+01**	2.124E+02	2.018E+03
	II	Mean	1.201E+03	9.352E+02	8.4588E+02	4.873E+03	8.739E+02
		std	4.302E+01	5.077E+01	8.4588E+02	1.395E+03	5.162E+01
F27	I	Mean	5.186E+02	5.307E+02	**4.831E+02**	5.078E+02	5.548E+02
		std	6.014E+01	4.193E+02	**3.729E+01**	4.824E+02	2.184E+02
	II	Mean	5.143E+02	5.062E+02	4.8815E+02	3.200E+03	4.981E+02
		std	5.165E+00	4.407E+00	6.6996E+00	264.5246	7.374E+00
F28	I	Mean	5.00E+02	5.292E+02	4.026E+02	4.089E+02	4.958E+02
		std	3.857E+02	2.387E+01	6.249E+01	3.758E+02	3.095E+02
	II	Mean	3.416E+02	3.172E+02	3.0000E+02	3.300E+03	**3.00E+02**
		std	5.602E+01	4.004E+01	3.8592E+01	1.683E+03	**2.241E+01**
F29	I	Mean	4.168E+11	7.576E+02	8.897E+02	1.215E+03	1.296E+03
		std	6.846E+06	2.017E+01	6.624E+01	1.145E+02	1.624E+02
	II	Mean	4.471E+02	**4.152E+02**	4.1850E+02	4.542E+03	4.462E+02
		std	7.024E+01	**1.206E+01**	7.3625E+00	1.922E+03	2.157E+01
F30	I	Mean	2.904E+12	**8.473E+02**	6.836E+03	7.590E+03	1.272E+07
		std	2.913E+08	**6.148E+01**	3.489E+02	6.425E+02	1.241E+06
	II	Mean	2.684E+03	2.001E+03	1.9414E+03	1.495E+06	2.675E+03
		std	6.875E+01	7.409E+01	4.1663E+01	3.223E+08	5.912E+01

Table 4: Parameter assumptions in the simulation of considered algorithms

Algorithm	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
MHDA	SA	30	β	1.5	w_{max}	0.9	w_{min}	0.2		
DA	SA	50	β	1.5	w_{max}	0.9	w_{min}	0.4		
DE	SA	50	F	0.5	CR	0.9				
PSO	SA	50	C1	0.12	C2	1.2	w_{max}	0.9	w_{min}	0.4
GWO	SA	50	r1	[0,1]	r2	[0,1]				
FEP	SA	100	SD	3						
SSA	SA	30	C2	[0,1]	C3	[0,1]				
GSA	SA	50	Rnorm	2						
WOA	SA	30	p	[0,1]	$\lvert\bar{A}\rvert$	>1				
DMPSADE	SA	50	Setp	0.175	Msp	0.02				
L-SHADE	SA	30	F	[0,1]	M_{CR}	0.5	M_F	0.5		

Table 5: Summary of Friedman's Test and Wilcoxon Test of functions F1 to F23

Functions 1 to 23 (Friedman Test)			Functions 1 to 23 (Wilcoxon Test)			
Algorithm	**Mean Rank**	**Rank**	**BIMA Vs**	−	+	**p-value**
MHDA	3.9783	3	MHDA	12	8	0.37109
DA	5.4783	4	DA	15	4	0.01162
PSO	6.8913	8	PSO	15	4	0.01162
GWO	5.5217	5	GWO	19	4	0.00176
DE	3.3261	2	DE	9	8	0.47102
FEP	6.7826	7	FEP	21	2	0.00007
SSA	6.5000	9	SSA	15	4	0.01162
WOA	6.2174	6	WOA	18	3	0.00106
GSA	7.0435	10	GSA	18	3	0.00106
BIMA	3.2609	1				

Table 6: Statistical results of BIMA on the 30D CEC2017 Benchmark function Suit, Average over 51 independent runs

Function	Best	Worst	Median	Mean	Std
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	5.786E+01	7.21E+01	6.26E+01	6.48E+01	0.00E+00
F5	8.628E+01	8.92E+00	8.78E+00	8.76E+00	1.891E+00
F6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F7	3.743E+01	3.826E+01	3.75E+01	3.79E+01	1.835E+00
F8	9.765E+00	9.87E+00	9.79E+00	9.82E+00	3.759E+00
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	1.534E+03	1.87E+03	1.59E+03	1.58E+03	1.638E+02
F11	3.847E+01	4.08E+01	3.89E+01	3.99E+01	1.326E+01
F12	9.462E+01	9.70E+01	9.47E+01	9.64E+01	5.892E+01
F13	1.832E+01	1.96E+0l	1.86E+0l	1.89E+01	3.261E+00
F14	2.571E+01	2.60E+01	2.59E+01	2.59E+01	1.250E+00
F15	8.896E-01	9.32E-01	8.92E-01	8.95E-01	7.826E-01
F16	4.537E-01	4.77E-01	4.61E-01	4.66E-01	6.724E-01
F17	2.569E+02	2.69E+02	2.63E+02	2.58E+02	6.361E+00
F18	2.031E+01	2.35E+01	2.12E+01	2.19E+01	2.936E+00
F19	3.793E+00	4.36E+01	4.14E+00	4.04E+00	5.825E-01
F20	1.408E+02	1.56E+02	1.49E+02	1.53E+02	1.026E+01
F21	2.154E+02	2.67E+02	2.33E+02	2.39E+02	1.157E+00
F22	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.202E-03
F23	4.412E+02	4.60E+02	4.51E+02	4.51E+02	7.971E+00
F24	4.180E+02	4.38E+02	4.32E+02	4.28E+02	6.802E+00
F25	3.871E+02	3.92E+02	3.87E+02	3.87E+02	1.027E+00
F26	8.739E+02	9.24E+02	8.80E+02	8.97E+02	5.162E+01
F27	4.981E+02	5.14E+02	5.04E+02	5.06E+02	7.374E+00
F28	3.00E+02	4.54E+02	4.21E+02	4.28E+02	2.241E+01
F29	4.462E+02	4.72E+02	4.50E+02	4.58E+02	2.157E+01
F30	2.675E+03	2.91E+03	2.77E+03	2.79E+03	5.912E+01

Table 7: Statistical results of BIMA on the 50D CEC2017 Benchmark function Suit, Average over 51 independent runs

Function	Best	Worst	Median	Mean	Std
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	1.95E-01	0.00E+00	0.00E+00	2.00E-01
F3	1.02E+01	1.63E+01	1.37E+01	1.31E+01	2.87E+01
F4	1.64E+01	1.72E+02	2.85E+01	3.25E+0l	2.34E+0l
F5	3.72E+00	2.04E+01	7.95E+00	6.47E+00	1.35E+00
F6	1.02E+00	5.53E+01	2.73E+00	2.31E+01	7.43E+00
F7	3.47E+01	7.23E+01	4.67E+01	6.16E+01	2.63E+00
F8	5.22E+01	6.37E+01	6.98E+01	6.87E+01	3.41E+01
F9	2.00E+00	7.03E+00	3.01E+00	2.08E+00	1.01E+00
F10	1.12E+02	3.17E+02	2.17E+02	1.81E+02	3.27E+02
F11	1.93E+01	3.64E+01	2.82E+01	2.36E+01	2.52E+00
F12	3.25E+03	7.34E+03	2.67E+03	3.67E+03	5.27E+02
F13	4.31E+01	5.26E+02	5.42E+01	4.83E+01	4.27E+01
F14	1.27E+01	5.57E+01	4.04E+01	3.37E+01	6.18E+00
F15	7.32E+01	2.07E+02	8.92E+01	8.94E+01	2.27E+01
F16	1.85E+01	5.91E+01	3.63E+01	3.46E+01	1.25E+01
F17	5.31E+01	6.69E+01	5.67E+01	6.01E+01	5.33E+01
F18	2.67E+01	4.31E+01	3.35E+01	3.74E+01	6.13E+00
F19	1.38E+01	4.26E+01	3.72E+01	2.94E+01	4.37E+00
F20	6.12E+01	9.61E+01	6.97E+01	6.53E+01	1.34E+0l
F21	3.01E+02	3.72E+02	3.29E+02	3.24E+02	2.16E+01
F22	1.00E+02	3.14E+02	1.91E+02	2.63E+02	6.24E+02
F23	3.86E+02	4.13E+02	3.97E+02	4.00E+02	2.57E+01
F24	3.81E+02	4.92E+02	3.89E+02	4.01E+02	3.05E+01
F25	5.31E+02	5.83E+02	5.61E+02	5.46E+02	6.19E+0l
F26	2.10E+02	3.57E+02	2.31E+02	2.24E+02	3.84E+02
F27	7.39E+02	8.23E+02	7.41E+02	7.47E+02	1.51E+01
F28	3.94E+02	4.13E+02	4.00E+02	4.03E+02	2.72E+01
F29	3.12E+02	3.89E+02	3.53E+02	3.24E+02	5.18E+01
F30	4.32E+04	4.79E+04	4.76E+04	4.63E+04	3.95E+03

Table 8: Comparison between BIMA and the other algorithms

Algorithm						
	I	**DA**	**GWO**	**PSO**	**DE**	**SSA**
	II	**L-SHADE**	**L-SHADE SPACMA**	**L-SHADE-cnEpSin**	**CMA-ES**	**BIMA**
w/l/t	I	0/30/0	1/29/0	6/24/0	0/30/0	0/30/0
	II	0/26/4	3/23/4	7/18/5	1/29/0	7/18/5

4.2 Statistical Analysis

In general, the performance of the algorithm is tested with the results of mean and standard deviation. But in order to check the variation of the results in comparison to other algorithms, we also perform a statistical analysis of the results. In order to find the statistical significance of the results, we perform Friedman's test and Wilcoxon ranksum test. The Friedman's test is used for finding the rank of the algorithm [50] where the best performing algorithm got the lowest rank and the worst one got the highest rank. On the other hand, Wilcoxon's ranksum test [51] is performed considering the lowest ranking algorithm resulting from Friedman's as the control algorithm. The result of the tests indicates the number of times the proposed BIMA offers a better result (+) or a worse result (−) in comparison to other algorithms presented in Table 5.

From the statistical result of the Friedman's test, it is clear that BIMA outperforms other state-of-the-art algorithms with a mean rank of 3.2609 for a 5% level of significance in benchmark functions listed in Table 1. Thus, BIMA's rank is 1 among the 10 optimization algorithms presented in Table 2. From the results of Wilcoxon Test as shown in Table 5, it can be observed that the performance of BIMA is significantly superior to the considered well-known algorithms.

The statistical results of BIMA are shown in Table 6 and Table 7 for D=30 and 50, respectively. Results show the best, worst, median, mean and standard deviation over the 51 runs of the error value. Friedman's Test and Wilcoxon Test of functions F1 to F23 are performed with a 0.05 level of significance in order to assess the importance of the results as shown in Table 5. Table 8 shows the outcome of win/lose/tie (*w/l/t*) or *scalability test* performance comparison of the proposed BIMA with some of the other state-of-the-art algorithms in literature. It is observed that BIMA wins in 7 cases; in 23 cases it performs inferior and having zero tie with other algorithms presented here. The result indicates that BIMA is the superior algorithm among the most popular and standard optimization algorithms in literature.

4.3 Analysis of BIMA

The superior performance of BIMA in a uni-modal function depicts its significant exploration rate. Tracking the particle best (pB_{est}) of each buyer to evaluate the quality of the product is the key component responsible for exploitation capability of the algorithm. The random initialization of population, levy flight search, particle best (pB_{est}), global best (gB_{est}) and global worst solution (gW_{orst}), are the prime components for balancing the exploration and exploitation rate of the algorithm. The performance of the algorithm in multi-modal and fixed dimension multi-modal functions shows the ability of the algorithm in balancing the exploration and exploitation rate.

On the other hand, the performance of the algorithm in CEC-2017 benchmark functions shows the capability of the algorithm to avoid local convergence in the search space and finding the global optima. The evolutionary selection of the finest product from the best shop and the high exploration and exploitation rate of BIMA plays an important role in achieving the global optimal solution.

5 Convergence Analysis

In this section, the performance of the proposed BIMA is tested for observing the convergence of the benchmark test functions, namely Quadric (F3), Rosenbrock (F5), Function F7, Griewank (F11) and Levy (F13) functions.

Here, we consider 100 search agents (population of buyers) in 2, 10 and 30 dimensions and run the evaluation for 300 iterations, considering the average best-so-far in each iteration over 30 runs. We compare the BIMA algorithm for convergence analysis with DA, GWO, PSO and WOA as shown in the Figures 6-15.

- From Figure 7, it is observed that the convergence of BIMA is superior to DA and WOA algorithms with a little behind the GWO and PSO algorithms in the short run. Over the long run, the convergence of BIMA to global optima is better than the other algorithms presented.

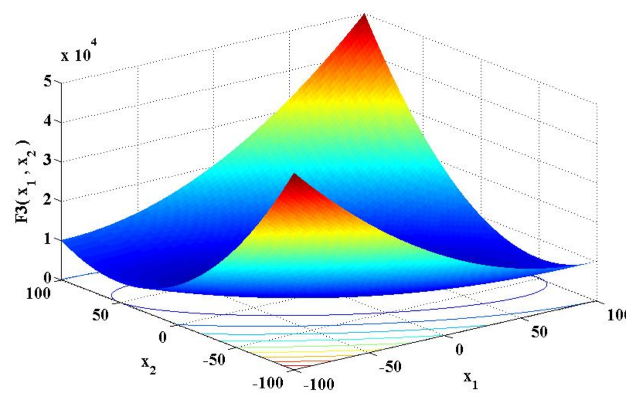

Figure 6: Parameter space of function F3

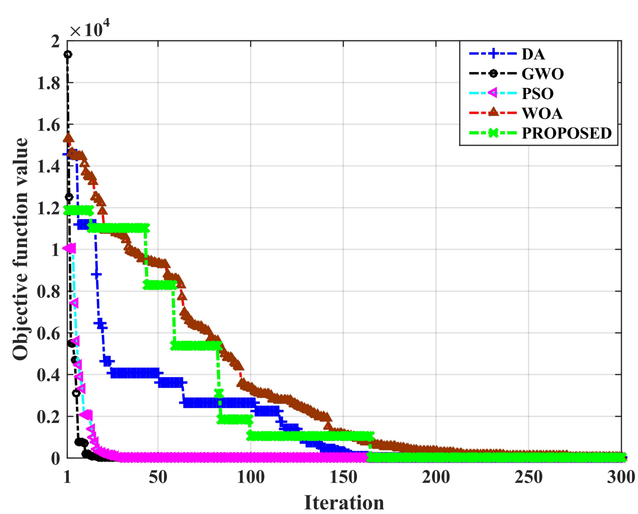

Figure 7: Convergence graph of F3 in 30D

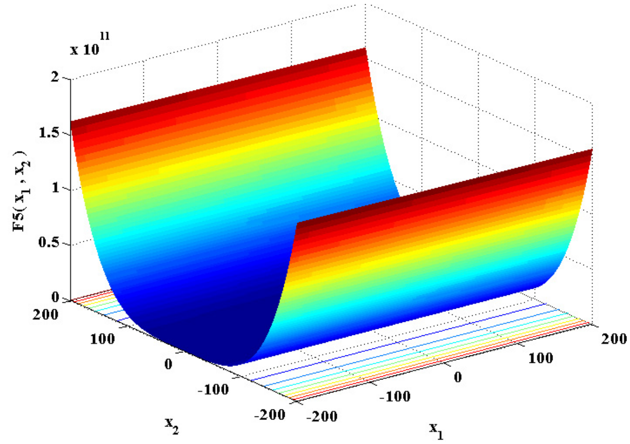

Figure 8: Parameter space of function F5

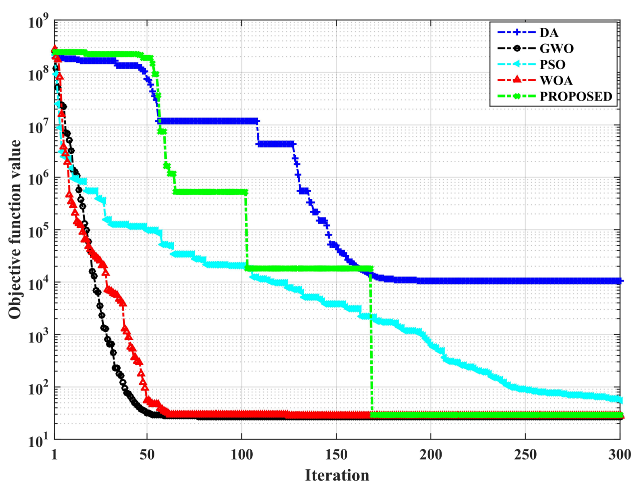

Figure 9: Convergence graph of F5 in 30D

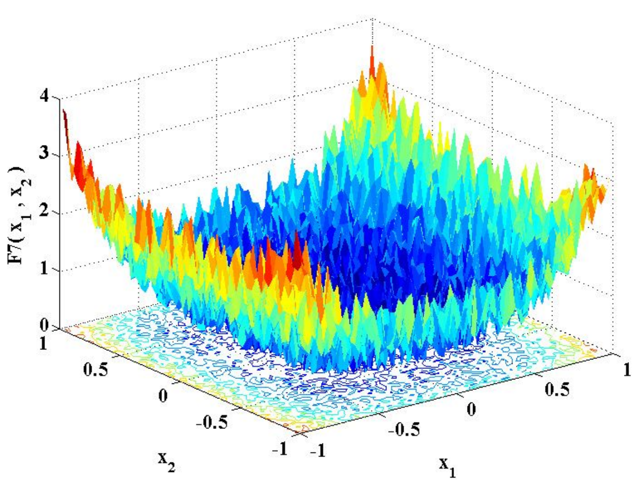

Figure 10: Parameter space of function F7

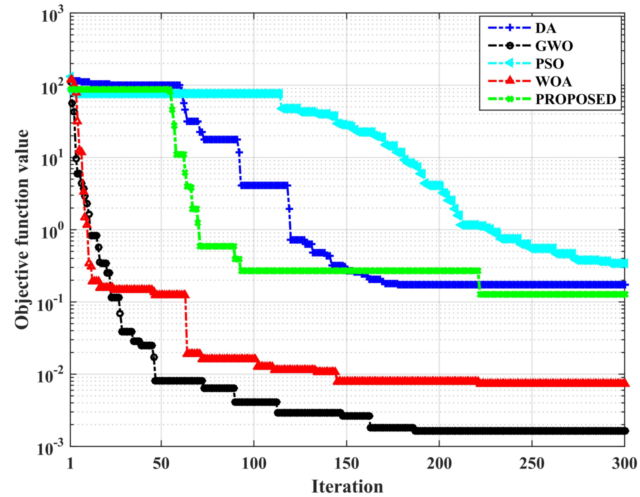

Figure 11: Convergence graph of F7 in 30D

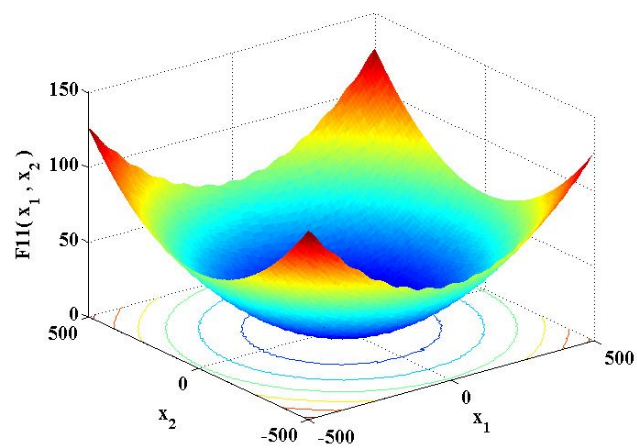

Figure 12: Parameter space of function F11

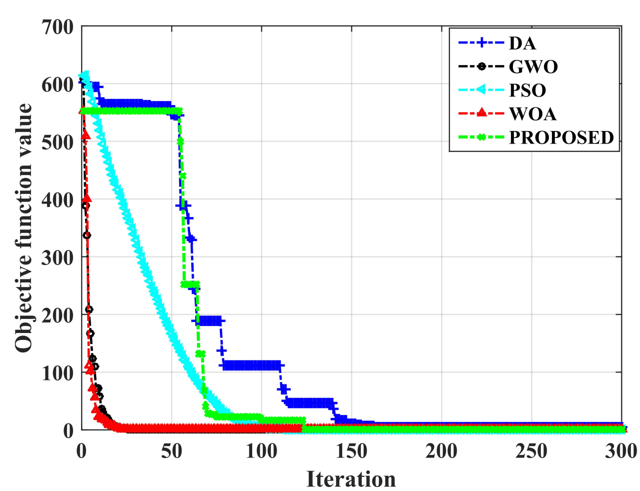

Figure 13: Convergence graph of F11 in 30D

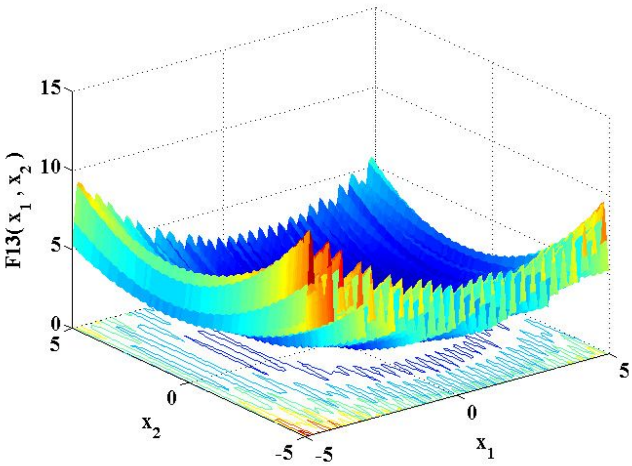

Figure 14: Parameter space of function F13

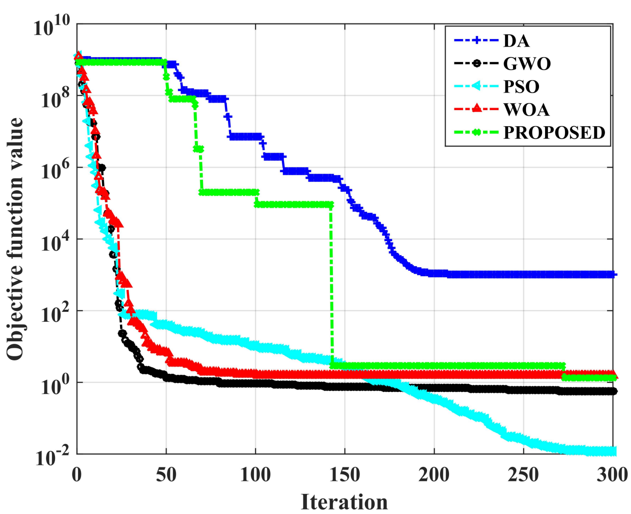

Figure 15: Convergence graph of F13 in 30D

- In Figure 9, for Rosenbrock (F5) function we find that the convergence characteristics of the proposed BIMA is considerably better than all the algorithms in the long run but lags in the short run behind GWO, WOA and PSO algorithms.
- In function F7 as shown in Figure 11, it can be observed that the convergence rate of BIMA is better than DA and PSO.
- The results of the algorithms on Griewank (F11) function is represented in Figure 13. The figure shows that BIMA offers significant convergence in comparison to all the algorithms presented here. However, in the initial phase of the evaluation, GWO and WOA are able to converge better than BIMA but in the long run of 300 iteration BIMA able to perform better than the other algorithms used for comparison.
- Lastly, the test results on Levy (F13) function is shown in Figure 15, show the performance of BIMA for 300 iterations, where it can be observed that the performance of BIMA is better than DA and WOA in 30D but the characteristics of BIMA are worse in comparison to GWO and PSO.

The low convergence rate of the algorithm is the initial phase of the iteration is the main weakness of BIMA. The convergence of the algorithm mainly depends on the fast searching of the global solution in the search space. In BIMA, every buyer has their own set of neighborhood buyers in the respective zone created by the neighborhood radius. Each buyer in the search space learns from their neighbor buyers and then share the right information with all the buyers in the search space to find the global optimal solution, which requires considerable time to reach the global optima. An Individual buyer performs self-learning through pB_{est} and mutual learning through gB_{est} and gW_{orst}. The self-learning strategy is comparatively slow but has good impact in the long run, whereas mutual learning becomes fast but reliant on buyer information. In BIMA, as a buyer undergoes both the self and mutual learning, the convergence rate of the algorithm is slightly sluggish. However, this comes with an advantage of increasing the exploration ability of BIMA in finding the global optimal solution in the large search space. Though initially it is sluggish, the algorithm offers a very competitive exploration rate in general. Therefore the overall performance of the algorithm is found to be satisfactory in terms of global optima. The convergence performance of BIMA in 3 uni-modal and 2 multimodal functions depicts the acceptability of the algorithm.

6 Testing of the Proposed Algorithm on benchmark design problems

To validate the performance of any optimization algorithm, it needs to be checked with benchmark engineering design problems [52]. The performance of BIMA is tested on three popular constraint engineering design problems, namely Welded Beam, Tension/Compression Spring, and Speed Reducer Design Problems and the results are compared with the well-known optimization techniques namely MHDA [40], DA [19], DE [27], PSO [3], GWO [13], WOA [12], SSA [18], DMPSADE [53] and LSHADE [46]. The parameters preset for simulation of all the algorithms including BIMA are consistent and are as follows: 100

search agents, 30 independent evaluations and 1000 iterations per evaluation. Also, results of 30 independent evaluations are used for statistical analysis of the engineering design problem by Friedman and Wilcoxon test.

6.1 Welded beam design problem

This engineering problem is formulated for minimizing the fabrication cost for a given bar length (l), height (t), thickness (b) and the thickness of the welding part (h) as parameters in the function. In the formulation of the design, the major constraint of the design is formed on the basis of shear (τ) and bending stress in the beam (b), beam deflection (δ) and the load corresponding to buckling (Pc). As shown in Figure 16 [54], the variable corresponding to h is x_1, l is x_2, t is x_3, and b is $x_{4,}$ respectively.

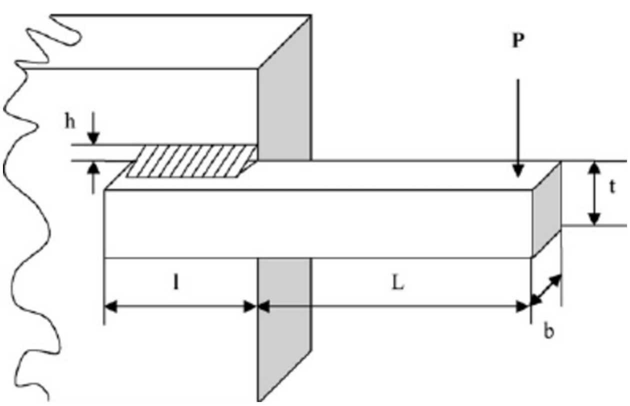

Figure 16: Welded beam design problem

The cost function for welded beam optimization problem can be written as

$$f(\vec{X}) = 1.10471x_2x_1^2 + 0.04811x_3x_4(14 + x_2)$$

Subject to constraints

$$g_1(\vec{X}) = \tau(\vec{X}) - \tau_{\max} \le 0, \quad g_2(\vec{X}) = \sigma((\vec{X})) - \sigma_{\max} \le 0,$$
$$g_3((\vec{X})) = \delta(\vec{X}) - \delta_{\max} \le 0, \quad g_4(\vec{X}) = x_1 - x_4 \le 0,$$
$$g_5(\vec{X}) = P - P_c(\vec{X}) \le 0, \quad g_6(\vec{X}) = 0.125 - x_1 \le 0,$$
$$g_7(\vec{X}) = 1.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0$$

Where, $\tau(\vec{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{P}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$, $M = P\left(L + \frac{x_2}{2}\right)$, $R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$, $J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$, $\sigma(\vec{X}) = \frac{6PL}{x_4x_3^2}$, $\delta(\vec{X}) = \frac{6PL^3}{Ex_4x_3^2}$,

$$P_c(\vec{X}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

Where, $P = 6000$ lb, $L = 14$ in, $\delta_{max} = 0.25$ in, $E = 30E6$ *psi*, G = 12E6 psi, $\tau'_{max} = 30{,}000$ psi.

Where, $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$and $0.1 \le x_4 \le 2$.

Table 9 shows the comparison of the results obtained for the proposed BIMA algorithm with some of the popular and most widely used algorithms in the literature. The result shows that BIMA outperforms other algorithms in terms of finding the optimal cost of the problem.

From the Friedman Test result, as shown in Table 10, it can be observed that BIMA ranks first among all the compared algorithms and from the result of the Wilcoxon Test it can be concluded that the performance of BIMA is significantly better compared to all the considered state-of-the-art algorithms. In Wilcoxon Test the (+) sign is used to show the significantly better and (–) sign shows the significantly worse test count among the total test performance.

Table 9: Results of the welded beam design problem of various optimization algorithms

Algorithm	Variables				Optimum cost
	h	*l*	*t*	*b*	
BIMA	**0.194288**	**3.16681**	**9.03743**	**0.205695**	**1.6675**
MHDA	0.2057296	3.2531200	9.0366239	0.2057296	1.6952471
DA	0.194288	3.46681	9.04543	0.205695	1.70808
DE	0.20573	3.470489	9.0336624	0.205730	1.724852
PSO	0.20573	3.47049	9.03662	0.20573	1.7248508
GWO	0.1990	3.1632	9.0304	0.2060	1.6746
WOA	0.205396	3.484293	9.037426	0.206276	1.730499
SSA	0.2057	3.4714	9.0366	0.2057	1.7249
DMPSADE	0.20573	3.47049	9.03662	0.20573	1.72485084
LSHADE	0.194288	3.16681	9.03743	0.205695	1.6675

Table 10: Statistical Test of the welded beam design problem

Friedman Test			Wilcoxon Test			
Algorithm	**Mean Rank**	**Rank**	**BIMA Vs**	−	+	**p-value**
MHDA	6.075	3	MHDA	66	0	.00338
DA	6.850	5	DA	210	0	.00008
DE	5.225	2	DE	210	0	.00008
PSO	6.050	4	PSO	210	0	.00008
GWO	8.250	7	GWO	210	0	.00008
WOA	9.100	9	WOA	210	0	.00008
SSA	6.900	6	SSA	210	0	.00008
DMPSADE	8.625	8	DMPSADE	210	0	.00008
LSHADE	1.950	1	LSHADE	0	0	1.00
BIMA	1.950	1				

6.2 Tension/compression spring design problem

The schematic of the problem is shown in Figure 17, where the design variables are wire diameter d (x_1), mean coil diameter D (x_2), and number of active coils P (x_3) and the objective is to minimize the weight $f(x)$ [55].

The objective function of the said problem can be defined as:

$$f(\vec{X}) = (x_3 + 2)x_2x_1^2$$

Subject to constraints

$$g_1(\vec{X}) = 1 - \left(\frac{x_2^3x_3}{71785x_1^4}\right) \le 0,$$

$$g_2(\vec{X}) = \left(\frac{4x_2^3 - x_1x_2}{12566(x_2x_1^3 - x_1^4)}\right) + \left(\frac{1}{5108x_1^2}\right) - 1 \le 0,$$

$$g_3(\vec{X}) = 1 - \left(\frac{140.45x_1}{x_2^2x_3}\right) \le 0,$$

$$g_4(\vec{X}) = \left(\frac{x_1 + x_2}{1.5}\right) - 1 \le 0$$

Where, $0.05 \le x_1 \le 2$, $0.25 \le x_2 \le 1.30$, and $2 \le x_3 \le 15$

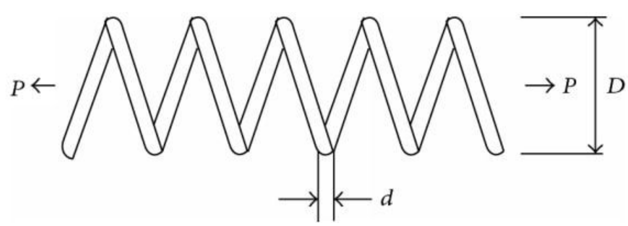

Figure 17: Tension/compression spring design problem

The results in Table 11 show that the performance of the proposed BIMA algorithm is considerably better than some of the popular algorithms presented in this article. Here, from the Friedman and Wilcoxon tests, it can be observed that the statistical performance of BIMA in the Ten-

Table 11: Comparison Results of Tension/compression spring design problem

	Variables			
Algorithm	Wire Diameter (d)	Coil Diameter (D)	Active Coils (P)	**Optimum cost**
BIMA	**0.05**	**0.4759**	**4.1634**	**0.00727**
MHDA	0.05	0.4797	4.0640	0.00727
DA	0.05	0.47998	4.0574	0.00727
DE	0.05	0.4798	4.0693	0.00728
PSO	0.05	0.5001	3.5846	0.0070
GWO	0.05	0.4797	4.0640	0.00727
WOA	0.0516	0.3525	11.3332	0.01266
SSA	0.05	0.345	12.004	0.0125
DMPSADE	0.05	0.48	4.0557	0.0073
LSHADE	0.05	0.4768	4.0796	0.00727

Table 12: Statistical test of Tension/compression spring design problem

Friedman Test			Wilcoxon Test			
Algorithm	**Mean Rank**	**Rank**	**BIMA Vs**	−	+	**p-value**
MHDA	4.075	2	MHDA	19	2	N/A
DA	7.875	5	DA	210	0	.00008
DE	2.950	1	DE	0	0	1.00
PSO	6.450	3	PSO	210	0	.00008
GWO	9.650	7	GWO	210	0	.00008
WOA	8.300	6	WOA	210	0	.00008
SSA	6.850	4	SSA	210	0	.00008
DMPSADE	6.450	3	DMPSADE	210	0	.00008
LSHADE	2.950	1	LSHADE	0	0	1.00
BIMA	2.950	1				

sion/Compression spring design problem is superior when compared to all the considered algorithms.

6.3 Speed Reducer Design Problem

The aim of the speed reducer design problem [56] is to minimize the weight of speed reducer subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shafts. The variables x_1 to x_7 represent the face width (b), module of teeth (m), number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2), and the diameter of first (d_1) and second shafts (d_2), respectively. The schematic of the problem is shown in Figure 18.

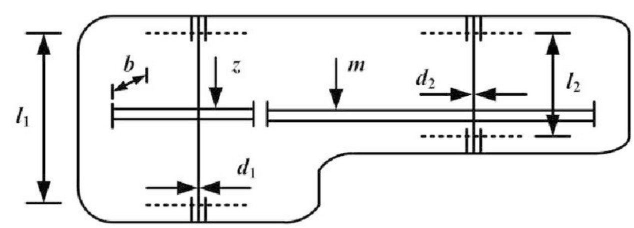

Figure 18: Speed Reducer Design Problem

$$f(\vec{X}) = 0.7854x_1x_1^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854((x_4x_6^2 + x_5x_7^2))$$

Subject to

$$g_1(\vec{X}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0, \quad g_2(\vec{X}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$

$$g_3(\vec{X}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0, \quad g_4(\vec{X}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,$$

$$g_5(\vec{X}) = \frac{\left[(745(x_4/x_2x_3))^2 + 16.9 \times 10^6\right]^{1/2}}{110x_6^3} - 1 \le 0,$$

Table 13: Results of Speed Reducer Design Problem of various optimization algorithms

Algorithm	Variables							Optimum cost
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	
BIMA	**2.6**	**0.7**	**17**	**7.3**	**7.3**	**2.9**	**5.00**	**2352.403**
MHDA	2.6	0.7	17	7.3	7.3	2.9	5.00	2352.4481
DA	2.6	0.7	17	7.529	7.38	2.9	5.00	2854.596
DE	3.5	0.71	17	7.48	7.715	3.35	5.21	2992.2
PSO	3.5	0.71	17	7.3	7.79	3.345	5.286	2996.35
GWO	2.6	0.7	17	7.38	7.3	2.9	5.00	2851.4936
WOA	3.5	0.7	17	7.35	7.715	3.35	5.286	2994.499
SSA	2.6	0.7	17	7.3	7.3	2.9	5.00	2352.4478
DMPSADE	3.5	0.7	17	7.3	7.715	3.35	5.286	2994.710
LSHADE	2.6	0.7	17	7.3	7.3	2.9	5.00	2352.4478

Table 14: Statistical test of Speed Reducer Design Problem

Friedman Test			Wilcoxon Test			
Algorithm	**Mean Rank**	**Rank**	**BIMA Vs**	**−**	**+**	**p-value**
MHDA	4.87500	5	MHDA	15	6	N/A
DA	5.10000	8	DA	15	6	N/A
DE	4.87500	7	DE	19	2	N/A
PSO	5.77500	9	PSO	49	6	.02852
GWO	4.87500	6	GWO	15	6	N/A
WOA	4.87500	4	WOA	11	3	N/A
SSA	4.87500	3	SSA	11	4	N/A
DMPSADE	10.0000	10	DMPSADE	210	0	.00008
LSHADE	4.87500	2	LSHADE	15	6	N/A
BIMA	4.87500	1				

$$g_6(\vec{X}) = \frac{[(745(x_5/x_2x_3))^2 + 157.5 \times 10^6]^{1/2}}{85x_7^3} - 1 \le 0,$$

$$g_7(\vec{X}) = \frac{x_2x_3}{40} - 1 \le 0, \quad g_8(\vec{X}) = \frac{5x_2}{x_1} - 1 \le 0,$$

$$g_9(\vec{X}) = \frac{x_1}{12x_2} - 1 \le 0, \quad g_{10}(\vec{X}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,$$

$$g_{11}(\vec{X}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

where $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.3 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5 \le x_7 \le 5.5$

The comparison results are given in Table 13.

From Table 13, it can be observed that among the compared optimization algorithms BIMA, DA and GWO optimization have the best minimum results and detected best solution with considerably less function evaluations. The statistical performance result is shown in Table 14, where from Friedman test it can be observed that the rank of BIMA is one among all the 10 considered algorithms.

7 Conclusion

In this paper, we proposed a novel optimization technique BIMA inspired by the bargaining nature of human beings in buying the required product with the best quality and at optimum price. The buyer hops between shops in order to purchase products and bargains to ensure the best quality at minimum cost with the choicest product based on the reviews made by neighbor buyers about the product. The proposed BIMA thus provides an appropriate balance between exploration and exploitation which are necessary and essential features of optimization algorithms. The performance of the proposed BIMA is tested on 23 (Unimodal, Multi-modal and Fixed dimensional multi-modal) well reported benchmark functions as well as on 30 benchmark functions of CEC2017 and the results are compared with some of the most popular optimization algorithms. The results show that the proposed BIMA outperformed some of the recent and most popular optimization algorithms as cited in the manuscript in the majority of the tests and offered very competitive results in the other cases. We tested the performance of BIMA on three well-known engineering design problems and the results show the superiority of BIMA in comparison to other prominent and recognized algorithms in the literature. The results comprehensively validate the applicability of the algorithm in solving real-life engineering problems like other standard meta-heuristic optimizers. The initial convergence rate of the proposed model is a little sluggish compared to some of the reported algorithms and may be improved by some innovative, fast and efficient search mechanisms in the future.

Acknowledgement: This work was supported by the Ministry of Electronics and Information Technology (MeitY), Govt. of India Reference Grant No.: 21(1)/2015-CC&BT.

References

[1] Dorigo M., Thomas S., Ant colony optimization, MIT Press eBooks, 2004

[2] Blum C., Li X., Swarm intelligence in optimization, In C. Blum, D.Merkle (Ed.), Swarm intelligence, Natural Computing Series, Springer, 2008, DOI: https://doi.org/10.1007/978-3-540-74089-6_2

[3] Kennedy J., Eberhart R., Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, 1995, 4, 1942-1948, DOI: 10.1109/ICNN.1995.488968.

[4] Omran M., Engelbrecht A.P., Salman A., Particle swarm optimization methods for image clustering." International Journal of Pat-

tern Recognition and Artificial Intelligence, 2005, 19(3), 297-321. DOI: https://doi.org/10.1142/S0218001405004083
[5] Al Rashidi M.R., El-Hawary M.E., A survey of particle swarm optimization applications in electric power systems, IEEE Transactions on Evolutionary Computation, 2008, 13(4), 913–918. DOI: 10.1109/TEVC.2006.880326
[6] He S., Prempain E., Wu Q.H., An improved particle swarm optimizer for mechanical design optimization problems, Engineering Optimization, 2007, 36(5), 585-605. DOI: 10.1080/03052150410001704854
[7] Nimtawat A., Nanakorn P., Simple Particle Swarm Optimization for Solving Beam-Slab Layout Design Problems, Procedia Engineering, 2011, 14, 1392-1398, DOI: https://doi.org/10.1016/j.proeng.2011.07.175
[8] Dorigo M., 2007, Ant colony optimization, Scholarpedia, 2(3), 1461. DOI:10.4249/scholarpedia.1461
[9] Akay B., Karaboga D., Artificial bee colonial algorithm for large-scale problems and engineering design optimization, Journal of Intelligent Manufacturing, 2012, 23(4), 1001–1014, DOI: https://doi.org/10.1007/s10845-010-0393-4
[10] Yang X.S., A new metaheuristic bat-inspired algorithm, In González J.R., Pelta D.A., Cruz C., Terrazas G., rasnogor N. (Ed.), Nature inspired cooperative strategies for optimization (NICSO 2010), Studies in Computational Intelligence, Springer, 2010, DOI: https://doi.org/10.1007/978-3-642-12538-6_6
[11] Yang X.S., Firefly algorithms for multimodal optimization." In Watanabe O., Zeugmann T. (Ed.), Stochastic algorithms: Foundations and applications: 5th international symposium, SAGA 2009, Springer, 2009, DOI: https://doi.org/10.1007/978-3-642-04944-6_14
[12] Mirjalili S., Lewis A., The whale optimization algorithm, Advances in Engineering Software, 2016, 95, 51–67. DOI: https://doi.org/10.1016/j.advengsoft.2016.01.008
[13] Mirjalili S., Mirjalili S.M., Lewis A., Grey wolf optimizer, Advances in Engineering Software, 2014, 69, 46–61. DOI: https://doi.org/10.1016/j.advengsoft.2013.12.007
[14] Sharma A., Sharma A., Panigrahi B.K., Kiran D., Kumar R., Ageist spider monkey optimization algorithm, Swarm and Evolutionary Computation, 2016, 28, 58–77. DOI: https://doi.org/10.1016/j.swevo.2016.01.002
[15] Wang G.G., Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems, Memetic Computing, 2016, 10(2), 151–164. DOI: https://doi.org/10.1007/s12293-016-0212-3
[16] Mirjalili S., Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge based systems, 2015, 89, 228-249. DOI: https://doi.org/10.1016/j.knosys.2015.07.006
[17] Mirjalili S., The ant lion optimizer, Advances in Engineering Software, 2015, 83, 80-98, DOI: https://doi.org/10.1016/j.advengsoft.2015.01.010
[18] Mirjalili S., Gandomi A.H., Mirjalili S.Z., Saremi S., Faris H., Mirjalili S.M., Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Advances in engineering software, 2017, 114, 163-191. DOI:10.1016/j.advengsoft.2017.07.002
[19] Mirjalili S., Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective discrete, and multi-objective problems, Neural Computing and Applications, 2016, 27(4), 1053–1073. DOI: https://doi.org/10.1007/s00521-015-1920-1
[20] Thangaraj R., Pant M., Abraham A., Bouvry P., Particle swarm optimization: Hybridization perspectives and experimental illustrations, Applied Mathematics and Computation, 2011, 217(12), 5208–5226. DOI: https://doi.org/10.1016/j.amc.2010.12.053
[21] Xuewen X., Xing Y., Wei B., Zhang Y., Li X., Deng X., Gui L., A fitness-based multi-role particle swarm optimization, Swarm and Evolutionary Computation, 2019, 44, 349-364, DOI: https://doi.org/10.1016/j.swevo.2018.04.006
[22] Tian D., Shi Z., MPSO: Modified particle swarm optimization and its applications, Swarm and Evolutionary Computation, 2018, 41, 49-68. DOI: https://doi.org/10.1016/j.swevo.2018.01.011
[23] Nobile M.S., Cazzaniga P., Besozzi D., Colombo R., Mauri G., Pasi G., Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization, Swarm and Evolutionary Computation, 2018, 39, 70-85. DOI: https://doi.org/10.1016/j.swevo.2017.09.001
[24] Chen W.N., Lin Y., Chen N., Zhan Z. H., Chung H.S.H., Li Y., Shi Y.H., Particle Swarm Optimization With an Aging Leader and Challengers, IEEE Transactions on Evolutionary Computation, 2013, 17(2), 241-258, DOI: 10.1109/TEVC.2011.2173577
[25] Nickabadi A., Ebadzadeh M.M., Safabakhsh R., A novel particle swarm optimization algorithm with adaptive inertia weight, Applied Soft Computing, 2011, 11(4), 3658–3670. DOI: https://doi.org/10.1016/j.asoc.2011.01.037
[26] Li Z., Wang W., Yan Y., Li Z., PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems, Expert Systems with Applications, 2015, 42(22), 8881–8895. DOI: https://doi.org/10.1016/j.eswa.2015.07.043
[27] Storn R., Price K., Differential Evolution – A simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 1997, 11, 341–359. DOI: https://doi.org/10.1023/A:1008202821328
[28] Wang Z., Zhan Z., Lin Y., Yu W., Yuan H., Gu T., Kwong S., Dual-Strategy differential evolution with affinity propagation clustering for multimodal optimization problems, IEEE Transactions on Evolutionary Computation, 2018, 22(6): 894-908. DOI: 10.1109/TEVC.2017.2769108
[29] Cui L., Li G., Luo Y., Chen F., Ming Z., Lu N., Lu J., An enhanced artificial bee colony algorithm with dual-population framework, Swarm and Evolutionary Computation, 2018, 43, 184-206, DOI: https://doi.org/10.1016/j.swevo.2018.05.002
[30] Xiang W., Meng X., Li Y., He R., An M., An improved artificial bee colony algorithm based on the gravity model, Information Sciences, 2018, 429, 49-71. DOI: https://doi.org/10.1016/j.ins.2017.11.007
[31] Chen X., Tianfield H., Li K., Self-adaptive differential artificial bee colony algorithm for global optimization problems, Swarm and Evolutionary Computation, 2019, 45: 70-91, DOI: https://doi.org/10.1016/j.swevo.2019.01.003
[32] Singh A., Deep K., Artificial Bee Colony algorithm with improved search mechanism, Soft Computing, 2019, 23, 12437–12460. DOI: https://doi.org/10.1007/s00500-019-03785-y
[33] Beheshti Z., Shamsuddin S.M.H., A review of population-based meta-heuristic algorithm, International Journal of Advances in Soft Computing and its Applications, 2013, 5(1), 1–35.
[34] Cao Y., Zhang H., Li W., Zhou M., Zhang Y., Chaovalitwongse W.A., Comprehensive Learning Particle Swarm Optimization Algorithm with Local Search for Multimodal Functions, IEEE Transactions on Evolutionary Computation, 2019, 23(4), 718-731. DOI: 10.1109/TEVC.2018.2885075

[35] Liu X., Zhan Z., Gao Y., Zhang J., Kwong S., Zhang J., Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization, IEEE Transactions on Evolutionary Computation, 2019, 23(4), 587-602. DOI: 10.1109/TEVC.2018.2875430
[36] Gupta S., Deep K., A novel Random Walk Grey Wolf Optimizer, Swarm and Evolutionary Computation, 2019, 44: 101-112. DOI: https://doi.org/10.1016/j.swevo.2018.01.001
[37] Singh N., Son L.H., Chiclana F., Magnot J.P., A new fusion of salp swarm with sine cosine for optimization of non-linear functions, Engineering with Computers, 2020, 36, 185–212. DOI: https://doi.org/10.1007/s00366-018-00696-8
[38] Long W., Cai S., Jiao J., Tang M., An efficient and robust grey wolf optimizer algorithm for large-scale numerical optimization, Soft Computing, 2020, 24, 997–1026, DOI: https://doi.org/10.1007/s00500-019-03939-y
[39] Yang X.S., Chapter 2- analysis of algorithms, In X.-S. Yang (Ed.), Nature-inspired optimization algorithms, Oxford: Elsevier, 2014.
[40] Sree Ranjini K.S, Murugan S., Memory based hybrid Dragonfly algorithm for numerical optimization problem, Expart System with Application, 2017, 83, 63-78, DOI: https://doi.org/10.1016/j.eswa.2017.04.033
[41] Liu H., Abraham A., Zhang W., A Fuzzy adaptive turbulent particle swarm optimization, International Journal of Innovative Computing and Applications, 2007, 1(1), 39-47, DOI: https://doi.org/10.1504/IJICA.2007.013400
[42] Yang X-S., Nature-inspired Metaheuristic Algorithms, 2^{nd} edition. Luniver Press, 2010.
[43] Yao X., Liu Y., Lin G., Evolutionary programming made faster, IEEE Transactions on Evolutionary Computation. 1999, 3(2), 82-102, DOI: 10.1109/4235.771163
[44] Saremi S., Mirjalili S., Lewis A., Grasshopper Optimisation Algorithm: Theory and application, Advances in Engineering Software, 2017, 105, 30-47. DOI: https://doi.org/10.1016/j.advengsoft.2017.01.004
[45] Awad N.H., Ali M.Z., Suganthan P.N., Liang J.J., Qu B.Y., Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Technical Report, Nanyang Technological University, Singapore, 2016.
[46] Tanabe R., Fukunaga A.S., Improving the search performance of SHADE using linear population size reduction, 2014 IEEE congress on evolutionary computation (CEC), 2014, 1658-1665. DOI: 10.1109/CEC.2014.6900380
[47] Mohamed A.W., Hadi A.A., Fattouh A.M., Jambi K.M., LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, 2017, 145-152. DOI: 10.1109/CEC.2017.7969307
[48] Awad N.H., Ali M.Z., Suganthan P.N., Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems, 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, 2017, 372-379. DOI: 10.1109/CEC.2017.7969336.
[49] Hansen N., Müller S.D., Koumoutsakos P., Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evolutionary Computation, 2003, 11(1), 1-18. DOI: https://doi.org/10.1162/106365603321828970
[50] Demsar J., Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research, 2006, 7, 1–30.
[51] Derrac J., Garcia S., Molina D., Herrera F., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation, 2011, 1(1), 3–18, DOI: https://doi.org/10.1016/j.swevo.2011.02.002
[52] Deb K., Optimization for engineering design: algorithm and example, prentice-Hall of India, 1998.
[53] Fan Q., Yan X., Self-adaptive differential evolution algorithm with discrete mutation control parameters, Expert Systems with Applications, 2015, 42(3), DOI: https://doi.org/10.1016/j.eswa.2014.09.046
[54] Rao S.S., Chapter 8, Engineering Optimization – theory and practice, 4th edition, Wiley publication, 2009.
[55] Arora J., Arora J., Introduction to Optimum Design, McGraw-Hill, 2011.
[56] Mezura-Montes E., Coello C.A.C., Useful infeasible solutions in engineering optimization with evolutionary algorithms, In: Gelbukh A., de Albornoz Á., Terashima-Marín H. (Ed) MICAI 2005: Advances in Artificial Intelligence Lecture Notes in Computer Science, 3789, 2005, 652–662. DOI: http://dx.doi.org/10.1007/11579427_66