Computer
& Robotics

An Adaptive Fuzzy-based Algorithm with Informed Mutation for Semi-supervised Clustering

Atiyeh Taghizabet[a], Amineh Amini[a,*], Jafar Tanha[b], Javad Mohammadzadeh[a]

[a] *Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran*

[b] *Department of Computer and Electrical Engineering Department, Tabriz University, Tabriz, Iran*

Received 1 August 2024, Accepted 22 December 2024

Abstract

A technique that blends semi-supervised learning and clustering is called semi-supervised clustering. In semi-supervised clustering, using labeled data can significantly enrich the results quality. However, clustering is an NP-hard and multi-objective problem that necessitates the utilization of multi-objective meta-heuristic algorithms to generate satisfactory solutions. The main challenge associated with these algorithms lies in their susceptibility to local optima and the manual adjustment of parameters. To overcome these issues, this study introduces an adaptive multi-objective cuckoo algorithm for semi-supervised clustering termed "Informed AdamCo," The motion coefficient parameter has been adjusted automatically using fuzzy control in this algorithm. The proposed algorithm can also increase the diversity of non-dominated solutions with a mutation capacity based on preference information to overcome the local optima trap problem. The adaptive adjustment of the migration coefficient parameter using fuzzy control considers three practical inputs simultaneously: Iteration, Error, and Distance. To evaluate the proposed approach, 10 UCI and six synthetic datasets, as well as the KDD Cup 1999 dataset were used in the experiments. Statistical and numerical analyses demonstrate the superiority of Informed AdamCo over the other five algorithms compared.

Keywords: clustering, semi-supervised, multi-objective, adaptive, swarm intelligence, fuzzy control, informed mutation

1. Introduction

Semi-supervised clustering plays an important function in machine learning. It has been extensively utilized as a pre-processing method in many real-world situations, such as anomaly identification [1], analysis of gene expression patterns [2], image segmentation [3], to name but a few.

Considering the NP-hard and multi-objective nature of the clustering problem, applying multi-objective optimization algorithms is favored for addressing the challenge. Multi-bjective algorithms, compared to single-objective ones, can also provide high-robustness results. Yet, both these algorithms (single-objective and multi-objective) suffer from the two challenges of manual parameter adjustment and getting stuck in the local optimum.

Semi-supervised clustering algorithms can be categorized into three fundamental sorts: distance-based, search-based, and hybrid approaches [4]. Within the distance-based approach, an existing clustering strategy is utilized,

a.taghizabet@gmail.com

and based on supervised information the distance measure is adjusted. This shift aims to decrease the distance between data points with identical clusters (Must-Link constraints: ML) and increase that distance for data points with separated clusters (Cannot-Link constraints: CL). [5]. In essence, the distance measure exploiting the prior knowledge (ML and CL) is parameterized [6]. However, it is important to note that the modified distance measure in the distance-based method may not always yield accurate results. For example, two samples with a Must-Link constraint may still be far apart and end up in different clusters. Some studies that have employed this method include Sanodiya et al. (2019), Zhang et al. (2003), and [9].

On the other hand, search-based approaches to lead the clustering process and ameliorate results modify existing clustering algorithms by incorporating prior knowledge (labeled data or constraints). This is achieved by converting the clustering algorithm objective function in various ways. For instance, constrained COBWEB [10], optimizes clustering objectives making use of constraints in the incremental partitioning process. Seeded K-means [11] embodies labeled data to the conventional K-means in the initialization step of the algorithm. Similarly, constrained K-means [11] benefits from prior knowledge in the two steps of the K-means algorithm: initialization and assignment steps. Combined methods incorporate distance and constraint-based perspectives and benefit from both in the effective solving of this problem.

As mentioned above, in search-based methods, prior knowledge is added to traditional clustering to optimize clustering performance, with one such technique being the modeling of clustering as a multi-objective optimization problem [5]. This technique has been employed in the studies conducted by Akbarzadeh et al. (2019), Ebrahimi and Abadeh (2012), and Kumar Alok et.al (2017). However, the literature shows that single-objective methods are still utilized (Dong et al., 2016; Dong et al., 2017; Lai et al., 2020). Therefore, to address the problem of semi-supervised clustering, this paper presents a fuzzy-based method, called Informed AdamCo, for the adaptive adjustment of the migration coefficient parameter of the multi-objective cuckoo algorithm aimed at achieving the optimal solution of the semi-supervised clustering problem. To achieve this objective, the single-objective cuckoo algorithm is first converted into a multi-objective cuckoo algorithm. Afterwards, the labeled data is used in the initialization phase and training process of the proposed approach. Informed mutations are also included in the algorithm to increase the diversity of solutions and create more efficient populations. Finally, the motion coefficient parameter of the cuckoo algorithm is adjusted using a fuzzy method, which consists of three input components: algorithm iterations, population diversity, and error. The main results of this study are summarized as follows:

- Adaptive adjustment of motion coefficient parameter in multi-objective cuckoo Algorithm using a fuzzy control technique.
- Improving the variety of non-dominant solutions and providing better direction for the entire Cuckoo population to reach an optimal solution through the establishment of a mutation algorithm that relies on preference data.
- Applying two coefficients to the cuckoo movement pattern to improve the convergence in the multi-objective development of Rajabioun’s cuckoo Algorithm.

The remainder of the paper is organized as follows. Section 2 reviews related research on semi-supervised clustering, and Section 3 presents the proposed algorithm. Section 4 reviews the experiments and results on the dataset and compares them with state-of-the-art algorithms. Finally, Section 5 concludes the study.

2. Related Work

The literature in this domain has been categorized into two distinct groups: 1) Papers using the fuzzy control of parameters within swarm intelligence algorithms (FC-SI group). 2) Papers employing the fuzzy control of parameters within evolutionary algorithms (FC-EA group). In the following, we will further examine only the first group, as it directly pertains to the cuckoo algorithm, a swarm intelligence algorithm.

2.1. FC-SI Group

In this section, we review only the articles that focus on adjusting the parameters of swarm intelligence algorithms using fuzzy control techniques.

[22] applied a fuzzy control strategy to regulate the evaporation rate ρ within the Ant Colony Algorithm. The control mechanism involves the use of an error metric to precisely adjust the parameter values.

Mishra (2005) introduced a fuzzy control technique to modify the step size C in Bacterial Foraging Optimization (BFO) by taking into account the fitness value of the best individual in the population. Similarly, [24] suggested a fuzzy control system for adapting this parameter in BFO, utilizing the error in the objective function and the C parameter current value as input. In the study by [25], a fuzzy method was hired to regulate the α and γ parameters of the Firefly algorithm, where α controls the random walks step sizes and γ controls the visibility of fireflies (and hence search modes). The strategy involves integrating an adaptive weight that dynamically updates these parameters based on their previous values, with the fuzzy engine directly managing the control of this weight.

[26] introduced a fuzzy adjustment technique for the *G*, which has an important effect on agent acceleration in the Gravitational Search Algorithm (GSA). This approach utilizes the best fitness value of an individual, the iteration number without improvement, and the variance in population fitness to fine-tune the *G* parameter. Similarly, [27] proposed a GSA variant with a fuzzy adaptation of the gravitational constant. The best fitness achieved at iteration *T*, the iteration number without fitness improvement, and the fitness variance at iteration *T* are used as inputs to dynamically tune *G*. Additionally, Kumar et al. (2013) presented a fuzzy adjustment method for the GSA algorithm, which utilizes an individual normalized fitness and the gravitational constant *G* current value to adapt and optimize the constant. In an alternative context, [29] suggested a fuzzy control strategy for parameter α, responsible for regulating the speed of convergence. They incorporated three measures: population diversity, population progress, and the iteration number to modify the parameter value through increments or decrements.[30] introduced a fuzzy control method for the parameters C_1 (the cognitive coefficient) and C_2 (the social coefficient) within the Particle Swarm Optimization (PSO) algorithm. This technique relies on three factors: iteration number, diversity, and an error measure to dynamically adjust the parameters. Similarly, [31] demonstrated a fuzzy control approach for the parameters ω (the inertia weight), C_1 and C_2 in PSO. The control strategy makes use of the finest fitness value and the iteration number without fitness improvement to adjust the parameters. Furthermore, Olivas et al. (2013) put forward a fuzzy control method for the parameters C_1 and C_2 in PSO. This method utilizes a measure of population diversity and the number of iterations to modify the parameter values. Additionally, they introduced an altered fuzzy control approach where the input variables, population diversity, and the number of generations are treated as fuzzy. In their

research, the second model produced superior outcomes. [33] introduced a fuzzy control technique for the adjustment of C_1 and C_2 parameters in PSO using population fitness values.

[34] proposed two fuzzy models to tune visual parameters in the fish field algorithm. The first model, named Fuzzy Uniform Fish, takes as input the number of iterations and the proportion of individuals with the best current physical condition. The second model, called Fuzzy Autonomous Fish, uses the distance, fitness rating, and number of iterations of the best individual as inputs.

[35] introduced a method that incorporates a novel definition of population diversity in the improved Cuckoo algorithm. This fuzzy control approach is employed to regulate parameter changes for determining the location of sensor network nodes. In our study, fuzzy adaptation of migration coefficient parameter is employed to regulate changes in Cuckoo's algorithm for semi-supervised clustering problem. A new informed mutation algorithm has also been used to increase the diversity of non-dominated solutions and to overcome the local optima trap problem with a mutation capacity based on preference information.

As previously highlighted, clustering poses challenges as an NP-hard and multi-objective problem [36]. The computational complexity of clustering remains high, even for reasonably sized datasets. It is necessary to employ meta-heuristic algorithms, such as swarm intelligence algorithms, which have shown success in solving clustering problems. Multi-objective clustering, in particular, aims to optimize clustering based on multiple criteria instead of a single objective. By leveraging multi-objective optimization algorithms, clustering methods effectively reduce the search space and strive to optimize diverse and complementary criteria. These algorithms are preferred over their single-objective counterparts due to their ability to generate robust results. Our investigation into swarm intelligence algorithms and their potential capabilities led us to the conclusion that the Cuckoo algorithm could be beneficial in solving the semi-supervised clustering problem.

On the other hand, the studies referenced above indicate that despite the challenges associated with defining thresholds and fuzzy rules, fuzzy control techniques continue to be utilized due to their promising outcomes. In particular, these techniques have been integrated into various swarm intelligence algorithms such as PSO, BFO, ACO, AFSA, GSA, FA, CS, BA, and ABC. Among these algorithms, fuzzy control within the PSO stands out for its simplicity and efficiency, making it a viable option for testing and implementation in other algorithms. Therefore, we deem these approaches worthy of consideration.

Through comprehensive analysis and review of the literature, we have identified crucial factors influencing parameter adjustment, including the time/number of repetitions, the error rate of each cuckoo (indicating the disparity between the objective function of each cuckoo and the best cuckoo of each generation), and the Euclidean distance among every cuckoo and the finest cuckoo of every generation. Consequently, appropriate definitions for these factors were devised. Subsequently, those elements were hired as inputs for the fuzzy system to permit fuzzy adaptation of the migration coefficient. Lastly, the fuzzy system was integrated into the multi-objective cuckoo algorithm to enhance its effectiveness and performance.

3. Proposed Informed AdamCo for Semi-supervised Clustering

In this section, we first define semi-supervised clustering problems and research objectives. We then explain each step to reach the proposed method and present our algorithm.

3.1. Research Objectives and Problem Definition

Clustering serves the purpose of classifying unmarked samples into distinct classes or clusters, based on their similarity [37]. In semi-supervised algorithms, there exists a combination of labeled data $X_l = (x_1.x_2.\dots.x_l)$ with labels $\{1,\dots,k\}$ and unlabeled data $X_u = (x_{l+1}.x_{l+2}.\dots.x_{l+u})$ with unknown labels, where the number of labeled instances (l) is significantly smaller than the number of unlabeled instances (u). Both datasets, labeled and unlabeled, are independently sampled from the same data distribution. The objective of semi-supervised methods is to utilize the available labeled data to guide the algorithm towards achieving improved outcomes.

- Our proposed framework, depicted in Fig. 1, presents an adaptive multi-objective method to solve the semi-supervised clustering problem. Following an examination of various techniques, we concluded that meta-heuristic algorithms are more suitable for solving clustering problems given their NP-hard nature.
- On the other hand, clustering is a multi-objective challenge, and different and complementary objectives must be considered simultaneously to narrow the search space and obtain a better result. Hence, the concept of transforming the single-objective cuckoo into its multi-objective counterpart was investigated. Concurrently, the initialization step of the algorithm utilized labeled data as cluster centers. To increase the diversity of non-dominated solutions and efficiently steer the entire Cuckoo population toward a superior solution, a mutation algorithm grounded on partiality information was implemented.
- Then, to achieve better results, we investigated the challenges and practical considerations in the performance of such algorithms. The challenges encompassed issues such as entrapment in local optima and the need for manual parameters. In this regard, the existing methods for automatic parameter adaptation were studied, and their practical components were consequently extracted. Finally, the extracted features were applied through a fuzzy system design.

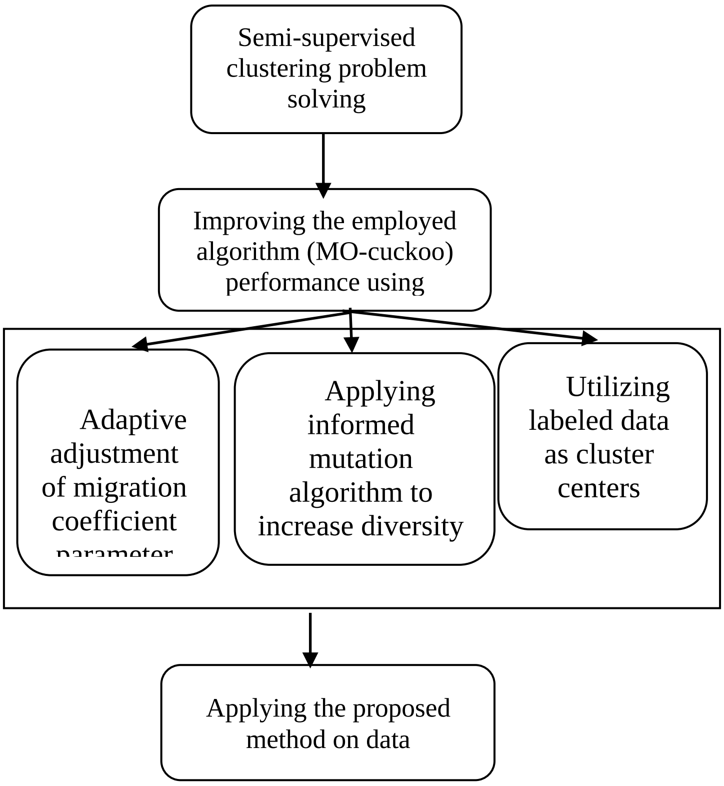

Semi-supervised clustering problem solving
Improving the employed algorithm (MO-cuckoo) performance using
Adaptive adjustment of migration coefficient parameter
Applying informed mutation algorithm to increase diversity
Utilizing labeled data as cluster centers
Applying the proposed method on data

Fig. 1. The framework to achieve informed fuzzy adaptation based multi-objective method for semi-supervised clustering problem.

3.2. The multi-objective Cuckoo algorithm

The multi-objective cuckoo algorithm is a generalization of the single-objective cuckoo algorithm to solve multi-objective problems. However, several adaptations have been incorporated as follows:

Determining the best cuckoo based on two objective functions is done using a concept called archive, which includes a population of non-dominated solutions found so far and is kept separate from the main population. This pool has a limited capacity and always contains solutions that approximate the Pareto front. Moreover, each cuckoo has two sources of information for movement: the current position and the position of the best cuckoo of each generation. Therefore, in the multi-objective cuckoo algorithm, the concept of the best position in the population has changed. The best position of the cuckoo population is not a fixed position. All the collections of cuckoos in the archive are among the best.

3.3. Cuckoos State Encoding and Cluster Centers Initialization

In the Informed AdamCo, a set of real numbers represents the state of cuckoos. These real numbers indicate the locations of the cluster centers and are initially chosen from labeled data of each cluster. Each cuckoo state is a matrix with '*k*' rows and '*d'* columns, where '*k*' is the number of clusters and '*d*' is the number of data dimensions.

3.4. Assignment of Points

The minimum Euclidean distance measure has been utilized to assign unlabeled data. A special point P is given to the cluster with the minimum Euclidean distance from its center (Equations 1 and 2).

$$P \in i|\, argmin_i\{\, d_e(C_i.P)\} \qquad (1)$$

$$i = 1...k \qquad (2)$$

C_i is the ith cluster center and d_e designates the Euclidean distance.

3.5. Objective Functions

Two objective functions are considered for optimization: The compactness and the Normalized Mutual Information (NMI) index. Compactness is obtained through the calculation of the Sum of Squares of Errors (SSE) of a solution according to Eq. 3 and represents the distance between an object and the closest cluster center. In calculating this error, labeled data are assigned to their respective cluster and not necessarily to the cluster from which they have a shorter distance:

$$SSE(\text{₵}) = \sum_{i=1}^{k} \sum_{\forall x_j \in c_i} ||x_j - \mu_i||^2 \qquad (3)$$

where ||.|| indicates the Euclidean distance, μ_i is the cluster center C_i, and x_j denotes the j_{th} element of the dataset. As the objective function, the SSE should be minimized.

The NMI index is a symmetric measure that calculates the shared information between the members of a pair of clusterings. Our optimization algorithm reaches the best result through minimization. Therefore, the NMI negation is minimized and defined as follows:

$$\mathrm{I(X;Y)=H(X)\text{-}H(X|Y)} \qquad (4)$$

$$\mathrm{NMI = I(X.Y)/\sqrt{H(X)H(Y)}} \qquad (5)$$

Eq. 4 represents the mutual information between *X* and *Y*. *X* and *Y* are two random variables described by clustering labels with different numbers of clusters. *H*(.) indicates the entropy according to Eq. 5 [38].

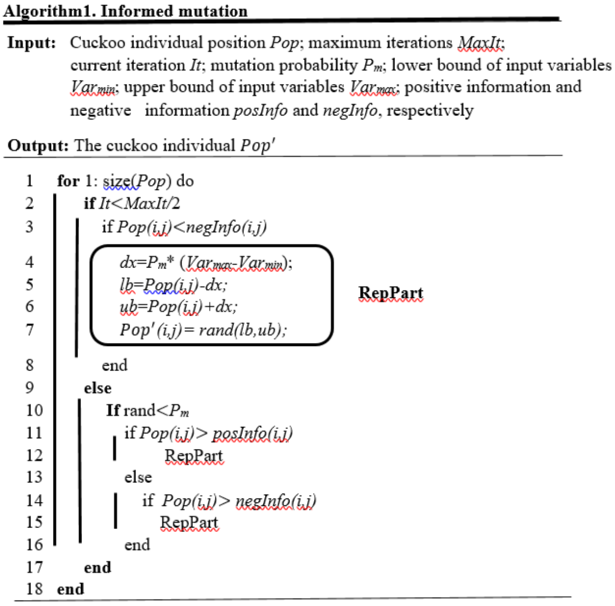
3.6. Informed Mutation

To expand the set of superior solutions and guide the whole population towards enhanced solutions, a well-informed mutation method is presented. This mutation approach relies on preference details, namely Negative and Positive information. The method is built on two key principles: Negative information, which involves the average positions of dominated solutions, and Positive information, which encompasses the average positions of non-dominated solutions at each iteration. In the first half of the evolutionary process, the negative information is used to manage the cuckoo population and quickly discard unfavorable cuckoo positions. Later, both positive and negative information are used to steer the overall population's evolution, facilitating the rapid attainment of optimal solutions and augmenting the diversity of these solutions during the latter half of the evolution process. This approach is mathematically formulated in Eqs. 6 and 7.

$$\text{Positive Information}=\sum_{i=1}^{|N|}\text{CuckooPop}(i).\text{Position}\Big/|N| \tag{6}$$

$$\text{NegativeInformation}=\sum_{j=1}^{|D|}\text{CuckooPop}(j).\text{Position}\Big/|D| \tag{7}$$

$|N|$ and $|D|$ represent the number of non-dominant and dominated cuckoos in each iteration, respectively. The Informed mutation algorithm is defined below (Algorithm 1).



Algorithm1. Informed mutation

Input: Cuckoo individual position *Pop*; maximum iterations *MaxIt*; current iteration *It*; mutation probability P_m; lower bound of input variables Var_{min}; upper bound of input variables Var_{max}; positive information and negative information *posInfo* and *negInfo*, respectively

Output: The cuckoo individual *Pop′*

```
for 1: size(Pop) do
    if It<MaxIt/2
        if Pop(i,j)<negInfo(i,j)
            dx=Pm* (Varmax-Varmin);
            lb=Pop(i,j)-dx;
            ub=Pop(i,j)+dx;                RepPart
            Pop′(i,j)= rand(lb,ub);
        end
    else
        If rand<Pm
            if Pop(i,j)> posInfo(i,j)
                RepPart
            else
                if Pop(i,j)> negInfo(i,j)
                RepPart
            end
    end
end
```

3.7. Fuzzy Parameter Adjustment

In this section, we discuss the design of the fuzzy system that is used in the adaptation of the migration coefficient parameter.

The equation of cuckoos' movement in each generation is defined as follows:

$$\mathrm{H_{ij}}^{\mathrm{new}} = \mathrm{C_1} * \mathrm{H_{ij}}^{\mathrm{old}} + \mathrm{C_2} * \mathrm{F} * \lambda(\mathrm{H}_{\mathrm{ij}}^{\mathrm{best}} - \mathrm{H}_{\mathrm{ij}}^{\mathrm{old}})$$

where $H_{ij}{}^{new}$ is a new position, H_{ij}^{old}indicates the current cuckoo position, F denotes the migration coefficient, λ is a uniform random number between zero and one, and H_{ij}^{best} represents the position of the best cuckoo in the best cluster in each generation. $\mathrm{C_1} = \left(1 - \exp\left(-\frac{1}{\text{current iteration}}\right)\right)$, and $\mathrm{C_2} = (0.5 + 1.5 * (1/\text{sqrt}(\text{current iteration})))\text{^}\text{Norm}(\text{NormalizedError})$ coefficients have been added to the cuckoo movement pattern of [39] to increase convergence inspired by [40]. $Norm(NormalizedError)$ will be described subsequently.

The migration coefficient (*F)* and λ help cuckoos explore a broader range of positions in all environments. F needs to be adjusted manually; we have adapted this coefficient automatically using fuzzy logic. According to [30], the recommended values of this parameter fall within the range of 0.5 to 2.5.

Since the practical components affecting parameter adaptation include the diversity of the swarm, the error, and the algorithm iterations, in our proposed method, all the mentioned components are used to adapt the migration coefficient parameter through the fuzzy system in each iteration dynamically. Mamdani's fuzzy system framework is designed according to Fig. 2.

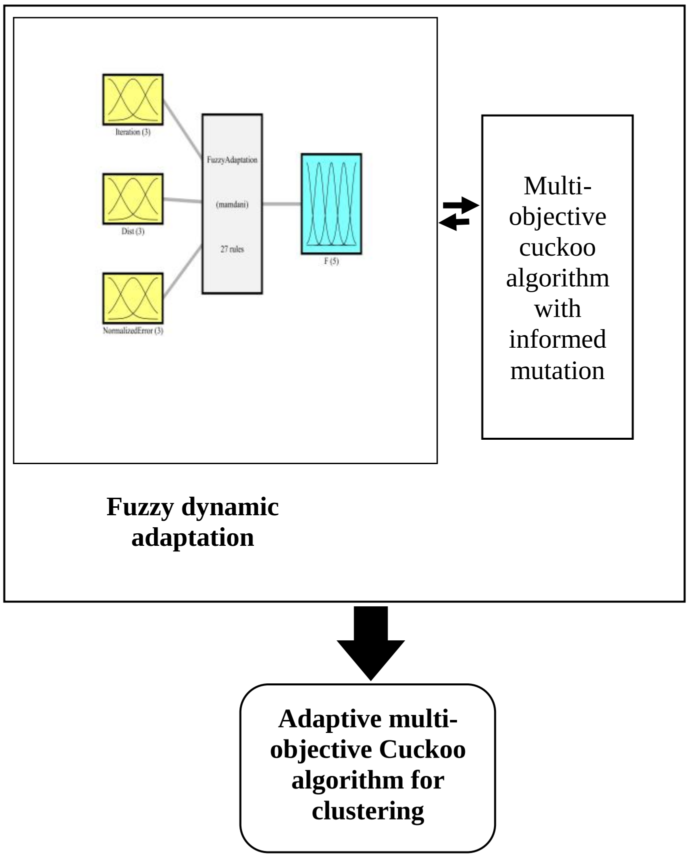

Iteration (3)
Dist (3)
NormalizedError (3)
FuzzyAdaptation
(mamdani)
27 rules
F (5)
Multi-objective cuckoo algorithm with informed mutation
Fuzzy dynamic adaptation
Adaptive multi-objective Cuckoo algorithm for clustering

Fig. 2. Fuzzy system design of adaptive parameter adjustment of the multi-objective cuckoo algorithm with informed mutation for semi-supervised data clustering.

To measure iteration, the percentage of repetitions has been considered: at the initial iterations of the algorithm, the percentage of repetitions is low, but when the repetitions are completed, the value for iteration increases, which is defined as Eq. 9:

$$\text{Iteration} = \frac{\text{Current iteration}}{\text{Maximum of iterations}} \quad (9)$$

To measure diversity, we used the Euclidean distance criterion so that the greater the distance of cuckoos from the best cuckoo of the best cluster is, the greater its diversity will be. The diversity is normalized as given in

$$NormalizedDist = \frac{Distance(Each\ Cuckoo\ from\ Best\ Cuckoo\ of\ the\ Best\ Cluster)}{Maximum\ Distance} \quad (10)$$

The Error criterion is also defined as the difference between the fitness value of each cuckoo and the fitness of the best cuckoo of the best cluster in each generation (Eq. 11).

$$Error = |Fitness(x_i) - Fitness(BestCuckoo)| \quad (11)$$

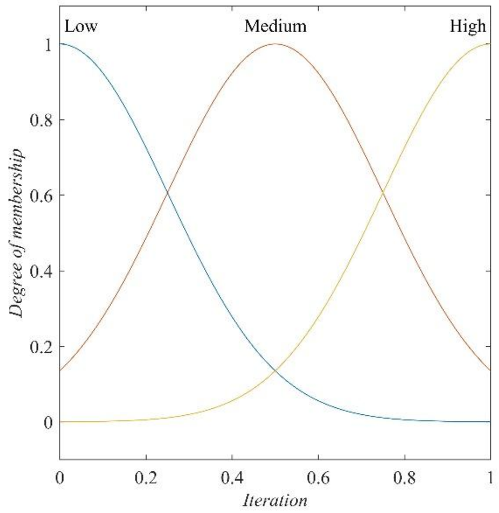
To have the Error value between 0 and 1, we normalize it as Eq. 12:

$$\text{NormalizedError} = \frac{\text{Error} - \min(\text{Error})}{\max(\text{Error}) - \min(\text{Error})} \quad (12)$$

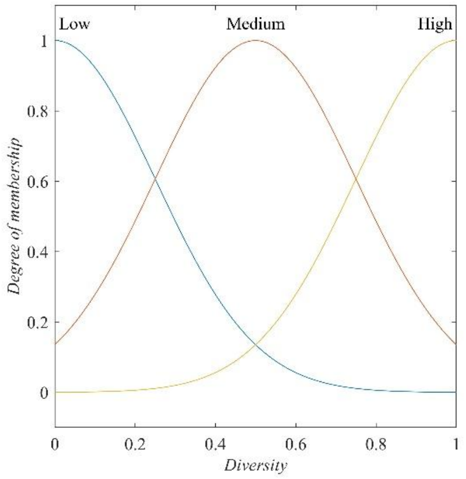
The Error vector is represented as a two-dimensional vector. To simplify the calculation and comparison of the results, the normalized form of Eq. 12 is adopted as the fuzzy system input, as indicated in Eq. 13.

$$\text{Norm}(\text{NormalizedError}) = \text{Norm2}(\text{NormalizedError}) \quad (13)$$

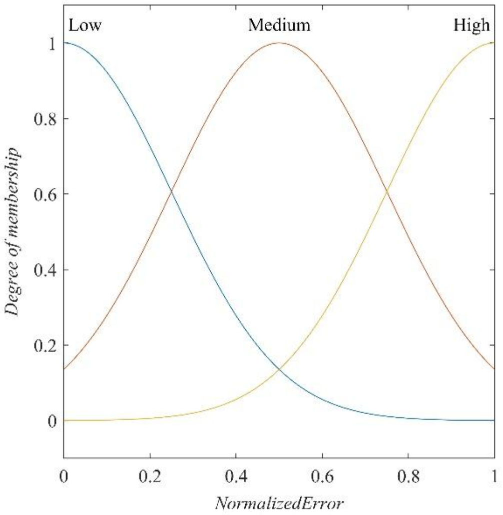
Therefore, three inputs are applied to the fuzzy system to tune the migration coefficient parameter dynamically. The outcome of the fuzzy system represents the modified migration coefficient. The input variables have been designed according to Figs. 3 (a), (b), and (c), which show iteration, distance-based diversity, and error, respectively. Each input is crumbled into three Gaussian membership functions. As mentioned earlier, the output variable's range of change is considered between 0.5 and 2.5, defined by five Gaussian membership functions and illustrated in Fig. 3 (d).



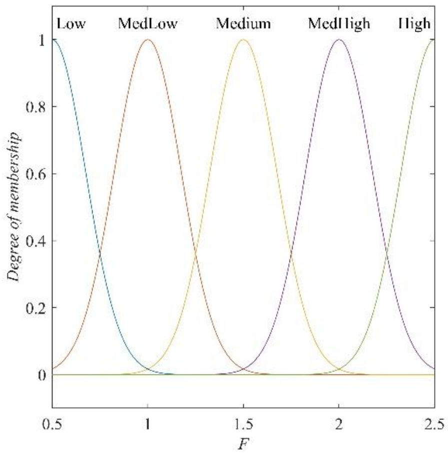
(a)



(b)



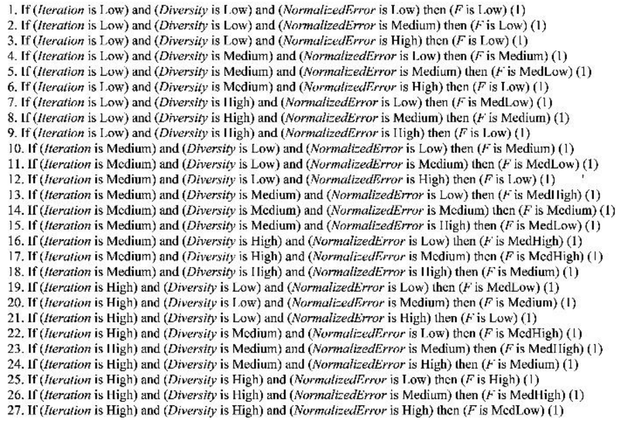
(c)



(d)

Fig. 3. a) Iteration input. b) Diversity input. c) Normalized Error input. d) Output (adjusted migration coefficient).

The rules of the fuzzy system are formulated based on the following logic: the exploration must be high in the initial iterations, and gradually, as the algorithm approaches the final iterations, exploitation should be increased due to the enrichment of the exploration. On the other hand, when the diversity of the population is low, the cuckoos are, accordingly, close to the best cuckoo. Therefore, the exploration should be increased. In scenarios of high diversity, however, the exploitation should be increased. The rules of the fuzzy system are shown in Fig. 4:

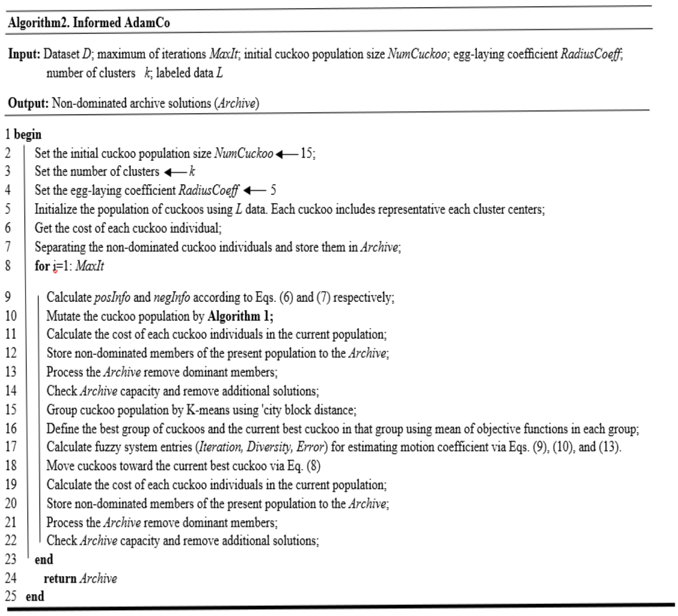


1. If (*Iteration* is Low) and (*Diversity* is Low) and (*NormalizedError* is Low) then (*F* is Low) (1)
2. If (*Iteration* is Low) and (*Diversity* is Low) and (*NormalizedError* is Medium) then (*F* is Low) (1)
3. If (*Iteration* is Low) and (*Diversity* is Low) and (*NormalizedError* is High) then (*F* is Low) (1)
4. If (*Iteration* is Low) and (*Diversity* is Medium) and (*NormalizedError* is Low) then (*F* is Medium) (1)
5. If (*Iteration* is Low) and (*Diversity* is Medium) and (*NormalizedError* is Medium) then (*F* is MedLow) (1)
6. If (*Iteration* is Low) and (*Diversity* is Medium) and (*NormalizedError* is High) then (*F* is Low) (1)
7. If (*Iteration* is Low) and (*Diversity* is High) and (*NormalizedError* is Low) then (*F* is MedLow) (1)
8. If (*Iteration* is Low) and (*Diversity* is High) and (*NormalizedError* is Medium) then (*F* is Medium) (1)
9. If (*Iteration* is Low) and (*Diversity* is High) and (*NormalizedError* is High) then (*F* is Low) (1)
10. If (*Iteration* is Medium) and (*Diversity* is Low) and (*NormalizedError* is Low) then (*F* is Medium) (1)
11. If (*Iteration* is Medium) and (*Diversity* is Low) and (*NormalizedError* is Medium) then (*F* is MedLow) (1)
12. If (*Iteration* is Medium) and (*Diversity* is Low) and (*NormalizedError* is High) then (*F* is Low) (1)
13. If (*Iteration* is Medium) and (*Diversity* is Medium) and (*NormalizedError* is Low) then (*F* is MedHigh) (1)
14. If (*Iteration* is Medium) and (*Diversity* is Medium) and (*NormalizedError* is Medium) then (*F* is Medium) (1)
15. If (*Iteration* is Medium) and (*Diversity* is Medium) and (*NormalizedError* is High) then (*F* is MedLow) (1)
16. If (*Iteration* is Medium) and (*Diversity* is High) and (*NormalizedError* is Low) then (*F* is MedHigh) (1)
17. If (*Iteration* is Medium) and (*Diversity* is High) and (*NormalizedError* is Medium) then (*F* is MedHigh) (1)
18. If (*Iteration* is Medium) and (*Diversity* is High) and (*NormalizedError* is High) then (*F* is Medium) (1)
19. If (*Iteration* is High) and (*Diversity* is Low) and (*NormalizedError* is Low) then (*F* is MedLow) (1)
20. If (*Iteration* is High) and (*Diversity* is Low) and (*NormalizedError* is Medium) then (*F* is Medium) (1)
21. If (*Iteration* is High) and (*Diversity* is Low) and (*NormalizedError* is High) then (*F* is Low) (1)
22. If (*Iteration* is High) and (*Diversity* is Medium) and (*NormalizedError* is Low) then (*F* is MedHigh) (1)
23. If (*Iteration* is High) and (*Diversity* is Medium) and (*NormalizedError* is Medium) then (*F* is MedHigh) (1)
24. If (*Iteration* is High) and (*Diversity* is Medium) and (*NormalizedError* is High) then (*F* is Medium) (1)
25. If (*Iteration* is High) and (*Diversity* is High) and (*NormalizedError* is Low) then (*F* is High) (1)
26. If (*Iteration* is High) and (*Diversity* is High) and (*NormalizedError* is Medium) then (*F* is MedHigh) (1)
27. If (*Iteration* is High) and (*Diversity* is High) and (*NormalizedError* is High) then (*F* is MedLow) (1)

Fig. 4. The rules of the designed fuzzy system to estimate the migration coefficient.

3.8. The Informed AdamCo Algorithm

This study introduces the Informed AdamCo algorithm, an adaptive multi-objective cuckoo algorithm for semi-supervised clustering. By utilizing fuzzy parameter adaptation and preference information, the algorithm aims to enhance the clustering process. To provide a comprehensive understanding of the Informed AdamCo algorithm, its pseudocode is presented in Algorithm 2. According to the pseudocode, the cuckoo population is initialized and the objective values of each cuckoo are obtained in Steps 5 to 6. Subsequently, the initial cuckoo population undergoes the non-dominated sorting method to extract and store the non-dominated individuals. Steps 9 to 11 involve the calculation of Positive and Negative Information, the application of the mutation algorithm to the current cuckoo population, and the computation of the objective values for the mutated population. The mutated population is then subjected to the non-dominated sorting algorithm to identify the non-dominated cuckoo individuals, which are stored as a non-dominated solution for updating in Steps 12 to 14. Steps 15 to 16 include grouping the cuckoo population by the K-means algorithm to find the best group using the mean of each group's objective values and, subsequently, defining the best cuckoo in the population by specifying the best one in the best group. Calculation of the fuzzy system inputs using Eqs. 9, 10, and 13 are carried out in Step 17, which leads to the estimation of the migration coefficient parameter. Then, in Step 18, the cuckoo population moves towards the best cuckoo via Eq. 8. In the final stages of the algorithm, the objective values of the migrated cuckoo population are calculated, and the archive is updated with non-dominated members. Steps 9 to 23 continue in a loop until the maximum iteration is reached and the archive of non-dominated members is finally returned.



Algorithm2. Informed AdamCo

Input: Dataset *D*; maximum of iterations *MaxIt*; initial cuckoo population size *NumCuckoo*; egg-laying coefficient *RadiusCoeff*; number of clusters *k*; labeled data *L*

Output: Non-dominated archive solutions (*Archive*)

```
1 begin
2  | Set the initial cuckoo population size NumCuckoo ⟵ 15;
3  | Set the number of clusters ⟵ k
4  | Set the egg-laying coefficient RadiusCoeff ⟵ 5
5  | Initialize the population of cuckoos using L data. Each cuckoo includes representative each cluster centers;
6  | Get the cost of each cuckoo individual;
7  | Separating the non-dominated cuckoo individuals and store them in Archive;
8  | for i=1: MaxIt
9  |   | Calculate posInfo and negInfo according to Eqs. (6) and (7) respectively;
10 |   | Mutate the cuckoo population by Algorithm 1;
11 |   | Calculate the cost of each cuckoo individuals in the current population;
12 |   | Store non-dominated members of the present population to the Archive;
13 |   | Process the Archive remove dominant members;
14 |   | Check Archive capacity and remove additional solutions;
15 |   | Group cuckoo population by K-means using 'city block distance;
16 |   | Define the best group of cuckoos and the current best cuckoo in that group using mean of objective functions in each group;
17 |   | Calculate fuzzy system entries (Iteration, Diversity, Error) for estimating motion coefficient via Eqs. (9), (10), and (13).
18 |   | Move cuckoos toward the current best cuckoo via Eq. (8)
19 |   | Calculate the cost of each cuckoo individuals in the current population;
20 |   | Store non-dominated members of the present population to the Archive;
21 |   | Process the Archive remove dominant members;
22 |   | Check Archive capacity and remove additional solutions;
23 | end
24    return Archive
25 end
```

4. Experimental Results and Analysis

4.1. Experimental Datasets and Parameter Settings

In this section, we delineate the experiments carried out to demonstrate the effectiveness of the proposed Informed AdamCo clustering approach. Furthermore, we conduct a numerical and statistical analysis of the results obtained. A total of seventeen datasets were selected to assess the performance of Informed AdamCo, with ten UCI datasets and six artificial datasets included in the evaluation. The artificial datasets were created using the software developed by Julia Handl and were formatted *as Xd-Xc-noX*, where *'d*' represented attributes, *'c'* indicated clusters, and *'no'* denoted the dataset number. For example, the dataset *2d-10c-no0* consisted of two attributes, ten clusters, and a dataset number of zero. The last dataset selected for the study i.e., *10kdd*, is a practical standard for comparing different intrusion detection methods. This dataset contains 494,021 records, 41 features, and 23 clusters (one cluster contains standard records, and the other clusters represent different types of network attacks). Since the proposed method can be applied to numerical data, the three features of this dataset are not numeric. First, the strings of these three features were merged into one. In the next step, a number between 0 and

208 was assigned to each unique string. Instead of 23 clusters, two clusters were considered. One cluster is for the standard records, and the other is for all the other 22 types of network attacks. The relief method was applied to select the 10 top features and showed that many records were repeated. As a result, duplicate records were removed, leaving a total of 41351 records in the end. The Informed AdamCo algorithm was applied to this collection. Table 1 provides detailed descriptions of these datasets. The number of features across the 17 datasets ranges from 2 to 200, while the number of samples ranges from 100 to 494021. All simulation experiments were conducted using MATLAB R2022a on a system equipped with Windows 10, an Intel (R) Core (TM) i7-6500 CPU running at 2.5 GHz, and 8.00 GB of RAM.

Table 1
Information on 17 Datasets.

Name	#Example	#Attribute d(D)	#Class
Iris	150	4	3
Wine	178	13	3
Jain	373	2	2
Bank Authentication	1372	4	2
Ecoli	336	7	8
Aggregati on	788	2	7
Zoo	101	17	7
WDBC	569	30	2
Seeds	210	7	3
GenPAN CAN-801*200	801	200	5
110d-3c-HighD	100	110	3
2d-10c-no	2972	2	10
2d-4c-no1	1623	2	4
2d-4c-no2	1064	2	4
2d-4c-no3	1123	2	4
2d-4c-no4	863	2	4
10kdd	49402 1	10	2

The partitioning results obtained were compared with various state-of-the-art algorithms, including semi-supervised NSGAII (semi-NSGAII), semi-supervised MOPSO (semi-MOPSO), semi-supervised multi-objective evolutionary algorithm based on decomposition (semi-MOEA/D), semi-supervised multi-objective grasshopper (semi-MOGHopper), and semi-supervised multi-objective adaptive guided differential evolution (semi-MOAGDE). The standard parameters for these comparative algorithms were consistent: population size *(nPop) = 30*, maximum iterations *(MaxIt) = 150*, and number of repository members *(nRep) = 100*. To account for the randomness of the optimization algorithm, each approach was executed 20 times on the datasets to calculate the average result. The specific parameter configuration for Informed AdamCo and the other comparative algorithms can be found in Table 2. To ensure an equitable comparison among different algorithms, all semi-supervised clustering algorithms utilized the same labeled dataset (*10%)* in our experiment. Moreover, the number of clusters '*k*' was set to correspond with the number of ground truth clusters.

Table 2
Parameter settings for all comparative algorithms.

Algorithm	Y	Parameters
MOPSO	2005	Inertia weight w = 0.7; maximum velocity Vmax = 0.6; inflation parameter of the grid is set to 0.1; and the acceleration constants c1=1, and c2 = 2.
NSGAII	2002	Crosser probability pCross = 0.7; mutation rate mu= 0.02;
MOEA/D	2007	Number of neighbors T=20;
MOGHopper	2017	Default parameter settings are used
MOAGDE	2 021	Default parameter settings are used

4.2. Clustering Results of Informed AdamCo and Five Other Algorithms

To evaluate the performance of Informed AdamCo with other algorithms, five state-of-the-art semi-supervised multi-objective algorithms, including semi-NSGAII [41], semi-MOPSO [42], semi-MOEA/D (Zhang & Li, 2007), semi-MOGHopper [44], semi-MOAGDE [45] were employed for comparison. The relevant parameter settings of these algorithms are outlined in Table 2. To assess the clustering performance of each algorithm, two commonly used evaluation metrics were used: the Adjusted Rand Index (ARI) [46], and the Accuracy (Acc) [47]. The Adjusted Rand Index is an external measure with a value between -1 and 1. The closer the ARI value to one, the better the clustering quality. The Accuracy is also an external measure and is equivalent to the ratio of accurate matching pair number to the total matching pair number. The experimental results of comparison with these five algorithms are shown in Table 3 with optimal results highlighted in bold. As seen in Table 3, the three metric values of Informed AdamCo are better than those of the other algorithms on 13 datasets, except for datasets Iris, Ecoli, Aggregation, and Zoo.

The results in Table 3 show that the Informed AdamCo algorithm improves the clustering performance for most datasets compared to the other five algorithms compared. The significance of the results is confirmed via the t-test. According to Table 3, the values of the criteria of Informed AdamCo are better than those of semi-NSGAII, semi-MOPSO, semi-MOEA/D, semi-MOGHopper, and semi-MOAGDE on 13 datasets, except for Ecoli, Aggregation, and Zoo datasets. Specifically, semi-MOAGDE exhibits superior performance on the Aggregation dataset, semi-MOEA/D achieves better results on the Ecoli dataset, and semi-MOGHopper demonstrates improved performance on the Zoo dataset.

In summary, Informed AdamCo has revealed superior clustering performance on most datasets. Thanks to its motion coefficient parameter adaptation, more efficient use of labeled data, and its informed mutation method, the Informed AdamCo algorithm can enhance the likelihood of obtaining superb candidate individuals. In examining the data in which the proposed method did not work better than other comparable methods, it seems that most of such data are highly cluttered.

Table 3
Comparing The Performance of The Proposed Method with 5 Other Algorithms.

Dataset **Algol**	semi-NSGAII		semi-MOPSO	
	ARI	Acc	ARI	Acc
Iris	0.53	70	0.53	70.27
Wine	0.38	68.06	0.39	68.06
Jain	0.62	90	0.63	90.07
BankAuthentication0402	0.69	90.6	0.7	90.67
Ecoli	0.44	64.2	0.45	64.27
Aggregation	0.57	69	0.58	69.1
Zoo	0.74	81	0.75	81.24
WDBC	0.73	93	0.74	93.18
Seeds	0.33	60	0.34	60.93
GenePANCAN-801*200	0.71	78	0.72	78.47
2d-10c-no0	0.82	87	0.83	87.27
2d-4c-no1	0.88	90	0.89	90.17
2d-4c-no2	0.87	93	0.88	93.13
2d-4c-no3	0.93	95	0.94	95.92
2d-4c-no4	0.94	96	0.95	96.44
110d-3c-HighD	0.85	90	0.86	90.2
110kdd-10-02	0.57	89.5	0.58	90

Table 3 Continued
Comparing The Performance of The Proposed Method with 5 Other Algorithms.

Dataset Algol	semi-MOEAD		semi-MOGHopper	
	ARI	Acc	ARI	Acc
Iris	0.81	90.77	0.28	56.9
Wine	0.39	72.53	0.37	47.16
Jain	0.72	92.64	0.29	75.88
BankAuthentication0402	0.54	85.65	0.2	56.49
Ecoli	**0.74**	**81.07**	0.26	52.53
Aggregation	0.82	90.54	0.45	60.93
Zoo	0.42	61.93	**0.84**	**88.12**
WDBC	0.71	92.12	0.3	64.16
Seeds	0.71	88.74	0.16	48.95
GenePANCAN-801*200	0.97	99.41	0.35	62.68
2d-10c-no0	0.87	90.16	0.43	57.35
2d-4c-no1	0.9	92.32	0.43	66.96
2d-4c-no2	0.89	94.57	0.35	59.04
2d-4c-no3	0.95	98.56	0.51	72.2
2d-4c-no4	0.97	99.64	0.33	60.97
110d-3c-HighD	0.88	95.75	0.2	66.92
110kdd-10-02	0.49	86.5	0.2	76.68

Table 3 Continued

Comparing The Performance of The Proposed Method with 5 Other Algorithms

Dataset Algol	semi-MOAGDE		Informed AdamCo	
	ARI	Acc	ARI	Acc
Iris	0.84	94.17	**0.85**	**95.27**
Wine	0.42	73.03	**0.46**	**74.78**
Jain	0.59	88.18	**0.79**	**94.93**
BankAuthentication0402	0.42	82.16	**0.78**	**93.97**
Ecoli	0.66	73.9	0.73	79.14
Aggregation	**0.86**	**93.27**	0.83	89.7
Zoo	0.81	87.33	0.83	87.57
WDBC	0.69	91.66	**0.75**	**93.5**
Seeds	0.71	88.83	**0.72**	**89.05**
GenePANCAN-801*200	0.98	99.44	**0.99**	**99.55**
2d-10c-no0	0.87	91.89	**0.88**	**92.56**
2d-4c-no1	0.89	93.09	**0.93**	**96.18**
2d-4c-no2	0.90	95.87	**0.91**	**96.12**
2d-4c-no3	0.96	98.29	**0.97**	**98.58**
2d-4c-no4	0.98	99.57	**0.99**	**99.69**
110d-3c-HighD	0.91	97.6	**0.92**	**96.95**
110kdd-10-02	0.51	87.18	**0.61**	**90.88**

5. Conclusion

The AdamCo demostrates high search capability, a simple structure, and good stability. Furthermore, most of the currently used optimization-based clustering techniques are prone to produce suboptimal solutions. The present study introduced the AdamCo algorithm based on preference information and adaptive adjustment of the migration coefficient parameter to address these clustering issues. First, the single-objective cuckoo algorithm was analyzed and then converted to a multi-objective cuckoo algorithm due to the multi-objective and NP-hard nature of clustering. Labeled data were employed in two steps (the initialization step and the learning phase) of the proposed method. To produce a better population, an informed mutation was also used in the algorithm to improve the diversity of the solutions. Finally, the migration coefficient parameter of the cuckoo algorithm was adapted using the fuzzy technique with three input components: algorithm iterations, population diversity, and error. AdamCo was tested on several UCI, artificial, and 10kdd datasets and compared with five other state-of-the-art algorithms. The experimental results illustrate that the AdamCo algorithm outperforms the other compared algorithms in terms of ARI and Accuracy. Future research can focus on the weaknesses of the Informed AdamCo algorithm. One such work can be improving the performance of the algorithm on datasets with mixed clusters.

References

[1] J. S. Mahmud, E. Birihanu, and I. Lendak, “A Semi-supervised Framework for Anomaly Detection and Data Labeling for Industrial Control Systems,” in *Disruptive Information Technologies for a Smart Society*, 2024, pp. 149–160. doi: 10.1007/978-3-031-50755-7_15.

[2] X. Shi, C. Yue, M. Quan, Y. Li, and H. Nashwan Sam, “A semi-supervised ensemble clustering algorithm for discovering relationships between different diseases by extracting cell-to-cell biological communications,” *J. Cancer Res. Clin.*

Oncol., vol. 150, no. 1, pp. 1–15, Jan. 2024, doi: 10.1007/s00432-023-05559-4.

[3] S. Adiga V., J. Dolz, and H. Lombaert, “Anatomically-aware uncertainty for semi-supervised image segmentation,” *Med. Image Anal.*, vol. 91, pp. 103–111, Jan. 2024, doi: 10.1016/j.media.2023.103011.

[4] Y. Qin, S. Ding, L. Wang, and Y. Wang, “Research progress on semi-supervised clustering,” *Cognit. Comput.*, vol. 11, pp. 599–612, 2019, doi: https://doi.org/10.1007/s12559-019-09664-w.

[5] D. Dinler and M. K. Tural, “A survey of constrained clustering,” in *Unsupervised Learning Algorithms*, M. Celebi and K. Aydin, Eds. Springer:Cham, 2016, pp. 207–235. doi: 10.1007/978-3-319-24211-8_9.

[6] S. J. Nanda and G. Panda, “A survey on nature inspired metaheuristic algorithms for partitional clustering,” *Swarm Evol. Comput.*, vol. 16, pp. 1–18, 2014.

[7] R. K. Sanodiya, S. Saha, and J. Mathew, “A kernel semi-supervised distance metric learning with relative distance: Integration with a MOO approach,” *Expert Syst. Appl.*, vol. 125, pp. 233–248, Jul. 2019, doi: 10.1016/j.eswa.2018.12.051.

[8] Z. Zhang, J. T. Kwok, and D.-Y. Yeung, “Parametric distance metric learning with label information,” 2003.

[9] M. Baghshah, S. Shouraki, and N. 2010, “Kernel-based metric learning for semi-supervised clustering,” *Neurocomputing*, vol. 73, no. 7–9, pp. 1352–1361, 2010, doi: https://doi.org/10.1016/j.neucom.2009.12.009.

[10] K. Wagstaff and C. Cardie, “Clustering with Instance-level Constraints,” *AAAI/IAAI*, vol. 1097, pp. 577–584, 2000.

[11] S. Basu, A. Banerjee, and R. Mooney, “Semi-supervised clustering by seeding,” in *Proceedings of the 19th International Conference on Machine Learning (ICML-2002)*, 2002, pp. 19–26. [Online]. Available: https://www.cs.utexas.edu/~ml/papers/semi-icml-02.pdf

[12] H. Akbarzadeh Khorshidi, U. Aickelin, G. Haffari, and B. Hassani-Mahmooei, “Multi-objective semi-supervised clustering to identify health service patterns for injured patients,” *Heal. Inf. Sci. Syst.*, vol. 7, no. 1, pp. 1–8, Dec. 2019, doi: 10.1007/s13755-019-0080-6.

[13] S. Saha, K. Kaushik, A. K. Alok, and S. Acharya, “Multi-objective semi-supervised clustering of tissue samples for cancer diagnosis,” *Soft Comput.*, vol. 20, no. 9, pp. 3381–3392, Sep. 2016, doi: 10.1007/s00500-015-1783-5.

[14] A. Kumar Alok, S. Saha, and A. Ekbal, “Multi-objective semi-supervised clustering for automatic pixel classification from remote sensing imagery,” *Soft Comput.*, vol. 20, no. 12, pp. 4733–4751, Dec. 2016, doi: 10.1007/s00500-015-1701-x.

[15] A. K. Alok, S. Saha, and A. Ekbal, “A new semi-supervised clustering technique using multi-objective optimization,” *Appl. Intell.*, vol. 43, no. 3, pp. 633–661, 2015, doi: 10.1007/s10489-015-0656-z.

[16] A. Kumar Alok, S. Saha, and A. Ekbal, “Semi-supervised clustering for gene-expression data in multiobjective optimization framework,” *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 2, pp. 421–439, Apr. 2017, doi: 10.1007/s13042-015-0335-8.

[17] J. Ebrahimi and M. S. Abadeh, “Semi supervised clustering: a pareto approach,” in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 2012, pp. 237–251. doi: https://doi.org/10.1007/978-3-642-31537-4_19.

[18] S. Saha, A. Ekbal, and A. K. Alok, “Semi-supervised clustering using multiobjective optimization,” in *Proceedings of the 2012 12th International Conference on Hybrid Intelligent Systems*, 2012, pp. 360–365. doi: 10.1109/HIS.2012.6421361.

[19] D. T. C. Lai, M. Miyakawa, and Y. Sato, “Semi-supervised data clustering using particle swarm optimisation,” *Soft Comput.*, vol. 24, no. 5, pp. 3499–3510, Mar. 2020, doi: 10.1007/s00500-019-04114-z.

[20] J. Dong, M. Qi, and F. Wang, “A two-stage semi-supervised clustering method based on hybrid particle swarm optimization,” in *1st International Conference on Electronics Instrumentation and Information Systems, EIIS 2017*, 2017, pp. 1–5. doi: 10.1109/EIIS.2017.8298609.

[21] J. Dong, M. Qi, and F. Wang, “An improved artificial bee colony algorithm for solving semi-supervised clustering,” in *5th International Conference on Computer Science and Network Technology*, 2016, vol. December, pp. 315–319. doi: 10.1109/iccsnt.2016.8070171.

[22] O. Castillo, H. Neyoy, J. Soria, M. García, and F. Valdez, “Dynamic fuzzy logic parameter tuning for aco and its application in the fuzzy logic control of an autonomous mobile robot,” *Int. J. Adv. Robot. Syst.*, vol. 10, no. 1, Jan. 2013, doi: 10.5772/54883.

[23] S. Mishra, “A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation,” *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 61–73, 2005, doi: 10.1109/TEVC.2004.840144.

[24] C. Venkaiah and D. Vinod Kumar, “Fuzzy adaptive bacterial foraging congestion management using sensitivity based optimal active power re-scheduling of generators,” *Appl. Soft Comput.*, vol. 11, no. 8, pp. 4921–4930, 2011, doi: https://doi.org/10.1016/j.asoc.2011.06.007.

[25] T. Hassanzadeh, M. R. Meybodi, and M. Shahramirad, “A new fuzzy firefly algorithm with adaptive parameters,” *Int. J. Comput. Intell. Appl.*, vol. 16, no. 3, Sep. 2017, doi: 10.1142/S1469026817500171.

[26] S. H. Zahiri, “Fuzzy gravitational search algorithm an approach for data mining,” *Iran. J. Fuzzy Syst.*, vol. 9, no. 1, pp. 21–37, 2012, doi: 10.22111/ijfs.2012.223.

[27] H. Askari and S. H. Zahiri, “Decision function estimation using intelligent gravitational search algorithm,” *Int. J. Mach. Learn. Cybern.*, vol. 3, no. 2, pp. 163–172, Jun. 2012, doi: 10.1007/s13042-011-0052-x.

[28] J. Kumar, D. Kumar, and K. Edukondalu, “Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market,” *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2445–2455, 2013, doi: https://doi.org/10.1016/j.asoc.2012.12.003.

[29] F. Saeidi-Khabisi and S. Rashedi, “Fuzzy gravitational search algorithm,” in *2nd International eConference on Computer and Knowledge Engineering*, 2012, vol. October, pp. 156–160. doi: 10.1109/icccke.2012.6395370.

[30] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, “Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic,” *Expert Syst. Appl.*, vol. 40, no. 8, pp. 3196–3206, Jun. 2013, doi: 10.1016/J.ESWA.2012.12.033.

[31] T. Niknam, E. Azadfarsani, and M. Jabbari, “A new hybrid evolutionary algorithm based on new fuzzy adaptive PSO and NM algorithms for distribution feeder reconfiguration,” *Energy Convers. Manag.*, vol. 54, no. 1, pp. 7–16, 2012, doi: https://doi.org/10.1016/j.enconman.2011.09.014.

[32] F. Olivas, F. Valdez, and O. Castillo, “Particle swarm optimization with dynamic parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions,” in *World Congress on Nature and Biologically Inspired Computing*, 2013, vol. August, pp. 36–40. doi: 10.1109/NaBIC.2013.6617875.

[33] M. Neshat, “FAIPSO: Fuzzy adaptive informed particle swarm optimization,” *Neural Comput. Appl.*, vol. 23, no. Suppl1, pp. 95–116, 2013, doi: 10.1007/s00521-012-1256-z.

[34] D. Yazdani, M. R. Meybodi, and A. N. Toosi, “Fuzzy adaptive artificial fish swarm algorithm,” in *AI 2010: Advances in Artificial Intelligence.*, 2010, vol. 6464, pp. 334–343. doi: 10.1007/978-3-642-17432-2_34.

[35] X. Ou, M. Wu, Y. Pu, B. Tu, G. Zhang, and Z. Xu, “Cuckoo search algorithm with fuzzy logic and Gauss–Cauchy for minimizing localization error of WSN,” *Appl. Soft Comput.*, vol. 125, pp. 109–211, Aug. 2022, doi: 10.1016/j.asoc.2022.109211.

[36] A. José-García and W. Gómez-Flores, “Automatic clustering using nature-inspired metaheuristics: A survey,” *Appl. Soft Comput. J.*, vol. 41, pp. 192–213, 2016, doi: 10.1016/j.asoc.2015.12.001.

[37] S. Wei, Z. Li, and C. Zhang, “A semi-supervised clustering ensemble approach integrated constraint-based and metric-based,” in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, Aug. 2015, pp. 1–6. doi: 10.1145/2808492.2808518.

[38] A. Strehl and J. Ghosh, “Cluster ensembles - A knowledge reuse framework for combining multiple partitions,” *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 583–617, Apr. 2003, doi: 10.1162/153244303321897735.

[39] R. Rajabioun, “Cuckoo optimization algorithm,” *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5508–5518, Dec. 2011, doi: 10.1016/J.ASOC.2011.05.008.

[40] Z. Cheng *et al.*, “Improvement and application of adaptive hybrid cuckoo search algorithm,” *IEEE Access*, vol. 7, pp. 145489–145515, 2019, doi: 10.1109/access.2019.2944981.

[41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” in *IEEE Transactions On Evolutionary Computation*, vol. 6, no. 2, M. Celebi and K. Aydin, Eds. 2002, pp. 182–197. doi: https://doi.org/10.1007/978-3-319-24211-8_9.

[42] U. Nacional de La Plata Argentina Cagnina,

C. Coello, L. Plata, L. Cagnina, S. Esquivel, and C. A. Coello Coello, "A particle swarm optimizer for multi-objective optimization universidad nacional de la plata a particle swarm optimizer for multi-objective optimization," *J. Comput. Sci. Technol.*, vol. 5, pp. 204–210, 2005, Accessed: Dec. 22, 2023. [Online]. Available: https://www.redalyc.org/articulo.oa?id=638067343017

[43] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007, doi: 10.1109/TEVC.2007.892759.

[44] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Appl. Intell.*, vol. 48, no. 4, pp. 805–820, Apr. 2018, doi: 10.1007/s10489-017-1019-8.

[45] H. T. Kahraman and S. Duman, "Multi-objective adaptive guided differential evolution for multi-objective optimal power flow incorporating wind-solar-small hydro-tidal energy sources," in *Studies in Computational Intelligence*, vol. 1009, B. V. Kumar, D. Oliva, and P. N. Suganthan, Eds. Springer, Singapore, 2022, pp. 341–365. doi: 10.1007/978-981-16-8082-3_13.

[46] A. Taghizabet, J. Tanha, A. Amini, and J. Mohammadzadeh, "A semi-supervised clustering approach using labeled data," *Sci. Iran.*, vol. 30, no. 1 D, pp. 104–115, Jan. 2023, doi: 10.24200/sci.2022.58519.5772.

[47] R. Janani, S. Vijayarani, and E. S. with A. 2019, "Text document clustering using spectral clustering algorithm with particle swarm optimization," *Elsevier*, vol. 134, pp. 192–200, 2019, doi: https://doi.org/10.1016/j.eswa.2019.05.030.