International Journal of Sensor Networks

ISSN online: 1748-1287 - ISSN print: 1748-1279
https://www.inderscience.com/ijsnet

Node WSN localisation based on adaptive crossover-mutation differential evolution

Trong-The Nguyen, Thi-Kien Dao, Truong-Giang Ngo, Trinh-Dong Nguyen

DOI: 10.1504/IJSNET.2024.10061733

Article History:

Received:	30 May 2023
Last revised:	30 September 2023
Accepted:	30 September 2023
Published online:	30 January 2024

Node WSN localisation based on adaptive crossover-mutation differential evolution

Trong-The Nguyen and Thi-Kien Dao*

Fujian Provincial Key Laboratory of Big Data Mining and Applications,
School of Computer Science and Mathematics,
Fujian University of Technology,
Fuzhou 350118, China
and
University of Information Technology,
Ho Chi Minh City, Vietnam
Email: thent@uit.edu.vn
Email: kiendt@uit.edu.vn
and
Vietnam National University,
Ho Chi Minh City 700000, Vietnam

Truong-Giang Ngo

Faculty of Computer Science and Engineering,
Thuyloi University,
Hanoi, Vietnam
Email: giangnt@tlu.edu.vn

Trinh-Dong Nguyen

University of Information Technology,
Ho Chi Minh City, Vietnam
Email: dongnt@uit.edu.vn
and
Vietnam National University,
Ho Chi Minh City 700000, Vietnam

Abstract: Accurate node positioning in wireless sensor networks (WSNs) is essential for optimising monitoring and tracking applications. This becomes increasingly challenging, especially in large-scale WSNs where precise location data is needed for unknown nodes. Traditional methods often struggle with computational complexity, particularly in enlarged network setups. To address this issue, we introduce an innovative adaptive optimisation approach called crossover mutation differential evolution (ACMDE) tailored for node localisation in WSNs. ACMDE rapidly localises unknown nodes by leveraging location data and employing adaptive optimisation strategies, including enhanced crossover, mutation, and reinitialisation techniques. The objective function is modelled for the WSNs node localisation to minimise localisation errors between actual and detected node positions that are obtained optimisation targets through ACMDE's superior capabilities. The ACMDE's effectiveness is evaluated in the test suits and node localisation through comprehensive comparisons with existing strategies using various metrics. Experimental results unequivocally demonstrate that ACMDE outperforms competing algorithms in node localisation within WSNs.

Keywords: node localisation; ACMDE algorithm; wireless sensor networks; differential evolution; DE; optimisation.

Reference to this paper should be made as follows: Nguyen, T-T., Dao, T-K., Ngo, T-G. and Nguyen, T-D. (2024) 'Node WSN localisation based on adaptive crossover-mutation differential evolution', *Int. J. Sensor Networks*, Vol. 44, No. 1, pp.1–22.

Biographical notes: Trong-The Nguyen obtained his PhD in Communication Engineering from the National Kaohsiung University of Applied Sciences, Taiwan, in 2016. He currently serves as a Lecturer in the School of Computer Science and Mathematics at Fujian University of Technology, China and a researcher with UIT-MMLab. His research focuses on computational intelligence, grid computing, signal processing, electrical engineering, and sensor networks.

Thi-Kien Dao earned her PhD in Electronics and Engineering from the National Kaohsiung University of Technology and Sciences, Taiwan, in 2019. Presently, she serves as a Lecturer in the School of Computer Science and Mathematics at Fujian University of Technology. Her research focuses on computational intelligence, data mining, sensor networks, and grid computing.

Truong-Giang Ngo earned his PhD in Mathematical Theory for Informatics from the Graduate University of Science and Technology, Vietnam Academy of Science and Technology, Vietnam, in 2017. He is currently a Lecturer at the Faculty of Computer Science and Engineering at Thuyloi University, Vietnam. His research focuses on machine learning, data mining, networking, and computational intelligence.

Trinh-Dong Nguyen obtained his PhD in Software Engineering from VNU Hanoi-University of Engineering and Technology in 2018. He currently holds a position as a lecturer in the Faculty of Software Engineering at the University of Information Technology, VNUHCM Vietnam. His research interests encompass software engineering, real-time systems, formal methods, computational intelligence, networking, and machine learning.

1 Introduction

Wireless sensor networks (WSNs) (Shiva et al., 2012) are critical assets deployed in diverse sectors, such as surveillance, military operations, healthcare, agriculture, and astronomy (Nguyen et al., 2016a). These networks, comprising small sensor nodes, offer advantages like self-organisation (Zhu et al., 2012), rapid deployment, and seamless integration with the internet or cloud environments for efficient data transfer (Nguyen et al., 2019a). Whether heterogeneous or homogeneous, sensor nodes are strategically positioned to monitor environmental and physical variables and are responsible for data observation, processing, wireless communication, and energy management (Singh et al., 2021). In some scenarios, sensor nodes are equipped with global positioning system (GPS) technology, broadening their utility in deepwater exploration, space missions, and various monitoring applications (Niewiadomska-Szynkiewicz, 2012). Precise sensor node localisation is crucial, particularly in mobility-aided WSNs (Zhang et al., 2018) where vehicles equipped with mobile sensor nodes collect data without relying on external communication infrastructure (Nguyen et al., 2019b).

Location information is the primary interest data in specific WSNs characterised by random sensor node deployment. Consequently, substantial research efforts have been devoted to reducing localisation errors, resulting in various algorithms and models. Two prominent categories of WSN positioning techniques, namely range-based and range-free methods, are employed. Range-based methods, prised for accuracy, calculate distances between beacon nodes (nodes with known coordinates) and unknown nodes. Conversely, range-free methods offer cost-effective solutions but with less precision, omitting distance measurements during placement determination. This study centres on precise node localisation within WSNs, an essential aspect of WSN functionality. Various meta-heuristic algorithms have been developed to address the challenges posed by node localisation.

Recent research has introduced strategies to enhance the accuracy of WSN node localisation, encompassing evolution-based approaches (Beheshti and Shamsuddin, 2013; Sivakumar and Venkatesan, 2015) such as the genetic algorithm (GA) (Holland, 1992) and differential evolution (DE) (Price et al., 2005), physics-based techniques like simulated annealing (SA) (Kirkpatrick et al., 1983), and swarm-based methods including particle swarm optimisation (PSO) (Kennedy and Eberhart, 1948), Gray Wolf optimisation (GWO) (Mirjalili et al., 2014), ant lion optimiser (ALO) (Mirjalili, 2015a), moth-flame optimisation (MFO) (Mirjalili, 2015b), and whale optimisation algorithm (WOA) (Mirjalili and Lewis, 2016).

DE (Price et al., 2005; Bilal et al., 2020), initially conceived by Storn and Price, has emerged as a robust stochastic population-based optimisation algorithm, renowned for its versatility and efficiency in addressing complex computational problems (Liu and Lampinen, 2005). However, its performance is heavily reliant on the selection of control parameters, with improper choices leading to issues like premature convergence and stagnation (Gong et al., 2011). Consequently, optimising DE and addressing these limitations hold significant academic importance, resulting in the development of strategies and enhancements to mitigate these challenges (Zhang and Sanderson, 2009; Qin and Suganthan, 2005; Omran et al., 2005; Zhang, 2023).

This study introduces an investigative approach named adaptive crossover mutation differential evolution (ACMDE) to enhance DE's capabilities, particularly in the context of the challenging node localisation problem within WSNs. The ACMDE integrates adaptive crossover, mutation, chaotic mapping, and inertia weighting techniques to mitigate issues related to local optima and stagnation inherent in the original DE algorithm. The ACMDE is meticulously engineered to navigate complex optimisation problems and surmount challenges in node localisation, ultimately delivering superior performance. The ACMDE's innovative features include chaotic mapping during population initialisation and an enhanced mutation operation that replaces the static scaling factor with a dynamic adaptive counterpart to address late convergence.

This research represents a significant advancement aimed at enhancing DE's performance, particularly in complex, multimodal optimisation scenarios. Key highlights of this investigation encompass:

- Strategic parameter adjustments within DE, including re-initialising pivotal individual locations, mutation and crossover variable refinements, and tuning weighting parameters to enhance population diversity and alleviate local optima.
- The development of a node localisation model encompassing objective functions that consider error derivation (Err), delay (del), energy derivation (Eng), path loss (PL), and received signal strength, thereby bolstering exploration capabilities for specific node localisation challenges.
- Comparative analyses that underscore the substantial performance improvements realised through the proposed strategies within ACMDE.

The remaining parts of the study include the following sections. Section 2 presents a literature review of the node localisation problem statement, the node localisation in WSN, and reviews the original DE approach. Section 3 introduces the application strategies mechanism to enhance the DE algorithm ACMDE, implement tests to verify the performance in comparison, and analyse the results. Section 4 presents the ACMDE for optimal WSN Node Localisation. Finally, section 5 draws a summary as the conclusion.

2 Literature review

2.1 Node localisation problem statement

The trend-scatter node in deploying WSN is commonly taken place in many industries since less expensive sensor nodes due to the development technology hardware (Nguyen et al., 2016b). Node localisation is one of the fundamental operations and challenges various monitoring or tracking applications because a large area deployed network allocates the acquired location information to unknown devices. A WSN's sensor nodes collect data such as humidity, temperature, and pressure depending on the application target in the region.

The proper localisation of sensor nodes in WSN is necessary when evaluating the quality of WSN applications (Najarro et al., 2022). Accurate node localisation is crucial for enhancing network performance in practical applications like monitoring, military operations, healthcare, agriculture, and astronomy. However, the dynamic movement and coverage connection of sensor nodes make it challenging to pinpoint their precise location. Consequently, various metaheuristics and research algorithms have been developed for WSN node location. Table 1 highlights a few of these works and briefly summarises their features and difficulties.

The elitist genetic algorithm (EGA) (Ren et al., 2020) chose a preservation strategy with an RSSI quantisation based on sensing disks of nodes, for optimal node localisation. The improved differential evolution (IDE) algorithm (Zhang et al., 2023) and the DE (Harikrishnan et al., 2014) are used to optimise the node position in a wireless sensor network. In order to address the drawback of the conventional positioning technique, the distance measurement error was lowered with the modified RSSI by Gaussian that was employed in a network large-ranging positioning accuracy.

The hybridised node localisation model and improved PSO were combined in the local optima issue by the improved particle swarm optimisation (IPSO) (Phoemphon et al., 2020) to enable clear communication between the anchor nodes and unknown nodes in the same group. Although it may be less precise for localising the unknown nodes, it may be more accurate when monitoring the actual positions of the unknown nodes at the convex hull outside of that. More precise localisation results and a reduction in localisation mistakes were provided by hybrid particle swarm optimisation (HPSO) (Lakshmi et al., 2022). However, it struggles to handle complex situations requiring real-world node location. The localisation latency and localisation error were decreased by ABC-BAT (Nithya and Jeyachidra, 2021). However, it did not consider the propagation mistake for future node localisation advancements.

Krill Herd optimisation algorithm (KHA) (Sabbella et al., 2021) reduced the error rate regarding the mean absolute error and root means square error, propagation error, and localisation error. But, it depends on the length of the communication radius to increase the success rate of localisation. Sequential greedy optimisation algorithm (SGO) (Shi et al., 2010) achieved good convergence efficiency and is also appropriate for distributed network optimisation. Yet, it only performs efficiently when the anchors are randomly placed inside the networks. Bio-inspired algorithms (BIA) (Kulkarni and Venayagamoorthy, 2010a) performed faster and more accurate localisation and reduced the sensor node count in deploying terrains without interest. On the other hand, it is not applicable for centralised localisation, which makes it particularly useful regarding energy awareness.

Chicken swarm optimisation (CSO) (Al Shayokh and Shin, 2017) is considered robust and efficient for determining the unknown nodes at a considerable rate of minimum error. However, it secures lower precision on node localisation since it needed to improve their rooster behaviours for making the velocity update properly. Butterfly optimisation algorithm (BOA) (Arora and Singh, 2017) provided effective performance regarding the computation time, localising the nodes, and localisation error. On the other hand, it does not consider the energy problems involved in the WSN and needs to reduce the location estimation error. These challenges in the existing localisation scheme in WSN motivate the development of a new heuristic strategy for localising the unknown nodes in WSN.

A good monitoring and tracking application relies heavily on location accuracy. Existing works have struggled with difficult conditions that require real-world node localisation and improving the accuracy of localising unknown nodes. They also did not consider propagation error for future advancements in node localisation. The execution time of the positioning system in a large-scale ranging network must be considered. The scheme contributed to energy consciousness by avoiding centralised localisation and working well with dispersed anchors. However, these works failed to address energy issues in WSN and reduce location estimation errors. This study aims to develop a new metaheuristic technique or enhance the existing algorithm to properly localise unknown nodes, addressing these issues with the current localisation scheme in WSN.

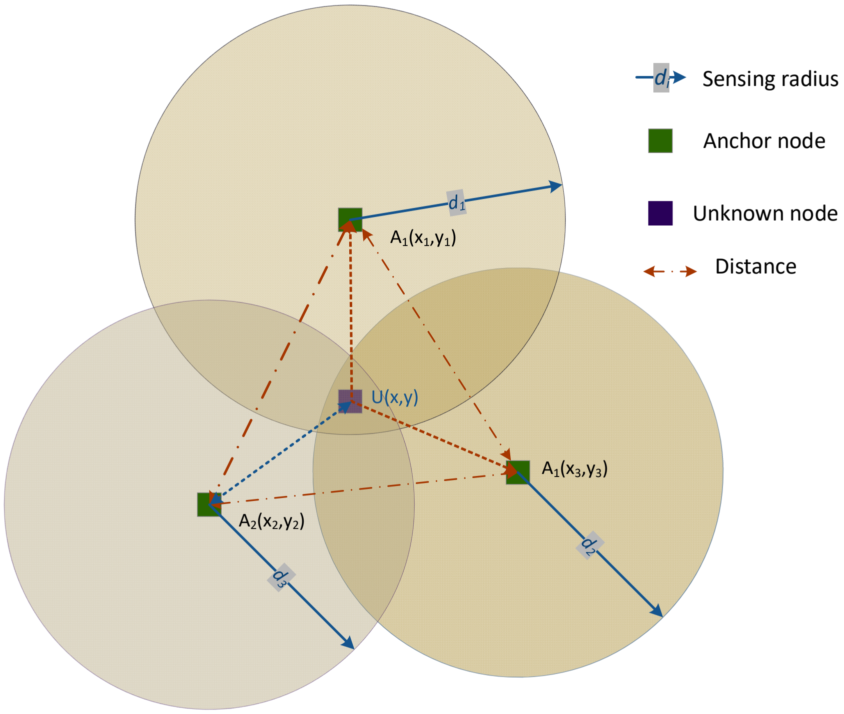
2.2 Node localisation in WSN

The problem of WSN node localisation is finding the correct node positions in a network with many sensor nodes, such as anchors or beacon nodes, unknown nodes or dumb nodes, and settled nodes, where every node has a communication range (Najarro et al., 2022). An anchor node is represented as a start-up node that understands its position in the network using coordinates. Unknown nodes are nodes that are unaware of their location in the network; localisation algorithms can subsequently be used to identify them as free nodes (Nguyen et al., 2017; Nguyen et al., 2020). An anchor node is considered a settled node initially that knows its position with coordinates in the network; afterward, it can somehow manage to determine the status by localisation scheme. Unknown node is represented as nodes unaware of their location in the network. To successfully carry out monitoring or tracking applications, which is a process of selecting or estimating a location known as localisation, the position must be aware of sensor nodes.

Table 1 Several existing WSN node localisation models with their features and challenges

Author [citation]	*Approach*	*Features*	*Challenges*
Ren et al. (2020)	EGA	It was precision still only appropriate due to the fitness function independent units. The overlapping of rings was figured out by calculating the binary code sequence	It quantised RSSI measurements from sensor nodes with irregular appearing areas that reduced the localisation error
Zhang et al. (2023)	IDE	It reduced the distance measurement error that was used in a network large-ranging positioning accuracy to overcome the disadvantage of the traditional positioning algorithm	It suffers from time consumption with a large-ranging network. The modified RSSI by Gaussian in the fitness function
Phoemphon et al. (2020)	IPSO	It does not undergo the local optima problem but ensures communication without obstructions between the anchor nodes and unknown nodes within the same group	It provides less accuracy when observing the actual positions of the unknown nodes at the convex hull outside, making less precision for localising the unknown nodes
Lakshmi et al. (2022)	HPSO	It provides more accurate localisation results and also decreases localisation errors	It suffers from handling certain challenging scenarios that require real-world node localisation
Nithya and Jeyachidra (2021)	ABC-BAT	It reduces the localisation delay and localisation error.	It does not consider the propagation error for further improvements in node localisation
Harikrishnan et al. (2014)	DE	It minimises the node's distances and considers the minimisation of localisation error problems in WSN	It does not consider the delay and propagation error for further improvements in network performance with node localisation
Shi et al. (2010)	SGO	It achieves good convergence efficiency. It is also appropriate for distributed optimisation over the networks	It does not perform efficiently when the anchors are randomly placed inside the networks
Kulkarni et al. (2010a)	BIA	It performs faster and more accurate localisation. It reduces the sensor nodes' count in deploying terrains without interest	It is not applicable for centralised localisation, which makes it particularly useful regarding energy awareness
Al Shayokh and Young (2017)	CSO	It is considered to be robust and efficient for determining the unknown nodes at a considerable rate of minimum error	It secures lower precision on node localisation since it does not improve their rooster behaviours for making the velocity update properly
Arora and Singh (2017)	BOA	It provides effective performance regarding the computation time, localising the nodes, and localisation error	It does not consider the energy problems involved in the WSN and also needs to reduce the location estimation error

Figure 1 A typical calculation in node localisation issue in WSN via anchor nodes to the unknown node (see online version for colours)



As a result, in the WSN setting, location finding presents a significant issue. Either the range stage or the estimating phases are included in the process. The angle of arrival, RSSI, Time of arrival, the former, or distances, is measured between the nodes (Cheung et al., 2004). The estimation stage is then completed by considering the range value and minimising the localisation error.

Figure 1 depicts a typical calculation in node localisation issues in WSN via anchor nodes to the unknown node. It is an expected WSN localisation issue with dashed and solid arrows, respectively, and indicates anchor-to-anchor and anchor-to-unknown node measures. The WSN deployment area are divided into grid cells with the node's communication radius. The adjacent grid cells must guarantee direct communication between two nodes. In order to determine which cell the node would belong to, it is assumed that it knows the location coordination of its neighbour.

Let us assume the WSN is a symmetric type, illustrated as a Euclidean graph $E = (A, B)$. Here, we could assign the vertices A as a set as $A = \{a_1, a_2, \ldots, a_z\}$ and B indicates the edges as $B = \{b_1, b_2, \ldots, b_m\}$ where b is set to coordinates (x, y). Subsequently, the communication is happened by computing the distance between the two nodes. Hence, the two nodes as a_x and a_y, and their estimated distance is $d_{x,y} \leq m$, in which m gives the maximum distance among the nodes. The communication is done only when the distance becomes less than the variable as m. Given $E = (A, B)$ of WSN, a set of anchor nodes with its known position coordinates as (x_i, y_i) for all i $\in I$, where I is a number of anchor nodes (max is M). Further, it aids in estimating the location or position of unknown nodes coordinated as (x_j, y_j) for all $j \in J$, where J a number of unknown nodes (max is N). Thus, influencing the localisation algorithm on the unknown node makes it a settled node by identifying the position as S solution.

The objective function is mainly designed with the fitness approaching value to validate the efficacy of the node localisation approach in WSN. Once the optimal location is determined, it aids in reducing the error factor in locating the sensor nodes. Here, the localisation error is mainly calculated by the distance estimation concerning anchor nodes and sensing ranges of the chosen dumb node and the beacon node. The mathematical expression of the objective function is given formula as follows.

$$ObF = \underset{S_z^*}{argmin}[E_{rr}] \tag{1}$$

where S_z^* is a resultant optimal position as a solution; Err denotes the error measure that is determined by using the following formula.

$$Err = del + \frac{1}{Eng} + \frac{1}{RSS} + PL \tag{2}$$

Here, Err is the formulated functional derivation, del specifies the delay, Eng derives the energy function, and the path loss is denoted as PL, and RSS (received signal strength) is 'the intensity of the acquired signal by the wireless access point. These factors are all closely related to the distance of the coordinates of the anchor and target nodes. The distance between the anchor nodes within the sensing range of the target node with the coordinate known as position (x, y) is used to identify the target node.

2.3 Differential evolution algorithm

The DE algorithm (Price et al., 2005) is a popular stochastic optimisation technique for tackling global optimisation issues. The principles of evolution are imitated, where potential solutions go through random mutation and selection processes. By merging and altering the current solutions, the DE generates a new population of candidate solutions at each generation loop, then picks the best ones to develop the following generation in population. This process of mutation and recombination is carried out by comparing the differences between randomly selected candidate solutions based on fitness values forwarding. The best-performing candidate solutions are chosen for the following iteration after the altered candidate solutions have been evaluated using a fitness function.

It is a productive and successful method of optimisation with several phase stages of the optimising process of the DE algorithm, e.g., initialisation, mutation, crossover, and selection of candidate solutions.

The initialisation population phase is implemented by starting to set the candidate solution as in an optimisation problem, each individual in a population represents a potential solution, and their position information is used to determine the candidate solutions. Before the optimisation process begins, the position of population members must be initialised to ensure even distribution throughout the spread of the D-dimensional optimisation space, which typically corresponds to a D-dimensional area of the target problem space. A random method is often employed to generate the initial population position information distribution. Specifically, the population size is denoted as N, and the initial population position distribution is calculated using the following formula.

$$x_{i,j}(0) = x_{i,j}^{L} + rand(0,1)\left(x_{i,j}^{L} - x_{i,j}^{M}\right) \tag{3}$$

The i^{th} individual in a population indicates a potential answer to an optimisation issue, and the j^{th} decision variable of that individual is denoted by $x_{i,j}(0)$. The i and j range from 1 to NP and 1 to D, respectively. Meanwhile, the function rand (0, 1) creates a random integer with a uniform distribution within the range [0, 1]. This formula is utilised to initialise the position information of the population.

The mutation is carried out with offset changes in population; one of the critical processes for exploring strategy in the DE algorithm called mutation strategy is that it allows the algorithm to explore new regions in the search space. This process creates new candidate solutions by calculating the differences between two selected random individuals from the current population. Common mutation operations include the following.

- Strategy 1: DE/rand/1/bin

$$V_i^{t+1} = x_{r1}^{t} + F*\left(x_r 2^t - x_{r3}^{t}\right) \tag{4}$$

- Strategy 2: DE/rand/2/bin

$$V_i^{t+1} = x_{r1}^{t} + F*\left(x_2^t - x_{r3}^{t}\right) + F*\left(x_{r4}^{t} - x_{r5}^{t}\right) \tag{5}$$

- Strategy 3 DE/best/1/bin

$$V_i^{t+1} = x_{best}^{t} + F*\left(x_{r1}^{t} - x_{r2}^{t}\right) \tag{6}$$

- Strategy 4: DE/best/2/bin

$$V_i^{t+1} = x_{best}^{t} + F_1*\left(x_{r1}^{t} - x_{r2}^{t}\right) + F_2*\left(x_{r3}^{t} - x_{r4}^{t}\right) \tag{7}$$

- Strategy 5: DE/current-to-best/1/bin

$$V_i^{t+1} = x_i^t + F_1*\left(x_b est^t - x_i^t\right) + F_2*\left(x_{r1}^{t} - x_{r2}^{t}\right) \tag{8}$$

In the formula, V_i^{t+1} is the experimental individual I in the $t + 1$ generation population after mutation, $i \in [1, N]$, and the population size is denoted by n; $x_{r1}^t, x_{r2}^t, x_{r3}^t, x_{r4}^t, x_{r5}^t$ are three individuals randomly selected in the T generation population, and r_1, r_2, r_3, r_4 and r_5 represent the identification numbers of different individuals in the same generation population; x_{best}^t denotes the best person in the population's g^{th} generation. F is the variation probability and the value is between 0 and 1.

The crossover population phase is carried out the combining solutions values or information from multiple candidate solutions to create new solutions with higher optimisation potential using a binomial crossover operator. The crossover operation in DE algorithm is expressed as follows.

$$U_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1}, rand(j) \le CR \ or \ j = rand(i) \\ x_{ij}^{t+1}, rand(j) > CR \ or \ j \ne rand(i) \end{cases} \tag{9}$$

In the formula, U_{ij}^{t+1} represents an updated individual obtained by crossing the j gene of the test individual; *rand*(j) is a random integer with a homogeneous distribution, with a number ranging from 0 to 1, and j represents the j^{th} gene; CR is the cross probability, and its value is between 0 and 1; *rand*(i) is the generated random integer, i takes the value in [1, D], and d represents the D-dimensional parameter (number of decision variables); x_{ij}^t represents the individual of t generation population without mutation operation; v_{ij}^{t+1} represents the individuals of t generation population after mutation operation. The value of CR in this paper is 0.9.

The selection process is getting new solutions for the individuals for the next generation. The procedure of obtaining a new individual solution for the subsequent generation is critical for determining which vector, the trial vector or the target vector, will be retained and transferred to the next generation. The individuals following the crossover operation and others in the population are selected using a greedy algorithm. The individuals of the $t + 1$ generation are selected by comparing their fitness to form a new population. Before selection, fitness should be determined for each trial vector using an objective function. The vector with inferior fitness will be eliminated, while the vector with superior fitness will be maintained. The following expression represents the selection operation.

$$x_{ij}^{t+1} = \begin{cases} u_i^{t+1}, f\left(u_i^{t+1}\right) \le f\left(x_i^t\right) \\ x_i^t, f\left(u_i^{t+1}\right) > f\left(x_i^t\right) \end{cases} \quad (10)$$

where $f\left(u_i^{t+1}\right)$ represent that fitness of the test individual through crossover; $f\left(x_i^t\right)$ indicates the fitness of the target individual.

3 Adaptive crossover-mutation differential evolution

This section presents the adaptive strategies in the ACMDE algorithm, consisting of contents, e.g., the chaotic sequence for adapting initialisation, mutation candidate solution for adapting exploration, and crossover for adapting exploitation. Adjusting certain modifying variables is necessary to enhance the optimisation capability of the DE algorithm (Price et al., 2005) and overcome the problem of being susceptible to local optima. The break subsections of adaptive strategies and experimental results are presented as follows.

3.1 Adaptive improvement strategies

Adaptive improvement strategies are carried out by adjusting and modifying parameters and variables to enhance the algorithm's optimisation capability and overcome the trap local optima problem. The key parameters adapting consideration are the initialisation population, mutation factor, and crossover probability. The adaptive strategies for improving algorithm performance are highlighted as follows.

- Chaotic sequence initialisation is one of the efficiencies of most current intelligent optimisation algorithms that is greatly influenced by population initialisation. The uniformly distributed population can appropriately broaden the algorithm's search scope, improving the algorithm's convergence speed and solution accuracy. Using chaos mapping to initialise the population, individuals can be distributed as evenly as possible in the search space. This feature can be used to improve the algorithm's performance. The primary concept is to map variables into the value range of the chaotic variable space using the properties of the variable c is set to constant. Then, convert the result into the ideal variable space linearly. There are currently many chaotic maps in the optimisation field, most notably the Tent, messy, and logistic maps. In this case, we use the chaotic process to create seate starting population. The definition of a chaotic map is as follows.

$$x_{i+1} = mod\left(x_i + 0.2 - \left(\frac{0.5}{2\pi}\right) sin(2\pi x_i), 1 \right), \quad (11)$$

where x_{i+1} and x_i the locations of the current and previous iterations of the individual solutions. When generating the original population, the circular mapping technique produces a more uniform spread of population locations than randomly dispersed populations. The algorithm search space area adapts and broadens the population locations for distributing close target solutions that address the local optima and improve the algorithm's optimisation efficiency.

- Adaptive mutation strategy is carried out with improved 'DE/rand/2/bin'; largely, mutation determines how well DE works. Using an adjusted mutation operator, *F* can increase good convergence performance in the later stages of the algorithm. A new mutation strategy can be proposed by using a dynamic adaptive factor to replace *F* with a new one to solve the problem of insufficient convergence performance in the later stages of the algorithm. The following is the adapting formula.

$$V_i^{t+1} = x_{r_5}^t + \gamma * \left(x_{r_1}^t - x_{r_2}^t\right) + F * \left(x_{r_3}^t - x_{r_4}^t\right), \quad (12)$$

$$\gamma = (\gamma \max - \gamma \min) * ((T - t) / T) + \gamma \min; \quad (13)$$

where *T* represents the maximum iteration count, and *t* represents the current iteration count; γ is an adaptive factor considered as dynamic mutation. The variation factors have upper and lower bounds, denoted by γmax and γmin respectively; $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$.

During the initial phase of evolution, the population explores a wider range of possibilities to discover the optimal solution, and a high value of *F* is preferable at this stage. As the evolution process advances, *F* should be gradually reduced to improve the population's ability to conduct accurate and focused local searches.

- Adaptive crossover parameter as the operator is carried out with the value in the crossover operator that determines the proportion of individual information from parents in the new individual. A significant value favours mutants and improves convergence speed, while a small value favours parents and enhances global optimisation. However, the standard DE algorithm uses a fixed value, which neglects the trade-off between global and local search. Therefore, an adaptive monotone-decreasing crossover operator is introduced to address this issue, and the formula is calculated as follows.

$$CR = Cr * \exp(-2 * (t / T)), \quad (14)$$

where *t* and *T* represent, the present stage of the iteration process and the maximum iteration, *Cr* is a crossover parameter that is set to 0.9. When the algorithm begins to operate, this operator takes a more considerable value, which can improve the algorithm's early convergence speed and decrease the value of the operator. This promotes the algorithm's global optimisation and lessens the likelihood of focusing on local optimisation later. Algorithm 1 shows the ACMDE pseudocode.

Algorithm 1 ACMDE pseudo-code

Input: population size: N_P, dimension: D, the Max_iter T, and boundaries: ub, lb.

Output: The best object for the entire population.

```
Initialisation: Initiate the population using circle
mapping P^G.G=1 and evaluate the fitness of each
individual in the initial population.
While t < T do
    For i = 1:Np do
    (1)Mutation operator
    Random selection:
        r1≠r2≠r3≠r4≠r5;
    γ = (γmax–γmin)*((T–t)/T)+γmin;
    Vi(g) = Xr5 (g) + γ* (Xr1 (g)–Xr2 (g)) + F*(Xr3(g)–
    Xr4(g))
    (2)Crossover operator
     jrand = rndint [1, D]
     For j = 1 to D do
        CR = Cr*exp(–2*(t/T));
     If rand[0, 1] < CR or j = jrand
        Ui (g) = Vi(g)
     Else
        Ui (g) = Xi (g)
     (3)Selection operator
        If f(Ui (g)) < f(Xi (g))) then Xi (g) = Ui (g)
        End if
     End for
       g = g + 1
     End While
Output: The best object for the entire population
```

3.2 *Experimental results on mathematic test functions*

By comparing it with the chosen popular algorithms, this subsection assesses the qualifying performance of the suggested ACMDE algorithm. 28 benchmark functions in CEC2013 are used to evaluate the ACMDE performance effectiveness quality of the improvement algorithm. There are several modal types of testing functions in the test suit, e.g., unimodal, multimodal, hybrid, and compound functions. The serial number functions are named CEC01 to CEC28, including unimodal (CEC01~CEC05), multimodal (CEC06~CEC015), hybrid (CEC16 ~ CEC21), and compound (CEC21 ~ CEC28) that are various complexity and dimensions settings in the selected functions in the test suite.

The test sets of comparison with the original optimisation algorithms are implemented by uniformly setting the same population size, number of iterations, and dimension, e.g., 50, and 100D. The obtained results of the algorithms for the test are the global optimum presented in the form of tables and figures. The achieved optimal results of the ACMDE compared with the variant improved DE types of adjusted parameters and with the other algorithms in the literature. The set of improved defferential evolution types included the EDE (Gong et al., 2011), JADE (Zhang and Sanderson, 2009), SaDE (Qin and Suganthan, 2005), and original DE (Price et al., 2005) algorithms and the set of the other algorithms are such as ALO (Mirjalili, 2015a), GWO, MFO (Mirjalili, 2015b), and PSO (Kennedy and Eberhart, 1948) algorithms for the test function with different dimensions. Table 2 lists the selected popular algorithms' parameter settings for the benchmark testing functions.

Table 2 Parameters settings for the algorithms

Algorithm	*Parameters settings*
ACMDE	a = 2 to 0, b = 1, l = [–1, 1], u, v, $g \in$ [0, 1], β = 1.5, NP = 60, Cr = 0.6 F = 0.6, Maxiter = 1000
DE (Price et al., 2005)	a = 2 to 0, b = 1, p ∈ [0, 1], NP = 60, Cr = 0.6 F = 0.6, Maxiter = 1000
EDE (Gong et al., 2011)	a = 2 to 0, b = 1, $p \in$ [0,1], NP = 60, Cr = 0.6 F = 0.6, Maxiter = 1000
JaDE (Zhang and Sanderson, 2009)	a = 2 to 0, b = 1, $p \in$ [0, 1], NP = 60, Cr = 0.6 F = 0.6, Maxiter = 1000
SaDE (Qin and Suganthan, 2005)	$\beta \in$ [0, 2], a = 2 to 0, b = 1, p ∈ [0,1], μ, v ∈ [0, 1], NP = 60, Cr = 0.6 F = 0.6, Maxiter = 1000
ALO (Mirjalili, 2015a)	$\omega \in$ [3 to 6], r = 1 or 0, NP = 60, Maxiter = 1000
PSO (Kennedy and Eberhart, 1948)	V_{max} = 10, V_{min} = –10, $\omega \in$ [0.9, 0.4], $c_1 = c_2$ = 1.489, NP=60, Maxiter=1000
MFO (Mirjalili, 2015b)	a = –1, b = 1, NP = 60, Maxiter = 1000
GWO (Mirjalili et al., 2014)	$\alpha \in$ [0, 2], $C \in$ [0, 2], r_1, r_2, $r_3 \in$ [0, 1], NP = 60, Maxiter = 1000

Tables A1 and A2 (in the index section) show the obtained optimal results of the ACMDE against the EDE, JADE, SaDE, and DE algorithms for the test function with 50 and 100D performances, respectively. The qualified performance of the suggested ACMDE algorithm is analysed by comparing it with the selected popular improved DE algorithms. Table A3 shows the achieved optimal results of the ACMDE against the ALO, GWO, MFO, and PSO algorithms for the test function on 100D performance.

Table A3 shows the other set of comparisons with the other algorithms in the literature that achieved optimal results of the ACMDE against the ALO, GWO, MFO, and PSO algorithms for the test function on 100D performance. The table's contents include two kinds of values of the best optimal 'BEST' and means of the average 'MEAN' of the best ones of 25 runs. The summarised statistics are set in the last of the table, e.g., '*win*,' '*lose*,' and '*draw*,' which means the suggested algorithm is better, less, or similar, respectively. The highlighted values are the best in comparing the tables' row.

The compared algorithms among the unimodal functions, except CEC4, the other seven functions achieved the best results, and the best values were found in CEC1 and CEC5. Among the 15 basic multimodal functions, the ACMDE has achieved the best results in nine test functions: CEC6, 8, 10, 11, and CEC16~20, which shows that the ACMDE can effectively jump out of local optima. Among the eight multimodal functions, CEC21, 24, 25, 27, and 28 perform well, showing that the ACMDE can apply to multimodal functions and complex optimisation problems. The tests also show that the ACMDE performs well in different dimensions, indicating that the ACMDE has good adaptability. It is seen that the number of wins belongs to the ACMDE algorithm.

The algorithm's convergence rate for the chosen functions, such as CEC1, 3, 4, 5, CEC10 through CEC19, CEC21, and CEC22, is shown in Figure A1. Because the CEC1 and CEC5 unimodal functions are simple, the ACMDE can find the optimum value quickly. After leaving the local optimal, the ACMDE has the highest convergence accuracy and the fastest convergence speed. In the most complex composition functions, the convergence curve of the approach is superior to that of the comparison algorithm, demonstrating the ACMDE algorithm's excellent global optimisation capability over other comparison methods. Specific algorithms slow down and enter local optimisation in their later stages of evolution. The statistical outcome demonstrates that the ACMDE has more 'Win' numbers than the others, indicating that the ACMDE performs exceptionally well.

4 ACMDE for optimisation WSN node localisation

This section presents the construction of optimal node localisation using the ACMDE technique in WSN. The progressive presentation subsections are specified following sequential descriptions, such as the description scheme, model objective function, established ACMDE scheme for optimal nodes' location, system parameter setting, and presentation of experimental results.

4.1 System schema description

In constructing a goal node localisation with the ACMDE technique in WSN, progressive presentation subsections are specified following sequential descriptions. By resolving the goal functional derivation concerning variables such as delay, PL, energy, and RSS for optimising the locations of the nodes based on the anchors to reach the target nodes in the sensor field simulation, the objective function frame as the effective localisation strategy in WSN would be stated with directing calculated fitness values. The optimal node localisation is the freshly established ACMDE technique scheme that optimises nodes' location for anchor nodes toward the target nodes. For a higher rate of agreement and statistics to determine the placement of the nodes' position based on the anchor nodes, update the candidate solution and get the value of the objective function in optimisation using the best candidates for the ACMDE solutions. In order to carry out the node localisation process, the findings employ simulation using scenarios of various WSN networks, including network area, number of target nodes, anchor nodes, and sensing range. The WSN solution with minor fitness is found to be the best localisation solution in terms of anchor node placement coordinates. The resolution achieves more effective localisation performance and a faster convergence rate than any other choice.

4.2 Modelling objective function frame

The node localisation strategy to reach the target nodes in the sensor field using a suggested ACMDE algorithm for minimising localisation errors with the optimised node positions with estimations based on anchor node provided with constraints limited resource of the solution element. As a result of resolving the objective functional derivation concerning variables, including latency, *PL*, energy, and *RSS*, are recognised as the effective localisation strategy in WSN. The designed node localisation model for WSN is derived in the following manner with the expression function *ObF* referred to in equation (1) mainly designed with the fitness approaching value to validate the efficacy of the suggested node localisation approach over deploying WSN as follows.

$$ObF = \underset{\{AN_{(j)}\}}{\arg min}\left(\frac{1}{Eng} + \frac{1}{RSS} + PL + del\right), \tag{15}$$

Here, *Eng* derives the energy function, del specifies the delay, and the *PL* is denoted as *PL*, *RSS* is 'the intensity of the acquired signal by the wireless access point', and $AN_{(j)}$ is a number of the anchor nodes.

The energy function of anchor nodes is estimated as derived as follows.

$$Eng = \left(dis^2 \cdot \rho \cdot F_s l + \left(TX_{eng} + AD_{eng}\right) \cdot \rho\right), \tag{16}$$

Here, F_{sl} and ρ are coefficients loss for free space derived of anchor nodes; TX_{eng} and AD_{eng} are the transmission energy and the acquired energy, respectively. Using a new ACMDE method on an optimised anchor node in an environment with constrained resources, the node localisation technique aims to reach the target nodes in the sensor field. Once the optimal location is established, energy-related technology is utilised to locate the sensor nodes, which contributes to reducing the error factors associated with distance estimates. Therefore, the ACMDE algorithm operates the position optimisation. The localisation error in the present scenario is determined mainly by estimating the distances between the anchor nodes and the sensing ranges of the selected dumb node and the beacon node in the WSN. Euclidean distance has specified the coordinate known as positions used to identify the target node ($rt_{(j)}$, $st_{(j)}$, and the location of the anchor node is referred to as $r_{(j)}$–$s_{(j)}$.

$$dis_{(j)} = \sqrt{\left(rt_{(j)} - r_{(j)}\right)^2 + \left(st_{(j)} - s_{(j)}\right)^2}, \tag{17}$$

Here, dis_j is the distance between the anchor nodes and target nodes. The recommended node localisation strategy gets the node information via the position of anchor nodes in the direction of the optimal target nodes or unknown nodes by applying the ACMDE algorithm, where the nearest target nodes are reached by minimising node localisation errors. The anchor nodes are specified as $AN_{(j)}$, the number of anchor nodes is considered as O, and $j = 1, 2, \ldots, O$ and the target nodes are represented as $TN(t)$, where $t = 1, 2, \ldots, T$, the in which a number of target nodes T.

The derivation concerning the variable of the RSS is given as follows.

$$RSS = \frac{RP}{NP}, \tag{18}$$

Here, the term NP specifies the noise power, and RP illustrates the receiving power that describes the RSS of anchor nodes; that is, in the experimental setting, it is set to –91 to –35 dBm. The path loss PL variable is modelled by formulating the normal log for defining the communication range of nodes.

$$PL = LN(Range_c), \tag{19}$$

where, $Range_c$ is the communication range of a node, in which the pass loss exponent is used for determination and setting a $Range_c = 10^{\left(\frac{pw_{tr} - pw_{\tau}}{10\mu}\right)}$, where the PL factor is denoted as μ, $Pw\tau$ derives the least threshold receiver power attained by nodes, and the transmitted power is derived as $Pwtr$. In the experimental section, it is set to 10 to 30 m. Localisation delay is known as del, which is defined as the time variation while broadcasting a request message by a sensor node and when it achieves its place as formulated in equation (20).

$$del = RM_{(time)} - LN_{(time)}, \tag{20}$$

where, $LN_{(time)}$, and $RM_{(time)}$ are the time while the position coordinates are achieved for a node and the time while a request message is broadcast, respectively. The best localisation solution in WSN in terms of location coordinates of anchor nodes is found via optimising objective functional derivation for minimum fitness solution for expecting taken as the optimal solutions that reach higher localisation performance with better convergence rate.

4.3 Node localisation schematic in WSN

As previously pointed out, the sensor nodes in WSN are used to collect information such as humidity, temperature, and pressure that depend on the application purpose, which is the place to be collected about WSN for the node localisation scheme since sensor nodes are less expensive. The WSN deployment zone must be separated into several virtual grid cells based on the node's coordination and communication radius. The grid cells next to one other must ensure direct contact between the nodes. It is suggested that the node is aware of its neighbour's location coordination to decide which cell it belongs to. This means that the mesh is surrounded by three rings covered, and multiple rings cover the actual location of the grid as equal to 3r (where r is the ring radius, as a grid unit length). As a result, the more covered grids there are, the more likely it is that there will be unknown nodes in the area.

The area where three subrings intersect is considered, which means that the mesh is surrounded by three rings covered with establishing a boundary condition for the optimisation algorithm constraint that must be set to regulate the forward updated solution the core concept is underlying creating an ideal model. It ensures that any two nodes in nearby cells can communicate directly with one another, eliminating the need for noise-cancelling device terminals and guaranteeing that expressing cell radius requirements are met as follows.

$$(3r)^2 \tau_i + (3r)^2 \tau_{i+1} R^2, \tag{21}$$

where R and r are the communication radius and the grid unit length; This is a set to grid unit length is met a condition. The formula can be drawn from the expression rewritten as follows.

$$r \le \frac{R}{3\sqrt{2}}, \tag{22}$$

where τ_i is an effective noise coefficient to node i, generally set to 1. Considering these constraints, the metaheuristic optimisation algorithms, e.g., BIA, swarm intelligence, and genetic-based heuristic approach, are applied for node localisation and formulated the equations for reducing the localisation error among the nodes in WSN. Over the iteration, the algorithm is deployed to find the position of unknown nodes that continues till the dumb nodes become settled nodes.

The suggested ACMDE algorithm is applied to WSN's node localisation scheme scenario to improve locating accuracy over the system node localisation model. Algorithm 2 displays an ACMDE – node localisation pseudo-code for optimisation localisation errors.

Algorithm 2 An ACMDE- Node localisation pseudo-code

Input: N_P: population size, D: dimension, the Max_iter T, and a built objective function ObF: equation (15) subject to its constraints equations (16) to (22)

Output: The best node localisation for the WNS deployment.

```
Initialisation: Initiate the pop using circle mapping P^G, evaluate the objective function ObF of each individual in the initial population, and set g to1
While t < T do
    For i = 1: N_p do
    (1)Mutation operator
      Random generating: r1≠r2≠;
    r3≠r4≠r5
```

Update dynamic mutation adaptive factor equation (12)
Update velocity vector equation (13)
(2)Crossover operator
$j_{rand} = rndint[1, D]$
For $j = 1$ to D **do**
Update a crossover adaptive factor equation (14);
If rand[0, 1] < CR or $j = j_{rand}$ **then**
$U_i(g) = V_i(g)$
Else
$U_i(g) = X_i(g)$
End if
$ObF(U_i(g))$
(3)Selection operator
If $ObF(U_i(g)) < ObF(X_i(g))$ **then**
$X_i(g) = U_i(g)$
End if
$ObF(X_i(g))$
End for
$g = g + 1$
End while
Output: The best node localisation $X_i(g)$ for the WNS in terms of minimised localisation errors.

The main goal of the proposed model is to address the node localisation problem within the WSN deployment and to construct the node localisation objective function based on the process optimisation of the ACMDE with distance computation to reduce localisation error. The unique method can reduce the localisation error by measuring the distance and range between nodes. The suggested approach ensures effective localisation performance by achieving fewer mistakes. The simulation findings are verified and contrasted with those of other current heuristic optimisations discussed in the following subsection.

4.4 Experimental setup

The obtained results of the node localisation framework in WSN and simulation setup from the proposed ACMDE are analysed to evaluate performance. Based on convergence analysis and statistical analysis, the performance of the suggested model is compared with several previous schemes in the literature with model condiction. Here, the total iteration count is set at 1,000, the number of populations is taken to be 40, and the number of dimensions is set to the number of anchors and unknown sensor nodes, along with the node's (x, y) coordinates.

Table 3 lists considered parameter settings simulating schemes. The range-free WSN localisation solution uses a directional antenna and four directional antennas connected by nodes. A straightforward processing method is used to find the sensor nodes and the coordinates that pick up the strobe messages, mitigating the impact of changes in the WSN nodes' transmission range on sensor node communication. Most routing procedures require probe messages to obtain neighbour information to communicate node parameters (such as energy, memory, and node id) and understand node states. Using the control packet as a piggyback, networks use the communication protocol to transmit signals and query messages, update the neighbour node on the node's state, or query a few neighbour nodes. Additionally, broadcast, unicast, multicast, and anycast are frequently used in communication networks. Due to the 'always broadcast nature' of messaging in wireless communication, broadcasting or beacon messages are prohibited until necessary.

Table 3 An experimental parameter setting

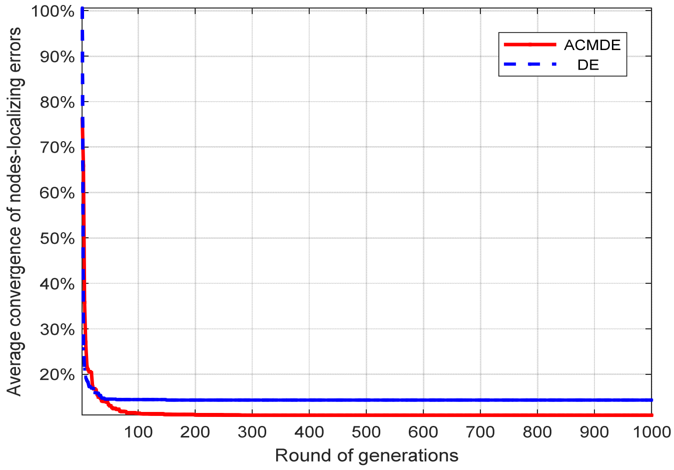
Description	*Parameter settings*	*Value settings*
Simulation area of the network of deployment	$W{\bullet}L$	100 m × 100 m, 150 m × 150 m
Initial node *i* energy	N_i	0.5 J
Transmission energy	TX_{Eng}	0.00000000 001J
Acquired energy	AD_{eng}	0.00000000 5J
Coefficients loss for free space	F_{sl} and ρ	0.00000000 001J, and 4,000
Anchor noise and receiving power	NP and RP	–95 and –35 dBm
Sensing radius ranges with directional antenna	R_s	20 m, 20 m, 30 m, 30 m
Number of anchor nodes	M	15, 20, 30, 35
Communication radius, with directional antenna	Rc	10 m, 10m, 12m, 20 m
Number of unknown sensor nodes	N	25, 40, 60, 80
The number of iterations – no. of rounds	Round iteration	500, 1000

4.5 Experimental results

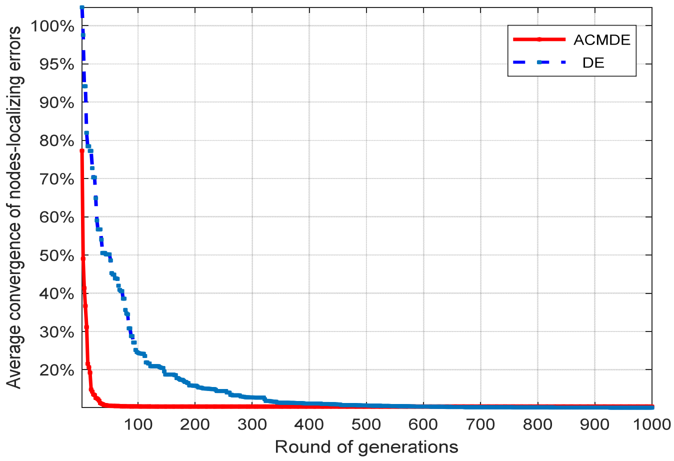
Figure 2 compares the obtained convergence by ACMDE with the DE (Harikrishnan et al., 2014) for the objective function as designed fitness localisation with 30/60 anchor/target nodes for two cases in areas 60 × 60 and 100 × 100 m^2, respectively. It can be seen that the ACMDE produces convergence faster than the DE in the same condition simulation settings.

Moreover, the obtained results of the suggested ACMDE method are compared with the previous scheme algorithms, including EGA (Ren et al., 2020), IPSO (Phoemphon et al., 2020), KHA (Sabbella et al., 2021), CSO (Al Shayokh and Shin, 2017), and IDE (Gou et al., 2021) algorithms. Experimental parameter settings are initialised for the scheme simulating in the compared fair of the algorithm of the WSN for node localisation.

Figure 2 Comparison of the obtained convergence by the ACMDE with the original DE for fitness localisation with 30/60 and 35/80 anchor/unknown nodes, (a) area of 80 × 80 m^2 with 30/60 anchor/target nodes, 100 × 100 m^2, respectively (b) area of 150 × 150 m^2 with 35/80 anchor/target nodes (see online version for colours)



(a)



(b)

Table 4 Statistical evaluation of the ACMDE approach for the node localisation scheme in WSN over classical optimisations

Metrics	*Competing approaches in comparison*						
	DE (Harikrishnan et al., 2014)	*EGA (Ren et al., 2020)*	*IPSO (Phoemphon et al., 2020)*	*KHA (Sabbella et al., 2021)*	*SCO (Al Shayokh and Shin, 2017)*	*IDE (Zhang et al., 2023)*	*ACMDE*
Best score	$4.48 \times 10^{+00}$	$4.48 \times 10^{+00}$	$4.81 \times 10^{+00}$	$4.76 \times 10^{+00}$	$4.99 \times 10^{+00}$	$4.90 \times 10^{+00}$	$4.01 \times 10^{+00}$
Worst score	$2.70 \times 10^{+01}$	$2.70 \times 10^{+01}$	$2.70 \times 10^{+01}$	$3.31 \times 10^{+01}$	$3.19 \times 10^{+01}$	$2.31 \times 10^{+01}$	$2.11 \times 10^{+01}$
Mean	$7.70 \times 10^{+00}$	$7.70 \times 10^{+00}$	$7.62 \times 10^{+00}$	$7.70 \times 10^{+00}$	$7.62 \times 10^{+00}$	$7.25 \times 10^{+00}$	$7.88 \times 10^{+00}$
Std. deviation	$6.98 \times 10^{+00}$	$6.98 \times 10^{+00}$	$6.87 \times 10^{+00}$	$6.91 \times 10^{+00}$	$7.38 \times 10^{+00}$	$8.57 \times 10^{+00}$	$6.54 \times 10^{+00}$
Time (s×10c)	$5.84 \times 10^{+00}$	$5.84 \times 10^{+00}$	$4.49 \times 10^{+00}$	$6.37 \times 10^{+00}$	$5.94 \times 10^{+00}$	$6.98 \times 10^{+00}$	$5.45 \times 10^{+00}$

Table 4 compares results obtained from the proposed ACMDE with the other methods: DE (Harikrishnan et al., 2014), EGA (Ren et al., 2020), IPSO (Phoemphon et al., 2020), KHA (Sabbella et al., 2021), SCO (Al Shayokh and Shin, 2017), IDE (Zhang et al., 2023) algorithms, in situations of rate percentage of coverage, executing times, round iterations to convergence reaching, and sensor nodes for monitoring region sizes.

The performance of the routing protocol is also impacted by and dependent on the deployment of WSN applications. Because the sensor nodes are dispersed at random, an ad hoc infrastructure is produced. To enable connection and energy-efficient network operation, optimum clustering is required if the resulting node distribution is not uniform. Inter-sensor communication typically takes place within small transmission ranges due to energy and bandwidth restrictions. For selecting a routing method suitable for the scheme of WSN localisation, it is possible that a route would have several wireless hops to meet this need. In this work, the Span (Chen et al., 2002) method is chosen as some nodes as coordinators based on their placements since it is the energy-efficient coordination mechanism for topology maintenance in ad hoc WSN. In the distributed, randomised method Span, nodes locally decide whether to go to sleep or to become a coordinator in a forwarding backbone. Each node bases its choice on estimating the number of neighbours that will profit from being awake and the energy supply.

4.6 Analysis and discussion results

Several metrics over iterations represent analysis for the node localisation scheme based on the ACMDE with the objective function, e.g., the best, worst, mean, standard deviation score values, and computation time of different optimisation approaches. A statistical evaluation of the proposed ACMDE for the node localisation scheme in WSN over classical optimisations. The ACMDE algorithm attains better quality performance in contrast with conventional algorithms such as DE EGA, IPSO, KHA, CSO, and IDE approaches.

Figure 3 shows the optimal graphical demonstration of the ACMDE for some node localisation under situations of the number of unknown and anchor nodes in the same deployment of a 100 × 100 m area, e.g., anchor/unknown nodes: 15/25, 20/35. 30/60, and 35/80 respectively.

Figure 4 shows the convergence analysis of the proposed node localisation scheme in WSN compared against various optimisations, e.g., De, EGA, IPSO, KHA, SCO, and IDE methods. Several scenarios are carried out in this comparison convergence analysis of the ACMDE scheme with previous algorithms for localisation errors in different deployment network ranges and different rates of distribution density, e.g., (a) deployed 60 × 60 m^2 area: rate 15/25 nodes, (b) deployed 80 × 80 m^2 area: rate 20/35 nodes, (c) deployed 100 × 100 m^2 area: rate 30/60 nodes, and (d) deployed 150 × 150 m^2 area: rate 35/80 nodes. Over the iteration, the objective function is gradually decreased. It implies that it has a propensity to achieve higher convergence rates. The improved model successfully establishes the unknown node's location in the WSN. The proposed node localisation approach's convergence evaluation over particular optimisations is shown. The superior, in the majority of circumstances, use the ACMDE technique. As a result, the convergence rate to locate the sensor nodes in WSN tends to be significantly improved by the lower value convergence.

The localisation error analysis of localisation errors of the proposed method compared with traditional algorithms concerning the variation of anchor nodes and sensor ranges. Figure 7 shows the localisation error analysis of the ACMDE scheme compared against various algorithms for different scenarios of areas network deployment, e.g.,

a 60 × 60 m^2

b 80 × 80 m^2

c 100 × 100 m^2

d 150 × 150 m^2 setting respectively.

In most cases of setting net area deployments, the localisation error analysis of the proposed ACMDE scheme is smaller than the other schemes' optimisations. In the error analysis with net deploying ranges of Figure 5, the ACMDE algorithm obtained the error output as less when compared to percentages of the other methods. Similarly, Figure 5(b), 5(c), and 5(d) represent the localisation error analysis of the proposed scheme with varying unknown nodes. The error value achieved by the suggested ACMDE algorithm is less localisation error than the others in comparison as acquired to improve the localisation performance over WSN.

Figure 3 The optimal graphical demonstration of the ACMDE for some node localisation under situations of the number of unknown and anchor nodes in the same deployment of a 100 × 100 m^2 area, (a) anchor/unknown nodes: 15/25 (b) anchor/unknown nodes: 20/40 (c) anchor/unknown nodes: 30/60 (d) anchor/unknown nodes: 35/80 (see online version for colours)

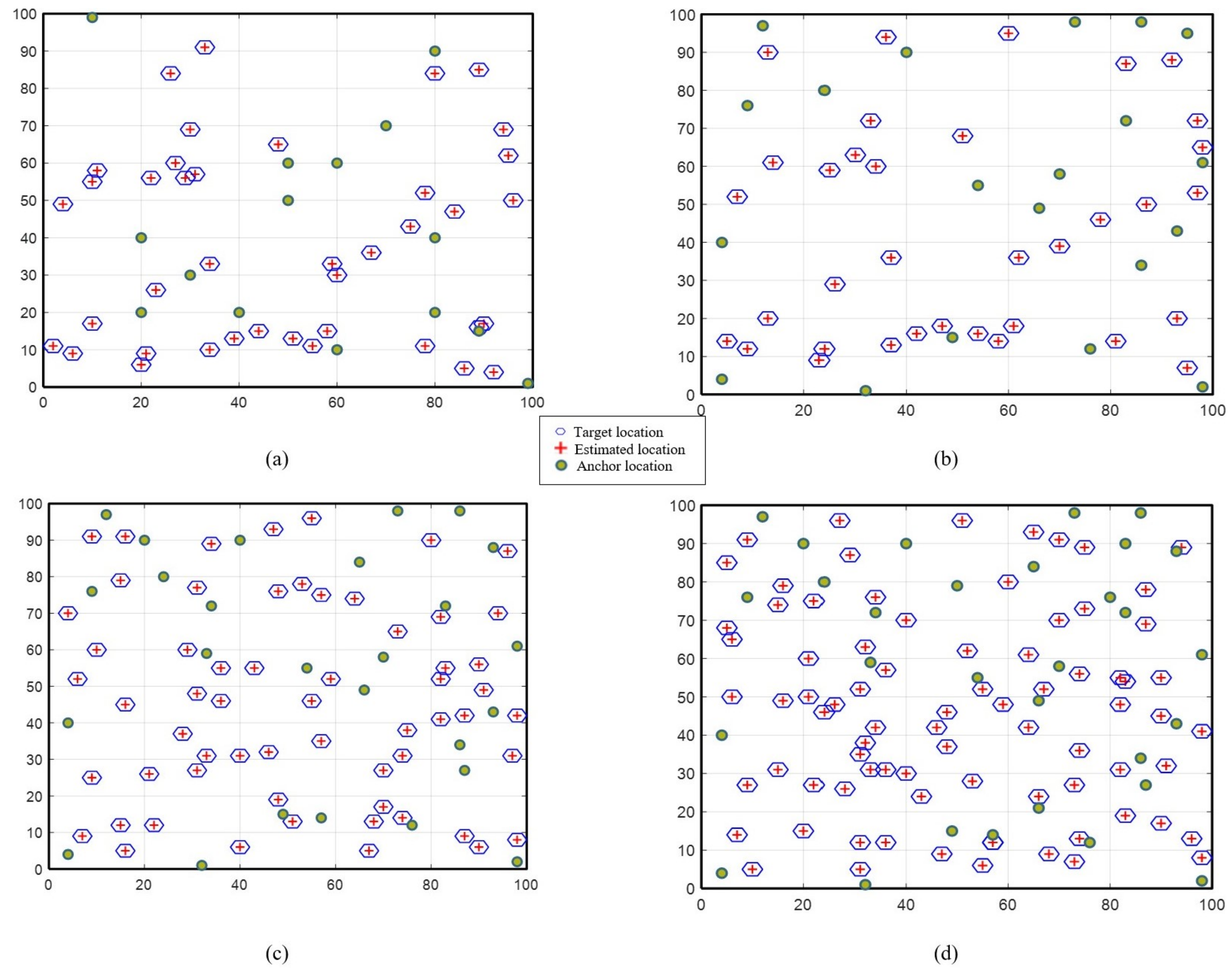


Figure 4 Convergence analysis of proposed node localisation scheme in WSN compared against various optimisations, (a) deployed 60 × 60 m^2 area: rate 15/25 nodes (b) deployed 80 × 80 m^2 area: rate 20/40 nodes (c) deployed 100 × 100 m^2 area: rate 30/60 nodes (d) deployed 150 × 150 m^2 area: rate 35/80 nodes (see online version for colours)

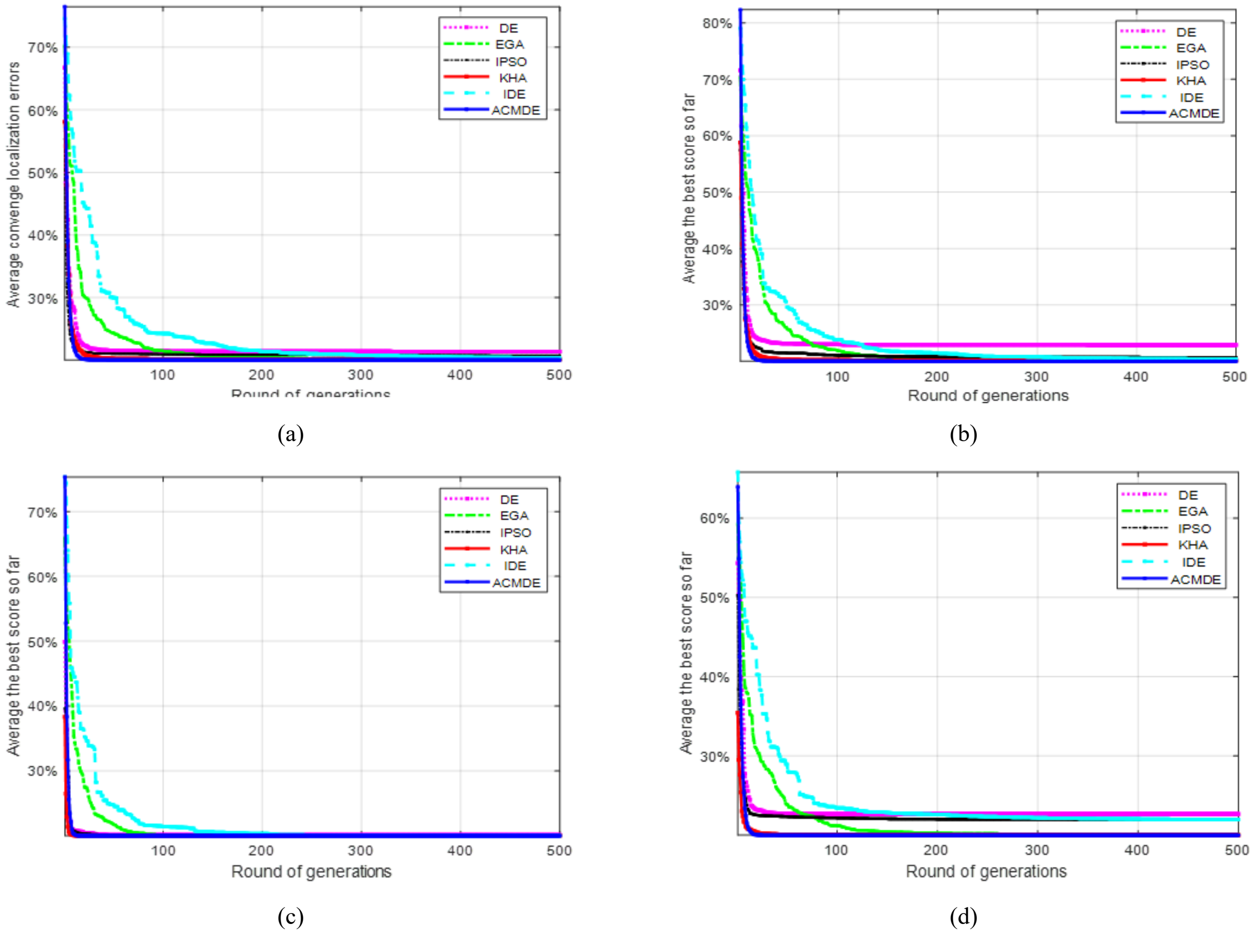


Figure 5 Localisation error analysis of the ACMDE node localisation scheme compared against various algorithms for different scenarios of areas network deployment, e.g., (a) 60 × 60 m^2 (b) 80 × 80 m^2 (c) 100 × 100 m^2 (d) 150 × 150 m^2 setting respectively (see online version for colours)

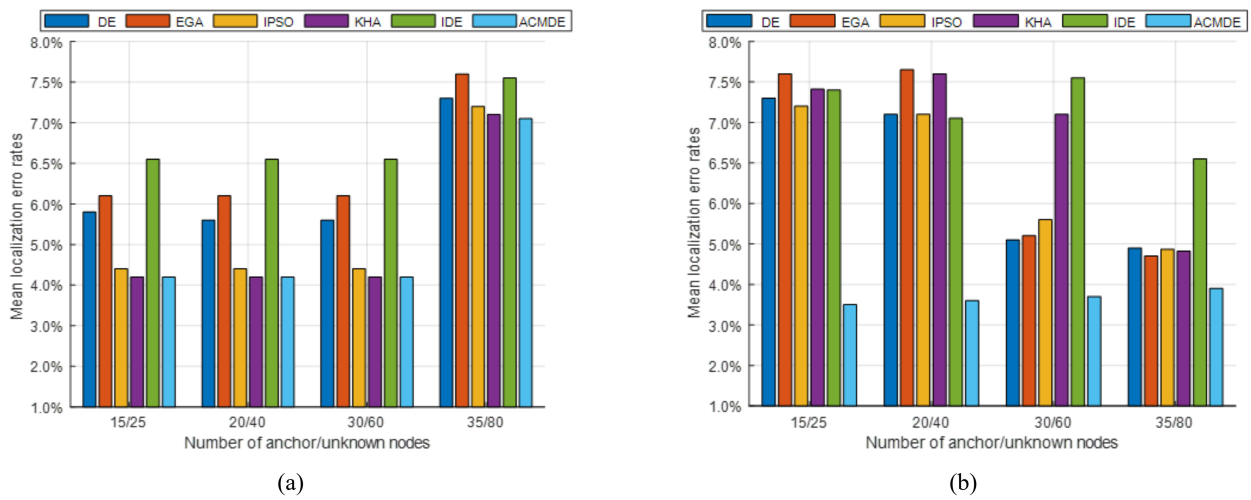
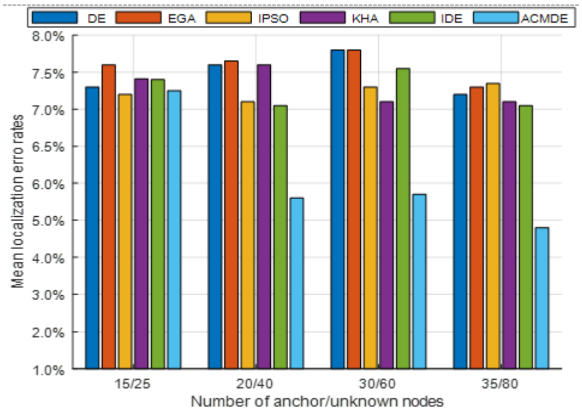
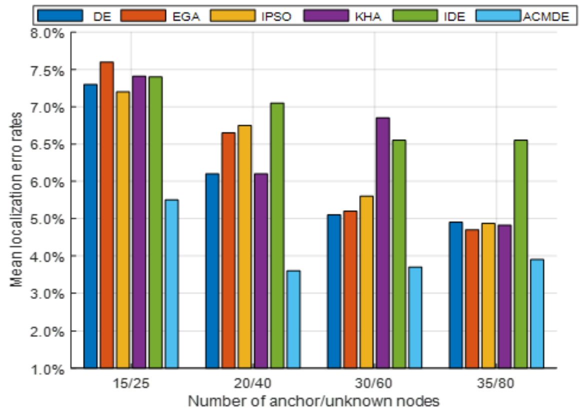


Figure 5 Localisation error analysis of the ACMDE node localisation scheme compared against various algorithms for different scenarios of areas network deployment, e.g., (a) 60 × 60 m² (b) 80 × 80 m² (c) 100 × 100 m² (d) 150 × 150 m² setting respectively (continued) (see online version for colours)



(c)



(d)

Table 5 Comparison of results obtained from the proposed ACMDE scheme with the other schemes: the EGA, IPSO, KHA, SCO, and IDE algorithms, in situations of rate percentage of coverage, executing times, round iterations to convergence reaching, and sensor nodes for monitoring region sizes

Approach	*Factor variables*	*60 m × 60 m*	*80 m × 80 m*	*100 m × 100 m*	*150 m × 150 m*
EGA (Ren et al., 2020)	Localisation errors	7.8%	9.0%	7.9%	7.1%
	Time execution (s)	2.26 × 10+00	5.84 10+00	8.01 E+00	7.25 E+00
	Round iterations for convergence reaching	354	459	554	735
	Average optimal converges	3.42 E+00	6.25 E+00	8.57E+00	7.76 E+00
IPSO (Phoemphon et al., 2020)	Localisation errors	8.7%	7.9%	7.9%	9.2%
	Time execution (s)	2.56 E+00	4.48 E+00	7.36 E+00	8.89 E+00
	Round iterations for convergence reaching	145	458	336	781
	Average optimal converges	4.08 E+00	7.48 E+00	1.03 E+01	1.18 E+01
KHA (Sabbella et al., 2021)	Localisation errors	7.6%	8.0%	1.9%	9.3%
	Time execution (s)	2.98 E+00	6.32 E+00	8.35 E+00	8.45 E+00
	Round iterations for convergence reaching	379	485	468	719
	Average optimal converges	3.19 E+00	6.76 E+00	8.93 E+00	9.04 E+00
CSO (Al Shayokh and Shin, 2017)	Localisation errors	7.7%	7.9%	2.0%	9.2%
	Time execution (s)	3.32 E+00	5.94 E+00	7.13 E+00	8.19 E+00
	Round iterations for convergence reaching	445	555	665	776
	Average optimal converges	4.18 E+00	1.06 E+01	1.39 E+01	1.41 E+01
DE (Harikrishnan et al., 2014)	Localisation errors	7.8%	7.9%	8.9%	9.1%
	Time execution (s)	2.92 E+00	6.98 E+00	7.40 E+00	8.24 E+00
	Round iterations for convergence reaching	665	473	595	824
	Average optimal converges	4.23 E+00	1.06 E+01	2.39 E+01	2.41 E+01
IDE (Zhang et al., 2023)	Localisation errors	7.8%	7.9%	8.9%	9.1%
	Time execution (s)	2.92 E+00	6.98 E+00	7.40 E+00	8.24 E+00
	Round iterations for convergence reaching	665	473	595	824
	Average optimal converges	4.23 E+00	1.06 E+01	2.39 E+01	2.41 E+01
ACMDE	Localisation errors	6.7%	6.9%	6.8%	7.0%
	Time execution (s)	2.81 E+00	5.45 E+00	7.01 E+00	7.19 E+00
	Round iterations for convergence reaching	231	463	556	765
	Average optimal converges	3.01E+00	5.83 E+00	7.50 E+00	7.69 E+00

By carrying out calculations and information transfer in a wireless context, sensor nodes can be consumed their limited energy supply without affecting accuracy. Utilising energy-efficient communication and analytical techniques is considered essential in the deployment period. Each sensor node, which functions as a router and a data emitter, has a limited battery life. Some of its sensor nodes break down when the power goes out or runs out, which may have a significant topological impact and necessitate packet rerouting and network reconstruction. Network efficiency is significantly impacted by the energy used during the receiving, demodulating, decapsulating, processing, encapsulating, modulating, transmission, and routing activities, which leads to congestion and lengthens delays. Cluster formation, routing tables, setup, and maintenance paths are examples of routing elements included in energy-aware routing. Solutions are required to lessen message broadcasting and beacon message exchange because of their essential role in these operations. In settings with severe energy factor limits, the routing method reduces broadcast. One common technique for addressing broadcast storm problems is packet sequencing. The broadcast protocol should use as little overhead, latency, and energy as feasible to send packets to every node in the network. The coordinator serves as the network's central hub for message transmission in the Span technique. A node should become a coordinator if two neighbours of a coordinator node cannot communicate directly or via one or more coordinators. Rotating coordinators demonstrate how constrained node selections lead to a constrained, capacity-preserving global topology.

The increase in system lifespan brought on by Span increases as network density climbs and the ratio of idle to sleep energy usage rises. For instance, the simulations show that a realistic energy model improves the system lifetime of an 802.11 network in power-saving mode with Span by two times compared to no model. Span effortlessly integrates with the 802.11 power-saving techniques when employed with them, increasing system endurance, capacity, and communication latency.

Table 5 compares the synthetic statistical analysis over parameters like errors, executed time, achieved converge at round generation, and average optimal converges that considered optimisation algorithms with the other DE, EGA, IPSO, KHA, SCO methods, and IDE algorithms. It can be seen that the proposed ACMDE has acquired a higher value than 1.2% to 4.1% for the DE, 1.1% to 4.2% for the EGA, 2.3% to 4.1% for the IPSO, 1.5% to 3.2% for the KHA, 1.7% to 4.1% for the SCO, and 2.3 to 4.1% for the IDE algorithms in statistical analysis mean of the measured localisation error, respectively.

From the results, the statistical estimation of the recommended node localisation model of the ACMDE offers better performance in the cases of setting deployments than the other schemes' optimisations. Significantly, the error and convergence values achieved by the ACMDE are less location error, faster in convergence and executed time than the others compared to at least a reduced 1.5% to 4.7% error rate, and quicker by at least 4% and 2.1% in convergence and executed time, respectively for the experimental scenarios.

5 Conclusions

This study proposed an enhanced version of the DE algorithm called ACMDE to address optimal node localisation issues in WSN. The ACMDE algorithm incorporates chaos initialisation and dynamic adaptive factors to improve mutation and linear crossover, increasing population diversity and avoiding local optimisation. Function tests and node localisation experiments demonstrate the superiority of the ACMDE compared to other algorithms in the literature. The objective of the localisation model is to determine the location of unknown nodes considering variables like delay, PL, energy and received signal strength. The ACMDE provides optimal unknown node localisation by optimising the objective function. Simulation results show that the ACMDE outperforms other algorithms regarding localisation error. The proposed method effectively estimates the location of unknown nodes by implementing adaptive optimisation strategies. In future work, the proposed ACMDE algorithm can be further improved and extended for broader applications in WSN deployment, e.g., optimal coverage, localisation, and routing (Dao et al., 2020). The algorithm's performance can be enhanced by exploring different adaptive strategies for crossover, mutation, and reinitialisation. Additionally, the algorithm can be integrated with cloud computing platforms to handle large-scale localisation tasks efficiently (Dao et al., 2022). Furthermore, applying the ACMDE algorithm to domains, e.g., cloud computing, autonomous driving, and vectorised mapping, can provide valuable insights and contribute to advancements in those areas. Overall, future research should focus on refining and expanding the ACMDE algorithm to address the challenges of WSN optimisation and explore its potential in various real-world applications.

References

Al Shayokh, M. and Shin, S.Y. (2017) 'Bio Inspired distributed WSN Localization based on chicken swarm optimization', *Wireless Personal Communications*, Vol. 9, pp.5691–5706, https://doi.org/10.1007/s11277-017-4803-1.

Arora, S. and Singh, S. (2017) 'Node localization in wireless sensor networks using butterfly optimization algorithm', *Arabian Journal for Science and Engineering*, Vol. 42, No. 8, pp.3325–3335, DOI: 10.1007/s13369-017-2471-9.

Beheshti, Z. and Shamsuddin, S.M.H. (2013) 'A review of population-based meta-heuristic algorithm', *International Journal of Advances in Soft Computing and its Applications*, Vol. 5, No. 1, pp.1–35.

Bilal, M., Pant, H., Zaheer, L., Garcia-Hernandez, and Abraham, A. (2020) 'Differential evolution: a review of more than two decades of research', *Engineering Applications of Artificial Intelligence*, Vol. 90, p.103479, https://doi.org/10.1016/j.engappai.2020.103479.

Chen, B., Jamieson, K., Balakrishnan, H. and Morris, R. (2002) 'Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks', *Wireless Networks*, Vol. 8, No. 5, pp.481–494, DOI: 10.1023/A:1016542229220.

Cheung, K.W., So, H.C., Ma, W-K. and Chan, Y.T. (2004) 'Least squares algorithms for time-of-arrival-based mobile location', *IEEE Transactions on Signal Processing*.

Dao, T., Yu, J., Nguyen, T. and Ngo, T. (2020) 'A hybrid improved MVO and FNN for identifying collected data failure in cluster heads in WSN', *IEEE Access*, Vol. 8, pp.124311–124322, DOI: 10.1109/ACCESS.2020.3005247.

Dao, T-K., Chi, S-C., Nguyen, T-T., Nguyen, T-D. and Nguyen, V-T. (2022) 'An optimal WSN node coverage based on enhanced archimedes optimization algorithm', *Entropy*, Vol. 24, No. 8, DOI: 10.3390/e24081018.

Gong, W., Cai, Z., Ling, C.X. and Li, H. (2011) 'Enhanced differential evolution with adaptive strategies for numerical optimization', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 41, No. 2, pp.397–413, DOI: 10.1109/TSMCB.2010.2056367.

Gou, P., He, B. and Yu, Z. (2021) 'A node location algorithm based on improved whale optimization in wireless sensor networks', *Wireless Communications and Mobile Computing*, Vol. 2021, pp.1–17, Article ID 7523938, https://doi.org/10.1155/2021/7523938.

Harikrishnan, R., Kumar, V.J.S. and Ponmalar, P.S. (2014) 'Differential evolution approach for localization in wireless sensor networks', in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pp.1–4, DOI: 10.1109/ICCIC.2014.7238536.

Holland, J.H. (1992) 'Genetic algorithms', *Scientific American*, Vol. 267, No. 1, pp.66–73.

Kennedy, J. and Eberhart, R. (1948) 'Particle swarm optimization', in *Proceedings of ICNN'95 – International Conference on Neural Networks*, 1995, Vol. 6, No. 4, pp.1942–1948, DOI: 10.1109/ICNN.1995.488968.

Kirkpatrick, S.C.G. Jr, Vecchi, M.P. and Gelatt, C.D. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220, No. 4598, pp.671–80, DOI: 10.1126/science.220.4598.671.

Kulkarni, R.V. and Venayagamoorthy, G.K. (2010) 'Bio-inspired algorithms for autonomous deployment and localization of sensor nodes', *IEEE Transactions on Systems Man and Cybernetics Part C*, Vol. 40, No. 6, p.675.

Lakshmi, Y.V., Singh, P., Abouhawwash, M., Mahajan, S., Pandit, A.K. and Ahmed, A.B. (2022) 'Improved chan algorithm based optimum UWB sensor node localization using hybrid particle swarm optimization', *IEEE Access*, Vol. 10, pp.32546–32565, DOI: 10.1109/ACCESS.2022.3157719.

Liu, J. and Lampinen, J. (2005) 'A fuzzy adaptive differential evolution algorithm', *Soft Computing*, Vol. 9, No. 6, pp.448–462, DOI: 10.1007/s00500-004-0363-x.

Lu, X.F., Tang, K., Sendhoff, B.H. and Yao X. (2014) 'A new self-adaptation scheme for differential evolution', *Neurocomputing*, Vol. 146, pp.2–16, https://doi.org/10.1016/j.neucom.2014.04.071.

Mirjalili, S. (2015a) 'The ant lion optimizer', *Advances in Engineering Software*, Vol. 83, pp.80–98, DOI: 10.1016/j.advengsoft.2015.01.010.

Mirjalili, S. (2015b) 'Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm', *Knowledge-based Systems*, Vol. 89, pp.228–249, https://doi.org/10.1016/j.knosys.2015.07.006.

Mirjalili, S. and Lewis, A. (2016) 'The whale optimization algorithm', *Advances in Engineering Software*, Vol. 95, pp.51–67, DOI: 10.1016/j.advengsoft.2016.01.008.

Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) 'Grey Wolf optimizer', *Advances in Engineering Software*, Vol. 69, pp.46–61, Mar. 2014, DOI: 10.1016/j.advengsoft.2013.12.007.

Najarro, L.A.C., Song, I. and Kim, K. (2022) 'Fundamental limitations and state-of-the-art solutions for target node localization in WSNs: a review', *IEEE Sensors Journal*, Vol. 22, No. 24, pp.23661–23682, DOI: 10.1109/JSEN.2022.3217335.

Nguyen, T.T., Pan, J.S. and Dao, T.K. (2019a) 'A compact bat algorithm for unequal clustering in wireless sensor networks', *Applied Sciences*, Vol. 9, No. 10, p.1973, Switzerland, May 2019, DOI: 10.3390/app9101973.

Nguyen, T.T., Pan, J.S. and Dao, T.K. (2019b) 'An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network', *IEEE Access*, Vol. 7, pp.75985–75998, DOI: 10.1109/ACCESS.2019.2921721.

Nguyen, T-T., Dao, T-K., Horng, M-F. and Shieh, C-S. (2016a) 'An energy-based cluster head selection algorithm to support long-lifetime in wireless sensor networks', *Journal of Network Intelligence*, Vol. 1, No. 1, pp.23–37.

Nguyen, T-T., Pan, J-S., Chu, S-C., Roddick, J.F. and Dao, T-K. (2016b) 'Optimization localization in wireless sensor network based on multi-objective firefly algorithm', *Journal of Network Intelligence*, Vol. 1, No. 4, pp.130–138.

Nguyen, T-T., Pan, J-S., Dao, T-K., Sung, T-W. and Ngo, T-G. (2020) 'Pigeon-inspired optimization for node location in wireless sensor network', in Sattler, K.U., Nguyen, D., Vu, N., Tien Long, B. and Puta, H. (Eds.): *Advances in Engineering Research and Application. ICERA 2019. Lecture Notes in Networks and Systems*, Vol. 104, Springer, Cham., https://doi.org/10.1007/978-3-030-37497-6_67.

Nguyen, T-T., Thom, H.T.H. and Dao, T-K. (2017) 'Estimation localization in wireless sensor network based on multi-objective Grey wolf optimizer', in Akagi, M., Nguyen, T-T., Vu, D-T., Phung, T-N. and Huynh, V-N. (Eds.): *Advances in Information and Communication Technology: Proceedings of the International Conference, ICTA 2016*, pp.228–237, Springer International Publishing, Cham.

Niewiadomska-Szynkiewicz, E. (2012) 'Localization in wireless sensor networks: classification and evaluation of techniques', *International Journal of Applied Mathematics and Computer Science*, Vol. 22, No. 2, pp.281–297, DOI: 10.2478/v10006-012-0021-x.

Nithya, B. and Jeyachidra, J. (2021) 'Hybrid ABC-BAT optimization algorithm for localization in HWSN', *Microprocessors and Microsystems*, Article Id: 104024, https://doi.org/10.1016/j.micpro.2021.104024.

Phoemphon, S., So-In, C. and Leelathakul, N. (2020) 'A hybrid localization model using node segmentation and improved particle swarm optimization with obstacle-awareness for wireless sensor networks', *Expert Systems with Applications*, Vol. 143, p.113044, https://doi.org/10.1016/j.eswa.2019.113044.

Price, K.V., Storn, R.M. and Lampinen, J.A. (2005) *Differential Evolution. A Practical Approach to Global Optimization*, Springer Science and Business Media, Berlin Heidelberg, Germany.

Qin, A.K. and Suganthan, P.N. (2005) 'Self-adaptive differential evolution algorithm for numerical optimization', in *2005 IEEE Congress on Evolutionary Computation*, Vol. 2, pp.1785-1791, DOI: 10.1109/CEC.2005.1554904.

Ren, Q., Zhang, Y., Nikolaidis, I., Li, J. and Pan, Y. (2020) 'RSSI quantization and genetic algorithm based localization in wireless sensor networks', *Ad Hoc Networks*, Vol. 107, p.102255, DOI: https://doi.org/10.1016/j.adhoc.2020.102255.

Sabbella, V.R., Edla, D.R., Lipare, A. and Parne, S.R. (2021) 'An efficient localization approach in wireless sensor networks using Krill Herd optimization algorithm', *IEEE Systems Journal*, Vol. 15, No. 2, pp.2432–2442, DOI: 10.1109/JSYST.2020.3004527.

Shi, Q., Chen, H., Chen, H. and Jiang, L. (2010) 'Distributed wireless sensor network localization via sequential greedy optimization algorithm', *IEEE Transactions on Signal Processing*, Vol. 58, No. 6, pp.3328–3340.

Shiva, S.G., Yadav, P. and Chitra, A. (2012) 'Wireless sensor networks - architectures, protocols, simulators and applications : a survey', *International Journal of Electronics and Computer Science Engineering*, Vol. 1, No. 4, pp.1941–1953.

Singh, A., Sharma, S. and Singh, J. (2021) 'Nature-inspired algorithms for Wireless Sensor Networks: a comprehensive survey', *Computer Science Review*, Vol. 39, p.100342, https://doi.org/10.1016/j.cosrev.2020.100342.

Sivakumar, S. and Venkatesan, R. (2015) 'Meta-heuristic approaches for minimizing error in localization of wireless sensor networks', *Applied Soft Computing*, Vol. 36, pp.506–518, https://doi.org/10.1016/j.asoc.2015.05.053.

Zhang, J. and Sanderson, A.C. (2009) 'JADE: adaptive differential evolution with optional external archive', *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 5, pp.945–958.

Zhang, X. (2023) 'Differential evolution without the scale factor and the crossover probability', *Journal of Mathematics*, Vol. 2023, p.8973912, DOI: 10.1155/2023/8973912.

Zhang, Y., Wu, H.T., Gulliver, A., Xian, J. and Liang, L. (2023) 'Improved differential evolution for RSSD-based localization in Gaussian mixture noise', *Computer Communications*, Vol. 206, pp.51–59, DOI: https://doi.org/10.1016/j.comcom.2023.04.021.

Zhang, Z., Mehmood, A., Shu, L., Huo, Z., Zhang, Y. and Mukherjee, M. (2018) 'A survey on fault diagnosis in wireless sensor networks', *IEEE Access*, Vol. 6, pp.11349–11364, DOI: 10.1109/ACCESS.2018.2794519.

Zhu, C., Zheng, C., Shu, L. and Han, G. (2012) 'A survey on coverage and connectivity issues in wireless sensor networks', *Journal of Network and Computer Applications*, Vol. 35, No. 2, pp.619–632.

Appendix

The results of the numerical experiment, which assess and contrast the accuracy and performance of the suggested method with alternative algorithms, are shown in the appendix as figures and tables.

Figure A1 The comparison of convergence curve outcomes of the ACMDE as achieved optimal results with the SaDE, JaDE, EDE, DE, PSO, MFO, GWO, and ALO algorithms for the selected test functions with 50D, (a) CEC1 (b) CEC3 (c) CEC5 (d) CEC7 (e) CEC13 (f) CEC22 (see online version for colours)

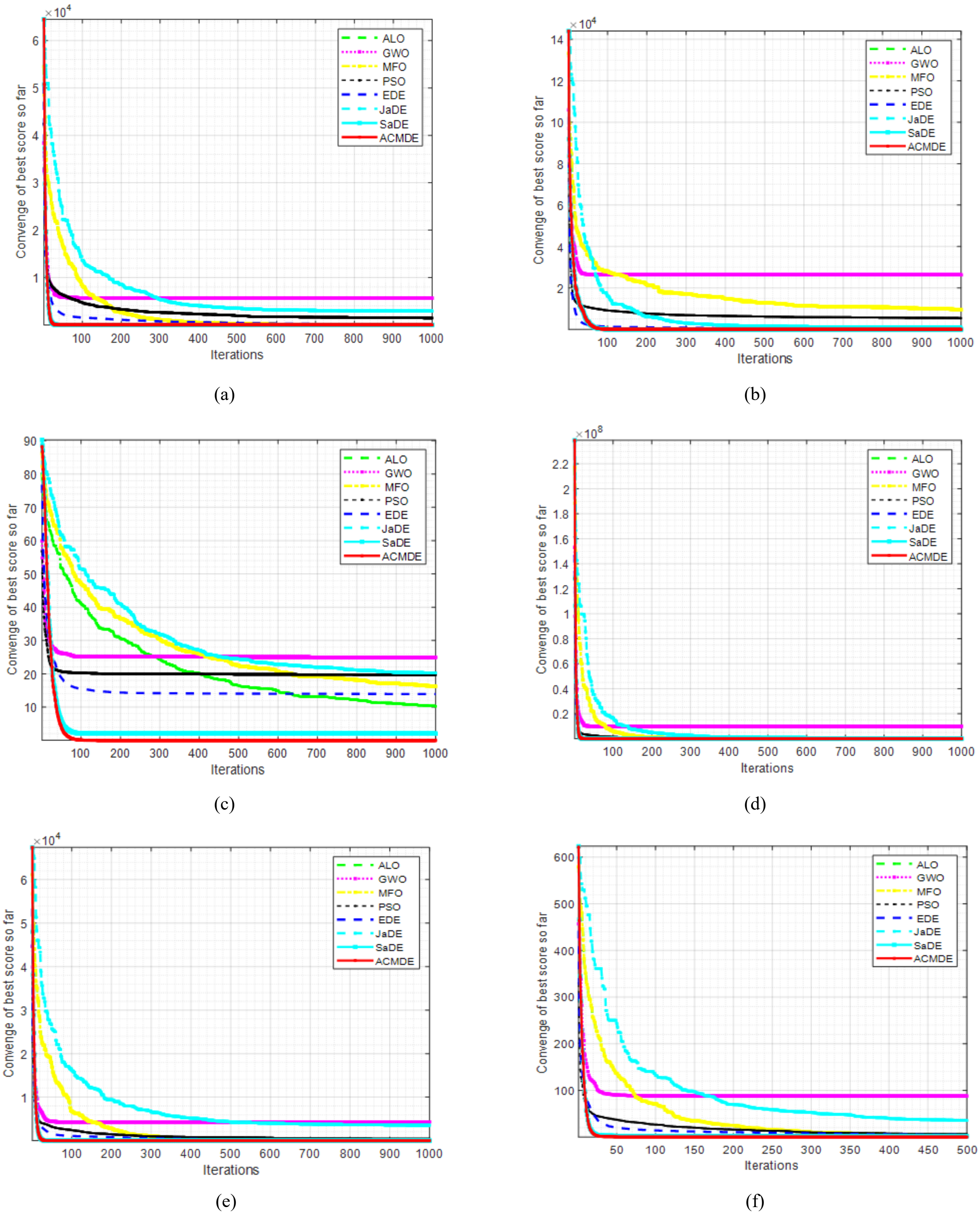


Table A1 The achieved optimal results of the ACMDE compares with the DE, EDE, JaDE, and SaDE algorithms for the test function on 50D performance

50D	*DE*		*EDE*		*JaDE*		*SaDE*		*ACMDE*	
	BEST	*MEAN*	*BEST*	*MEAN*	*BEST*	*MEAN*	*BEST*	*MEAN*	*BEST*	*MEAN*
CEC1	-1.25×10^{-03}	1.87×10^{-02}	$-1.37 \times 10^{+03}$	$2.12 \times 10^{+02}$	*-1.40×10^{-03}*	*-1.40×10^{-03}*	7.07×10^{-01}	1.39×10^{-04}	*-1.40×10^{-04}*	$-1.39 \times 10^{+04}$
CEC2	$4.32 \times 10^{+06}$	$2.46 \times 10^{+07}$	$1.04 \times 10^{+07}$	$3.74 \times 10^{+07}$	$1.24 \times 10^{+08}$	$1.71 \times 10^{+08}$	$8.28 \times 10^{+07}$	$1.14 \times 10^{+08}$	*1.86×10^{-06}*	*7.02×10^{-06}*
CEC3	$1.57 \times 10^{+09}$	$4.74 \times 10^{+09}$	$1.65 \times 10^{+10}$	$3.01 \times 10^{+03}$	$1.74 \times 10^{+09}$	$3.14 \times 10^{+09}$	$3.70 \times 10^{+10}$	$5.47 \times 10^{+10}$	*5.68×10^{-07}*	*2.35×10^{-05}*
CEC4	$2.13 \times 10^{+04}$	*$3.86 \times 10^{+04}$*	*$1.12 \times 10^{+04}$*	$3.98 \times 10^{+04}$	$7.22 \times 10^{+04}$	$9.00 \times 10^{+04}$	$6.42 \times 10^{+04}$	$9.62 \times 10^{+04}$	$5.87 \times 10^{+04}$	$1.11 \times 10^{+05}$
CEC5	$-8.61 \times 10^{+02}$	$-1.93 \times 10^{+02}$	$-8.51 \times 10^{+02}$	$7.72 \times 10^{+02}$	$-9.99 \times 10^{+02}$	$-9.98 \times 10^{+02}$	$5.20 \times 10^{+02}$	$2.89 \times 10^{+03}$	*$-1.00 \times 10^{+03}$*	*$-1.00 \times 10^{+03}$*
CEC6	$-8.10 \times 10^{+02}$	$-7.73 \times 10^{+02}$	$-8.48 \times 10^{+02}$	$-6.56 \times 10^{+02}$	$-8.54 \times 10^{+02}$	$-8.49 \times 10^{+02}$	$-5.39 \times 10^{+02}$	$-1.05 \times 10^{+02}$	$-8.84 \times 10^{+02}$	$-8.58 \times 10^{+02}$
CEC7	$-7.56 \times 10^{+02}$	$-7.31 \times 10^{+02}$	$-6.55 \times 10^{+02}$	$-3.76 \times 10^{+02}$	$-7.11 \times 10^{+02}$	$-7.03 \times 10^{+02}$	$-6.52 \times 10^{+02}$	$-6.17 \times 10^{+02}$	$-7.58 \times 10^{+02}$	$-7.14 \times 10^{+02}$
CEC8	$-6.79 \times 10^{+02}$	$-6.79 \times 10^{+02}$	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*
CEC9	$-5.87 \times 10^{+02}$	$-5.80 \times$ 10+02	$-5.71 \times 10^{+02}$	$-5.65 \times 10^{+02}$	$-5.63 \times 10^{+02}$	$-5.60 \times 10^{+02}$	$-5.71 \times 10^{+02}$	$-5.67 \times 10^{+02}$	$-5.82 \times 10^{+02}$	$-5.73 \times 10^{+02}$
CEC10	$-4.16 \times 10^{+02}$	$-2.19 \times 10^{+02}$	$-4.85 \times 10^{+02}$	$-1.03 \times 10^{+02}$	$-4.06 \times 10^{+02}$	$-3.60 \times 10^{+02}$	$6.66 \times 10^{+02}$	$1.35 \times 10^{+03}$	$-5.00 \times 10^{+02}$	*$-4.68 \times 10^{+02}$*
CEC11	$-3.49 \times 10^{+02}$	$-3.05 \times 10^{+02}$	$-2.10 \times 10^{+02}$	$-1.02 \times 10^{+02}$	$-2.24 \times 10^{+02}$	$-2.00 \times 10^{+02}$	-4.48×10^{-01}	$5.98 \times 10^{+01}$	$-3.61 \times$ 10+02	*$-3.39 \times 10^{+02}$*
CEC12	$-2.25 \times 10^{+02}$	$-1.59 \times 10^{+02}$	$-9.88 \times 10^{+01}$	$2.87 \times 10^{+00}$	$-9.16 \times 10^{+01}$	$-6.89 \times 10^{+01}$	$1.09 \times 10^{+02}$	$1.67 \times 10^{+02}$	$-2.66 \times$ 10+02	*$-2.00 \times 10^{+02}$*
CEC13	*$-8.90 \times 10^{+01}$*	$1.95 \times 10^{+01}$	$4.31 \times 10^{+01}$	$1.70 \times 10^{+02}$	$-8.62 \times 10^{+00}$	$3.01 \times 10^{+01}$	$2.12 \times 10^{+02}$	$2.72 \times 10^{+02}$	$-8.79 \times$ 10+01	*$-1.82 \times 10^{+01}$*
CEC14	*$1.80 \times 10^{+03}$*	$3.58 \times 10^{+03}$	$2.89 \times 10^{+03}$	$4.73 \times 10^{+03}$	$6.13 \times 10^{+03}$	$6.86 \times 10^{+03}$	$6.35 \times 10^{+03}$	$7.15 \times 10^{+03}$	$2.07 \times$ 10+03	$3.77 \times 10^{+03}$
CEC15	$2.69 \times$ 10+03	$5.20 \times 10^{+03}$	$3.36 \times 10^{+03}$	*$5.12 \times 10^{+03}$*	$6.85 \times 10^{+03}$	$7.72 \times 10^{+03}$	$5.61 \times 10^{+03}$	$6.77 \times 10^{+03}$	$4.33 \times 10^{+03}$	$7.15 \times 10^{+03}$
CEC16	$2.02 \times 10^{+02}$	*$2.03 \times 10^{+02}$*	*$2.01 \times 10^{+02}$*	*$2.03 \times 10^{+02}$*	$2.02 \times 10^{+02}$	*$2.03 \times 10^{+02}$*	$2.02 \times 10^{+02}$	*$2.03 \times 10^{+02}$*	$2.02 \times 10^{+02}$	$2.04 \times 10^{+02}$
CEC17	$4.03 \times 10^{+02}$	$4.77 \times 10^{+02}$	$5.21 \times 10^{+02}$	$6.73 \times 10^{+02}$	$5.38 \times 10^{+02}$	$5.61 \times 10^{+02}$	$8.76 \times 10^{+02}$	$9.33 \times 10^{+02}$	*$3.68 \times 10^{+02}$*	*$4.35 \times 10^{+02}$*
CEC18	$6.34 \times 10^{+02}$	$6.67 \times 10^{+02}$	$6.58 \times 10^{+02}$	$7.57 \times 10^{+02}$	$6.53 \times 10^{+02}$	$6.70 \times 10^{+02}$	$9.77 \times 10^{+02}$	$1.04 \times 10^{+03}$	*$5.03 \times$ 10+02*	*$6.09 \times 10^{+02}$*
CEC19	$5.05 \times 10^{+02}$	$6.00 \times 10^{+02}$	$5.19 \times 10^{+02}$	$1.01 \times 10^{+03}$	$5.21 \times 10^{+02}$	*$5.23 \times 10^{+02}$*	$2.35 \times 10^{+03}$	$7.14 \times 10^{+03}$	*$5.04 \times 10^{+02}$*	$5.24 \times 10^{+02}$
CEC20	*$6.12 \times 10^{+02}$*	*$6.13 \times 10^{+02}$*	*$6.12 \times 10^{+02}$*	$6.14 \times 10^{+02}$	$6.13 \times 10^{+02}$	*$6.13 \times 10^{+02}$*	$6.14 \times 10^{+02}$	$6.14 \times 10^{+02}$	*$6.12 \times 10^{+02}$*	*$6.13 \times 10^{+02}$*
CEC21	$1.14 \times 10^{+03}$	$1.67 \times 10^{+03}$	$9.43 \times 10^{+02}$	$1.13 \times 10^{+03}$	$1.01 \times 10^{+03}$	$1.03 \times 10^{+03}$	$2.65 \times 10^{+03}$	$2.79 \times 10^{+03}$	*$9.00 \times 10^{+02}$*	$9.82 \times 10^{+02}$
CEC22	*$2.29 \times 10^{+03}$*	$3.95 \times 10^{+03}$	$5.00 \times 10^{+03}$	$6.18 \times 10^{+03}$	$7.54 \times 10^{+03}$	$8.37 \times 10^{+03}$	$7.48 \times 10^{+03}$	$8.38 \times 10^{+03}$	$2.90 \times 10^{+03}$	*$3.84 \times 10^{+03}$*
CEC23	*$3.35 \times 10^{+03}$*	*$5.68 \times 10^{+03}$*	$5.53 \times 10^{+03}$	$7.02 \times 10^{+03}$	$8.29 \times 10^{+03}$	$8.92 \times 10^{+03}$	$6.79 \times 10^{+03}$	$8.05 \times 10^{+03}$	$5.33 \times 10^{+03}$	$8.08 \times 10^{+03}$
CEC24	*$1.24 \times 10^{+03}$*	$1.27 \times 10^{+03}$	$1.30 \times 10^{+03}$	$1.34 \times 10^{+03}$	$1.30 \times 10^{+03}$	$1.30 \times 10^{+03}$	$1.28 \times 10^{+03}$	$1.30 \times 10^{+03}$	$1.25 \times 10^{+03}$	*$1.26 \times 10^{+03}$*
CEC25	*$1.36 \times 10^{+03}$*	$1.39 \times 10^{+03}$	$1.40 \times 10^{+03}$	$1.44 \times 10^{+03}$	$1.40 \times 10^{+03}$	$1.41 \times 10^{+03}$	$1.41 \times 10^{+03}$	$1.43 \times 10^{+03}$	*$1.36 \times 10^{+03}$*	*$1.38 \times 10^{+03}$*
CEC26	*$1.40 \times 10^{+03}$*	$1.54 \times 10^{+03}$	$1.58 \times 10^{+03}$	$1.60 \times 10^{+03}$	$1.42 \times 10^{+03}$	*$1.43 \times 10^{+03}$*	$1.40 \times 10^{+03}$	*$1.43 \times 10^{+03}$*	*$1.40 \times 10^{+03}$*	$1.55 \times 10^{+03}$
CEC27	$2.01 \times 10^{+03}$	*$2.14 \times 10^{+03}$*	$2.43 \times 10^{+03}$	$2.67 \times 10^{+03}$	$2.56 \times 10^{+03}$	$2.61 \times 10^{+03}$	$2.43 \times 10^{+03}$	$2.52 \times 10^{+03}$	*$2.04 \times 10^{+03}$*	$2.23 \times 10^{+03}$
CEC28	$2.10 \times 10^{+03}$	$2.63 \times 10^{+03}$	$2.44 \times 10^{+03}$	$4.21 \times 10^{+03}$	$1.71 \times 10^{+03}$	*$1.74 \times 10^{+03}$*	$4.12 \times 10^{+03}$	$4.51 \times 10^{+03}$	*$1.70 \times 10^{+03}$*	$1.96 \times 10^{+03}$
Win	11	9	4	3	2	7	2	3	19	15
Lose	15	18	22	23	24	20	23	21	8	11
Draw	2	1	2	2	2	1	3	4	1	2

Table A2 The achieved optimal results of the ACMDE compares with the DE, EDE, JaDE, and SaDE algorithms for the test function on 100D performance

100D	*DE*		*EDE*		*JaDE*		*SaDE*		*ACMDE*	
	BEST	*MEAN*	*BEST*	*MEAN*	*BEST*	*MEAN*	*BEST*	*MEAN*	*BEST*	*MEAN*
CEC1	$1.74 \times 10^{+02}$	$2.35 \times 10^{+03}$	$-9.36 \times 10^{+02}$	$5.45 \times 10^{+02}$	$-1.25 \times 10^{+03}$	$-1.17 \times 10^{+03}$	$2.64 \times 10^{+04}$	$3.73 \times 10^{+04}$	*$-1.40 \times 10^{+01}$*	*$-1.31 \times 10^{+01}$*
CEC2	$2.06 \times 10^{+03}$	$5.17 \times 10^{+03}$	$2.30 \times 10^{+03}$	$9.17 \times 10^{+03}$	$4.87 \times 10^{+03}$	$6.25 \times 10^{+03}$	$1.94 \times 10^{+03}$	$2.97 \times 10^{+03}$	*$1.09 \times 10^{+01}$*	*$1.85 \times 10^{+01}$*
CEC3	$8.07 \times 10^{+03}$	$2.01 \times 10^{+4}$	$3.20 \times 10^{+4}$	$7.18 \times 10^{+5}$	$5.34 \times 10^{+4}$	$6.88 \times 10^{+5}$	$7.50 \times 10^{+5}$	$1.04 \times 10^{+5}$	*$3.17 \times 10^{+09}$*	*$1.68 \times 10^{+01}$*
CEC4	$3.99 \times 10^{+04}$	*$5.64 \times 10^{+04}$*	*$3.84 \times 10^{+04}$*	$7.30 \times 10^{+04}$	$1.41 \times 10^{+05}$	$1.65 \times 10^{+05}$	$1.34 \times 10^{+05}$	$1.58 \times 10^{+05}$	$9.85 \times 10^{+01}$	$2.01 \times 10^{+02}$
CEC5	$-4.79 \times 10^{+02}$	$1.01 \times 10^{+02}$	$-7.36 \times 10^{+02}$	$3.42 \times 10^{+02}$	$-9.15 \times 10^{+02}$	$-8.78 \times 10^{+02}$	$1.61 \times 10^{+03}$	$2.99 \times 10^{+03}$	*$-1.00 \times 10^{+03}$*	*$3.66 \times 10^{+01}$*
CEC6	$-7.29 \times 10^{+02}$	$-5.87 \times 10^{+02}$	$-7.87 \times 10^{+02}$	$-6.16 \times 10^{+02}$	$-8.19 \times 10^{+02}$	$-7.79 \times 10^{+02}$	$9.24 \times 10^{+02}$	$1.32 \times 10^{+03}$	*$-8.56 \times 10^{+02}$*	*$-8.23 \times 10^{+02}$*
CEC7	*$-7.46 \times 10^{+02}$*	*$-7.22 \times 10^{+02}$*	$-6.69 \times 10^{+02}$	$-3.91 \times 10^{+02}$	$-6.31 \times 10^{+02}$	$-6.13 \times 10^{+02}$	$-6.38 \times 10^{+02}$	$-5.96 \times 10^{+02}$	$-7.09 \times 10^{+02}$	$-6.48 \times 10^{+02}$
CEC8	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*
CEC9	*$-5.64 \times 10^{+02}$*	*$-5.58 \times 10^{+02}$*	$-5.44 \times 10^{+02}$	$-5.34 \times 10^{+02}$	$-5.29 \times 10^{+02}$	$-5.26 \times 10^{+02}$	$-5.45 \times 10^{+02}$	$-5.35 \times 10^{+02}$	$-5.63 \times 10^{+02}$	$-5.49 \times 10^{+02}$
CEC10	$3.81 \times 10^{+01}$	$3.27 \times 10^{+02}$	$-8.44 \times 10^{+01}$	$3.33 \times 10^{+02}$	$8.72 \times 10^{+02}$	$1.80 \times 10^{+03}$	$2.17 \times 10^{+03}$	$3.22 \times 10^{+03}$	$-4.97 \times 10^{+02}$	$-4.74 \times 10^{+02}$
CEC11	$-2.16 \times 10^{+02}$	$-1.38 \times 10^{+02}$	$1.14 \times 10^{+02}$	$3.42 \times 10^{+02}$	$8.47 \times 10^{+00}$	$5.53 \times 10^{+01}$	$3.41 \times 10^{+02}$	$3.95 \times 10^{+02}$	$-3.17 \times 10^{+02}$	$-2.17 \times 10^{+02}$
CEC12	$-1.16 \times 10^{+02}$	$2.40 \times 10^{+01}$	$2.28 \times 10^{+02}$	$3.89 \times 10^{+02}$	$1.38 \times 10^{+02}$	$2.20 \times 10^{+02}$	$4.88 \times 10^{+02}$	$5.88 \times 10^{+02}$	$-1.40 \times 10^{+02}$	$-8.56 \times 10^{+00}$
CEC13	*$8.81 \times 10^{+01}$*	*$2.15 \times 10^{+02}$*	$4.43 \times 10^{+02}$	$5.95 \times 10^{+02}$	$2.91 \times 10^{+02}$	$3.30 \times 10^{+02}$	$5.23 \times 10^{+02}$	$6.63 \times 10^{+02}$	$9.26 \times 10^{+01}$	$2.56 \times 10^{+02}$
CEC14	*$4.22 \times 10^{+03}$*	*$6.13 \times 10^{+03}$*	$6.34 \times 10^{+03}$	$9.49 \times 10^{+03}$	$1.23 \times 10^{+04}$	$1.32 \times 10^{+04}$	$1.34 \times 10^{+04}$	$1.40 \times 10^{+04}$	$4.52 \times 10^{+03}$	$6.08 \times 10^{+03}$
CEC15	*$5.37 \times 10^{+03}$*	*$8.40 \times 10^{+03}$*	$8.56 \times 10^{+03}$	$1.08 \times 10^{+04}$	$1.39 \times 10^{+04}$	$1.47 \times 10^{+04}$	$1.22 \times 10^{+04}$	$1.36 \times 10^{+04}$	$7.87 \times 10^{+03}$	$1.34 \times 10^{+04}$
CEC16	$2.03 \times 10^{+02}$	$2.04 \times 10^{+02}$	*$2.01 \times 10^{+02}$*	*$2.02 \times 10^{+02}$*	$2.03 \times 10^{+02}$	$2.04 \times 10^{+02}$	$2.03 \times 10^{+02}$	$2.04 \times 10^{+02}$	$2.03 \times 10^{+02}$	$2.04 \times 10^{+02}$
CEC17	$5.65 \times 10^{+02}$	$7.06 \times 10^{+02}$	$1.05 \times 10^{+03}$	$1.21 \times 10^{+03}$	$8.26 \times 10^{+02}$	$8.91 \times 10^{+02}$	$1.36 \times 10^{+03}$	$1.48 \times 10^{+03}$	*$4.53 \times 10^{+02}$*	*$5.49 \times 10^{+02}$*
CEC18	$7.79 \times 10^{+02}$	$9.66 \times 10^{+02}$	$1.08 \times 10+03$	$1.29 \times 10^{+03}$	$9.58 \times 10^{+02}$	$1.01 \times 10^{+03}$	$1.50 \times 10^{+03}$	$1.59 \times 10^{+03}$	*$7.23 \times 10^{+02}$*	*$9.34 \times 10^{+02}$*
CEC19	$5.55 \times 10^{+02}$	*$1.61 \times 10^{+03}$*	$6.54 \times 10^{+02}$	$1.79 \times 10^{+03}$	$2.75 \times 10^{+03}$	$5.56 \times 10^{+03}$	$9.76 \times 10^{+03}$	$1.63 \times 10^{+04}$	*$5.10 \times 10^{+02}$*	$5.68 \times 10^{+03}$
CEC20	*$6.20 \times 10^{+02}$*	*$6.22 \times 10^{+02}$*	$6.22 \times 10^{+02}$	$6.24 \times 10^{+02}$	$6.23 \times 10^{+02}$	$6.23 \times 10^{+02}$	$6.23 \times 10^{+02}$	$6.24 \times 10^{+02}$	$6.22 \times 10^{+02}$	$6.23 \times 10^{+02}$
CEC21	$1.86 \times 10^{+03}$	$2.87 \times 10^{+03}$	$1.56 \times 10^{+03}$	$2.06 \times 10^{+03}$	$1.58 \times 10^{+03}$	$2.57 \times 10^{+03}$	$4.73 \times 10^{+03}$	$4.88 \times 10^{+03}$	*$9.00 \times 10^{+02}$*	*$1.60 \times 10^{+03}$*
CEC22	$6.41 \times 10^{+03}$	$8.43 \times 10^{+03}$	$9.32 \times 10^{+03}$	$1.24 \times 10^{+04}$	$1.40 \times 10^{+04}$	$1.47 \times 10^{+04}$	$1.49 \times 10^{+04}$	$1.60 \times 10^{+04}$	*$5.36 \times 10^{+03}$*	*$8.12 \times 10^{+03}$*
CEC23	*$7.58 \times 10^{+03}$*	*$9.60 \times 10^{+03}$*	$8.64 \times 10^{+03}$	$1.34 \times 10^{+04}$	$1.51 \times 10^{+04}$	$1.61 \times 10^{+04}$	$1.45 \times 10^{+04}$	$1.61 \times 10^{+04}$	$9.17 \times 10^{+03}$	$1.44 \times 10^{+04}$
CEC24	$1.29 \times 10^{+03}$	$1.33 \times 10^{+03}$	$1.42 \times 10^{+03}$	$1.47 \times 10^{+03}$	$1.38 \times 10^{+03}$	$1.39 \times 10^{+03}$	$1.42 \times 10^{+03}$	$1.43 \times 10^{+03}$	*$1.28 \times 10^{+03}$*	*$1.32 \times 10^{+03}$*
CEC25	$1.43 \times 10^{+03}$	$1.46 \times 10^{+03}$	$1.52 \times 10^{+03}$	$1.56 \times 10^{+03}$	$1.50 \times 10^{+03}$	$1.51 \times 10^{+03}$	$1.53 \times 10^{+03}$	$1.54 \times 10^{+03}$	*$1.41 \times 10^{+03}$*	*$1.45 \times 10^{+03}$*
CEC26	$1.59 \times 10^{+03}$	$1.60 \times 10^{+03}$	*$1.41 \times 10^{+03}$*	$1.67 \times 10^{+03}$	$1.46 \times 10^{+03}$	*$1.50 \times 10^{+03}$*	$1.67 \times 10^{+03}$	$1.68 \times 10^{+03}$	$1.59 \times 10^{+03}$	$1.62 \times 10^{+03}$
CEC27	*$2.49 \times 10^{+03}$*	*$2.71 \times 10^{+03}$*	$3.35 \times 10^{+03}$	$3.70 \times 10^{+03}$	$3.42 \times 10^{+03}$	$3.49 \times 10^{+03}$	$3.41 \times 10^{+03}$	$3.56 \times 10^{+03}$	$2.65 \times 10^{+03}$	$2.85 \times 10^{+03}$
CEC28	$2.15 \times 10^{+03}$	$3.44 \times 10^{+03}$	$2.87 \times 10^{+03}$	$5.95 \times 10^{+03}$	$1.91 \times 10^{+03}$	*$2.00 \times 10^{+03}$*	$6.09 \times 10^{+03}$	$7.13 \times 10^{+03}$	*$1.80 \times 10^{+03}$*	$3.85 \times 10^{+03}$
Win	9	11	4	2	1	3	1	1	17	15
Lose	15	16	22	22	25	21	25	26	10	12
Draw	4	1	2	4	2	4	2	1	1	1

Table 3 The achieved optimal results of the ACMDE against the ALO, GWO, MFO, and PSO algorithms for the test function on 100D performance

100D	*ALO*		*GWO*		*MFO*		*PSO*		*ACMDE*	
	BEST	*MEAN*	*BEST*	*MEAN*	*BEST*	*BEST*	*MEAN*	*BEST*	*MEAN*	*BEST*
CEC1	$9.37 \times 10^{+03}$	$1.95 \times 10^{+04}$	$9.60 \times 10^{+03}$	$1.63 \times 10^{+04}$	$2.56 \times 10^{+04}$	$3.11 \times 10^{+04}$	$1.14 \times 10^{+05}$	$1.30 \times 10^{+05}$	*$-1.40 \times 10^{+03}$*	*$-1.12 \times 10^{+03}$*
CEC2	$1.11 \times 10^{+01}$	$1.87 \times 10^{+01}$	$1.52 \times 10^{+01}$	$7.27 \times 10^{+01}$	$2.01 \times 10^{+01}$	$2.84 \times 10^{+01}$	$1.41 \times 10^{+01}$	$1.99 \times 10^{+02}$	*$4.80 \times 10^{+0}$*	*$8.41 \times 10^{+0}$*
CEC3	$1.41 \times 10^{+2}$	$8.21 \times 10^{+2}$	$3.24 \times 10^{+2}$	$1.25 \times 10^{+3}$	$2.06 \times 10^{+2}$	$7.76 \times 10^{+2}$	$5.48 \times 10^{+5}$	$1.49 \times 10^{+2}$	*$5.10 \times 10^{+1}$*	*$1.14 \times 10^{+1}$*
CEC4	$1.11 \times 10^{+05}$	$1.35 \times 10^{+05}$	$1.54 \times 10^{+05}$	$2.22 \times 10^{+05}$	$2.64 \times 10^{+05}$	$3.80 \times 10^{+05}$	$2.59 \times 10^{+05}$	$3.54 \times 10^{+05}$	$2.79 \times 10^{+05}$	$4.24 \times 10^{+05}$
CEC5	$2.51 \times 10^{+03}$	$4.35 \times 10^{+03}$	$1.69 \times 10^{+03}$	$4.63 \times 10^{+03}$	$1.51 \times 10^{+03}$	2.35 × 10+03	$1.57 \times 10^{+04}$	$2.12 \times 10^{+04}$	*$-8.65 \times 10^{+02}$*	*$2.24 \times 10_{+03}$*
CEC6	$6.93 \times 10^{+02}$	$1.55 \times 10^{+03}$	$7.64 \times 10^{+02}$	$1.80 \times 10^{+03}$	$2.01 \times 10^{+03}$	$2.60 \times 10^{+03}$	$1.86 \times 10^{+04}$	$2.31 \times 10^{+04}$	*$-6.27 \times 10^{+02}$*	*$-4.50 \times 10^{+02}$*
CEC7	*$-6.56 \times 10^{+02}$*	$-4.17 \times 10^{+02}$	$9.34 \times 10^{+03}$	$1.89 \times 10^{+05}$	$-3.35 \times 10^{+02}$	$4.81 \times 10^{+02}$	$2.95 \times 10^{+04}$	$1.72 \times 10^{+05}$	$-6.47 \times 10^{+02}$	*$-5.45 \times 10^{+02}$*
CEC8	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*	$-6.79 \times 10^{+02}$	$-6.79 \times 10^{+02}$	$-6.79 \times 10^{+02}$	$-6.79 \times 10^{+02}$	$-6.79 \times 10^{+02}$	$-6.79 \times 10^{+02}$	*$-6.79 \times 10^{+02}$*	*$-6.79 \times 10^{+02}$*
CEC9	*$-5.12 \times 10^{+02}$*	*$-5.00 \times 10^{+02}$*	$-4.62 \times 10^{+02}$	$-4.51 \times 10^{+02}$	$-4.42 \times 10^{+02}$	$-4.37 \times 10^{+02}$	$-4.67 \times 10^{+02}$	$-4.48 \times 10^{+02}$	*$-5.12 \times 10^{+02}$*	$-4.80 \times 10^{+02}$
CEC10	$1.40 \times 10^{+03}$	$2.46 \times 10^{+03}$	$2.62 \times 10^{+03}$	$4.21 \times 10^{+03}$	$1.11 \times 10^{+04}$	$1.32 \times 10^{+04}$	$1.35 \times 10^{+04}$	$1.60 \times 10^{+04}$	*$-2.53 \times 10^{+02}$*	*$2.05 \times 10^{+02}$*
CEC11	$1.82 \times 10^{+02}$	$4.27 \times 10^{+02}$	$1.24 \times 10^{+03}$	$1.77 \times 10^{+03}$	$8.90 \times 10^{+02}$	$1.00 \times 10^{+03}$	$1.59 \times 10^{+03}$	$1.82 \times 10^{+03}$	*$6.85 \times 10^{+01}$*	*$3.10 \times 10^{+02}$*
CEC12	*$3.63 \times 10^{+02}$*	*$5.60 \times 10^{+02}$*	$1.45 \times 10^{+03}$	$1.88 \times 10^{+03}$	$1.18 \times 10^{+03}$	$1.31 \times 10^{+03}$	$1.78 \times 10^{+03}$	$2.07 \times 10^{+03}$	$3.81 \times 10^{+02}$	$5.73 \times 10^{+02}$
CEC13	*$6.54 \times 10^{+02}$*	*$9.23 \times 10^{+02}$*	$1.24 \times 10^{+03}$	$2.17 \times 10^{+03}$	$1.31 \times 10^{+03}$	$1.44 \times 10^{+03}$	$1.99 \times 10^{+03}$	$2.21 \times 10^{+03}$	$8.41 \times 10^{+02}$	$1.11 \times 10^{+03}$
CEC14	$1.39 \times 10^{+04}$	*$1.74 \times 10^{+04}$*	$2.10 \times 10^{+04}$	$2.46 \times 10^{+04}$	$3.06 \times 10^{+04}$	$3.15 \times 10^{+04}$	$2.94 \times 10^{+04}$	$3.10 \times 10^{+04}$	*$1.17 \times 10^{+04}$*	$1.75 \times 10^{+04}$
CEC15	*$1.36 \times 10^{+04}$*	*$1.87 \times 10^{+04}$*	$2.17 \times 10^{+04}$	$2.44 \times 10^{+04}$	$3.08 \times 10^{+04}$	$3.18 \times 10^{+04}$	$2.58 \times 10^{+04}$	$2.97 \times 10^{+04}$	$1.62 \times 10^{+04}$	$2.62 \times 10^{+04}$
CEC16	$2.04 \times 10^{+02}$	$2.05 \times 10^{+02}$	$2.03 \times 10^{+02}$	*$2.04 \times 10^{+02}$*	$2.04 \times 10^{+02}$	$2.05 \times 10^{+02}$	$2.04 \times 10^{+02}$	*$2.04 \times 10^{+02}$*	*$2.02 \times 10^{+02}$*	$2.05 \times 10^{+02}$
CEC17	$1.34 \times 10^{+03}$	$1.55 \times 10^{+03}$	$2.44 \times 10^{+03}$	$3.06 \times 10^{+03}$	$3.03 \times 10^{+03}$	$3.27 \times 10^{+03}$	$3.50 \times 10^{+03}$	$3.82 \times 10^{+03}$	*$9.59 \times 10^{+02}$*	*$1.25 \times 10^{+03}$*
CEC18	$1.73 \times 10^{+03}$	*$2.00 \times 10^{+03}$*	$2.84 \times 10^{+03}$	$3.22 \times 10^{+03}$	$3.06 \times 10^{+03}$	$3.37 \times 10^{+03}$	$3.63 \times 10^{+03}$	$3.98 \times 10^{+03}$	*$1.68 \times 10^{+03}$*	$2.18 \times 10^{+03}$
CEC19	$7.50 \times 10^{+03}$	$2.57 \times 10^{+04}$	$1.35 \times 10^{+04}$	$3.36 \times 10^{+04}$	$8.59 \times 10^{+05}$	$1.52 \times 10^{+06}$	$1.42 \times 10^{+05}$	$2.90 \times 10^{+05}$	*$6.01 \times 10^{+02}$*	*$7.24 \times 10^{+03}$*
CEC20	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*	*$6.50 \times 10^{+02}$*
CEC21	$3.53 \times 10^{+03}$	$5.57 \times 10^{+03}$	$2.80 \times 10^{+03}$	$4.91 \times 10^{+03}$	$7.53 \times 10^{+03}$	$8.23 \times 10^{+03}$	$8.89 \times 10^{+03}$	$9.48 \times 10^{+03}$	*$1.13 \times 10^{+03}$*	*$1.36 \times 10^{+03}$*
CEC22	$1.60 \times 10^{+04}$	$2.15 \times 10^{+04}$	$2.70 \times 10^{+04}$	$2.96 \times 10^{+04}$	$3.23 \times 10^{+04}$	$3.31 \times 10^{+04}$	$3.24 \times 10^{+04}$	$3.37 \times 10^{+04}$	*$1.28 \times 10^{+04}$*	*$1.71 \times 10^{+04}$*
CEC23	*$1.71 \times 10^{+04}$*	*$2.35 \times 10^{+04}$*	$2.50 \times 10^{+04}$	$2.88 \times 10^{+04}$	$3.30 \times 10^{+04}$	$3.41 \times 10^{+04}$	$3.11 \times 10^{+04}$	$3.40 \times 10^{+04}$	2.01 ×10+04	$2.98 \times 10^{+04}$
CEC24	*$1.45 \times 10^{+03}$*	*$1.49 \times 10^{+03}$*	$1.67 \times 10^{+03}$	$2.03 \times 10^{+03}$	$1.61 \times 10^{+03}$	$1.62 \times 10^{+03}$	$1.71 \times 10^{+03}$	$1.74 \times 10^{+03}$	*$1.45 \times 10^{+03}$*	*$1.49 \times 10^{+03}$*
CEC25	$1.65 \times 10^{+03}$	$1.68 \times 10^{+03}$	$1.83 \times 10^{+03}$	$1.94 \times 10^{+03}$	$1.75 \times 10^{+03}$	$1.76 \times 10^{+03}$	$1.82 \times 10^{+03}$	$1.84 \times 10^{+03}$	*$1.61 \times 10^{+03}$*	*$1.65 \times 10^{+03}$*
CEC26	*$1.73 \times 10^{+03}$*	*$1.75 \times 10^{+03}$*	$1.87 \times 10^{+03}$	$1.93 \times 10^{+03}$	$1.91 \times 10^{+03}$	$1.91 \times 10^{+03}$	$1.87 \times 10^{+03}$	$1.90 \times 10^{+03}$	$1.74 \times 10^{+03}$	$1.79 \times 10^{+03}$
CEC27	$4.14 \times 10^{+03}$	*$4.38 \times 10^{+03}$*	$5.94 \times 10^{+03}$	$6.80 \times 10^{+03}$	$5.67 \times 10^{+03}$	$5.81 \times 10^{+03}$	$5.91 \times 10^{+03}$	$6.08 \times 10^{+03}$	*$4.12 \times 10^{+03}$*	$4.44 \times 10^{+03}$
CEC28	$8.51 \times 10^{+03}$	$1.07 \times 10^{+04}$	$1.75 \times 10^{+04}$	$2.09 \times 10^{+04}$	$1.18 \times 10^{+04}$	$1.31 \times 10^{+04}$	$1.84 \times 10^{+04}$	$1.96 \times 10^{+04}$	*$4.77 \times 10^{+03}$*	*$8.53 \times 10^{+03}$*
Win	11	13	2	3	2	2	2	3	21	17
Lose	16	14	24	22	24	24	22	23	6	9
Draw	1	1	2	3	2	2	3	2	1	2