Optimized localization of sensor nodes in 3D WSNs using modified learning enthusiasm-based teaching learning based optimization algorithm

Prabhjot Singh  |  Nitin Mittal

Electronics and Communication Engineering, Chandigarh University, Gharuan, India

Correspondence
Nitin Mittal, Department of Electronics and Communication Engineering, Chandigarh University, Mohali-160413, Punjab, India
Email: mittal.nitin84@gmail.com

Abstract
Localization in wireless sensor networks (WSNs) is used to determine the coordinates of the sensor nodes deployed in the sensing field. It is the process that determines the location of the target nodes relative to the location of deployed anchor nodes. These anchor nodes are deployed at known locations having GPS installed in them. However, mostly in all 3D applications, the area under observation may have a complexity in the sensing environment. In this work, a modified learning enthusiasm-based teaching learning based optimization algorithm (LebTLBO) is proposed to deal with the 3D localization problem using single anchor and moving target nodes in anisotropic network with DOI 0.01. LebTLBO is a metaheuristic inspired by the classroom teaching and learning method of teaching learning based optimization algorithm. An improved LebTLBO algorithm aims to achieve enhanced performance by balancing the exploration and exploitation capabilities of conventional LebTLBO to improve its global performance. On the CEC2019 benchmark functions, the suggested technique is assessed, and computational findings show that it provides promising outcomes over other competitive algorithms. Also, mLLebTLBO outperforms well in terms of localization error in 3D environment. The proposed technique is useful to cope up in case of rescue operations.

1 | INTRODUCTION

The advancement in wireless technology inspired many researchers to work in the field of wireless sensor networks (WSNs). In these networks, the nodes deployed in the region of interest can sense, measure, and communicate with other nodes to share the useful information among each other [1, 2]. There are numerous applications of WSNs such as forest fire detection, military applications, air pollution control, safety surveillance etc. in which sensor nodes might be placed randomly or at predefined locations [3, 4]. WSNs are categorized into structured and unstructured types. A network that consists of lesser or fewer nodes is referred to as structured WSN. There are certain challenges in unstructured WSNs, such as retaining the network and making the node connectivity with each other.

The main challenges in WSNs consist of localization problems, routing, network topology, memory and power constraints etc. [5–7]. Amongst all, the localization is a huge impacted problem, because, the reporting event location must be known; otherwise, the information will be useless. Further, localization itself has several challenges that include node failure, self-localization within the nodes etc. [8–10].

Localization is the main issue in WSNs and it can be applied to different scenarios, i.e. for static and dynamic WSNs. To locate all sensors in bi-dimensional (2D) and multi-dimensional (3D) environments, the key objective is to identify their specific coordinates [11]. But when deployed in large areas and the sensors have movement abilities, so that the localization problem becomes more challenging. However, mostly in all 3D applications, the area under observation may have a complexity in the sensing environment (i.e. different battery backups,
non-uniform pattern of radiation, and degree of irregularity (DOI)). The challenges to be addressed in mobility-based scenarios are networking, coverage, and energy usage. An effective localization of the target node is one of the key challenges in dynamic WSNs. In space, atmosphere and underwater applications, where network height is most required, the assumptions made for 2D are violated. Nodes are often distributed randomly in three-dimensional space for these types of applications.

In this paper, we introduced the concept of deploying single anchor and assuming this node virtually in six distinct directions for targeting the unknown nodes using a modified learning enthusiasm-based teaching learning based optimization (LebTLBO) algorithm. As these target nodes drop under the range of anchor node, then at 60 degrees angles the virtual anchors are projected in six distinct directions with same range inside the circle, anchor node itself and three virtual nodes are nominated (as minimum four SNs are required to estimate 3D positions) of unknown nodes.

In this paper, the mLebTLBO algorithm is proposed to evaluate the performance of the localization problem of 3D WSNs with various meta-heuristics. The main problem with teaching learning based optimization (TLBO) is that all learners have the same opportunity to acquire knowledge in the teacher or learner phase. In reality, learners are distinct in actual, and they have a distinct passion for learning. When studying, learners with elevated learning excitement are more focused; therefore, they have a great opportunity to gain understanding from others. On the other hand, learners with low learning enthusiasm have relatively little chance of learning from others. This research, motivated by this behaviour, will bring to the TLBO the notion of the mechanism of learning enthusiasm to suggest a novel strategy called LebTLBO [12].

LebTLBO's optimization method can be considered as the continuous conversion of its search space. The search space is hard to contain the optimum solution when it drops into the local optimum. Therefore, guiding the present solution space approximation to the optimum solution space is very important.

It has been found that the proper balance between the diversification and intensification process is missing in LebTLBO and more research is needed to achieve the balance. Furthermore, a probability that defines the degree of exploration and exploitation is a very important factor and no proper study was done to find this parameter. However, the algorithm is very effective in exploring the search space; it takes a lot to improve its exploitative tendencies. Motivated by these, we developed a new improved version of LebTLBO in the current work that has good properties for exploration and exploitation. To achieve these goals, a new variant was introduced to achieve the above-said balance. The concept inspired by Grey wolf optimization [13] is implemented in a bid to improve LebTLBO's worldwide search capability (i.e. exploration capability). LebTLBO uses the present best solution in the local stage and random alternatives to improve local search. The local neighbourhood search (LNS) model [14] is introduced to improve the local search capability to enhance convergence characteristics. LNS's primary concept is to use the finest solution discovered in a tiny neighbourhood of the present solution to update the present solution. To validate the proposed algorithm in solving some extremely difficult data sets, benchmark problems are applied to CEC 2019 and comparative analysis is provided in relation with well-known algorithms.

The paper is structured as follows: In Section 2 literature review on 3D localization in WSNs is presented. Section 3 discusses the LebTLBO and suggests the LebTLBO variant in detail. Section 4 performs the performance assessment and thorough analysis of mLebTLBO. Section 5 presents single anchor deployment structure in 3D WSN. Section 6 deals with results and discussions. Conclusion and future scope are discussed in section 7.

2 | LITERATURE REVIEW

There are many works related to wireless networks. Yang Liu et al. [15, 16] have many works in varied wireless networks, for example, Mobile Ad Hoc Social Networks and Mobile Opportunistic Networks. In contrast to their network, this paper focuses on WSNs that contain many sensor nodes to monitor a physical region. Recently, various localization schemes have been proposed, while most research proposals are dealing with 2D localization problems, where the sensing area is assumed flat. However, mostly in all 3D applications, the area under observation may have a complexity in the sensing environment and large differences in altitude levels. As a result, this problem in WSNs presents new challenges for 3D localization system design.

A 3D localization algorithm based on the received signal strength indicator (RSSI) concept [17], follows parameter matching between each pair of reference node and unlocated node and then evaluating the loss in a wireless signal using various models when there are obstacles (walls) between them. Cheng et al. [18] presented a technique that is applied for underwater localization application for 3D acoustic sensor networks. Chen et al. [19] enhanced the conventional centroid localization algorithm in 2D space to 3D, that uses a greater number of anchor nodes and shows an inverse relationship between localization accuracy and deployed anchor nodes. Besides, the range-free algorithm DV-HOP is also extended to 3D space by Wang et al. [20]. Shi et al. [21] introduced a 3D localization method for WSN. Their localization approach uses a mobile beacon that transmits ultra-wideband (UWB) signals to determine the exact position. During the reception of these signals, every target node by using the TOA technique determines the distance to the anchor node. Graefenstein et al. [22] suggested a technique that is energy efficient in which the RSSI method is utilized to calculate the approximate distance among the reference and target nodes deployed in the field using a trilateration approach. Sumathi and Srinivasan [23] have suggested a method that requires a single anchor node for localizing the unknown nodes using the RSS method. The least-square approach for locating the fixed target nodes is in this paper. Guo et al. [24] developed a perpendicular intersection (PI) mobile-based approach that
does not specifically map RSS distances. The measurement of the node position is carried out using the geometric PI relation. Shi et al. [25] used ultra-wide band, ToA technique to compute node location in a 3D environment. By using this technique, the distance between reference and an unknown node is calculated more efficiently. Wang et al. [26] introduced distance vector-hop based approach for locating sensor nodes. The main reason behind the failure of this algorithm is the complexity and increased cost. Xu et al. [27] introduced another enhanced 3D localization technique that adopted DV-distance with a quasi-Newton optimize method for optimizing the results. By considering localization accuracy and coverage, the authors further tested the effectiveness of the proposed algorithm. Li et al. [28] suggested the 3D WSN localization method based on an irregular RSSI model. To calculate the relation among DOIs and the variability in signal transmission range, the authors proposed this model. Ahmad et al. [29] suggested a parametric loop-division algorithm for 3D localization where the deployed sensors are located in an area surrounded by a group of anchor nodes. In this method, the network shrinks toward the center accurately and provides accurate localization results.

Gopakumar and Jacob [30] suggested a new and computationally efficient swarm intelligence approach for locating static nodes with easier implementation and low memory requirements. Chuang and Wu [31] use RSS ranging technique for locating sensor nodes efficiently using PSO-based approach. The scheme has a higher success rate in terms of localization. Kulkarni et al. presented a distributed iterative localization algorithm called PSO-iterative [32, 33]. For each target node, there exist more than three anchors and PSO is exploited to minimize the localization error. Kumar et al. [34] suggested localization techniques that use the concepts of HPSO and BBO by taking minimal hardware requirement, referred to as range free HPSO and BBO respectively. By using PSO and BBO applications, the edge weights are further optimized. Arora and Singh [35] suggested an optimization algorithm called BOA to optimize the location of unknown sensor nodes. PSO and FA performances in 2D scenarios are compared and contrasted with the performance of BOA. Their approach outperforms in terms of convergence time and location accuracy as compared to other metaheuristic algorithms. Due to the higher accuracy, the range-based methods are commonly used but flip ambiguity is the major drawback of range-based methods.

The method adopted to locate target nodes in 3D environment has the following benefits.

a. A new technique for projecting virtual anchors having paraboloid projection are considered using the proposed mLebTLBO algorithm to determine the exact positions of deployed sensor nodes in a 3D environment.

b. Virtual anchor nodes can mitigate Line of Sight (LoS) issues to a greater extent.

c. The problem of flip ambiguity in range-based approaches is also reduced.

In this paper, mLebTLBO has been proposed to deal with the localization concept used in WSNs. The main objective is to explore the efficiency of the mLebTLBO algorithm in 3D WSNs localization problem and to compare the performance with various other algorithms. The basic idea behind mLebTLBO is explained in the next section.

3 PROPOSED MODIFIED LEARNING ENTHUSIASM-BASED TLBO ALGORITHM

3.1 Teaching learning-based optimization (TLBO)

A relatively new metaheuristic inspired by the behaviours of the teaching and learning process in a classroom called TLBO [36]. To search good solutions TLBO exploits two creative operators, called the teacher and learning phase.

In the teacher phase, by raising the mean outcome of the classroom (xmean) towards his/her position xteacher, the teacher attempts to improve the outcomes of other learners (xi). To preserve stochastic search characteristics, the solution xi uses two randomly generated parameters rand and TF as:

\[
x_{i,new} = x_{i,old} + \text{rand} \cdot (x_{\text{teacher}} - TF \cdot x_{\text{mean}})
\]

where \( x_{i,new} = (x_{i,new}^1, \ldots, x_{i,new}^f, \ldots, x_{i,new}^D) \) and \( x_{i,old} = (x_{i,old}^1, \ldots, x_{i,old}^f, \ldots, x_{i,old}^D) \) are the \( i \)th learner’s current and previous positions, respectively. TF is a teacher factor, and its value may be either 1 or 2. \( x_{i,new} \) is accepted, If it is better than \( x_{i,old} \), otherwise \( x_{i,old} \) remains intact.

In the learner phase, to further enhance its efficiency, a learner \( x_j \) randomly chooses other learner \( x_i (j \neq i) \) for interaction and the learning process can be expressed as follows:

\[
x_{i,new} = \begin{cases} x_{i,old} + \text{rand} \cdot (x_{i,new} - x_{i,old}) & \text{if } f(x_{i,new}) < f(x_{i,old}) \\ x_{i,old} + \text{rand} \cdot (x_{i,new} - x_{i,old}) & \text{otherwise} \end{cases}
\]

where \( f(x) \) is the fitness function and \( x_{i,old} \) is the old position of the \( i \)th learner. If \( x_{i,new} \) is better than \( x_{i,old} \), \( x_{i,new} \) is used to replace \( x_{i,old} \).

3.2 Learning enthusiasm based TLBO (LebTLBO)

LebTLBO is a modified version of basic TLBO. Modification in TLBO enhances search capability for an optimum solution. LebTLBO introduced two new modules one is learning enthusiasm-based teacher part and second is learner phase [12]. These are conferred to boost the search potency, to enhance the standard of the poor learners using poor student tutoring phase. In basic TLBO, every learner has the same capability of acquiring knowledge from others [36]. LebTLBO is motivated from a real-world learning enthusiasm mechanism, where every learner
has different capabilities and enthusiasm to learn. LebTLBO is motivated by this learning enthusiasm mechanism.

The algorithm in its initial stage consists of a population of \( NP \) learners (where the total population is denoted by \( x \)), initialized as

\[
x'_i = x'_{\text{min}} + a_i \times \left( x'_{\text{max}} - x'_{\text{min}} \right)
\]  

(3)

where \( i \in \{1, 2, \ldots, NP\}, \) \( j \in \{1, 2, \ldots, D\}, \) \( x_{ij} \) is the \( j \)th solution in the \( i \)th dimension; \( a_i \) is random number between [0, 1]; \( x_{\text{min}} \), \( x_{\text{max}} \) are lower and upper bound respectively.

After the population initialization of learners, the fitness of each learner is evaluated. The learner with the highest fitness is termed as teacher \( x_{\text{teacher}} \). The details and description of different phases of the LebTLBO algorithm are discussed as follows.

**Learning enthusiasm based teacher phase:** LebTLBO utilizes a learning enthusiasm based model in that the students with decent evaluations have higher learning excitement, and they have an extensive likelihood of getting the information from the instructor. The students with terrible evaluations have less learning energy and little likelihood of getting the information from the educator.

In this phase, all learners are sorted depending upon their fitness value as follows:

\[
f(x_1) \leq f(x_2) \leq \ldots \leq f(x_{NP}) 
\]  

(4)

Then, the learner’s learning enthusiasm value is defined as

\[
LE_i = LE_{\text{max}} + (LE_{\text{max}} - LE_{\text{min}}) \frac{NP - i}{NP}, \quad i = 1, 2, \ldots, NP
\]  

(5)

where \( LE_{\text{max}} \) is maximum learning enthusiasm and \( LE_{\text{min}} \) is minimum learning enthusiasm with suggested values of \( LE_{\text{max}} = 1 \) and \( LE_{\text{min}} \in [0.1, 0.5] \). The learning enthusiasm curve is plotted in Figure 1. It is well depicted from Figure 2 that the highest learning enthusiasm is shown by the best learner, while lowest learning enthusiasm is displayed by the worst learner.

In the wake of characterizing the learning enthusiasm, depending upon learning enthusiasm value \( LE_i \) every student is classified into the category of gain or learn from the instructor or not learn from the instructor. For student \( x_i \), create an irregular number \( r_i \in [0, 1] \), if \( r_i \leq LE_i \), at that point student \( x_i \) will gain from the educator; else a student \( x_i \) will disregard generally the learning of the instructor. If student \( x_i \) can procure the information from the instructor, it refreshes the position utilizing an assorted variety upgraded showing methodology, and the accompanying condition:

\[
x'_{i,\text{new}} = \begin{cases} 
  x'_{i,\text{old}} + \text{rand}_2 \times (x'_{\text{teacher}} - T_F \times x'_{\text{mean}}) & \text{if } \text{rand}_1 < 0.5 \\
  x'_i + F \times (x'_{r_2} - x'_{r_3}) & \text{otherwise}
\end{cases}
\]  

(6)

where \( r_1, r_2 \) and \( r_3 (r_1 \neq r_2 \neq r_3 \neq i) \) are random integers chosen from \( \{1, 2, \ldots, NP\} \); \( d \in \{1, 2, \ldots, D\} \); \( \text{rand}_1 \) and \( \text{rand}_2 \) are randomly generated numbers which are uniformly distributed within [0, 1]; and \( F \) is a scale factor in [0, 1]. It is well clear from Equation (6) that it may be seen as a hybrid model of TLBO and DE given by Figure 2.

**3.2.1 Learning enthusiasm based learner phase**

The learning strategy of the learner is also learning enthusiasm based in LebTLBO. Same as with the teaching strategy it includes higher learning enthusiasm for learners with good grades and those learners lie on high probability region for getting knowledge from others and vice versa. Based on the performance of grades all learners are sorted in the case of learning enthusiasm inspired learner phase as defined in Equation (5).

A randomly generated number between \( r_i \in [0, 1] \) for learner \( x_i \), if \( r_i \leq LE_i \), then learner \( x_i \) will learn from the other learner; otherwise knowledge of the learner will be neglected by the learner \( x_i \). If learner \( x_i \) obtains the knowledge from the
teacher, then based on diversity enhanced teaching strategy its position will be updated as

$$x_{i,\text{new}} = \begin{cases} x_{i,\text{old}} + \text{rand.}(x_i - x_j), & \text{if } f(x_j) \geq f(x_i) \\ x_{i,\text{old}} + \text{rand.}(x_j - x_i), & \text{if } f(x_j) < f(x_i) \end{cases}$$  \hspace{1cm} (7)

where $f(x_j)$ is the objective function, $x_{i,\text{old}}$ is the previous position of the $i$th learner. If $x_{i,\text{new}}$ is fitter than $x_{i,\text{old}}$ then $x_{i,\text{new}}$ is accepted, otherwise $x_{i,\text{old}}$ is unchanged.

### 3.2.2 Poor student tutoring phase

This phase is not available in basic TLBO, the main motive behind this phase is to improve the grades of weak students. The same procedure is followed in this phase too based on their grades the learners are arranged in the range of best to worst.

The learners are considered as a poor learner if they are in the bottom 10%. This phase will randomly choose learner $x_f$ from each poor student $x_i$ whose rank lies in the top 50% and the learning is based on the following equation:

$$x_{i,\text{new}}^d = x_{i,\text{old}}^d + \text{rand.}(x_f^d - x_{i,\text{old}}^d)$$  \hspace{1cm} (8)

If $x_{i,\text{new}}^d$ is better than $x_{i,\text{old}}^d$, $x_{i,\text{new}}^d$ is accepted, otherwise $x_{i,\text{old}}^d$ is unchanged. The students having bad grades have less probability to update the position in the category of good students whereas students with good grades have relatively more probability to update their position. Poor student tutoring phase plays a keen role in the improving grades of the poor student to good student. This strategy is relevant to a real-time teaching-learning process where poor students always badly require tutorials for their improvement as compared to other good students. The pseudo-code for LebTLBO is given in Algorithm 1 as shown in the Appendix.

### 3.3 The proposed modified LebTLBO

LebTLBO has gained attention among researchers in the recent past due to its linear nature. The revised improved LebTLBO seeks to achieve enhanced performance by improving its basic exploration and exploitation capabilities both. Exploration has been improved by the introduction of search equations inspired from Grey wolf optimization (GWO) [13], exploitation has been enhanced by local neighbourhood search (LNS) driven by the knowledge of the finest solution so far found in the current solution's small neighbourhood [14]. The concept of LNS is discussed as follows:

#### 3.3.1 Local neighbourhood search (LNS)

LebTLBO uses the current solution and local information to improve the search space in the teacher phase. The local neighbourhood search (LNS) model is introduced in an attempt to further strengthen the search functionality to increase convergence speed.

The main idea is to use the finest solution found until now to upgrade the current solution in the neighbourhood of the current solution. For updating the position of the individual, the history of the neighbourhood of that individual is evaluated, and the graph of their interconnection is termed as the neighbourhood structure.

Assume that in the current LebTLBO population $X = \{X_1, X_2, X_3, \ldots, X_{NP}\}$, $X_i (i \in [1, NP])$ is a vector. The neighbourhood of radius $(2r + 1 < NP)$ is defined for each $X_i$ vector; that is, $X_i$ neighbourhood consists of $X_{i-r}, \ldots, X_{i}, \ldots, X_{i+r}$. For analysis, we assume that according to their indices, the vectors are organized into a ring topology. Figure 3 shows the local neighbourhood model concept. Therefore, the topology of the neighbourhood is static all the time and about the description of the set of vectors. The model for LNS is defined in

$$L_i = X_i + m \times (X_{i,\text{opt}} - X_i) + n \times (X_p - X_q)$$  \hspace{1cm} (9)

where $X_{i,\text{opt}}$ is the best solution in the $X_i$ neighbourhood and $p, q \in i - r, i + r \setminus \{i \neq q \neq i\}$, and $m, n \in \text{rand()}$ are the scaling factors. The new best solution is updated according to Equation (9) in the improved version of LebTLBO, and the updated solution performs the teacher phase.

$$X_i^{t+1} = L_i + r \times (X_{k}^{t} - X_{m}^{t})$$  \hspace{1cm} (10)

where $L_i$ is the best solution updated by LNS and $X_{k}^{t}$ and $X_{m}^{t}$ are two random solutions for $k$th and $m$th learner where $k \neq m$ and $r$ is a scaling factor, where $r \in \text{rand()}$.
STATISTICAL TESTING OF MODIFIED LEARNING
ENTHUSIASM-BASED TLBO ALGORITHM

A set of 10 CEC 2019 benchmark functions termed as the “100-Digit Challenge” [37] is selected for the proposed modified LebTLBO evaluation. These are all scalable test functions.

\[ L_t' = x_t' + u \times (x_{n, opt} - x_t') + v \times (x_p' - x_t') \] (16)

where \( x_{n, opt} \) is the best solution in the neighbourhood of \( x_t' \), and \( u, v \in \text{rand}() \) are the scaling factors. The updated solution performs the teacher phase using LNS is given by

Comparison with FPA [38], GWO [13], CS [39], FDO [40], SCA [41], TLBO [36], and LebTLBO [12] is done to check the effectiveness of its proposed variant. For every algorithm, a total of 500 iterations are performed using 30 agents. For all the algorithms under evaluation, the results are described in terms of the best, worst, mean, and std. dev. values for 30 independent runs. mLebTLBO achieves better than competitive algorithms, except in CEC06, CEC07, and CEC10 as shown in Table 1.

Figure 4 displays the convergence graph and box plot of various algorithms for the different benchmark functions. It has been found that the mLebTLBO algorithm’s results for function CEC01 are better in terms of the best value obtained in comparison to other algorithms. However, for the worst, average, and median values achieved, the results of TLBO are better. Most of the algorithms can find an optimal solution for function CEC02, but it is found that mLebTLBO is better for minimum standard deviation values attained. For the CEC03 function, all the algorithms achieve an almost globally optimal solution; but overall, in terms of minimum standard deviation, mLebTLBO is found to be better than other algorithms. Results of mLebTLBO are better than others in the case of CEC04 and CEC05, in terms of the best, worst, mean, and median fitness performance obtained.

Results of mLebTLBO are better than other algorithms for CEC06, in terms of the best fitness performance obtained. However, for worst, average, median and standard deviation values achieved; the results of CS are better than competitive algorithms. CS algorithm’s results for function CEC07 are better in terms of all performance metric. For CEC08 and CEC09 functions, mLebTLBO is found to be better than other algorithms. mLebTLBO’s results are competitive and better than others for function CEC10 in terms of the best value obtained.

Figure 4 also draws the box-plot of competitive algorithms for CEC2019 benchmark functions. The box-plots here are used to measure algorithm efficiency in terms of fitness values. It can be shown that in most cases the proposed mLebTLBO algorithm is cost-effective in fitness values as their median of fitness values is lower. Therefore, their overall performance is found to be consistent with other algorithms for optimization.
| Function | Parameters | Best     | Worst     | Average  | Median  | Std. dev. |
|----------|------------|----------|-----------|----------|---------|-----------|
| CEC01    | FPA        | 8.87E+07 | 1.41E+09  | 4.54E+08 | 4.10E+08| 2.69E+08  |
|          | GWO        | 4.98E+05 | 1.17E+09  | 1.87E+08 | 6.88E+07| 2.75E+08  |
|          | CS         | 1.00E+10 | 1.00E+10  | 1.00E+10 | 1.00E+10| 0.00E+00  |
|          | FDO        | 9.01E+07 | 8.03E+09  | 9.63E+08 | 4.82E+08| 1.48E+09  |
|          | SCA        | 3.11E+06 | 3.26E+10  | 6.43E+09 | 4.28E+09| 8.03E+09  |
|          | TLBO       | 2.50E+07 | 8.54E+08  | 1.55E+08 | 1.19E+08| 1.64E+08  |
|          | LebTLBO    | 5.33E+07 | 1.52E+09  | 5.75E+08 | 5.03E+08| 3.62E+08  |
|          | mLebTLBO   | 4.09E+04 | 1.53E+09  | 2.52E+08 | 1.50E+08| 2.90E+08  |
| CEC02    | FPA        | 1.85E+01 | 3.04E+01  | 2.26E+01 | 2.22E+01| 2.78E+00  |
|          | GWO        | 1.73E+01 | 1.73E+01  | 1.73E+01 | 1.73E+01| 2.57E+04  |
|          | CS         | 1.73E+01 | 1.73E+01  | 1.73E+01 | 1.73E+01| 2.77E+05  |
|          | FDO        | 1.73E+01 | 1.73E+01  | 1.73E+01 | 1.73E+01| 1.38E+04  |
|          | SCA        | 1.74E+01 | 1.77E+01  | 1.75E+01 | 1.75E+01| 8.54E+02  |
|          | TLBO       | 1.73E+01 | 1.73E+01  | 1.73E+01 | 1.73E+01| 7.14E-15  |
|          | LebTLBO    | 1.73E+01 | 1.73E+01  | 1.73E+01 | 1.73E+01| 7.23E-15  |
|          | mLebTLBO   | 1.73E+01 | 1.73E+01  | 1.73E+01 | 1.73E+01| 7.14E-15  |
| CEC03    | FPA        | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 1.06E-07  |
|          | GWO        | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 3.04E-04  |
|          | CS         | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 8.19E-11  |
|          | FDO        | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 2.00E-11  |
|          | SCA        | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 6.70E-05  |
|          | TLBO       | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 2.58E-09  |
|          | LebTLBO    | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 8.98E-08  |
|          | mLebTLBO   | 1.27E+01 | 1.27E+01  | 1.27E+01 | 1.27E+01| 4.05E-15  |
| CEC04    | FPA        | 7.72E+01 | 2.05E+02  | 1.36E+02 | 1.39E+02| 2.78E+01  |
|          | GWO        | 2.16E+01 | 2.46E+03  | 1.34E+02 | 5.02E+01| 4.41E+02  |
|          | CS         | 1.65E+01 | 3.89E+01  | 2.91E+01 | 2.88E+01| 5.66E+00  |
|          | FDO        | 1.40E+01 | 1.32E+02  | 4.90E+01 | 4.38E+01| 2.58E+01  |
|          | SCA        | 4.99E+02 | 3.64E+03  | 1.30E+03 | 1.20E+03| 6.66E+02  |
|          | TLBO       | 5.97E+00 | 8.85E+01  | 2.63E+01 | 2.30E+01| 1.69E+01  |
|          | LebTLBO    | 1.66E+01 | 5.84E+01  | 3.32E+01 | 3.32E+01| 9.03E+00  |
|          | mLebTLBO   | 2.87E+00 | 3.08E+01  | 1.65E+01 | 1.65E+01| 7.26E+00  |
| CEC05    | FPA        | 1.47E+00 | 1.71E+00  | 1.58E+00 | 1.58E+00| 6.60E-02  |
|          | GWO        | 1.06E+00 | 1.90E+00  | 1.32E+00 | 1.25E+00| 2.35E-01  |
|          | CS         | 1.04E+00 | 1.09E+00  | 1.06E+00 | 1.06E+00| 1.44E-02  |
|          | FDO        | 1.01E+00 | 1.25E+00  | 1.13E+00 | 1.13E+00| 6.75E-02  |
|          | SCA        | 2.01E+00 | 2.37E+00  | 2.18E+00 | 2.19E+00| 8.42E-02  |
|          | TLBO       | 1.02E+00 | 1.24E+00  | 1.09E+00 | 1.09E+00| 4.92E-02  |
|          | LebTLBO    | 1.02E+00 | 1.35E+00  | 1.13E+00 | 1.10E+00| 8.99E-02  |
|          | mLebTLBO   | 1.00E+00 | 1.08E+00  | 1.04E+00 | 1.03E+00| 2.20E-02  |
| CEC06    | FPA        | 8.83E+00 | 1.17E+01  | 1.06E+01 | 1.06E+01| 6.49E-01  |
|          | GWO        | 9.41E+00 | 1.20E+01  | 1.08E+01 | 1.07E+01| 6.76E-01  |
|          | CS         | 7.87E+00 | 1.02E+01  | 9.20E+00 | 9.19E+00| 5.39E-01  |
|          | FDO        | 8.59E+00 | 1.19E+01  | 1.08E+00 | 1.10E+00| 9.34E-01  |
|          | SCA        | 8.42E+00 | 1.19E+01  | 1.09E+01 | 1.09E+01| 7.43E-01  |
|          | TLBO       | 9.13E+00 | 1.17E+01  | 1.04E+01 | 1.05E+01| 5.82E-01  |
(Continues)
TABLE 1 (Continued)

| Function | Parameters | Best     | Worst    | Average  | Median   | Std. dev. |
|----------|------------|----------|----------|----------|----------|-----------|
|          | LebTLBO    | 8.99E+00 | 1.17E+01 | 1.07E+01 | 1.10E+01 | 7.15E-01  |
|          | mLLebTLBO  | 5.53E+00 | 1.12E+01 | 9.71E+00 | 9.73E+00 | 1.21E+00  |
| CEC07    | FPA        | 4.67E+01 | 4.08E+02 | 2.61E+02 | 2.80E+02 | 8.75E+01  |
|          | GWO        | 7.47E+00 | 9.76E+02 | 3.55E+02 | 2.99E+02 | 2.29E+02  |
|          | CS         | -1.21E+02| 1.71E+02 | 4.27E+01 | 3.11E+01 | 7.20E+01  |
|          | FDO        | -5.95E+01| 7.52E+02 | 2.77E+02 | 2.87E+02 | 2.10E+02  |
|          | SCA        | 4.82E+02 | 1.08E+03 | 7.71E+02 | 7.78E+02 | 1.44E+02  |
|          | TLBO       | 1.91E+02 | 7.93E+02 | 4.93E+02 | 4.85E+02 | 1.60E+02  |
|          | LebTLBO    | 4.40E+02 | 9.33E+02 | 7.26E+02 | 7.43E+02 | 1.37E+02  |
|          | mLebTLBO   | 3.79E+02 | 8.67E+02 | 6.62E+02 | 6.68E+02 | 1.26E+02  |
| CEC08    | FPA        | 5.11E+00 | 6.23E+00 | 5.79E+00 | 5.84E+00 | 3.11E-01  |
|          | GWO        | 3.48E+00 | 6.37E+00 | 5.01E+00 | 5.08E+00 | 8.12E-01  |
|          | CS         | 4.93E+00 | 5.95E+00 | 5.31E+00 | 5.31E+00 | 2.26E-01  |
|          | FDO        | 3.10E+00 | 6.50E+00 | 4.87E+00 | 4.93E+00 | 8.18E-01  |
|          | SCA        | 5.27E+00 | 6.84E+00 | 6.03E+00 | 5.95E+00 | 3.85E-01  |
|          | TLBO       | 2.03E+00 | 5.70E+00 | 4.13E+00 | 4.13E+00 | 9.70E-01  |
|          | LebTLBO    | 3.45E+00 | 6.05E+00 | 5.01E+00 | 5.12E+00 | 6.65E-01  |
|          | mLebTLBO   | 2.02E+00 | 5.33E+00 | 3.70E+00 | 3.69E+00 | 9.47E-01  |
| CEC09    | FPA        | 3.43E+00 | 7.28E+00 | 4.88E+00 | 4.79E+00 | 1.01E+00  |
|          | GWO        | 3.14E+00 | 6.40E+00 | 4.47E+00 | 4.42E+00 | 8.57E-01  |
|          | CS         | 2.55E+00 | 3.06E+00 | 2.77E+00 | 2.79E+00 | 1.27E-01  |
|          | FDO        | 2.35E+00 | 3.63E+00 | 2.54E+00 | 2.46E+00 | 2.38E-01  |
|          | SCA        | 1.59E+01 | 3.87E+02 | 9.10E+01 | 7.29E+01 | 7.43E+01  |
|          | TLBO       | 2.34E+00 | 2.38E+00 | 2.36E+00 | 2.36E+00 | 9.53E-03  |
|          | LebTLBO    | 2.36E+00 | 2.48E+00 | 2.41E+00 | 2.41E+00 | 3.12E-02  |
|          | mLebTLBO   | 2.34E+00 | 2.38E+00 | 2.35E+00 | 2.35E+00 | 6.92E-03  |
| CEC10    | FPA        | 2.01E+01 | 2.05E+01 | 2.04E+01 | 2.04E+01 | 9.96E-02  |
|          | GWO        | 2.03E+01 | 2.06E+01 | 2.05E+01 | 2.05E+01 | 8.52E-02  |
|          | CS         | 1.65E+01 | 2.04E+01 | 2.00E+01 | 2.03E+01 | 9.37E-01  |
|          | FDO        | 1.95E+01 | 2.00E+01 | 2.00E+01 | 2.00E+01 | 9.05E-02  |
|          | SCA        | 1.71E+01 | 2.06E+01 | 2.03E+01 | 2.05E+01 | 6.19E-01  |
|          | TLBO       | 2.58E+00 | 2.05E+01 | 1.89E+01 | 2.04E+01 | 4.70E+00  |
|          | LebTLBO    | 1.52E+01 | 2.05E+01 | 2.01E+01 | 2.04E+01 | 1.06E+00  |
|          | mLLebTLBO  | 4.30E-02 | 2.05E+01 | 1.90E+01 | 2.04E+01 | 4.86E+00  |

5 SINGLE ANCHOR NODE DEPLOYMENT IN 3D WSN

In the problem based on 3D localization, we considered a single sensor node whose location is defined prior. Using the knowledge of this anchor, the locations of mobile nodes are identified that are randomly deployed in the sensor field. These nodes are divided into three separate layers with an anchor placed at the top position, and unlocated nodes are placed at the middle and bottom layers are positioned using Parasol projection as shown in Figure 5.

The anchor node transmits a beacon signal that helps the target nodes to identify themselves Figure 6. Once these target nodes drop under the range of their respective anchor, they will listen to beacon first and use this information to retrieve the RSSI information. Then using RSSI information, the Euclidean distance between target and anchor node is calculated. The pseudo code representing parasol and localization using parasol projection are described in Algorithm 3 and 4 as shown in Appendix.

The steps mentioned below are used for locating target nodes in 3D scenario:
a. $N$ numbers of target nodes are assumed for this 3D localization problem that are randomly deployed at middle layer and bottom layer of a chaotic cubic structure having three layers. On the topmost layer, a single anchor is deployed. The transmission range for both target and anchor nodes are different (i.e. variation in battery backups, radiation patter, DOI).

b. Whenever a moving target node comes under the range of a particular anchor, a target node is said to be localizable. The target nodes maintain a list of their in-range anchors which contains the useful distance information. Along with an anchor node, three other virtual anchors are projected in surrounding of target node to locate itself as (minimum four anchors condition is required to locate target nodes in 3D scenario).

c. Using radio irregularity model, distance between the anchor and target node is calculated based on obtained RSSI \[42\] using $d_i^\Lambda = d_i - (D_p + F)$, as $d_i$ is calculated using Equation (17). $D_p$ is DOI path loss and $F$ represents fading.

$$d_i = \sqrt{(x_t - x)^2 + (y_t - y)^2 + (z_t - z)^2} \quad (17)$$

where $(x_t, y_t, z_t)$ represents the target node position for a 3D scenario, $(x, y, z)$ represent anchor node’s current location for 3D scenario. The centroid $(x_c, y_c, z_c)$ is calculated using Equation (18) for 3D scenario and is represented in Figure 7.

$$\begin{align*}
(x_c, y_c, z_c) &= \left(\frac{x + x_1 + x_2 + x_3}{3}, \frac{y + y_1 + y_2 + y_3}{3}, \frac{z + z_1 + z_2 + z_3}{3}\right)
\end{align*} \quad (18)$$
FIGURE 4  Continued
Figure 8 shows that proposed mLebTLBO is applied for the coordinate determination of target node represented by \((x_s, y_s, z_s)\). The objective function used in Equation (19) will minimize distance among the deployed and estimated target node coordinates.

\[
f(x_s, y_s, z_s) = \frac{1}{M} \sum \left( \sqrt{(x_e - x_i)^2 + (y_e - y_i)^2 + (z_e - z_i)^2 - d_i^2} \right)^2
\] (19)

Here, \((x_e, y_e, z_e)\) represents the estimated target node position, \((x_i, y_i, z_i)\) represents the location of beacon node \(i\) and estimated location of target nodes \((M > 4\) to compute 3D location\) respectively for the 3D scenario.

\(E_t\), represents error taking place in the process of localization and is calculated using Equation (20) and represented in Figure 9 for 3D scenario.

\[
E_t = \frac{1}{N_L} \sum \sqrt{(x_e - x_t)^2 + (y_e - y_t)^2 + (z_e - z_t)^2}
\] (20)

6 | RESULTS AND DISCUSSIONS FOR 3D SCENARIO

A novel proposed mLebTLBO is applied for a 3D localization problem using the principle of deploying a single anchor and six virtual anchors presumed to be positioned in six
### TABLE 2  Parameter settings

| Algorithm   | Parameters                                                                 |
|-------------|-----------------------------------------------------------------------------|
| PSO         | $NP = 20; D = 3; G_{\text{max}} = 100; c_1, c_2, c_3 = 1.494; w = 0.729$     |
| HPSO        | $NP = 30; D = 3; G_{\text{max}} = 100; c_1, c_2, c_3 = 1.494; \eta = 0.1; w = 0.729$ |
| BBO         | $NP = 30; D = 3; G_{\text{max}} = 100; p_w = 0.05$                           |
| FA          | $NP = 20; D = 5; G_{\text{max}} = 100; \alpha = 0.2; \gamma = 0.96$         |
| mLbTLBO     | $NP = 20; D = 3; G_{\text{max}} = 50; F = 0.9; CR = 0.5; \min_{\text{LE}} = 0.3$ |

Here, $NP$ is number of population, $D$ is dimension of problem, $G_{\text{max}}$ is number of iterations.

**FIGURE 10** Target node movement using BBO

**FIGURE 11** Target node movement with PSO

**FIGURE 12** Target node movement with FA

**FIGURE 13** Target node movement with HPSO

**FIGURE 14** Target node movement with mLbTLBO

Directions at an angle of approximately $60^\circ$ to locate all mobile sensor nodes. The 3D localization simulations in dynamic WSN are performed in MATLAB environment. These simulations are performed on MacBook Air having 4 GB of RAM. Only a single anchor is used as a reference node to locate all 80 target nodes in the 3D network. For a 3D scenario, a cubic structure is considered where the structure is divided into three layers. An anchor node is located at the top layer, and target nodes are deployed at the middle and bottom layers (40 each). Whenever the moving target nodes fall within the range of anchor nodes the above-mentioned scenario is considered. With the help of...
### Table 3: Comparison of meta-heuristic algorithms

| Optimization | Number of movements | Localization error (Max) | Localization error (Min) | Average error | Number of located targets |
|--------------|---------------------|--------------------------|--------------------------|---------------|--------------------------|
| PSO          | 1                   | 3.9358                   | 0.0554                   | 0.9861        | 80                       |
|              | 2                   | 5.3379                   | 0.0831                   | 0.9764        | 80                       |
|              | 3                   | 5.0108                   | 0.0800                   | 0.9124        | 80                       |
|              | 4                   | 5.1655                   | 0.0367                   | 0.9741        | 80                       |
|              | 5                   | 5.1655                   | 0.0800                   | 0.9267        | 80                       |
| HPSO         | 1                   | 3.1204                   | 0.1044                   | 0.6742        | 80                       |
|              | 2                   | 5.0134                   | 0.0647                   | 0.4876        | 80                       |
|              | 3                   | 4.8279                   | 0.0976                   | 0.4032        | 80                       |
|              | 4                   | 5.2356                   | 0.0247                   | 0.5576        | 80                       |
|              | 5                   | 5.2179                   | 0.0376                   | 0.5232        | 80                       |
| BBO          | 1                   | 5.8904                   | 0.1822                   | 1.1892        | 80                       |
|              | 2                   | 5.3500                   | 0.3318                   | 1.2560        | 80                       |
|              | 3                   | 5.5989                   | 0.1822                   | 1.1585        | 80                       |
|              | 4                   | 5.6356                   | 0.1547                   | 1.2576        | 80                       |
|              | 5                   | 5.9179                   | 0.1976                   | 1.1932        | 80                       |
| FA           | 1                   | 6.1904                   | 0.1992                   | 2.1792        | 80                       |
|              | 2                   | 6.3500                   | 0.3518                   | 2.2670        | 80                       |
|              | 3                   | 6.5989                   | 0.1922                   | 2.4585        | 80                       |
|              | 4                   | 6.6356                   | 0.1747                   | 2.5576        | 80                       |
|              | 5                   | 6.9529                   | 0.2976                   | 2.1932        | 80                       |
| mLebTLBO     | 1                   | 3.2904                   | 0.0992                   | 0.6312        | 80                       |
|              | 2                   | 4.7500                   | 0.0518                   | 0.4570        | 80                       |
|              | 3                   | 4.3989                   | 0.0972                   | 0.3785        | 80                       |
|              | 4                   | 4.4356                   | 0.0447                   | 0.4132        | 80                       |
|              | 5                   | 4.5529                   | 0.0476                   | 0.3815        | 80                       |

An anchor node, a parasol projection is formed, and the virtual anchors are assumed for determining 3D locations of target nodes. It is also theoretically possible to deploy more than six virtual anchors, but in this work, we assumed only six virtual anchors in a 3D environment for locating target nodes by selecting only four nearby anchor nodes.

Various optimization algorithms are discussed below in the mobility-based scenario. Anchor node is kept fixed for all target nodes, while mobility is applied for all target nodes. The fitness function is calculated using Equation (19). The strategic settings considered for various meta-heuristic optimization algorithms is given in Table 2.

In a mobility-based scenario, various optimization algorithms are evaluated discussed above. The target nodes are deployed randomly at the bottom and middle layers, and the topmost layer is equipped with anchor node. Anchor node is kept fixed while mobility is applied for all target nodes. The fitness function is calculated using the average localization error given in Equation (16). The results of these different optimization algorithms are drawn in Figures 10–14. By using a virtual anchor concept, LoS problem is also minimized to a greater extent. From these results, it is quite clear that mLebTLBO outperforms in the context of other meta-heuristic optimization algorithms and has a faster convergence rate.

In Table 3 the average localization error is calculated for all competitive algorithms and is presented in Figure 15. The average localization error of mLebTLBO is quite less as compared to other competitive algorithms tested for the same scenario.

Algorithmic complexity can be described as the standard notions (big O notation) of computational complexity in time. That is how long a localization algorithm runs before estimating the positions of all the nodes in the network and how much memory (storage) is needed for such calculations. The time complexity of the proposed method is obtained by the following analysis. The method is built on two major components: localization and optimization algorithm.

In localization, the cost of communication is the cost incurred by the network to spread out the information about the hop size between every connected pair of the nodes. In the algorithm, each anchor node out of total anchor nodes ‘n’ has to inform all the target nodes ‘N’ of the network about its hop size.
value. So, every node informs every other node in the network about the hop size of an anchor node. Repeatedly this controlled flooding takes place the same number of times as that of the number of anchor nodes. Therefore, the communicational cost is \(O(mN^2)\). Here in our algorithm, only one anchor node is considered, therefore, the overall communicational cost is \(O(N^2)\).

The computational complexity of mLebTLBO according to the major steps of the algorithm is given by:

- For a population size of \(NP\), the computational complexity for initialization is given by \(O(NP)\).
- After initialization, the second step is to evaluate the current population and hence the complexity is given by \(O(NP)\).
- Ranking the population and finding the current best solution has a complexity given by \(O(NP^2)\).
- In the iterative search, the positions of existing solutions is changed continuously and their fitness is calculated as \(O(2NP)\).
- Even in the iterative search, the population is sorted and the best individual is calculated. The complexity for this operation is \(O(NP^2)\).

Thus, based on the above points, the total run time complexity of the mLebTLBO for \(G_{max}\) iterations is given by:

\[
O(2NP) + O(NP^2) + G_{max} \left( O(2NP) + O(NP^2) \right) = (G_{max} + 1) \left( O(NP^2 + 2NP) \right) \tag{21}
\]

Overall, the generalized run time complexity of the mLebTLBO for a \(D\)-dimensional search space is then given by \(DG_{max}(O(NP^2 + 2NP))\).

Therefore, we can say that the time complexity of localization algorithm using mLebTLBO algorithm is \(O(N^2) + DG_{max}(O(NP^2 + 2NP))\).

7 | CONCLUSIONS

The range-based technique along with the proposed mLebTLBO optimization technique was used in this research to obtain 3D target node coordinates using the concept of a single anchor node. The concept of anchor and virtual anchor node forms a parasol projection to locate all target nodes. Whenever the moving targets fall within the anchor node; three virtual anchors and anchor itself operate together to locate the target nodes (as to obtain 3D locations at least 4 nodes are required). There are various applications like logistics, underwater scenario, localization of occurring events in remote and hilly areas, and tracking of workers in the coal mine industry where localization of sensor nodes is essential. The proposed mLebTLBO algorithm performs in comparison with other competitive algorithms to estimate node location precisely. The results achieved show the accuracy of mLebTLBO is higher and it has faster convergence characteristics.

One can implement another hybrid optimization approach in the future to achieve better accuracy and faster convergence.

REFERENCES

1. Aldeer, M. M. N.: A summary survey on recent applications of wireless sensor networks. In: Proceeding of IEEE Student Conference on Research and Development (SCORed). IEEE, Putrajaya, Malaysia, pp. 485–490 (2013)
2. Kulkarni, R.V., Venayagamoorthy, G.K.: Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 41(2), 262–267 (2011)
3. Akyildiz, I.F., Sankarasubramaniam, Y., Cayirli, E.: Wireless sensor networks: A survey. J. Comput. Networks 38, 393–422 (2002)
4. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. J. Comput. Networks 52, 2292–2302 (2008)
5. Bizagwira, H., Toussaint, J., Misson, M.: Synchronization protocol for dynamic environment: Design and prototype experiments. In: Proceeding of 23rd International Conference on Telecommunications (ICT). IEEE, Thessaloniki, Greece, pp. 1–7 (2016)
6. Kalnoor, G., Agrahakud, J.: QoS based Multipath Routing for Intrusion Detection of Sinkhole Attack in Wireless Sensor Networks. In: Proceeding of International Conference on Circuit, Power and Computing Technologies (ICCPCT). IEEE, Nagercoil, India, pp. 1–6 (2016)

7. Khan, A. A., Agrawal, H.: Optimization of delay of data delivery in wireless sensor network using genetic algorithm. In: Proceeding of International Conference on Computation of Power, Energy Information and Communication (ICCPEIC). IEEE, Chennai, India, pp. 159–164 (2016)

8. Shieh, C-S., et al.: Improved node localization for WSN using heuristic optimization approaches. In: Proceeding of International conference on Networking and Network Applications. IEEE, Hokkaido, Japan, pp. 95–98 (2016)

9. Liu, Z., Liu, Z.: Node self-localization algorithm for wireless sensor networks based on modified particle swarm optimization. In: Proceeding of 27th Chinese Control and Decision Conference (CCDC). IEEE, Qingdao, China, pp. 5968–5971 (2015)

10. Pei, B., et al.: Firefly algorithm optimization based WSN localization algorithm. In: Proceeding of International Conference on Information and Communications Technologies (ICT (2015)). IEEE, Ahmedabad, India, pp. 1–5 (2015)

11. Mao, G., Fidan, B., Anderson, B. D. O.: Wireless sensor network localization techniques. J. Comput. Netw. 51, 2529–2553 (2007)

12. Xu, C., et al.: Teaching-learning-based optimization with learning enthusiasm mechanism and its application in chemical engineering. J. Appl. Math. 2018, 1806947 (2018). https://doi.org/10.1155/2018/1806947

13. Mirjalili, S., Mirjalili, S. M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Software 69, 46–61 (2014)

14. Das, S., et al.: Differential evolution using a neighborhood-based mutation operator. IEEE Trans. Evol. Comput. 13(3), 526–553 (2009)

15. Liu, Y., et al.: Efficient quality-of-service (QoS) support in mobile opportunistic networks. IEEE Trans. Veh. Technol. 63(9), 4574–4584 (2014)

16. Liu, Y., et al.: Efficient data query in intermittently-connected mobile Ad hoc social networks. IEEE Trans. Parallel Distrib. Syst. 26(5), 1301–1312 (2015)

17. Gopakumar, A., Jacob, J.: Localization in wireless sensor networks using particle swarm optimization. In: Proceedings of IET International Conference on Wireless, Mobile Multimedia Networks. Beijing, pp. 227–230 (2008)

18. Kulkarni, R. V., Venayagamoorthy, G. K., Cheng, M. X.: Bio-inspired node localization in wireless sensor networks. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics. San Antonio, TX, pp. 205–210 (2009)

19. Low, K. S., Nguyen, H. A., Guo, H.: A particle swarm optimization approach for the localization of a wireless sensor network. In: Proceedings of the International Symposium on Industrial Electronics, ISIE 2008. IEEE, Cambridge, UK, pp. 1820–1825 (2008)

20. Low, H. S., Nguyen, A., Guo, H.: Optimization of sensor node locations in a wireless sensor network. In: Proceedings of 4th International Conference on Natural Computation. Jinan, China, pp. 286–290 (2008)

21. Elberhartz, R. S., Shi, Y. H.: Particle swarm optimization: Developments, applications and resources. In: Proceedings of IEEE Congress on Evolutionary Computation. Seoul, Korea, pp. 81–86 (2001)

22. Alberti, A., Biber, P., Schilling, A.: Wireless node localization based on rssi using a rotating antenna on a mobile robot. In: 6th Workshop on Positioning, Navigation and Communication, WPNC2009. IEEE, Hannover, Germany, pp. 253–9 (2009)

23. Sumathi, R., Srinivasan, R.: RSS-based location estimation in mobility assisted wireless sensor networks. In: IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS). IEEE, Prague, Czech Republic, pp. 848–52 (2011)

24. Guo, Z., et al.: Perpendicular intersection: Locating wireless sensors with mobile beacon. IEEE Trans. Veh. Technol. 59(7), 3501–3509 (2010)

25. Shi, Q., et al.: A 3d node localization scheme for wireless sensor networks. IEICE Electron Exp. 6(3), 167–172 (2009)

26. Wang, L., Zhang, J., Cao, D.: A new 3-dimensional dv-hop localization algorithm. J. Comput. Inf. Syst. 8(6), 2463–2475 (2012)

27. Xu, Y., Yi, Z., Gu, J. J.: An improved 3D localization algorithm for the wireless sensor network. Int. J. Distrib. Sens. Networks 11(6), 1–9 (2015)

28. Li, J., Zhong, X., Lu, I-T.: Three-dimensional node localization algorithm for wsn based on differential rss irregular transmission model. J. Commun. 9(5), 391–397 (2014)

29. Ahmad, T., Li, X. J., Seet, B.-C.: Parametric loop division for 3d localization in wireless sensor networks. Sensors 17(7), 1697 (2017)

30. Gopakumar, A., Jacob, J.: Localization in wireless sensor networks using particle swarm optimization. In: 2008 IET International Conference on Wireless, Mobile and Multimedia Networks. Beijing, pp. 227–230 (2008)

31. Chuang, P.-J., Wu, C.-P.: An effective PSO-based node localization scheme for wireless sensor networks. In: Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2008. IEEE, Otago, pp. 187–194 (2008)

32. Kennedy, J.: Bare bones particle swarms. In: Swarm Intelligence Symposium, SIS’03. Indianapolis, IN, pp. 80–87 (2003)

33. Kulkarni, R. V., Venayagamoorthy, G. K.: Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 41(2), 262–267 (2011)

34. Kumar, A., et al.: Meta-heuristic range-based node localization algorithm for wireless sensor networks. In: International Conference on Localization and GNSS (ICL-GNSS). Stamberg, Germany, pp. 1–7 (2012)

35. Arora, S., Singh, S.: Node localization in wireless sensor networks using butterfly optimization algorithm. Arab J. Sci. Eng. 42, 3325–3333 (2017)

36. Rao, R. V., Savsani, V. J., Vakharia, D. P.: Teaching-learning-based optimization: An optimization method for continuous non-linear large-scale problems. Inf. Sci. 183, 1–15 (2012)

37. Brest, J., Mauˇcek, M. S., Boškovi´c, B.: The 100-Digit Challenge: Algorithm Operators and Their ASM Mechanism and Its Application in Chemical Engineering. J. Appl. Math. and Computation. Berlin Heidelberg: Springer, pp. 240–249 (2012)

38. Yang, X. S.: Flower pollination algorithm for global optimization. In: International Conference on Unconventional Computing and Natural Computation. Berlin Heidelberg: Springer, pp. 240–249 (2012)

39. Yang, X. S., Deb, S.: Cuckoo search via Lévy flights. In: 2009 World Congress on Nature and Biologically Inspired Computing (NaBIC). Coimbatore, India, pp. 210–214 (2009)

40. Abdullah, J. M., Ahmed, T.: Fitness dependent optimizer: Inspired by the bee swarming reproductive process. IEEE Access 7, 43473–43486 (2019)

41. Mirjalili, S.: SCA: A sine cosine algorithm for solving optimization problems. Knowledge-Based Syst. 96, 120–133 (2016)

42. Singh, P., Mittal, N.: An efficient localization approach for WSNs using hybrid DA-FA algorithm. IET Commun. 14(12), 1975–1991 (2020)

How to cite this article: Singh, P., Mittal, N.: Optimized localization of sensor nodes in 3D WSNs using modified learning enthusiasm-based teaching learning based optimization algorithm. IET Commun. (2021):15:1223–1239. https://doi.org/10.1049/cm.12155
APPENDIX

ALGORITHM 1 Pseudo code of LebTLBO

1 Start
2 Initialize learners and evaluate them
3 while
4 Select best learner as x_{teacher} and estimate the mean x_{mean} of all learners
5 Sort all the learners from best to worst; (Learning Enthusiasm Based Teacher Phase)
6 Calculate the learning enthusiasm of all learners using equation
7 \[ \text{LE}_i = \text{LE}_{\text{max}} + (\text{LE}_{\text{max}} - \text{LE}_{\text{min}})(\frac{\text{NP} - i}{\text{NP}}) \]
8 for each learner x_i
9 Generate a random \( r_i \) in \([0, 1]\)
10 Update learner x_i according to equation
11 \( x_{i,new} = \begin{cases} x_{i,old} + \text{rand}(d) (x_{i,old} - x_{j,old}) & \text{if rand1} < 0.5 \\ x_{i,old} + F \times (x_{d,old} - x_{i,old}) & \text{otherwise} \end{cases} \)
12 Evaluate the new learner x_{i,new}
13 Accept x_{i,new} if it is better than the old one x_{i,old}
14 Select another learner at random that is different from that
15 end if
16 end while
17 Sort all the learners from best to worst; (Learning Enthusiasm Based Learner Phase)
18 Calculate the learning enthusiasm of all learners using equation
19 \[ \text{LE}_i = \text{LE}_{\text{max}} + (\text{LE}_{\text{max}} - \text{LE}_{\text{min}})(\frac{\text{NP} - i}{\text{NP}}) \]
20 for each learner x_i
21 Generate a random \( r_i \) in \([0, 1]\)
22 Update learner x_i according to equation
23 \( x_{i,new} = \begin{cases} x_{i,old} + \text{rand}(d) (x_{i,old} - x_{j,old}) & \text{if } f(x_{i,new}) < f(x_{j,old}) \\ x_{i,old} + \text{rand}(d) (x_{i,old} - x_{j,old}) & \text{otherwise} \end{cases} \)
24 Evaluate the new learner x_{i,new}
25 Accept x_{i,new} if it is better than the old one x_{i,old}
26 end if
27 Sort learners from best to worst; (Poor Student Tutoring Phase)
28 The learners ranked at the bottom 10\% are considered as the poor students
29 for each poor learner x_i
30 Randomly select a student x_T ranked at the top 50\%
31 Update learner x_i according to equation
32 \( x_{i,new} = x_{i,old} + \text{rand}(T) (x_T - x_{i,old}) \)
33 Evaluate the new learner x_{i,new}
34 Accept x_{i,new} if it is better than the old one x_{i,old}
35 end for
36 Finished

ALGORITHM 2 Pseudo code of mLbTLBO

1 Start
2 Initialize learners and evaluate them
3 while
4 Select best learner as x_{teacher} and estimate the mean x_{mean} of all learners
5 Sort all the learners from best to worst; (Learning Enthusiasm Based Teacher Phase)
6 Calculate the learning enthusiasm of all learners using equation
7 \[ \text{LE}_i = \text{LE}_{\text{max}} + (\text{LE}_{\text{max}} - \text{LE}_{\text{min}})(\frac{\text{NP} - i}{\text{NP}}) \]
8 for each learner x_i
9 Generate a random \( r_i \) in \([0, 1]\)
10 Update learner x_i according to local neighbourhood search (LNS) using Equation (16)% For 2nd half of iterations
11 Update learner x_i according to Equation (14)
12 \( x_{i,new} = \begin{cases} x_{GWO} + \text{rand} (x_{i,new} - x_{teacher} - F \times \text{rand}_d) & \text{if } \text{rand1} < 0.5 \\ x_{i,old} + F \times (x_{i,old} - x_{j,old}) & \text{otherwise} \end{cases} \)
13 where \( x_{GWO} = \frac{x_{i,old} + x_{j,old}}{2} \) using Equation (12)
14 Evaluate the new learner x_{i,new}
15 Accept x_{i,new} if it is better than the old one x_{i,old}
16 Select another learner at random that is different from that
17 end if
18 end while
19 Sort all the learners from best to worst; (Learning Enthusiasm Based Learner Phase)
20 Calculate the learning enthusiasm of all learners using equation
21 \[ \text{LE}_i = \text{LE}_{\text{max}} + (\text{LE}_{\text{max}} - \text{LE}_{\text{min}})(\frac{\text{NP} - i}{\text{NP}}) \]
22 for each learner x_i
23 Generate a random \( r_i \) in \([0, 1]\)
24 if \( r_i \leq \text{LE}_i \)
25 Update learner x_i according to equation
26 \( x_{i,new} = \begin{cases} x_{i,old} + \text{rand}(d) (x_{i,old} - x_{j,old}) & \text{if } f(x_{i,new}) < f(x_{j,old}) \\ x_{i,old} + \text{rand}(d) (x_{i,old} - x_{j,old}) & \text{otherwise} \end{cases} \)
27 Evaluate the new learner x_{i,new}
28 Accept x_{i,new} if it is better than the old one x_{i,old}
29 end if
30 end for
31 Sort learners from best to worst; (Poor Student Tutoring Phase)
32 The learners ranked at the bottom 10\% are considered as the poor students
33 for each poor learner x_i
34 Randomly select a student x_T ranked at the top 50\%
35 Update learner x_i according to equation
36 \( x_{i,new} = x_{i,old} + \text{rand}(T) (x_T - x_{i,old}) \)
37 Evaluate the new learner x_{i,new}
38 Accept x_{i,new} if it is better than the old one x_{i,old}
39 end for
40 Finished
**Algorithm 3** Pseudo code of parasol projection

1. **Start**
2. for every target node do
3. Use RSS information to determine distance between target and anchor node
4. Six virtual anchors are projected at calculated distance with an angle difference of 60 degree
5. Using a single anchor and three virtual anchors for every target node, centroid is calculated and then deploying the particles around the centroid using
   \[(X_{\text{centroid}} + \text{radius}.\cos\theta + Y_{\text{centroid}} + \text{radius}.\sin\theta + Z_{\text{centroid}} + \text{radius}.\cos\theta\cdot\sin\theta)\]
6. **end for**

**Algorithm 4** Pseudo code for localization using parasol projection

1. Deploy target node randomly at middle and bottom layer
2. Deploy a single anchor at topmost layer
3. for movement start for target node do
4. for every target node do
5. Use RSS information to determine distance between target and anchor node
6. for every target node in range with anchor node do
7. Using a single anchor and three virtual anchors for every target node, centroid is calculated
8. if number of anchor and virtual anchor nodes \(\geq 4\) then
9. Calculate centroid
10. Deploy the particles around the centroid using
    \[(X_{\text{centroid}} + \text{radius}.\cos\theta + Y_{\text{centroid}} + \text{radius}.\sin\theta + Z_{\text{centroid}} + \text{radius}.\cos\theta\cdot\sin\theta)\]
11. Apply various meta heuristic algorithm along with novel mLebTLBO
12. **end if**
13. **end for**
14. **end for**
15. **end for**