Kufa Journal of Engineering
Vol. 17, No. 2, April 2026, P.P. 61-78
Article history: Received 11 November 2024, last revised 3 March 2025, accepted 21 March 2025

ADAPTIVE EXPLORATION ARTIFICIAL BEE COLONY ON EVOLUTIONARY CLUSTERING

Doaa Alsamee[1], Hazim Albedran[2], and Reza Ramezani[3]

[1] Faculty of Computer Science and Mathematics, Department of Computer Science, University of Kufa, Iraq, Email:doaam.alsamee@uokufa.edu.iq.

[2] Department of Mechanical Engineering, Faculty of Engineering, University of Kufa, Iraq, Email:hazimn.bedran@uokufa.edu.iq.

[3] Department of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, Email:r.ramezani@eng.ui.ac.ir.

https://doi.org/10.30572/2018/KJE/170205

ABSTRACT

The current study was an attempt to examine the new application of the Adaptive Exploration Artificial Bee Colony algorithm in evolutionary clustering. An optimized version of the traditional Artificial Bee Colony (ABC) algorithm, AEABC relies heavily on a probability parameter that is based on distance. In this regard, a balanced relationship of exploration and exploitation leads to the dynamic parameter adaption, resulting in the spatial distribution of solutions. As a result, AEABC provides an efficient performance for solving dynamic clustering tasks, involving consistent data changes and constant reformation of cluster structures. To conduct the present study, the performance of AEABC on evolutionary clustering was compared with that of other metaheuristic algorithms, namely Genetic Algorithm (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO). The results confirmed that AEABC produced lower performance error rates than competing algorithms in all three scenarios representing best case, worst case and average case performance. This established both its reliability as well as robust efficiency. AEABC indicated convergence performance because the algorithm initiates with quick development in the first stages then reached stable points effectively, which shows its ability to position itself well in changing data environments. Moreover, the findings could confirm AEABC as an effective means for evolutionary clustering, which makes it a suitable candidate for applications involving dynamic systems, such as social network analysis, customer segmentation, and anomaly detection. By applying

This work is licensed under a Creative Commons Attribution 4.0 International License.

AEABC to evolutionary clustering, this paper contributes to expanding the range of applications for swarm intelligence algorithms in handling real-world, dynamic data challenges.

KEYWORDS

Evolutionary Clustering; Optimization Algorithms; Metaheuristics; Swarm Intelligence; Unsupervised Learning.

1. INTRODUCTION

Artificial intelligence AI has wide range of application especially in robotics (Ghafil, László and Jármai, 2019), object detection (Alaa, Hussein and Al-libawy, 2024) control systems (Albedran and Jármai, 2023), and predictive systems in engineering (Alsamia, Albedran and Mahmood, 2023). This domain dominates numerical analysis in almost all fields nowadays. One of the branches of AI is machine learning where machines are learning decision-making from predefined manners. In turn, clustering or unsupervised learning is one of the main domains of machine learning that has exceptional applications in industry. Unsupervised learning (Shen et al., 2022) has obtained a great deal of attention in recent years as a result of the increasing availability of large and complex datasets. Unsupervised learning aims to discover hidden patterns and structures in data without the need for explicit supervision or guidance from labeled data. Optimization algorithms (Abasi, Khader and Al-Betar, 2022;Alsamia, Ibrahim and Ghafil, 2021), which are widely used in various fields, can also be applied to unsupervised learning problems. In this paper, we presented a comparative study of seven optimization algorithms, namely, particle swarm optimization, dynamic differential annealed optimization, artificial bee colony, differential evolution, genetic algorithm, cuckoo search, and fertilization optimization algorithm, for the purpose of unsupervised learning on an unlabeled dataset. In doing so, the performance of these algorithms on a real-world dataset was evaluated and statistical analysis was run to compare their efficiency. The obtained results of this study can help researchers and practitioners choose the right optimization algorithm for their unsupervised learning applications. The application of some optimization algorithms for unsupervised learning on unlabeled datasets was also explored in this study since unsupervised learning is an integral part of machine learning, aiming to find data patterns without using labels. This method is of use where it is not easy, cost-effective, or time-efficient to label data. Some unsupervised learning tasks for which optimization algorithms are used include clustering and dimensionality reduction (Ghafil and Jármai, 2019). These algorithms can be used to adjust various objective functions that measure similarity or difference between data points. Optimization algorithms (Alkafaji and Swadi, 2024; Ghafil and Jármai, 2020b) also serve as smart means to boost the performance of machine learning algorithms. As mentioned, this current study was performed to compare the performance of seven metaheuristics, namely particle swarm optimization, genetic algorithm, artificial bee colony, differential evolution, dynamic differential annealed optimization, cuckoo search, and fertilization optimization on an unlabeled dataset. The reasons to choose these algorithms were their efficiency, popularity, and proven ability to solve optimization problems. In this regard, this study sought to provide

insights into the relative performance of these algorithms for unsupervised learning tasks. The algorithms in this study were evaluated in terms of their ability to minimize the objective function and suggest an appropriate representation of the data. Moreover, the robustness and convergence properties of these algorithms were analyzed.

The remainder of this paper is organized as follows: Section 2 describes the dataset and the metaheuristic algorithms used for clustering, including the proposed AEABC algorithm. Section 3 explains the evolutionary clustering approach and the objective function used to optimize cluster formation. Section 4 presents a detailed discussion of the results where AEABC's performance is compared with other optimization techniques. Finally, Section 5 concludes the study by summarizing key findings, discussing limitations, and outlining potential future research directions.

2. RELATED WORKS

Clustering is the problem of sorting observed data into distinguished groups that can have rate of similarities among their population, and have rate of dissimilarities among each other (Majhi and Biswal, 2018). Clustering is useful to discover the nature of the structures in a dataset, and it is performed by different methods that that share the same key factor which is the distance between elements of the data. The distance term in clustering refer to different criterion that depend on the nature of the problem. The distance may refer to Euclidean or non-Euclidean like Hamming distance (Gao, Yue and Li, 2021). Clustering algorithms can be classified into partitional clustering (Nanda and Panda, 2014) , hierarchical methods (Ferreira and Hitchcock, 2009), grid based methods (Cheng, Wang and Batista, 2018), overlapping clustering (Das, Abraham and Konar, 2009) and density based methods (Campello et al., 2020), and each method has its advantages and disadvantages. In many real-life dataset, clusters have an overlapping nature, and that was tackled by fuzzy clustering (Sato and Jain, 2006). Metaheuristic clustering was an interesting manner to cluster data in separated groups, and type of clustering is widely existing in the literature (Nanda and Panda, 2014). Metaheuristics are excellent optimization techniques to find an unknown solution in unsupervised dataset or generally search space. Optimization algorithms, or metaheuristics (Ghafil H.N., 2020) are very efficient techniques to find local or global solutions for wide range of applications in wide range of fields of science. Metaheuristics, which are relay on random number generators in their search engines, are the mostly used in vast and diverse applications.

3. MATERIALS AND METHODS

3.1. Dataset

An unsupervised dataset can be defined as a dataset lacking any predefined labels or categories.

The major uses of unsupervised learning techniques are in discovering relationships, structures, and patterns within the data. Some uses include

1. Customer Segmentation: run unsupervised clustering to group algorithms can be used for clustering customers with regard to based on their behavior, preferences, behavior, preference, or demographics using unsupervised for the purpose of running clustering in businesses. This helps with creating personalized marketing campaigns for each group.
2. Image and Video Analysis: The algorithm is capable of classifying and analyzing images and videos considering visual features like shapes, colors, and textures. This helps with content-based image retrieval, video surveillance, and object recognition.
3. Anomaly Detection: The algorithm can identify odd data patterns that may facilitate the recognition of 'fraud', 'errors', or other abnormal performance. This involving help with fraud detection, cybersecurity, and fault diagnosis.
4. Natural Language Processing: As the name suggests, algorithm can sort text data and make the analysis easy in terms of semantic or syntactic features. This helps with topic modeling, sentiment analysis, and text clustering.
5. Bioinformatics: Biological data like DNA sequences or gene expression profiles can be analyzed and classified using unsupervised learning. These algorithms can of use in gene function prediction, drug discovery, and disease diagnosis.

Supervised learning relies heavily on labeled datasets, which train the algorithm. Therefore, there is a corresponding label for each data point or target variable in the dataset. The algorithm realizes how to predict the target variable from the input data by considering the labeled examples as a guide. Image classification datasets, sentiment analysis datasets, and speech recognition datasets are common examples of labeled datasets. What constitutes a dataset is a set of features plus a label (or target variable) for each dataset instance. Features describe each instance based on its attributes or characteristics, while the label shows the output or the class used to predict or categorize that instance. Labeled datasets are used in supervised learning to train machine learning models to predict or classify the label for new instances using their features. The clustering output supports pattern recognition by identifying data patterns and structures. It can used for classifying in the pre-processing step to put similar data points together before using any classification algorithm. To provide an example, the stroke dataset clustering can classify patients into different risk categories considering their health attributes. This grouping can be narrowed down further and lead to a classification model that predicts stroke risk for new patients. In this study, an unlabeled dataset (Yarpiz, no date) was used to compare seven metaheuristic algorithms. Fig. 1 shows the data.

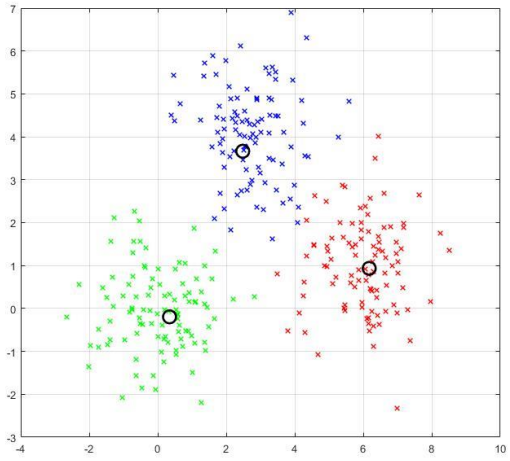

Fig. 1. Unlabeled dataset of three clusters

The dataset used in this study consists of a set of points represented in a two-dimensional Cartesian space, where each data point is defined by its x and y coordinates. The dataset is structured in a tabular format, with the first column representing the x-coordinates and the second column representing the y-coordinates of the points. These points form clusters of varying densities and distributions, providing a suitable test environment for evaluating the effectiveness of the AEABC algorithm in evolutionary clustering. The dataset presents challenges in terms of dynamic cluster formations, making it an appropriate benchmark for assessing the adaptability and robustness of AEABC in handling spatially distributed data. By optimizing cluster centers based on the spatial distribution of points, the algorithm aims to minimize intra-cluster distances while maintaining a well-structured grouping of data points. The dataset can be downloaded online:
https://docs.google.com/spreadsheets/d/1XLRxsyWYleIzDUUeknrwH75TUmRsZ3vF/edit?usp=sharing&ouid=101969985739565212436&rtpof=true&sd=true

3.2. Metaheuristics

3.2.1. Particle swarm optimization

One computational optimization technique to gain an optimal solution of a problem is called particle swarm optimization (PSO) (Eberhart and Kennedy, 1995). The operation of this algorithm relies on letting the population of particles moving around in a landscape to arrive at the best solution. A candidate solution is defined as each particle, and its position within the search space stands for a solution. The velocity of the particles is adjusted during the motion over the search space considering their own best position and the best global position by the swarm. The reasons to recommend PSO for use in various optimization problems is bound to it being easy to implement and simple yet the convergence properties are good. PSO is helpful

for continuous, discrete problems, and complex optimization problems. By finding the best number of clusters and their positions for the centroids, PSO can improve clustering analysis.

3.2.2. Dynamic differential annealed optimization

Another algorithm to solve complex optimization problems is called Dynamic Differential Annealed Optimization (DDAO) (Ghafil and Jármai, 2020a;Rajesh et al., 2022). DDAO functions on the basis of the simulated annealing principles. Therefore, a search is defined by a temperature parameter for acceptance of new solutions. The temperature parameter of this algorithm is dynamically adjusted, which means it is adaptively changing to the running process of the optimization algorithm (Hussien et al., 2024), and so the balance between exploration and exploitation within a space helps to provide better performance to jump from local optima. As a result, DDAO is popular for resolving optimization problems like the ones that include function optimization, parameter tuning, and feature selection. Moreover, it can be used effectively and efficiently in obtaining high quality solutions in complex search spaces.

3.2.3. Artificial Bee Colony

Inspired by the foraging behavior of honey bees, called artificial bee colony (ABC) is a swarm intelligence algorithm (Ghafil and Jármai, no date). The processing of the algorithm begins with the initial random generation of an initial population of food sources, indicating potential solutions to the problem of optimization (Albedran, Alsamia and Koch, 2025). ABC categorizes the population into three groups, namely employed bees, onlooker bees, and scout bees. In the first population, the food most recently found by the current neighboring of a food source is searched for and the quality of the solutions is evaluated. Second population observing the dances of employed bees chooses a food source according to the solution quality. The final population including the scout bees are responsible for seeking new food sources by exploring unexplored regions of the search space. The ABC algorithm balances the exploration and exploitation utilizing a probabilistic selection method, which facilitates to converge to the global optimum. This algorithm can be effectively used for feature selection, image processing, and clustering. In addition, a set of candidate solutions can be used as the initial cluster centroids, and it can be employed for clustering. The evaluation of the quality of each solution at this step is made using a fitness function. Local search and global search are then used to choose the best solutions to generate new candidate solutions. ABC continues functioning, and an optimized clustering solution is gained.

3.2.4. Differential Evolution

One optimization algorithm designed to solve problems related to non-linearity and non-

differentiability is named Differential Evolution (DE) (Song et al., 2024; Alsamia, Ibrahim and Ghafil, 2021). This algorithm, considered as a class of evolutionary algorithms, functions on the basis of population (Al-Kadimy and Hamza, 2014). It works based on "individuals" that are a population of candidate solutions and follows a simple mutation strategy. The first stage in DE is initializing a population of random solutions, followed by their iterative improvement through a combination of crossover and mutation operators. In the crossover operation, the information of two or more parent individuals is combined to generate a new solution. The mutation operator, however, involves small random changes to maintain diversity within the population. Solution fitness is gauged considering an objective function, aiming to optimize. Those individuals that are ranked high in fitness are invited to the next generation, and the rest are eliminated. This algorithm is popular for being simple, robust, and efficient in resolving complex optimization problems. Exchanging information between individuals in the population is the basis of DE to maintain a population of candidate solutions and improve them iteratively. While being used for clustering, the algorithm is used to find the configuration that optimizes a given objective function as the candidate solutions act as different cluster configurations.

3.2.5. Genetic Algorithm

Inspired by the process of natural selection and genetics, genetic algorithm (GA) (Ghafil and Jármai, 2020c) is a metaheuristic search algorithm, which imitates the natural process of evolution to find a solution for optimization problems (Fadhil, Abed and Jasim, 2021). GA functionality is based on chromosomes, which are a population of potential solutions, evolving over generations using genetic operators like selection, crossover, and mutation. The initial stage in GA involves a random generation of the initial population of chromosomes. Fitness is a criterion considered for the evaluation of each chromosome, indicating the solution quality. Chromosomes with higher levels of fitness are highly likely to be chosen for the next generation. The selection is performed considering fittest survival, where the best solutions are maintained and regenerated. Crossover operation involves the exchange of genetic information related to two selected chromosomes in order to generate new offspring chromosomes. On the other hand, mutation operation embraces random changes in the chromosomes to gain new solutions. To create a new generation of chromosomes, the mentioned genetic operators are applied to the population, which is again subjected to the fitness evaluation. The process lasts until all the termination criteria (reaching a maximum number of generations or achieving a satisfactory fitness level) are met. GA is commonly used in engineering, finance, and biology fields. It is efficient in searching a large solution space and arriving at near-optimal solutions

to complex problems. Moreover, this algorithm can have a crucial role in optimizing clustering process through determining the appropriate fitness function that evaluates the quality of the clustering solution. Criteria, such as intra-cluster distance, inter-cluster distance, compactness, and separation can determine fitness function.

3.2.6. Cuckoo Search

Inspired by the unique reproduction behavior of cuckoo (that is laying eggs in the neighboring nest), Cuckoo Search (Ghafil and Jármai, 2020c) was introduced in 2009 as an algorithm for optimizing problems. It can be used for feature selection, clustering, and image processing through randomly generating a set of solutions, named cuckoo eggs. In the next step, eggs are laid in a randomly chosen nest, matching a potential solution. An objective function applies to assess the quality of each solution. Afterward, the best solutions are maintained for the next iteration. While iterating, Cuckoo Search provides each cuckoo egg with a chance to be replaced by a new egg generated using a Lévy flight, a random walk with a long-tailed probability distribution. Named cuckoo egg discovery, this process can enhance diversity in the population and reduce the risk of getting stuck in local optima. Of note, Cuckoo Search uses a nest destruction strategy that involves the replacement of solutions in the worst nests with new ones generated by a random walk. This strategy encourages the exploration of new areas within the search space and avoids being stuck in local optima. Cuckoo Search lasts until the termination criterion is met. The output of this algorithm is the best solution found during the iterations.

3.2.7. Adaptive Exploration Artificial Bee Colony

An enhanced version of Artificial Bee Colony algorithm is called Adaptive Exploration Artificial Bee Colony (AEABC) algorithm (Shaymaa Alsamia , Edina Koch , Hazim Albedran, 2024). The primary aim of designing AEABC was to enhance search efficiency through improving the exploration and exploitation balance during the optimization process. In doing so, this algorithm defines a new probability parameter based on distance, which helps bees (candidate solutions) make adaptive changes in their spatial relationships within the population. Therefore, this algorithm is suitable for finding optimal groupings in clustering problems where both global exploration and local refinement are of significant importance.

AEABC works based on the basics of the ABC algorithm, that is employed bees, onlooker bees, and scout bees. The roles of these three types are bees are briefly reviewed below.

1. Employed Bees: Bees in this group are assigned to a specific food source named a candidate solution and try to explore the neighborhood to find better positions. Distance-based probability parameter can affect the local search, meaning that the parameter can define the possibility of

moving to neighboring solutions considering their proximity. As a result, adaptive exploration is enhanced, and employed bees are encouraged to move to a promising solution when they are nearby. Consequently, the local search phase, which is an integral aspect of clustering, is improved.

2. Onlooker Bees: These bees are responsible for choosing food sources by observing the waggle dances of employed bees. Onlooker bees take into account fitness and proximity while making a decision. Distance from other solutions is a factor that affects the decision of selecting a food source in AEABC, which enhances the balance between exploration and exploitation. Regarding clustering, well-performing clusters can be refined while different areas of the solution space are still being explored.

3. Scout Bees: Scout bees appear when several iterations cannot help a fit solution (food source) and the associated employed bees leave. In this phase, scout bees define new random solutions into the population aiming to provide a chance for exploration of untouched regions of the search space. This approach becomes important when we are dealing with evolutionary clustering as the search space may entail many local optima, and maintaining diversity prevents the algorithm from converging prematurely.

The main feature of AEABC is its distance-based probability parameter P_d, which is designed to adapt the behavior of the algorithm with regard to the spatial distribution of solutions. This parameter is calculated using the Euclidean distance between solutions

$$P_d = e^{\frac{-1}{d}} \quad (1)$$

Where,

d denotes the Euclidean distance between the current solution and a randomly selected solution in the population. The distance parameter P_d indicates the likelihood of shifting the current solution based on its proximity to other solutions. When solutions are far apart, P_d is close to 1, resulting in a low probability of shifting the solution. Exploration is enhanced in this algorithm through keeping distant solutions stable, leading to the search for broader areas of the solution space. When solutions are in neighborhood, P_d approaches 0, increasing the probability of moving to nearby solutions, which in turn improves exploitation by focusing on refining clusters around high-quality solutions. Therefore, AEABC is suitable for clustering tasks since it can widely explore potential cluster centers in initial stages and stay focused on refining clusters as convergence progresses.

4. EVOLUTIONARY CLUSTERING

Similar to other algorithms, the objective function in the AEABC algorithm is designed to

minimize the Sum of Within-Cluster Distances (WCD) for clustering. WCD measures the cluster compactness through assessing by calculating how close each data point is to its corresponding cluster center. Therefore, well-defined clusters are shaped as a result of reduced intra-cluster variability. Two inputs should be given to the objective function, one is m, indicating the dataset points expected to be clustered and the other is X representing a set of cluster centers (solutions), each center is being represented by its coordinates (x,y). As three clusters are available, the size of X is 3×2=6 variables. The objective function measures the distance matrix d between each data point in m and each cluster center in X. In the next step, the algorithm finds the minimum distance for each data point and the index of the closest cluster center, leading to the effective correspondence of each data point to the nearest cluster. The Sum of Within-Cluster Distance (WCD) measured as the sum of all minimum distance values, determines the total intra-cluster distance across all clusters, which the optimization algorithm seeks to minimize. When the points are closer to their respective cluster centers, WCD is lower and therefore, the clusters are more compact. The algorithm should also have a structure that groups data points by cluster, ensuring that each point is appropriately associated with its cluster center. Although this information is beneficial for visualization of clustering outcome, it has no direct effect on the objective function value or WCD. Regarding optimization, the optimization process is directed by this objective function when solutions with minimum intra-cluster distances are suggested and consequently, clustering quality is enhanced. The role of optimization algorithm is to iteratively adjust the cluster centers (the variable X in the function) to reduce WCD, refining the clustering outcome with each generation. Mathematically, the objective function can be defined as:

$$WCD = \sum_{i=1}^{N} min\left(m_i - X_j\right) \quad (2)$$

where, N denotes the total number of data points, m_i is i-th data point, X_j refers to the center of the j-th cluster. The AEABC algorithm aims to reach the minimum WCD in order to improve clustering compactness and therefore more cohesive clusters.

Clustering applications can benefit from this objective function when they need compact clusters since it can directly reduce the total distance between points and their nearest cluster centers. As a result, optimization algorithm continuously refines the cluster centers toward an optimal configuration to efficiently optimize clustering solutions. Fig. 2 shows the flowchart that represent general evolutionary clustering procedure that can be followed by any optimizer.

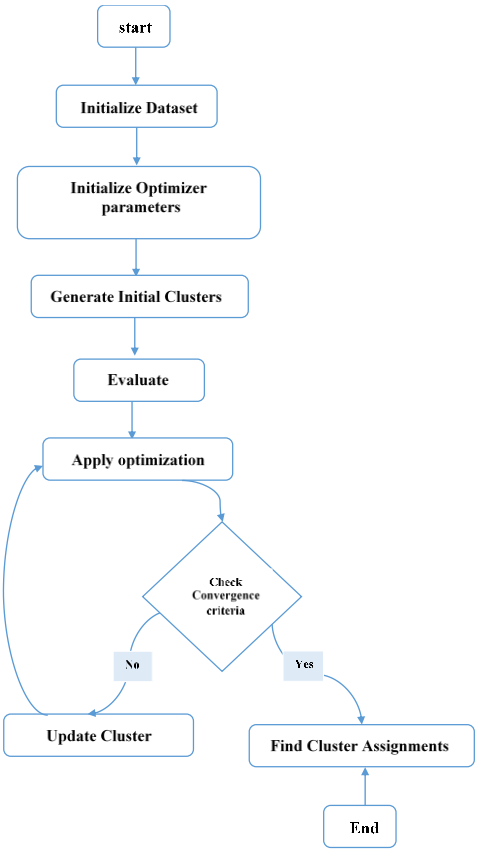

Fig. 2. Evolutionary clustering technique

5. RESULTS

This section presents the obtained results of AEABC algorithm on evolutionary clustering. The findings are then compared with those of other metaheuristic algorithms, namely PSO, DDAO, ABC, DE, GA, and CS. Table 1 provides statistical comparisons based on the best, worst, average, and standard deviation values achieved by each algorithm over multiple runs. Fig.3 shows the convergence behavior of AEABC across 150 iterations. The details of the comparative study can be summarized as follows:

1. Compared to other algorithms, the lowest best value in AEABC was achieved at 3.7060×10^2, which indicates its superiority in finding high-quality clustering solutions. PSO could be ranked as the second best algorithm (a best value of 3.7551×10^2), followed by GA with 3.8386×10^2. As indicated, AEABC is more effective, compared to its counterparts in achieving optimal solutions for the benchmark clustering task.
2. Regarding the worst-case performance, AEABC indicated the worst value of 3.9270×10^2, which was close to the optimal values across runs. Other algorithms, such as DDAO and ABC showed significantly higher worst values of 6.5242×10^2 and 6.3676×10^2, respectively. It can

be concluded that AEABC is more robust and generates fewer variations in performance and produces high-quality solutions.

3. Considering the average value, the obtained results for AEABC was 3.7552×10^2, which was again better than other algorithms in terms of mean performance. PSO with an average of 3.9243×10^2 was in second place, followed by DE (5.1050×10^2) and CS (5.3049×10^2). The average performance of AEABC confirms its capability to consistently find solutions close to the global optimum.

4. With regard to standard deviation, AEABC had the lowest value (4.7311), which indicates a high level of consistency across iterations. As a result, solutions found by AEABC are tightly clustered around the mean, which shows a reliable convergence behavior. Higher standard deviation values in other algorithms like DDAO (5.1841×10) and ABC (5.2359×10) are indicative of higher variation in their outcomes.

AEABC convergence curve shows rapid progress toward the optimal solution within the first 50 iterations before gradual stabilization as shown in Fig.3, indicating its efficient search behavior. In the initial iterations, AEABC quickly explores the solution space (emphasizing exploration) and then gradually focuses on fine-tuning the solutions (enhancing exploitation) as convergence is about to occur. A stable solution is gained after almost 100 iterations. Minimal changes in the objective function value while running the iteration indicate the efficiency of AEABC in reaching near-optimal solutions. The findings highlighted the higher efficiency of AEABC in best, worst, and average values, as well as standard deviation when compared to other algorithms. The findings also indicated higher convergence behavior of AEABC as a result of the effective balance between exploration and exploitation, thereby high-quality clustering solutions with significant consistency as revealed in Fig. 3. It can be inferred that AEABC is an appropriate choice for tasks with high levels of accuracy and stability due to its effectiveness and robustness.

Table 1 Statistical results of AEABC and other metaheuristics on evolutionary clustering

Algorithm	Best	Worst	Average	Standard deviation
PSO	3.7551E+02	5.1033E+02	3.9243E+02	2.4880E+01
DDAO	4.3026E+02	6.5242E+02	5.7386E+02	5.1841E+01
ABC	4.0415E+02	6.3676E+02	4.9890E+02	5.2359E+01
DE	4.0626E+02	6.3799E+02	5.1050E+02	5.1510E+01
GA	3.8386E+02	5.8501E+02	4.7704E+02	4.0894E+01
CS	4.3642E+02	6.2544E+02	5.3049E+02	4.7011E+02
AEABC	3.7060E+02	3.9270E+02	3.7552E+02	4.7311E+00

A comparative analysis of the metaheuristic algorithms used in this study is presented in Table2. This table highlights the key advantages and disadvantages of each method, their primary research applications, and their best performance in terms of the WCD. The results confirm that

AEABC achieved the lowest WCD value, demonstrating its superior clustering efficiency compared to other algorithms.

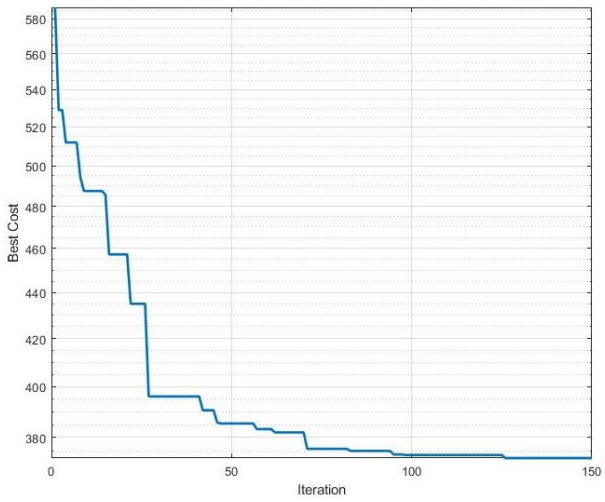

Fig.3 AEABC convergence curve

Table 2 Comparison of metaheuristic algorithms used in this study

Algorithm	Advantages	Disadvantages	Key Research Applications	Best WCD Metric Value (Lower is Better)
PSO	Fast convergence, easy to implement, good for continuous optimization	Can get trapped in local minima, struggles with high-dimensional spaces	Feature selection, clustering, function optimization	375.51
DDAO	Adaptive mechanism to escape local minima, good for parameter tuning	High computational cost, sensitive to initial temperature setting	Feature selection, neural network training	430.26
ABC	Balances exploration and exploitation, flexible for various applications	Slow convergence in high-dimensional search spaces	Image processing, clustering, routing problems	404.15
DE	Good global search ability, simple mutation strategy	Slow in fine-tuning solutions, requires careful parameter tuning	Signal processing, bioinformatics	406.26
GA	Well-suited for discrete problems, good diversity preservation	High computational cost, requires careful parameter tuning	Robotics, scheduling, machine learning	383.86
CS	Efficient in global optimization, uses Lévy flight for exploration	Premature convergence in some cases, requires tuning of parameters	Feature selection, anomaly detection	436.42

Algorithm	Advantages	Disadvantages	Key Research Applications	Best WCD Metric Value (Lower is Better)
AEABC	Improved exploration-exploitation balance, adaptive distance-based search, high stability	Requires additional computation for distance-based adaptation	Evolutionary clustering, dynamic data analysis	370.60 (Best)

6. CONCLUSION

This study aimed to introduce Adaptive Exploration Artificial Bee Colony (AEABC) algorithm as a new solution for evolutionary clustering since it can improve consistency as well as clustering quality. AEABC can run the search process effectively as it uses a distance-based probability parameter. This parameter helps AEABC dynamically adapt its behavior based on the spatial distribution of solutions. When trying to refine solutions in appropriate regions, AEABC does not converge prematurely, which leads to the clustering results being accurate and consistent. As mentioned in the result section of the article, AEABC could outperform other metaheuristic algorithms (PSO, DDAO, ABC, DE, GA, and CS) in terms of a series of metrics. Firstly, AEABC produced lower performance error rates than competing algorithms in all three scenarios representing best case, worst case and average case performance. Moreover, AEABC may be considered more reliable than other algorithms since the results exhibited only little variations between different runs. As a good alternative to traditional algorithms, AEABC could be thought of a good means for dynamic environments, where data evolves over time. AEABC is appropriate since its adaptation is also suitable for fast continuous clustering adjustments like social network analysis, anomaly detection or customer segmentation. Therefore, expanding the applicability of AEABC to larger and more complex datasets can be the focus of future studies. These studies can integrate adaptive parameter tuning or hybrid strategies to further enhance the flexibility and robustness of AEABC. While AEABC has shown strong performance, we acknowledge that its evaluation in this study was conducted on a single dataset. Clustering performance can be problem-dependent, and results may vary with different data characteristics. The dataset used in this study was selected because it presents a challenging, dynamic clustering scenario, effectively testing the adaptability of AEABC. However, further validation on diverse datasets is necessary to generalize its effectiveness. As a direction for future work, we plan to extend the experimental evaluation by testing AEABC on multiple benchmark datasets, including high-dimensional clustering problems and real-world applications such as bioinformatics and financial analytics. Additionally, exploring hybrid approaches that combine AEABC with other optimization techniques could further

enhance its adaptability to dynamic clustering environments. Future research could also investigate adaptive parameter tuning strategies to refine the balance between exploration and exploitation dynamically. In conclusion, AEABC can considered a novel approach in the field of evolutionary clustering since it implements a mechanism, where exploration and exploitation are balanced. It allows the AEABC to provide high quality, stable and interpretable clustering solutions to dynamic data environments with high efficiency. The findings of the current study can help other researchers conduct much more work toward more adaptive clustering techniques, and generate new ways to obtain reliable and high performing clustering in complicated and dynamic data sets.

7. REFERENCES

Abasi, A., Khader, A. T. and Al-Betar, M. A. (2022) 'An improved multi-verse optimizer for text documents clustering', Kufa Journal of Engineering, 13(2), pp. 28–42.

Alaa, R., Hussein, E. and Al-libawy, H. (2024) 'Object detection algorithms implementation on embedded devices: Challenges and suggested solutions', Kufa Journal of Engineering, 15(3), pp. 148–169.

Albedran, H. and Jármai, K. (2023) 'Evolutionary control system of asymmetric quadcopter', International Review of Applied Sciences and Engineering, 14(3), pp. 374–382.

Albedran, H., Alsamia, S. and Koch, E. (2025) 'Flower fertilization optimization algorithm with application to adaptive controllers', Scientific Reports, 15(1), p. 6273. doi: 10.1038/s41598-025-89840-1.

Al-Kadimy, A. M. and Hamza, Z. A. H. (2014) 'OPTIMIZATION DESIGN OF GRANULAR ACTIVAED CARBON USING GENETIC ALGORITHM', Kufa Journal of Engineering, 6(1), pp. 45–56.

Alkafaji, E. and Swadi, H. L. (2024) 'MULTI-OBJECTIVE OPTIMIZATION OF TRAFFIC SIGNAL TIMINGS FOR MINIMIZING WAITING TIME, CO2 EMISSIONS, AND FUEL CONSUMPTION AT INTERSECTIONS', Kufa Journal of Engineering, 15(3), pp. 21–31.

Alsamia, S., Albedran, H. and Mahmood, M. S. (2023) 'Contamination depth prediction in sandy soils using fuzzy rule-based expert system', International Review of Applied Sciences and Engineering, 14(1), pp. 87–99.

Alsamia, S., Ibrahim, D. S. and Ghafil, H. N. (2021) 'Optimization of drilling performance using various metaheuristics', Pollack Periodica.

Campello, R. J. G. B. et al. (2020) 'Density-based clustering', Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(2), p. e1343.

Cheng, W., Wang, W. and Batista, S. (2018) 'Grid-based clustering', in Data clustering. Chapman and Hall/CRC, pp. 128–148.

Das, S., Abraham, A. and Konar, A. (2009) Metaheuristic clustering. Springer.

Eberhart, R. and Kennedy, J. (1995) 'Particle swarm optimization', in Proceedings of the IEEE international conference on neural networks. Citeseer, pp. 1942–1948.

Fadhil, G., Abed, I. and Jasim, R. (2021) 'Genetic algorithm utilization to fine tune the parameters of PID controller', Kufa Journal of Engineering, 12(2), pp. 1–12.

Ferreira, L. and Hitchcock, D. B. (2009) 'A comparison of hierarchical methods for clustering functional data', Communications in Statistics-Simulation and Computation, 38(9), pp. 1925–1949.

Gao, Y., Yue, Q. and Li, F. (2021) 'The minimum Hamming distances of repeated-root cyclic codes of length $ $6 p^ s $$6 ps and their MDS codes', Journal of Applied Mathematics and Computing, 65(1), pp. 107–123.

Ghafil H.N., J. K. (2020) 'Optimization', in Optimization for Robot Modelling with MATLAB. first edit. Cham: Springer. doi: https://doi.org/10.1007/978-3-030-40410-9_2.

Ghafil, H. and Jármai, K. (no date) 'Comparative study of particle swarm optimization and artificial bee colony algorithms', in Multiscience XXXII. MicroCAD International Multidisciplinary Scientific Conference, Miskolc-Egyetemváros, Hungary. Miskolc: Miskolci Egyetem, pp. 1–6. Available at: http://real.mtak.hu/84332/1/D1_Hazim_Nasir_Ghafil.pdf.

Ghafil, H. N. and Jármai, K. (2019) 'Optimum dynamic analysis of a robot arm using flower pollination algorithm', in Advances and Trends in Engineering Sciences and Technologies III-Proceedings of the 3rd International Conference on Engineering Sciences and Technologies, ESaT 2018.

Ghafil, H. N. and Jármai, K. (2020a) 'Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications', Applied Soft Computing, p. 106392.

Ghafil, H. N. and Jármai, K. (2020b) 'Optimization Algorithms for Inverse Kinematics of Robots with MATLAB Source Code', in Vehicle and Automotive Engineering. Springer, pp. 468–477.

Ghafil, H. N. and Jármai, K. (2020c) Optimization for Robot Modelling with MATLAB. Springer Nature.

Ghafil, H. N., László, K. and Jármai, K. (2019) 'Investigating three learning algorithms of a neural networks during inverse kinematics of robots', in Solutions for Sustainable Development. CRC Press, pp. 33–40.

Hussien, A. G. et al. (2024) 'An enhanced dynamic differential annealed algorithm for global optimization and feature selection', Journal of Computational Design and Engineering, 11(1), pp. 49–72.

Majhi, S. K. and Biswal, S. (2018) 'Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer', Karbala International Journal of Modern Science, 4(4), pp. 347–360.

Nanda, S. J. and Panda, G. (2014) 'A survey on nature inspired metaheuristic algorithms for partitional clustering', Swarm and Evolutionary computation, 16, pp. 1–18.

Rajesh, P. et al. (2022) 'Leveraging a dynamic differential annealed optimization and recalling enhanced recurrent neural network for maximum power point tracking in wind energy conversion system', Technology and Economics of Smart Grids and Sustainable Energy, 7(1), p. 19.

Sato, M. and Jain, L. C. (2006) 'Innovations in Fuzzy Clustering'. Springer, Heidelberg.

Shaymaa Alsamia , Edina Koch , Hazim Albedran, R. R. (2024) 'Adaptive Exploration Artificial Bee Colony for Mathematical Optimization', AI, 5(4). doi: https://doi.org/10.3390/ai5040109.

Shen, Q. et al. (2022) 'Unsupervised learning of accurate Siamese tracking', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8101–8110.

Song, H. et al. (2024) 'Hybrid algorithm of differential evolution and flower pollination for global optimization problems', Expert Systems with Applications, 237, p. 121402.

Yarpiz (no date) Evolutionary Data Clustering. Available at: https://yarpiz.com/64/ypml101-evolutionary-clustering (Accessed: 16 April 2023).