Soft Computing (2024) 28:991–1008
https://doi.org/10.1007/s00500-023-09367-3

DATA ANALYTICS AND MACHINE LEARNING

# Hybrid multi-objective metaheuristic and possibilistic intuitionistic fuzzy c-means algorithms for cluster analysis

R. J. Kuo[1] · C. C. Hsu[1,2] · Thi Phuong Quyen Nguyen[3] · C. Y. Tsai[4]

Accepted: 10 October 2023 / Published online: 4 November 2023

## Abstract

This study proposes a hybrid multi-objective meta-heuristics and possibilistic intuitionistic fuzzy *c*-means (PIFCM) algorithms for cluster analysis. The PIFCM algorithms combine Atanassov's intuitionistic fuzzy sets (IFSs) with possibilistic fuzzy *c*-means (PFCM) algorithms. In this study, three metaheuristic algorithms are used to improve the clustering results, specifically a genetic algorithm (GA), a particle swarm optimization (PSO) algorithm, and a gradient evolution (GE) algorithm. Therefore, three clustering algorithms are proposed, including multi-objective GA-based PIFCM (MOGA–PIFCM), multi-objective PSO-based PIFCM (MOPSO–PIFCM), and multi-objective GE-based PIFCM (MOGE–PIFCM). Their performance results are compared with those of other clustering algorithms, such as intuitionistic fuzzy *c*-means (IFCM) algorithm, possibilistic intuitionistic fuzzy *c*-means (PIFCM) algorithm, single-objective GA–PIFCM algorithm, single-objective PSO–PIFCM algorithm, and single-objective GE–PIFCM algorithm using 15 benchmark datasets. Adjusted Rand Index and accuracy measures are employed as performance indices for comparison. Experiment results shows that the MOGE–PIFCM obtains better solutions than the other clustering algorithms in terms of all performance validation indices.

**Keywords** Cluster analysis · Meta-heuristics · Genetic algorithm · Particle swarm optimization algorithm · Gradient evolution algorithm · Intuitionistic fuzzy sets · Possibilistic fuzzy *c*-means algorithm · Multiple objectives

**Abbreviations**

| | |
|---|---|
| FCM | Fuzzy *c*-means |
| IFS | Intuitionistic fuzzy sets |
| IFCM | Intuitionistic fuzzy *c*-means |
| PFCM | Possibilistic fuzzy *c*-means |
| PIFCM | Possibilistic intuitionistic fuzzy *c*-means |
| GA | Genetic algorithm |
| PSO | Particle swarm optimization |
| GE | Gradient evolution |
| MOGA–PIFCM | Multi-objective genetic algorithm-based PIFCM |
| MOPSO–PIFCM | Multi-objective particle swarm optimization-based PIFCM |
| MOGE–PIFCM | Multi-objective gradient evolution-based PIFCM |
| $\pi$ | Global cluster compactness |
| *Sep* | Fuzzy separation |
| ARI | Adjusted Rand Index |

## 1 Introduction

Clustering analysis can be used to explore the structure of data to understand the characteristics of that data for many practical issues. Therefore, clustering analysis has been applied in many different fields such as customer segmentation in marketing (Kuo et al. 2002), bill of material clustering in manufacturing industries (Agard and Penz 2009), and characteristic clustering in various fields

✉ Thi Phuong Quyen Nguyen
ntpquyen@dut.udn.vn

[1] Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei, Taiwan

[2] Ingrasys Technology Inc., No. 1188, Nanqing Rd., Luzhu Dist., Taoyuan, Taiwan

[3] Faculty of Project Management, The University of Danang - University of Science and Technology, 54 Nguyen Luong Bang, Danang, Vietnam

[4] Department of Industrial Engineering and Management, Yuan Ze University, No. 135, Yuan-Tung Road, Chungli, Taoyuan, Taiwan

(Kahali et al. 2019; Kannan et al. 2017; Milligan and Sokol 1980). Many different clustering algorithms have been proposed to date, including the *K*-means algorithm (MacQueen 1967), the fuzzy *c*-means (FCM) algorithm (Bezdek 1981; Dunn 1973), the fuzzy *K*-prototype algorithm (Ji et al. 2012), and so on. Each algorithm has its own pros and cons, making each appropriate for application in certain cases.

The FCM algorithm is one of the most popular cluster analysis methods. In most databases, an uncertainty commonly exists. To deal with uncertainty and hesitation, fuzzy sets and intuitionistic fuzzy sets (IFS) were proposed by Zadeh (1965) and Atanassov (1986), respectively. IFS deal with the uncertainty and hesitation which commonly occur in human thinking. Thus, the intuitionistic fuzzy *c*-means (IFCM) algorithm was proposed and extended for clustering interval-valued datasets (Xu and Wu 2010). The IFCM algorithm has been proven to achieve better performance than the FCM algorithm (Chaira 2011; Lin 2014).

In addition, the possibilistic fuzzy *c*-means (PFCM) algorithm is also an improvement of the FCM algorithm, which simultaneously calculates memberships and possibilities (Pal et al. 2005). The PFCM algorithm overcomes the problems of noise and coincident clusters. The algorithm is interesting and provides a good solution. However, the PFCM algorithm still has a drawback of sensitivity to the initialization. Moreover, the PFCM classifies the unlabeled dataset according to feature vectors without take into account the qualitative information. Therefore, it may lead to an incorrect assignment of membership (typicality) values to their clusters during the clustering process (Verma and Agrawal 2015).

Motivated by the contributions of the IFS, and the potential of the PFCM algorithm, this study proposes a possibilistic intuitionistic fuzzy *c*-means (PIFCM) algorithm, which combines both IFS and possibility concepts, to address the improper assignment of membership and typicality values to their clusters.

To improve clustering results and overcome the drawback of initialization sensitivity, multi-objective fuzzy clustering algorithms, which combine meta-heuristic algorithms and the proposed PIFCM algorithm, are also developed to consider multi-objective functions simultaneously. Numerous multi-objective fuzzy clustering algorithms have been proposed. Nguyen and Kuo (2019) combined Non-Dominated Sorting Particle Swarm Optimization to fuzzy clustering for categorical data in which global cluster compactness and fuzzy separation are the two objective functions. Kuo et al. (2022) proposed a framework of sequential clustering and classification that combined deep learning technique and multi-objective sine–cosine algorithm to explore the structure of data. Belhor et al. (2023) combined the Strength Pareto Evolutionary Algorithm (SPEA2) and Non-dominated Sorting Genetic Algorithm (NSGA-II) with *k*-means clustering and applied it to the routing and scheduling of home health care. Besides, multiple multi-objective fuzzy clustering algorithm have been developed and applied for image segmentation (Singh et al. 2022; Singh and Muchahari 2023; Zhao et al. 2022a, 2022b).

In this study, three metaheuristic algorithms are employed, i.e., genetic algorithm (GA), particle swarm optimization (PSO) algorithm, and gradient evolution (GE) algorithm. GA is a heuristic search algorithm inspired by the theory of evolutionary genetics. Generally, GA is used to generate high-quality solutions to optimization and search problems by mutation, crossover and selection processes. Similarly, PSO algorithms are also an evolutionary method that try to look for an optimal solution by moving all the particles in a population around a search space. At every move, the position and velocity of the particles are updated. The best-known position of each step is found. The particles will move toward the best solutions. Correspondingly, GE is a new meta-heuristic optimization algorithm inspired by the gradient-based search technique. GE algorithm consists of three operators, namely vector updating, jumping, and refreshing, which are updated based on a gradient theorem. GE provides a promising result compared with those of other meta-heuristic methods such as PSO or differential evolution (Kuo and Zulvia 2015).

The proposed algorithms integrate several meta-heuristic approaches with the PIFCM algorithm. Herein, three meta-heuristic algorithms, i.e., GA, PSO, and GE are employed. First, the PIFCM algorithm is implemented. Thereafter, the clustering result (the cluster centroids) is used as the initial population of the meta-heuristic algorithms. Moreover, multi-objective optimization, which uses global cluster compactness ($\pi$), and 2) fuzzy separation (*Sep*) as two objective functions, is applied to balance the clustering result. Thus, these combinations form three multi-objective evolutionary methods with the PIFCM algorithms, i.e., multi-objective genetic algorithm-based PIFCM (MOGA–PIFCM), multi-objective particle swarm optimization-based PIFCM (MOPSO–PIFCM), and multi-objective gradient evolution-based PIFCM (MOGE–PIFCM). The proposed algorithms are expected to improve the quality of clustering performance.

The remainder of this paper is organized as follows. Section 2 presents the related literature survey. Section 3 describes the proposed clustering algorithm, while the model evaluation results are illustrated in Sect. 4. Finally, concluding remakes are made in Sect. 5.

## 2 Literature review

This section provides some concepts related to data clustering, which include several clustering algorithms. Moreover, some metaheuristic approaches for both single-objective optimization and multi-objective optimization are also reviewed.

### 2.1 Fuzzy *c*-means algorithm

Fuzzy sets are like sets whose elements have degrees of membership, and were proposed to deal with vagueness and uncertainty (Zadeh 1965). The fuzzy set theory allows one piece of data to belong to two or more clusters. This represents the similarity of a point shared with each cluster with a membership function whose values are between zero and one (Dunn 1973; Tan 2006; Zadeh 1965).

FCM is one of the most popular clustering algorithms. The FCM algorithm was developed by Dunn (1973) to detect well-separated clusters, and was improved by Bezdek (1981) to solve the clustering problem. The algorithm can be described as follows. Given a dataset of $n$ data instancces $X = \{x_1, x_2, \ldots, x_n\}$ and $c$ clusters $V = \{v_1, v_2, \ldots, v_c\}$, the FCM algorithm partitions X into $c$ fuzzy clusters, forming a fuzzy partition in data instances. A fuzzy partition can be conveniently represented as a matrix $U = [u_{ij}]_{n\times c}$ whose elements $u_{ij} \in [0, 1]$ are the membership degree of data instances $x_i$ in cluster$v_j$. The sum of $u_{ij}$ for a data instance $x_i$ on $c$ clusters should be equal 1. These two constraints are presented in the following equations:

$$0 \le u_{ij} \le 1 i = 1, 2, \ldots, nj = 1, 2, \ldots, c, \tag{1}$$

$$\sum_{j=1}^{c} u_{ij} = 1 i = 1, 2, \ldots n. \tag{2}$$

The FCM minimizes the distance-based objective function which is defined as follows:

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c} [u_{ij}]^m \times {d_{ij}}^2(x_i, v_j), \tag{3}$$

where $m$ is a weighting exponent ($m \in [1, +\infty)$ that control the influence of membership degree, $d_{ij}(x_i, v_j)$ is the Euclidean distance between $x_i$ and cluster center$v_j$. Suppose that the given dataset contains $p$ attributes. A data instance $x_i$ and a cluster center $v_j$ are presented as $x_i = \{x_{1i}, x_{2i}, \ldots, x_{pi}\}$ and $v_j = \{v_{1j}, v_{2j}, \ldots, v_{pj}\}$, respectively. The distance $d_{ij}(x_i, v_j)$ is calculated as follows:

$$d_{ij}(x_i, v_j) = \sqrt{\sum_{a=1}^{p} (x_{ia} - v_{ja})^2}, \tag{4}$$

where $x_{ia}$ is the $a^{\text{th}}$ attribute value of data instance $x_i$, and $v_{ja}$ is the $a^{\text{th}}$ attribute value of cluster center $v_j$, $a = 1, 2, \ldots, p$.

To minimize $J$, membership $u_{ij}$ and the centers of cluster $v_j$ are updated as:

$$ {u_{ij}}^{(t+1)} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}(x_i, v_j)}{d_{ij}(x_i, v_k)}\right)^{\frac{2}{m-1}}} \text{ and} \tag{5}$$

$$v_j = \frac{\sum_{i=1}^{n} [u_{ij}]^m x_i}{\sum_{i=1}^{n} [u_{ij}]^m} j = 1, 2, \ldots, c. \tag{6}$$

This iteration will stop when $\left|J^{(t+1)} - J^{(t)}\right| \le \varepsilon$, where $\varepsilon$ is a termination condition, whereas $t$ is the iteration step.

The FCM algorithm does, however, have some drawbacks, i.e., its sensitivity to the initialization, and its sensitivity to noise and outlier data.

### 2.2 Intuitionistic fuzzy *c*-means algorithm

This section describes intuitionistic fuzzy theory and the intuitionistic fuzzy *c*-means (IFCM) algorithm.

#### 2.2.1 Intuitionistic fuzzy theory

Intuitionistic fuzzy set (IFS) theory was proposed by Atanassov (1986) to handle uncertainty. Intuitionistic fuzzy sets extend Zadeh's fuzzy sets and consider both membership and non-membership.

An IFS, $A$, in a fixed set, $E$, is an objective of the expression. The model of intuitionistic fuzzy sets is defined by Atanassov (1989) as:

$$A = \{x, u_A(x), v_A(x) | x \in E\}, \tag{7}$$

where the function $u_A(x) \to [0, 1]$ denotes the membership, and $v_A(x) \to [0, 1]$ denotes the non-membership degree of the element in set $A$.

When $u_A(x) + v_A(x) = 1$ for every $x \in E$, then the IFS becomes a fuzzy set. For all IFS, Atanassov also considers a hesitation degree, $\pi_A(x)$. Therefore, the IFS function is defined as:

$$\pi_A(x) = 1 - u_A(x) - v_A(x), 0 \le \pi_A(x) \le 1. \tag{8}$$

With regard to intuitionistic fuzzy sets, Yager proposed an intuitionistic fuzzy generator by defining the complement of an intuitionistic fuzzy set (Burillo and Bustince 1996). Hence, an intuitionistic fuzzy complement with Yager-generating functions can be written as:

$$N(x) = (1 - x^\alpha)^{1/\alpha}, \alpha > 0 \quad \textit{where} \quad N(0) = 1 \quad \textit{and} \quad N(1) = 0, \tag{9}$$

where α is the parameter in Yager-generating functions, for each value of parameter $\alpha \in (0, \propto)$.

With the help of Yager's intuitionistic fuzzy compliment, the intuitionistic fuzzy set becomes:

$$A = \left\{x, u_A(x), (1 - u_A(x)^{\alpha})^{1/\alpha} | x \in E\right\} \tag{10}$$

and the hesitation degree is:

$$\pi_A(x) = 1 - u_A(x) - (1 - u_A(x)^{\alpha})^{1/\alpha}. \tag{11}$$

### 2.2.2 Intuitionistic fuzzy *c*-means algorithm

To integrate intuitionistic fuzzy theory into a conventional FCM algorithm, the intuitionistic fuzzy membership values are obtained as:

$$u_{ij}{}^{*} = u_{ij} + \pi_{ij}, \tag{12}$$

where $u_{ij}{}^{*}(u_{ij})$ denotes the intuitionistic (conventional) fuzzy membership degree of the data point $i$ in cluster $j$.

The cluster center of the IFCM is updated as:

$$v_j{}^{*} = \frac{\sum_{i=1}^{n} [u_{ij}{}^{*}] x_i}{\sum_{i=1}^{n} [u_{ij}{}^{*}]} j = 1, 2, \ldots, c. \tag{13}$$

Using Eq. (13), the cluster center and the degree of membership matrix are updated simultaneously. The IFCM optimizes the objective function by continuously updating the membership functions and centers of clusters until stopping criteria ε is satisfied.

The IFCM algorithm can deal with uncertainty and vagueness; however, it cannot solve the problems of clustering noise and outlier data.

## 2.3 Possibilistic fuzzy *c*-means algorithm

Possibilistic fuzzy *c*-means (PFCM) is an improvement of FCM which calculates memberships and possibilities simultaneously (Pal et al. 2005). PFCM can overcome the sensitivity to noise and outlier data of the FCM algorithm. Moreover, the drawback of coincident cluster problems is also solved in the PFCM algorithm. PFCM has a parameter which controls the influence of outliers on centroids. Therefore, the clustering results are expected to be more compact. PFCM introduces the typicality of data $x_i$ with respect to cluster $v_j$, notated by $t_{ij}$. The objective function of PFCM is defined in the following equation as:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} \left(a u_{ij}^{m} + b t_{ij}^{\eta}\right) d_{ij}(x_i, v_j)^2. \tag{14}$$

Here, $t_{ij}$ and $v_j$ are calculated based on Eqs. (15) and (16), respectively, as follows:

$$t_{ij} = \frac{1}{1 + \left(\frac{b}{\gamma_j} d(x_i, v_j)^2\right)^{\frac{1}{(\eta - 1)}}} \text{and} \tag{15}$$

$$v_j = \frac{\sum_{i=1}^{n} \left(a u_{ij}^{m} + b t_{ij}^{\eta}\right) x_i}{\sum_{i=1}^{n} \left(a u_{ij}^{m} + b t_{ij}^{\eta}\right)}. \tag{16}$$

The sum of all of a cluster center's $t_{ij}$ vlaues should be 1:

$$0 \le t_{ij} \le 1 i = 1, 2, \ldots, n j = 1, 2, \ldots, c, \tag{17}$$

$$\sum_{i=1}^{n} t_{ij} = 1 j = 1, 2, \ldots c, \tag{18}$$

where $a$ and $b$ are parameters defining the relative importance of fuzzy membership and typicality values. If $a > b$, then the centroids have a higher influence than the membership values. Therefore, to reduce the effect of the outliers, the value of $b$ should be higher than $a$. $\eta$ is a real number that governs the influence of typicality values, and $\eta$>1. In addition, $\gamma_j$>0 is defined as:

$$\gamma_j = K \frac{\sum_{i=1}^{n} u_{ij}^{m} \times d_{ij}(x_i, v_j)^2}{\sum_{i=1}^{n} u_{ij}^{m}}, \tag{19}$$

where $K > 0$, and the most common is $K = 1$.

The PFCM algorithm has been applied in many areas. Verma and Agrawal (2015) proposed an algorithm integrating Atanassov's IFS and the PFCM algorithm, used in the segmentation of human brain images. Emary et al. combined a cuckoo search algorithm with the PFCM algorithm for retinal vessel segmentation (Emary et al. 2014). Ji et al. (Ji et al. 2011) proposed a modified PFCM algorithm which employed the novel adaptive method to deal with intensity inhomogeneities and noise for brain magnetic resonance (MR) image segmentation. Similarly, Aparajeeta et al. also extended the PFCM algorithm for MR image segmentation with a bias field (Ji et al. 2011). Sarkar et al. developed a Rough Possibilistic Type-2 Fuzzy C-Means clustering to cluster MR images with overlapping areas (Sarkar et al. 2016). Recently, Memon et al. combined PFCM with kernel function and local information for image segmentation (Memon et al. 2019). Chen et al. (2021) improved the PFCM performance by considering weight component to assign membership and typicality. Wu and Zhang (2022) proposed an iterative weighted PFCM that not only adopted the principle of maximum entropy construct a new objective function and redesigned the weight coefficient but also determined the optimal number of clusters. Wu and Peng (2023) combined PFCM with type-2 fuzzy set to enhance the initilization and noise sensitivity of the original PFCM. Besides, the PFCM is also integrated with instance-level infomation, i.e., labeled patterns, to become semi-supervised clustering (Antoine et al. 2022).

The PFCM algorithm can achieve good performance in many real-world applications. However, the combination of evolutionary methods with conventional clustering algorithms always provides better performance. Thus, the next sub-section will review some meta-heuristic algorithms that are used in the proposed methods.

## 2.4 Multi-objective optimization algorithm

Multi-objective optimization deals with more than one objective function. In many real-world cases, several objectives must be optimized simultaneously. In considering multi-objective optimization, it is difficult to compare one solution with another. Therefore, a domination term is applied. Suppose that there are two solutions, $x_1$ and $x_2$. Solution $x_1$ dominates $x_2$ if $x_1$ has a better objective value for at least one of the objective functions and is not worse with respect to the remaining objective functions. A non-dominated solution which has no solution can be found that dominates it. These non-dominated solutions are recorded in Pareto optimal solutions. The goal of multi-objective optimization is to find the entire Pareto front. The non-dominated and Pareto optimality are shown in Fig. 1. The two objectives F1 and F2 are minimized simultaneously. Solutions X1, X2, X3, X4, X5, and X6 are not dominated by any other solution. Therefore, the set represents the Pareto optimal set.

### 2.4.1 Multi-objective genetic algorithm

Genetic algorithm (GA) is an optimization technique that simulates the phenomenon of natural selection, described by Charles Darwin. The basic theory of GA was proposed by Holland (1975). In GA, the characteristics of the population are expressed using genotypes, and are called chromosomes. Each chromosome represents a potential solution to a problem. GA has three main steps: (1) selection, (2) crossover, and (3) mutation.

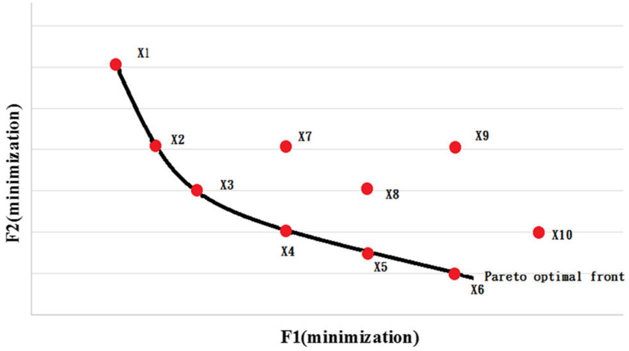

Fig. 1 Non-domination and Pareto optimality

Deb et al. (2002) proposed a fast and elitist multi-objective GA (NSGA-II) which not only contains non-dominated sorting, but also uses elitism to accelerate convergence. NSGA-II uses the fast non-dominated sorting process to compare each solution. The parent and offspring are merged and compared to find all non-dominated solutions. If identical non-dominated fronts are produced, then values are compared to pick a better solution as the next parent using crowding distance.

The non-dominated sorting process for selecting elitist offspring is illustrated in Fig. 2.

### 2.4.2 Multi-objective particle swarm optimization algorithm

PSO is a heuristic technique proposed by Eberhart and Kennedy (1995). In the PSO algorithm, potential solutions are called particles, and each particle corresponds to a fitness value. Each particle has its velocity and position, and these are corrected by individual and group search experience.

In PSO, particles' velocities are dynamically adjusted according to their historical behaviors, and the function is evaluated with the new coordinate for every iteration.

For multi-objective applications, Li (2003) proposed the non-dominated sorting particle swarm optimizer (NSPSO). The NSPSO compares all personal best solutions of all particles and their offspring in the population to find all dominant solutions. If two first non-dominated fronts are produced, then values are compared to pick a better solution as the next particle using crowding distance.

### 2.4.3 Multi-objective gradient evolution algorithm

The gradient evolution (GE) algorithm was proposed by Kuo and Zulvia (2015). It was developed from the Newton–Raphson algorithm. The GE algorithm consists of three operators, namely vector updating, jumping, and refreshing. Vector updating is the main operator in the GE algorithm, and is responsible for exploitation. The function used in vector updating was inspired by the updating rule in the Newton–Raphson algorithm. Vector updating is continued by vector jumping, which performs wider exploration.

The vector updating rule controls the vector movement to reach a better position. It consists of two parts. The first part is a gradient-based updating rule, and the second part is the acceleration factor. To expand the exploration, a random number $r_g$ is added to the updating rule. The updating rule by GradientMove is:

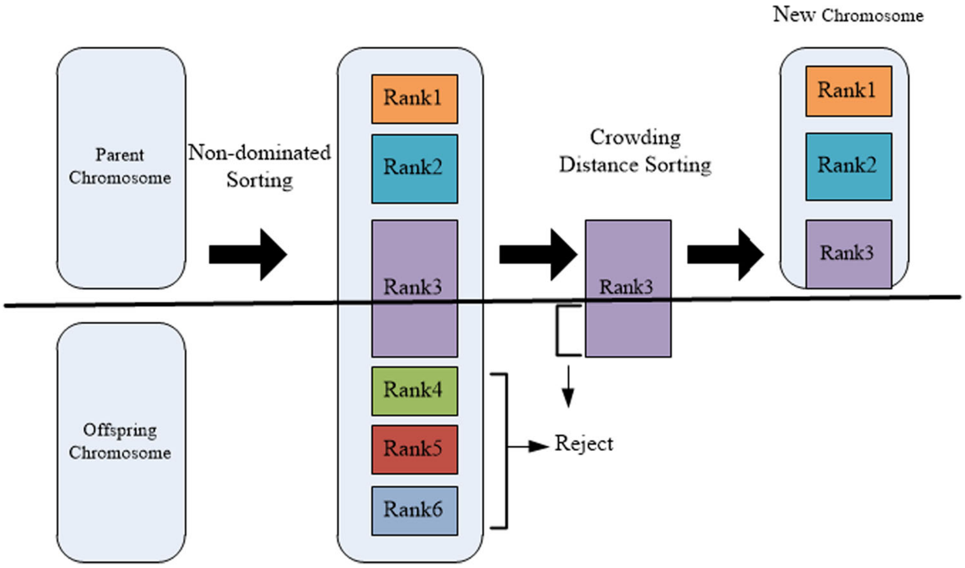

**Fig. 2** The non-dominated sorting process for selecting elitist offspring

$$\text{Gradient Move} = \left(r_g \cdot \frac{\Delta x_{ij}^t}{2}\right) \cdot \left(\frac{x_{ij}^W - x_{ij}^B}{x_{ij}^W - 2x_{ij}^t + x_{ij}^B}\right), \forall \mathrm{j} = 1, \ldots, \mathrm{D}, \tag{20}$$

$$\Delta x_{ij}^t = \left(\frac{\left|x_{ij}^t - x_{ij}^B\right| + \left|x_{ij}^W - x_{ij}^t\right|}{2}\right), \forall j = 1, \ldots, D, \tag{21}$$

where $X_i^B$ is a neighbor of vector $i$ that has better fitness, and $X_i^W$ is a neighbor of vector $i$ that has worse fitness.

The acceleration factor *Acc* is also embedded to accelerate the convergence of each vector. The *Acc*, defined in Eq. (22), is also multiplied by a random number $r_a \sim \mathrm{N}$ (0, 1). This ensures that each vector has a different step size:

$$Acc = r_a \cdot \left(y_j - x_{ij}^t\right), \forall \mathrm{j} = 1, \ldots, \mathrm{D}, \tag{22}$$

where Y is the best vector.

Finally, vector updating is written as:

$$u_{ij}^t = x_{ij}^t - \text{GradientMove} + \text{Acc}, \tag{23}$$

$$u_{ij}^t = x_{ij}^t - \left(r_g \cdot \frac{\Delta x_{ij}^t}{2}\right) \cdot \left(\frac{x_{ij}^W - x_{ij}^B}{x_{ij}^W - 2x_{ij}^t + x_{ij}^B}\right) + r_a \cdot \left(y_j - x_{ij}^t\right), \forall \mathrm{j} = 1, \ldots, \mathrm{D}, \tag{24}$$

where $U_i^t$ is the vector transition.

Because this process involves better and worse vectors, it requires an additional procedure for best and worst vectors. If $X_i^t$ is the best vector $B^t$, then $x_{ij}^B$ is replaced with $b_j$, as defined in Eq. (25):

$$b_j = x_{ij}^t - \Delta x_{ij}^t, \forall \mathrm{j} = 1, \ldots, \mathrm{D} \text{ and} \tag{25}$$

$$\Delta x_{ij}^t = \left(\frac{\gamma + \left|x_{ij}^W - x_{ij}^t\right|}{2}\right), \forall \mathrm{j} = 1, \ldots, \mathrm{D}. \tag{26}$$

Similarly, if the worst vector is $W^t$, then $x_{ij}^W$ is replaced with $w_j$, as defined in Eq. (27):

$$w_j = x_{ij}^t + \Delta x_{ij}^t, \forall \mathrm{j} = 1, \ldots, \mathrm{D} \text{ and} \tag{27}$$

$$\Delta x_{ij}^t = \left(\frac{\left|x_{ij}^t - x_{ij}^B\right| + \gamma}{2}\right), \forall \mathrm{j} = 1, \ldots, \mathrm{D}, \tag{28}$$

where $\gamma$ is a pre-defined parameter, and is defined as the initial step size. Therefore, it is a static or dynamic number, which decreases as the number of iterations increases.

Vector jumping can avoid local optima. The GE algorithm sets a jumping rate $J_r$ to determine whether a vector jumps or not. Vector jumping is defined as the following equation:

$$u_{ij}^t = -u_{ij}^t + r_m \cdot \left(u_{ij}^t - x_{kj}^t\right), \forall \mathrm{j} = 1, \ldots, \mathrm{D}, \tag{29}$$

where $x_{kj}^t$ is any random neighbor vector, $\forall \mathrm{i} \neq k$, and $r_m$ is a random number within [0,1].

Vector refreshing is performed on a vector which is stuck in a position. The GE also records the updating history of vector $i$, which is rotated by $s_i \in [0, 1]$. If $s_i$ is less than the refreshing rate, vector $i$ must be regenerated, and the newly generated vector, $s_i$, is set to one. $s_i$ is reduced by the following equation:

$$s_i = s_i - \varepsilon \cdot s_i, \tag{30}$$

where $\varepsilon$ is the reduction rate.

Kuo and Zulvia (2020) proposed a GE-based $K$-means algorithm which considers two objectives. It tries to minimize dissimilarity within the cluster, and maximize cluster separateness.

# 3 Methodology

This section introduces the proposed multi-objective clustering algorithms and the possibilistic intuitionistic fuzzy $c$-means (PIFCM) algorithm.

## 3.1 Objective functions

The proposed methods consider two objective functions: (1) global cluster compactness (π), and (2) fuzzy separation (*Sep*) (Yang et al. 2015), which will be optimized simultaneously. Given the information of $n$ data points and $c$ clusters, the global compactness $\pi$ of a solution represented by the cluster centers is defined as:

$$\pi = \sum_{j=1}^{c} \frac{\sum_{i=1}^{n} u_{ij}^{m} d_{ij}(x_{i,} v_j)}{\sum_{j=1}^{n} u_{ij}}. \tag{31}$$

In Eq. (31), the term $\sum_{i=1}^{n} u_{ij}^{m} d_{ij}(x_{i,} v_j)$ and $\sum_{j=1}^{n} u_{ij}$ are represented for the variation $\sigma_j$ and the fuzzy cardinality $n_j$ of the $j^{th}$ cluster, respectively (Mukhopadhyay et al. 2009). Herein, $u_{ij}$ is the membership degree of data instance $x_i$ to cluster $j$, and $d_{ij}(x_{i,} v_j)$ is the distance between $x_i$ and cluster center $v_j$, which is defined in Sect. 2.1. For each cluster, the cluster compactness is the ratio of $\sigma_j / n_j$. Thus, the global compactness is calculated as a summation of cluster compactness for all clusters. A good partition is indicated by a low global cluster compactness.

To calculate the fuzzy separation, the cluster center $v_j$ is considered as the center of a fuzzy set that contains the remaining cluster centers $v_k$ in which $\{v_k | 1 \le k \le c, k \ne j\}$. The membership degree of each cluster center $v_k$ to $v_j, j \,/\!\!= k$ is calculated as follows:

$$\mu_{jk} = \left( \sum_{l=1, l\ne k}^{c} \left( \frac{d_{kj}(v_k, v_j)}{d_{kl}(v_k, v_l)} \right)^{\frac{1}{m-1}} \right)^{-1}, j \ne k. \tag{32}$$

Subsequently, the fuzzy separation *Sep* is a distance-based measure between the cluster centers which is defined by the product of membership degree and distance of each cluster center $v_k$ to $v_j$:

$$Sep = \sum_{j=1}^{c} \sum_{k=1, j\ne k}^{c} \mu_{jk}^{m} d_{jk}(v_j, v_k). \tag{33}$$

To obtain compact clusters, the measure π must be as small as possible or minimized. Conversely, to obtain well-separated clusters, *Sep* should be as large as possible or maximized.

## 3.2 Possibilistic intuitionistic fuzzy $c$-means algorithm

This study proposes the PIFCM algorithm which integrates the advantages of the FCM, possibilistic $c$-means (PCM) algorithms, and the IFS. Thereafter, the PIFCM is combined to the multi-objective meta-heuristic algorithms to improve the clustering results. The steps of PIFCM are described below.

**Step 1:** Initialize the parameters. Set $T = 0$. Set up number of clusters $c$, membership degree $m$, possibilistic $\eta$, $\gamma_j$, tolerance ε, Yager index α, and possibilistic parameters $a$and$b$, and generate initial membership functions U and $t$.

**Step 2:** Calculate hesitation degree $\pi_{ij}$ by Eq. (11), and $u_{ij}{}^{*}$ by Eq. (12).

**Step 3:** Calculate $v_j{}^{*}$ by Eq. (16), where $u_{ij}$ is replaced by $u_{ij}{}^{*}$.

**Step 4:** Calculate $J^{(T)}$ by Eq. (14).

**Step 5:** Update $t_{ij}{}^{*(T+1)}$ by Eq. (15).

**Step 6:** Update $u_{ij}{}^{*(T+1)}$ by Eq. (5).

**Step 7:** Update $v_j{}^{*(T+1)}$ by Eq. (16).

**Step 8:** Calculate $J^{(T+1)}$ by Eq. (14).

**Step 9:** Compare $J^{(T)}$ and $J^{(T+1)}$. If $\left| J^{(T+1)} - J^{(T)} \right| \le \varepsilon$, then stop; otherwise, increase $T$ by one and return to Step 5.

## 3.3 Multi-objective meta-heuristic algorithm-based PIFCM algorithm

PIFCM is probably not robust because the dimension dataset is large. Thus, to improve the accuracy, this study proposes meta-heuristic algorithm-based PIFCM algorithm. The main idea is to provide the meta-heuristic algorithm with a set of better initial centroids so that it can obtain good clustering results. Moreover, this study considers two objective functions as the cluster validity measurements to provide a better clustering result.

### 3.3.1 Multi-objective genetic algorithm-based PIFCM algorithm

This sub-section describes MOGA–PIFCM. As in GA, the MOGA–PIFCM procedure includes population initialization, fitness computation, selection, crossover, mutation, and termination criteria. In this study, the MOGA–PIFCM uses the clustering result of PIFCM to generate the initial chromosomes. Herein, each chromosome represents a set of alternative centroids. Figure 3 illustrates the solution representation. It shows that a chromosome consists of $D$

× $K$ bits, where $D$ represents the number of attributes, and $K$ represents the number of centroids.

The selection strategy process is based on the roulette wheel method (Zhao et al. 1996). Selection of parents is in accordance with the probability distribution of the fitness values. The selected chromosomes are put into the mating pool for further genetic operations. This study uses the method introduced by Krishna and Murty (1999). The probability of roulette wheel selection is defined as the following equation:

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j}, \tag{34}$$

where the $P_i$ is the selection probability of chromosome $i$ $(1 \leq i \leq N)$, $f_i$ and $f_j$ are the fitness of chromosome $i$ and chromosome $j$ $(1 \leq j \leq N)$, respectively.

The well-known rank-based evaluation function is used to evaluate the fitness functions (Gan et al. 2009):

$$f_i = \beta(1-\beta)^{r_i - 1}, \tag{35}$$

where the $r_i$ is the rank of chromosome $i$, and $\beta \in [0, 1]$ is a parameter which is the selective pressure of the algorithm.

In the crossover procedure, this study generates two new chromosomes using the following function (Michielssen et al. 1992):

$$X^{\text{new}} = \alpha X + (1-\alpha) Y \text{and} \tag{36}$$

$$Y^{\text{new}} = \beta X + (1-\beta) X, \tag{37}$$

where $X^{\text{new}}$ and $Y^{\text{new}}$ are the genes of offspring after the crossover process, $\alpha$ and $\beta$ is a random number between 0 and 1, and $X$ and $Y$ are the genes of parents which are selected from all of the chromosomes.

In the mutation procedure, this study applies a function to change the genes of chromosomes. The function is represented as follows (Sumathi et al. 2008):

$$X^{\text{new}} = X + s \times r \times a, \text{and} \tag{38}$$

$$a = 2^{-uk}, \tag{39}$$

where $X^{\text{new}}$ is the genes of offspring after the mutation process, $s \in \{-1, 1\}$ is uniform at random, $r$ is mutation range and the standard is 10%, $u$ is a random number between 0 and 1, and $k \in \{4, 5, \ldots .20\}$ is mutation precision.

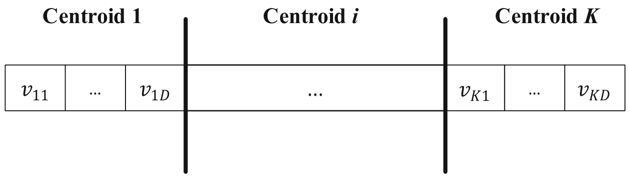

**Fig. 3** Solution representation

The MOGA–PIFCM procedure can be described as follows:

**Step 1:**Set up parameters of GA including population size, crossover rate, mutation rate and cluster number.

**Step 2:** Initialization: Initial chromosomes are derived from PIFCM.

**Step 3:** Compute the two objective values as fitness functions.

**Step 4:** Apply non-dominated sorting to rank the current chromosomes.

**Step 5:** Use roulette wheel selection to select $N$ chromosomes.

**Step 6:** Randomly select two chromosomes to be parent chromosomes, and check the probability of crossover to decide whether the crossover process must be applied.

**Step 7:** Check the probability of mutation to determine whether mutation should take place.

**Step 8:** Compute the two objective values as fitness functions.

**Step 9:** Implement non-dominated sorting for offspring and parent chromosomes.

**Step 10:** Select good chromosomes using crowding distance.

**Step 11:** Stop if the termination criterion is satisfied; otherwise, return to Step 5.

**Step 12:** Select the optimal solution from the set of non-dominated solutions.

### 3.3.2 Multi-objective particle swarm optimization-based PIFCM algorithm

Van der Merwe and Engelbrecht (2003) proposed two methods using a PSO algorithm for cluster analysis. The first method is the PSO clustering algorithm (PSOC), and the second method is the PSO-based $K$-means clustering algorithm (PSOKC). In this study, MOPSO–PIFCM is a hybrid of PSOC which combines the multi-objective optimization algorithm and PIFCM.

In this study, MOPSO–PIFCM uses the clustering result of PIFCM to set the initial particles, and uses inertia weight to control the influence of the previous velocity. The solution representation of each particle is the same as that used in the GA.

For inertia weight, this study equilibrates the global search ability and local search ability. When the inertia weight is large, PSO tends to exploitation, and is similar to global search methods. When inertia weight is smaller, PSO tends to exploration, and can find the best value in the local area. This is also similar to local search methods.

Initially, a large inertia weight is used to ferform an overall search. Then, it is reduced as the number of iterations increases to achieve convergence (Shi and Eberhart 1998). The inertia weight is as follows (Li et al. 2011):

$$w_t = \frac{t^{\max} - t}{t^{\max}} \times \left(w^{\max} - w^{\min}\right) + w^{\min}, \quad (40)$$

where $w_t$ is the inertia weight of the number of iterations $t$, $t^{\max}$ is the maximum of number of iterations t, $w^{\max}$ is the maximum inertia weight, and $w^{\min}$ is the minimum inertia weight.

In this study, the MOPSO–PIFCM uses position and velocity adjustment to obtain the optimal solution. The velocity and position of a particle are updated as follows:

$$v_i^{t+1} = w_t \times v_i^t + c_1 \times r_1 \times \left(\text{pBest}_i - x_i^t\right) + c_2 \times r_2 \times \left(\text{gBest} - x_i^t\right)\text{and} \quad (41)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (42)$$

where $r_1$ and $r_2$ are two random numbers uniformly distributed between 0 and 1, and $c_1$ and archive as a repository for non-dominated solutions and choosing members of the archive to direct further search. The elite archive accepts a new position of a particle if it is non-dominated by all the stored solutions. All dominated members of the archive are deleted.

The selection of pBest simply replaces the previous best experience with the current position if the former does not dominate the latter. This is implemented using the archive members that dominate the fewest particles in the current iteration as the gBest (Mostaghim and Teich 2003).

The MOPSO–PIFCM procedure is as follows:

**Step 1:**Set up parameters including population size, maximum velocity, inertia weight, two learning factors, and cluster number $c$.

**Step 2:** Set up the initial particles, from PIFCM.

**Step 3:** Compute the two objective values as fitness functions.

**Step 4:** Implement non-dominated sorting.

**Step 5:** Find pBest and gBest.

**Step 6:** Calculate inertia weight.

**Step 7:** Update the position and velocity of each particle by Eqs. (41) and (42).

**Step 8:** Compute the two objective values as fitness functions.

**Step 9:** Implement non-dominated sorting including new particles and pBest.

**Step 10:** Generate elitist particles.

**Step 11:** Stop if the termination criterion is satisfied; otherwise, return to Step 5.

**Step 12:** Select the optimal solution.

### 3.3.3 Multi-objective gradient evolution-based PIFCM algorithm

In this study, the proposed MOGE–PIFCM improves the conventional GE algorithm and combines PIFCM. The improvements of the GE algorithm include the modification of vector updating to be applied to multi-objective problems, and the crowding distance to manage the diversity of the solutions in the archive. Similarly, this study also considers two objective functions as the cluster validity measurements.

PIFCM is responsible for providing a good initial solution for MOGE–PIFCM. The solution representation of each vector is the same as that used in GA. To deal with more than one objective, an archive management strategy is applied, which uses a crowding distance concept to maintain a good spread of solutions.

The MOGE–PIFCM procedure is as follows:

**Step 1:**Set up parameters including population size, jumping rate, refreshing rate, and cluster number $c$.

**Step 2:** Set up the initial vectors, from PIFCM.

**Step 3:** Compute the two objective values as fitness functions.

**Step 4:** Perform non-dominated sorting.

**Step 5:** Find the best vector and worst vector, and record the best vector.

**Step 6:** Calculate the vector updating by Eq. (24).

**Step 7:** Calculate the vector jumping by Eq. (29).

**Step 8:** Calculate the vector refreshing.

**Step 9:** Compute the two objective values as fitness functions.

**Step 10:** Implement non-dominated sorting including new vectors and old vectors.

**Step 11:** Generate elitist vectors.

**Step 12:** Stop if the termination criterion is satisfied; otherwise, return to Step 5.

**Step 13:** Select the optimal solution.

## 3.4 Selecting the optimal solution

In multi-objective optimization problems, the solution in the Pareto front usually contains multiple options which do not dominate each other. The final solution is usually picked up randomly from the set of non-dominated solutions, since these solutions are considered equal. This study adopts a method for selecting the optimal solution from Pareto non-inferior solutions, proposed by Wang et al. (2017). The performance-price ratio is used as a reference to construct the average variability with adjacent non-inferior solutions corresponding to the objective function values.

The average variability is the average value of the slopes of the lines connecting any point except endpoints with two adjacent points. Let ${k_1}^{(m)}$ and ${k_2}^{(m)}$ be the average variability of the two objective values ${f_1}^{(m)}$ and ${f_2}^{(m)}$ of Pareto non-inferior solution $x^{(m)}$, respectively. The average variability is defined as:

$$k_1{}^{(m)} = \frac{1}{2}\left(\frac{f_2{}^{(m)} - f_2{}^{(m-1)}}{f_1{}^{(m)} - f_1{}^{(m-1)}} + \frac{f_2{}^{(m+1)} - f_2{}^{(m)}}{f_1{}^{(m+1)} - f_1{}^{(m)}}\right), m = 2, 3\ldots(M-1)\text{and} \tag{43}$$

$$k_2{}^{(m)} = \frac{1}{2}\left(\frac{f_1{}^{(m)} - f_1{}^{(m-1)}}{f_2{}^{(m)} - f_2{}^{(m-1)}} + \frac{f_1{}^{(m+1)} - f_1{}^{(m)}}{f_2{}^{(m+1)} - f_2{}^{(m)}}\right), m = 2, 3\ldots(M-1). \tag{44}$$

The average variability of the endpoints in the Pareto front is defined as:

$$k_1{}^{(1)} = \frac{f_2{}^{(2)} - f_2{}^{(1)}}{f_1{}^{(2)} - f_1{}^{(1)}}, \tag{45}$$

$$k_2{}^{(1)} = \frac{f_1{}^{(2)} - f_1{}^{(1)}}{f_2{}^{(2)} - f_2{}^{(1)}}, \tag{46}$$

$$k_1{}^{(M)} = \frac{f_2{}^{(M)} - f_2{}^{(M-1)}}{f_1{}^{(M)} - f_1{}^{(M-1)}}, \text{and} \tag{47}$$

$$k_2{}^{(M)} = \frac{f_1{}^{(M)} - f_1{}^{(M-1)}}{f_2{}^{(M)} - f_2{}^{(M-1)}}. \tag{48}$$

Then, the sensitivity ratio, which is similar to the performance-price, ratio is obtained, and a quantitative method is developed to evaluate Pareto non-inferior solutions. The sensitivity ratio reflects the sensitivity degree of average variability in the objective function values. The sensitivity ratio is defined as:

$$\delta_1{}^{(m)} = \frac{k_1{}^{(m)}}{f_1{}^{(m)}}, m = 1, 2\ldots M \text{ and} \tag{49}$$

$$\delta_2{}^{(m)} = \frac{k_2{}^{(m)}}{f_2{}^{(m)}}, m = 1, 2\ldots M. \tag{50}$$

For comparison, the sensitivity ratio must be non-dimensionalized. It is defined as:

$$\varepsilon_1{}^{(m)} = \frac{\delta_1{}^{(m)}}{\sum_{i=1}^{M} \delta_1{}^{(i)}}, m = 1, 2\ldots M \text{ and} \tag{51}$$

$$\varepsilon_2{}^{(m)} = \frac{\delta_2{}^{(m)}}{\sum_{i=1}^{M} \delta_2{}^{(i)}}, m = 1, 2\ldots M. \tag{52}$$

Finally, the solution corresponding to $(\Delta\varepsilon)_{min}$ is the unbiased and good solution which is simultaneously most suitable for $f_1{}^{(m)}$ and $f_2{}^{(m)}$. The $(\Delta\varepsilon)_{min}$ is defined as:

$$(\Delta\varepsilon)_{min} = \min\left\{\Delta\varepsilon^1, \Delta\varepsilon^2\ldots\Delta\varepsilon^M\right\} \text{ and} \tag{53}$$

$$\Delta\varepsilon^{(m)} = \left|\varepsilon_1{}^{(m)} - \varepsilon_2{}^{(m)}\right|, m = 1, 2\ldots M. \tag{54}$$

## 4 Experimental results

This section presents the results of experiments conducted in this study to evaluate the performance of the proposed clustering algorithms. Fifteen datasets from the UCI machine learning repository were used to test the functions of the proposed algorithms.

### 4.1 Datasets

This research used fifteen datasets including Ukm, Wine, Wbc, Tae, Vehicle, Pima, Iris, Breast, Liver, Banknote, Audit, Fertility, Seed, Haberman, and Vertebral to validate the clustering performance. In addition, the Wbc, Wine, and Pima datasets have outliers. Of these three datasets, Wbc has 239 noise points (35%), Wine has 10 noise points (5.6%), and Pima has 268 noise points (34.9%). Table 1 shows the details of each dataset.

### 4.2 Performance measurement

To compare clustering results, this study used two clustering validation indices as the performance measurement. The first index is the clustering accuracy, which depends on the correct cluster (Graves and Pedrycz, 2010).

The second index is the Adjusted Rand Index (*ARI*) which is a popular measure for comparing clustering results (Hubert and Arabie 1985; Rand 1971; Vinh et al. 2010). Given a set of $n$ objects $S$, suppose $U = \{u_1, u_2, \ldots, u_c\}$ and $V = \{v_1, v_2, \ldots, v_c\}$ represent two different partitions of the objects in $S$. Let $n_{ij}$ be the count

**Table 1** The parameters of datasets

| Dataset | # of instances | # of attributes | # of clusters |
|---|---|---|---|
| Ukm | 403 | 5 | 4 |
| Wine | 178 | 13 | 3 |
| Wbc | 683 | 9 | 2 |
| Tae | 151 | 5 | 3 |
| Vehicle | 846 | 18 | 4 |
| Pima | 768 | 8 | 2 |
| Iris | 150 | 4 | 3 |
| Breast | 106 | 9 | 6 |
| Liver | 345 | 6 | 2 |
| Banknote | 1372 | 4 | 2 |
| Audit | 772 | 17 | 2 |
| Fertility | 100 | 9 | 2 |
| Seed | 210 | 7 | 3 |
| Haberman | 306 | 3 | 2 |
| Vertebral | 310 | 6 | 3 |

in both the predicted cluster $u_i$ and the correct cluster $v_j$. The notations are described in the contingency table (Table 2).

The *ARI* is computed as follows:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right] - \left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]\Big/\binom{n}{2}}. \quad (55)$$

The two selected validation indices belong to the group of the external indices. The external validation indices measure the similarity between the clustering result obtained by the proposed algorithms and the true class labels. *ARI* and accuracy are not an exception. However, the *ARI* are calculated based on the pair-counting measures in the contingency table, while the accuracy is computed by the sum of all the objects of one-to-one mapping between the obtained clusters and true labels. ARI and accuracy have been used simultaneously for clustering evaluation in many studies (Cao et al. 2017, 2013; Chen et al. 2016; Chen and Huang 2019).

### 4.3 Parameter settings

The parameters should be pre-determined before the computational experiment is conducted because different parameters will influence the experiment results. First, the level of each parameter was determined based on previous studies. Thereafter, this study used the Taguchi method to determine the parameters for the PIFCM algorithm and the multi-objective meta-heuristic algorithms (Kackar 1985; Taguchi 1986). Every algorithm was run 30 times, and the number of iterations was set as 100. The other parameters were determined using Taguchi methods for each dataset.

The $\alpha$ in the Yager-generating function was set as 2 (Lin 2014) to calculate the hesitation degree. Four parameters must be determined for the PIFCM algorithm, namely the relative importance of the fuzzy membership $a$, relative importance of typicality values $b$, influence of membership grades $m$ and influence of typicality values $\eta$. The relative importance of fuzzy membership and typicality values were constrained to $a + b = 1$. Thus, $b = 1\text{-}a$ and the relative importance of fuzzy membership were set as 0.2, 0.5, and 0.8. Table 3 shows the three different levels of factors for the PIFCM algorithms.

In GA, the population size, crossover rate, and mutation rate must be determined. According to previous work, this study determined three different levels of factors (Michielssen et al. 1992).

Four parameters must be determined for the PSO algorithm, namely the number of particles ($n$), learning factor 1 ($c1$), learning factor 2 ($c2$), and inertia weight ($W$). The values for $c1$ and $c2$ were 0.5 and 1.495, and 2 (Jiang et al. 2014; Kuo et al. 2012). The maximum inertia weight was set as 0.9, and reduced until it reached 0.4, according to a previous study (Shi 2001).

For the GE algorithm, the number of vectors, jumping rate and refreshing rate must be determined. Table 4 summarizes all the parameters se tup for multi-objective-based clustering algorithms. The parameters of the proposed algorithms were determined by Taguchi method. The $L_9(3^3)$ orthogonal array was used to analyze the parameters in this study.

The Taguchi method using the Minitab 17 package can show the best combination of parameters for the proposed algorithms. Tables 5 and 6 show the best combination of parameters for each dataset.

### 4.4 Computational results

Since each algorithm is a heuristic, each run will give a different result. Therefore, running the algorithm multiple times will yield a better chance of finding a good result. The average result is adopted after multiple runs. In this study, each algorithm was run 30 times for each dataset, and the average accuracy and the *ARI* as the performance measurement were then calculated. Three proposed multi-objective algorithms were compared with other existing algorithms. Table 7 shows the results in terms of accuracy for each dataset, while Table 8 shows the results in terms of ARI.

According to the results, the three proposed multi-objective algorithms based on PIFCM exhibited better performances than those of the other clustering algorithms on

**Table 2** Contingency table

| $U$<br>$V$ | $u_1$ $u_2$ …$u_c$ | Sum |
|---|---|---|
| $v_1$ | $n_{11}$ $n_{12}$ … $n_{1c}$ | $a_1$ |
| $v_2$ | $n_{21}$ $n_{22}$ … $n_{2c}$ | $a_2$ |
| … | … … … | … |
| $v_c$ | $n_{c1}$ $n_{c2}$ …$n_{cc}$ | $a_c$ |
| Sum | $b_1$ $b_2$ …$b_c$ | $n$ |

**Table 3** Parameters setup for the PIFCM algorithm

| Factors | Level 1 | Level 2 | Level 3 |
|---|---|---|---|
| $a$ | 0.2 | 0.5 | 0.8 |
| $m$ | 1.05 | 1.1 | 2 |
| $\eta$ | 0.6 | 2 | 7 |

**Table 4** Parameters setup for all metaheuristic-based PIFCM clustering algorithms

| Method | Factors | Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| MOGA–PIFCM | Population size ($N_s$) | 50 | 80 | 100 |
| | Crossover rate ($p_c$) | 0.85 | 0.9 | 0.95 |
| | Mutation rate ($p_m$) | 0.001 | 0.01 | 0.1 |
| MOPSO–PIFCM | Number of particles ($N_p$) | 50 | 80 | 100 |
| | $c1$ | 0.5 | 1.495 | 2 |
| | $c2$ | 0.5 | 1.495 | 2 |
| MOGE–PIFCM | Number of vectors ($N_v$) | 50 | 80 | 100 |
| | Jumping rate ($J_r$) | 0.85 | 0.9 | 0.95 |
| | Refreshing rate ($R_r$) | 0.7 | 0.75 | 0.8 |

**Table 5** The parameters of PIFCM for all datasets

| | $a$ | $b$ | $m$ | $\eta$ |
|---|---|---|---|---|
| Ukm | 0.5 | 0.5 | 1.1 | 7 |
| Wine | 0.8 | 0.2 | 1.1 | 7 |
| Wbc | 0.8 | 0.2 | 2 | 2 |
| Tae | 0.2 | 0.8 | 2 | 7 |
| Vehicle | 0.5 | 0.5 | 1.05 | 2 |
| Pima | 0.8 | 0.2 | 1.05 | 7 |
| Iris | 0.8 | 0.2 | 2 | 2 |
| Breast | 0.5 | 0.5 | 1.1 | 2 |
| Liver | 0.8 | 0.2 | 1.05 | 7 |
| Banknote | 0.2 | 0.8 | 2 | 2 |
| Audit | 0.8 | 0.2 | 2 | 2 |
| Fertility | 0.8 | 0.2 | 2 | 2 |
| Seed | 0.8 | 0.2 | 2 | 1.1 |
| Haberman | 0.8 | 0.2 | 2 | 2 |
| Vertebral | 0.5 | 0.5 | 2 | 2 |

most of the tested datasets. For instance, the three proposed multi-objective algorithms obtained better results on 12 (Ukm, Wine, Wbc, Tae, Vehicle, Pima, Liver, Audit, Fertility, Seed, Haberman, and Vertebral) of the 15 tested datasets in terms of all performance validation indices.

For outlier datasets, the MOGE–PIFCM achieved better results for the Wine and Pima datasets, while the MOPSO–PIFCM and MOGA–PIFCM achieved better results for the Wbc dataset, with respect to accuracy comparison. Compared with the MOGE–PIFCM and IFCM performances with outlier datasets, the MOGE–PIFCM always achieved better results than the IFCM. This is reasonable, since hybridizing evolutionary methods (GA, PSO, and GE) with PIFCM can remove the drawback caused by initialization of PIFCM, and enhance the clusteirng result. Of the three algorithms, i.e., MOGE, MOGA, and MOPSO, the combination of MOGE with PIFCM achieved the best results due to its advantage of vector jumping, which can handle outliers. Therefore, the MOGE–PIFCM algorithm can address the outlier problem.

Because GA was originally proposed for binary coding, and was later modified for continuous coding certain objectives (π or *Sep*) often easily dominate the results. Therefore, GA is affected by continuous code, which causes it to generate unstable results.

Next, the performances of multi-objective-based PIFCM and single-objective-based PIFCM algorithms were compared. MOGA–PIFCM and GA–PIFCM were compared first. The accuracy values of the all the tested datasets showed that there is no significant difference between these GA-based algorithms. Because GA was originally proposed for binary coding and was later modified for continuous coding, the GA–PIFCM algorithm with single objective (π or *Sep*) often easily dominates the results.Thus MOGA–PIFCM algorithm is affected by continuous coding, which causes it to generate unstable results.

In the comparison between the MOPSO–PIFCM and PSO–PIFCM algorithms, the MOPSO–PIFCM produced better results in eleven datasets (Ukm, Wbc, Tae, Vehicle, Iris, Breast, Liver, Audit, Fertility, Seed, and Vertebral) in terms of accuracy. Compared with GE–PIFCM, the MOGE–PIFCM achieves better results in ten datasets in terms of accuracy (Ukm, Wine, Wbc, Tae, Vehicle, Pima, Breast, Banknote, Audit, and Fertility). For the five remaining datasets, there was not much difffence between MOGE and GE-based PIFCM algorithms. It can be concluded that the clustering results of both MOPSO–PIFCM and MOGE–PIFCM algorithms are better than those of single objective PSO–PIFCM and GE–PIFCM, respectively. The multi-objective-based algorithms achieve this because they can balance the cluster compactness and fuzzy seperation to enhance the clustering result. Moreover, employing the proposed technique to select the optimal solution in the Pareto front contributes to improving the obtained results. In addition, it can be recognized that the MOGE–PIFCM algorithm is the best in terms of accuracy with a dominant result in seven datasets (Ukm, Wbc, Wine, Tae, Vehicle, Pima, and Audit).

**Table 6** The parameters of meta-heuristic-based PIFCM algorithms for all datasets

| | MOGA–PIFCM | | | MOPSO–PIFCM | | | MOGE–PIFCM | | |
|---|---|---|---|---|---|---|---|---|---|
| | $N_s$ | $p_c$ | $p_m$ | $N_s$ | *c1* | *c2* | $N_v$ | $J_r$ | $R_r$ |
| Ukm | 80 | 0.85 | 0.01 | 100 | 1.495 | 2 | 100 | 0.95 | 0.75 |
| Wine | 50 | 0.9 | 0.01 | 50 | 2 | 2 | 100 | 0.85 | 0.8 |
| Wbc | 100 | 0.85 | 0.1 | 100 | 0.5 | 0.5 | 100 | 0.85 | 0.8 |
| Tae | 50 | 0.85 | 0.001 | 100 | 0.5 | 0.5 | 100 | 0.9 | 0.7 |
| Vehicle | 50 | 0.9 | 0.001 | 100 | 0.5 | 2 | 100 | 0.95 | 0.75 |
| Pima | 100 | 0.95 | 0.01 | 100 | 0.5 | 2 | 100 | 0.85 | 0.8 |
| Iris | 100 | 0.85 | 0.001 | 100 | 0.5 | 2 | 50 | 0.95 | 0.8 |
| Breast | 100 | 0.95 | 0.01 | 50 | 1.495 | 1.5 | 100 | 0.9 | 0.7 |
| Liver | 100 | 0.85 | 0.1 | 100 | 2 | 1.5 | 100 | 0.85 | 0.8 |
| Banknote | 80 | 0.9 | 0.1 | 100 | 0.5 | 1.5 | 100 | 0.9 | 0.7 |
| Audit | 80 | 0.85 | 0.001 | 100 | 0.5 | 2 | 100 | 0.85 | 0.8 |
| Fertility | 100 | 0.85 | 0.001 | 100 | 0.5 | 2 | 100 | 0.85 | 0.8 |
| Seed | 100 | 0.85 | 0.01 | 100 | 0.5 | 2 | 100 | 0.9 | 0.8 |
| Haberman | 100 | 0.85 | 0.001 | 100 | 0.5 | 2 | 100 | 0.85 | 0.8 |
| Vertebral | 100 | 0.85 | 0.01 | 100 | 0.5 | 2 | 100 | 0.85 | 0.8 |

**Table 7** The computation results in terms of accuracy

| | IFCM | PIFCM | GA–PIFCM (π) | GA–PIFCM (*Sep*) | PSO–PIFCM (π) | PSO–PIFCM (*Sep*) | GE–PIFCM (π) | GEPIFCM (*Sep*) | MOGA–PIFCM (π, *Sep*) | MOPSO–PIFCM (π, *Sep*) | MOGE–PIFCM (π, *Sep*) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ukm | 0.498 | 0.507 | 0.470 | 0.506 | 0.454 | 0.401 | 0.473 | 0.439 | 0.503 | 0.491 | **0.513** |
| Wine | 0.955 | 0.955 | 0.955 | 0.955 | 0.955 | 0.840 | 0.957 | 0.959 | 0.955 | 0.889 | **0.962** |
| Wbc | 0.956 | **0.974** | 0.973 | **0.974** | 0.969 | 0.966 | 0.972 | 0.966 | **0.974** | **0.974** | **0.974** |
| Tae | 0.497 | 0.472 | 0.404 | 0.427 | 0.442 | 0.473 | 0.452 | 0.432 | 0.409 | 0.512 | **0.515** |
| Vehicle | 0.368 | 0.378 | **0.384** | **0.384** | 0.376 | 0.377 | **0.384** | 0.381 | **0.384** | **0.384** | **0.384** |
| Pima | 0.667 | 0.667 | 0.665 | 0.666 | 0.668 | 0.655 | 0.666 | 0.663 | 0.667 | 0.651 | **0.682** |
| Iris | 0.893 | **0.920** | **0.920** | 0.914 | 0.901 | 0.886 | **0.920** | 0.902 | 0.919 | 0.915 | 0.916 |
| Breast | 0.500 | 0.489 | 0.567 | **0.614** | 0.553 | 0.562 | 0.573 | 0.438 | 0.595 | 0.595 | 0.591 |
| Liver | 0.513 | 0.539 | 0.539 | 0.542 | 0.521 | 0.565 | 0.525 | 0.558 | 0.541 | **0.574** | 0.555 |
| Banknote | 0.607 | 0.675 | 0.818 | 0.609 | 0.669 | 0.639 | 0.640 | 0.643 | **0.792** | 0.649 | 0.657 |
| Audit | 0.687 | 0.720 | 0.746 | 0.775 | 0.765 | 0.695 | 0.757 | 0.757 | 0.775 | 0.773 | **0.781** |
| Fertility | 0.510 | 0.533 | 0.552 | 0.553 | 0.516 | 0.517 | 0.531 | 0.532 | **0.554** | 0.526 | 0.533 |
| Seed | 0.891 | 0.898 | **0.899** | **0.899** | 0.891 | 0.830 | 0.894 | 0.873 | **0.899** | 0.892 | 0.885 |
| Haberman | 0.520 | **0.524** | 0.523 | 0.522 | 0.518 | 0.523 | 0.517 | **0.524** | **0.524** | 0.518 | 0.516 |
| Vertebral | 0.474 | 0.561 | 0.642 | 0.625 | 0.640 | 0.516 | 0.640 | 0.552 | 0.637 | **0.643** | 0.632 |

The bold values show the best accuracy

The experiment results also show that the comparison results in terms of *ARI* are similar to those of comparison in terms of accuracy. The three proposed multi-objective-based PIFCM algorithms had the highest *ARI* value for seven datasets. The MOGE–PIFCM was also the best clustering algorithm in this experiment, achieving the best results for nine of the fifteen datasets.

Next, computational times were compared, as shown in Table 9. According to the comparison, the meta-heuristic-based PIFCM algorithms require more time than the IFCM and PIFCM algorithms. This is reasonable because the meta-heuristic-based PIFCM algorithms use the clustering result of PIFCM to generate the initial clusters. However, the computational time required by the meta-heuristic-based PIFCM algorithms is acceptable since the computational time of most datasets is less than two minutes. Only one dataset, Vehicle, which contains 846 instances and 18 attributes, requires more comptation time. In

**Table 8** The computation results in terms of *ARI*

| | IFCM | PIFCM | GA–PIFCM (π) | GA–PIFCM (*Sep*) | PSO–PIFCM (π) | PSO–PIFCM (*Sep*) | GE–PIFCM (π) | GE–PIFCM (*Sep*) | MOGA–PIFCM (π, *Sep*) | MOPSO–PIFCM (π, *Sep*) | MOGE–PIFCM π, *Sep*) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ukm | 0.193 | 0.194 | 0.190 | 0.182 | 0.191 | 0.086 | 0.193 | 0.154 | 0.188 | 0.185 | **0.194** |
| Wine | 0.867 | 0.867 | 0.867 | 0.867 | 0.868 | 0.585 | 0.873 | 0.876 | 0.867 | 0.757 | **0.886** |
| Wbc | 0.830 | 0.897 | 0.894 | 0.897 | 0.881 | 0.867 | 0.890 | 0.869 | 0.897 | **0.899** | 0.897 |
| Tae | 0.050 | 0.054 | 0.013 | 0.029 | 0.036 | 0.072 | 0.042 | 0.050 | 0.016 | 0.080 | **0.081** |
| Vehicle | 0.079 | 0.082 | **0.090** | **0.090** | 0.087 | 0.072 | **0.090** | 0.087 | **0.090** | **0.090** | **0.090** |
| Pima | 0.101 | 0.101 | 0.100 | 0.100 | 0.102 | 0.026 | 0.104 | 0.038 | 0.101 | 0.050 | **0.105** |
| Iris | 0.729 | **0.786** | **0.786** | 0.773 | 0.751 | 0.714 | 0.786 | 0.749 | 0.783 | 0.777 | 0.778 |
| Breast | 0.295 | 0.308 | 0.423 | **0.481** | 0.413 | 0.376 | 0.417 | 0.245 | 0.468 | 0.446 | 0.451 |
| Liver | – 0.005 | – 0.005 | – 0.005 | – 0.005 | – 0.002 | – 0.007 | 0.002 | – 0.010 | – 0.005 | **0.008** | – 0.006 |
| Banknote | 0.045 | 0.155 | **0.416** | 0.055 | 0.165 | 0.078 | 0.119 | 0.082 | 0.360 | 0.136 | 0.126 |
| Audit | 0.130 | 0.224 | 0.251 | 0.299 | 0.311 | 0.144 | 0.285 | 0.261 | 0.299 | 0.295 | **0.313** |
| Fertility | – 0.006 | 0.000 | – 0.002 | 0.009 | – 0.004 | – 0.005 | – 0.002 | – 0.001 | **0.010** | – 0.005 | – 0.003 |
| Seed | 0.705 | 0.725 | 0.726 | 0.726 | 0.707 | 0.590 | 0.715 | 0.673 | **0.727** | 0.711 | 0.696 |
| Haberman | – 0.001 | 0.001 | – 0.001 | – 0.001 | – 0.002 | – 0.001 | – 0.002 | – 0.001 | **0.002** | – 0.002 | – 0.002 |
| Vertebral | 0.206 | 0.205 | 0.297 | 0.285 | 0.290 | 0.231 | 0.289 | 0.227 | 0.286 | 0.300 | **0.302** |

The bold values show the best ARI

addition, because the GE algorithm has more complex updating rules, it requires more time. Thus, the MOGE–PIFCM requires the greatest anount of time of all the compared algorithms. To reduce the required computational time, it may be feasible to conduct feature selection. In addition, from the current computational result, it seems that most meta-heuristics can improve

**Table 9** The computational time (in second)

| | IFCM | PIFCM | GA–PIFCM (π) | G–PIFCM (*Sep*) | PSO–PIFCM (π) | PSO–PIFCM (*Sep*) | G–PIFCM (π) | GE–PIFCM (*Sep*) | MOGA–PIFCM (π, *Sep*) | MOPSO–PIFCM (π, *Sep*) | MOGE–PIFCM (π, *Sep*) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ukm | 0.21 | 0.53 | 66 | 65 | 82 | 82 | 72 | 85 | 66 | 82 | 92 |
| Wine | 0.14 | 0.38 | 58 | 58 | 29 | 29 | 51 | 61 | 59 | 29 | 66 |
| Wbc | 0.25 | 0.70 | 106 | 106 | 105 | 106 | 94 | 111 | 107 | 106 | 119 |
| Tae | 0.06 | 0.15 | 23 | 22 | 22 | 22 | 19 | 23 | 23 | 22 | 24 |
| Vehicle | 1.13 | 3.31 | 249 | 247 | 494 | 494 | 434 | 517 | 249 | 494 | 552 |
| Pima | 0.25 | 0.71 | 108 | 108 | 108 | 107 | 96 | 113 | 110 | 108 | 122 |
| Iris | 0.05 | 0.12 | 19 | 19 | 19 | 19 | 8 | 10 | 20 | 19 | 11 |
| Breast | 0.24 | 0.45 | 53 | 52 | 26 | 26 | 46 | 55 | 53 | 26 | 58 |
| Liver | 0.09 | 0.25 | 38 | 38 | 38 | 38 | 33 | 40 | 39 | 38 | 42 |
| Banknote | 0.28 | 0.72 | 90 | 90 | 112 | 112 | 95 | 116 | 90 | 112 | 109 |
| Audit | 0.43 | 1.63 | 144 | 145 | 181 | 181 | 163 | 190 | 145 | 179 | 212 |
| Fertility | 0.08 | 0.10 | 35 | 35 | 35 | 35 | 31 | 37 | 35 | 35 | 39 |
| Seed | 0.03 | 0.01 | 13 | 14 | 14 | 14 | 12 | 14 | 14 | 15 | 14 |
| Haberman | 0.11 | 0.11 | 46 | 46 | 46 | 46 | 40 | 48 | 46 | 46 | 51 |
| Vertebral | 0.04 | 0.06 | 18 | 18 | 18 | 18 | 15 | 19 | 18 | 18 | 19 |

**Table 10** The comparison results of single objective and multiple objectives

| | MOGA–PIFCM vs.GA–PIFCM (π) | MOGA–PIFCM vs.GA–PIFCM (*Sep*) | MOPSO–PIFCM vs. PSO–PIFCM (π) | MOPSO–PIFCM vs. PSO–PIFCM (*Sep*) | MOGE–PIFCM vs. GE–PIFCM (π) | MOGE–PIFCM vs. GE–PIFCM (*Sep*) |
|---|---|---|---|---|---|---|
| Ukm | + | = | + | + | + | + |
| Wine | = | = | − | + | + | = |
| Wbc | + | = | = | + | = | + |
| Tae | = | − | + | + | + | + |
| Vehicle | + | = | + | + | = | + |
| Pima | + | + | − | = | + | + |
| Iris | = | + | + | + | − | + |
| Breast | + | − | + | + | + | + |
| Liver | + | − | + | + | + | − |
| Banknote | = | + | = | = | = | = |
| Audit | + | = | = | + | = | + |
| Fertility | = | = | = | = | = | = |
| Seed | = | = | + | = | = | + |
| Haberman | = | = | = | = | = | − |
| Vertebral | = | = | = | + | = | + |
| $\sum +$ | 7 | 3 | 7 | 10 | 6 | 10 |
| $\sum =$ | 8 | 9 | 6 | 5 | 8 | 3 |
| $\sum -$ | 0 | 3 | 2 | 0 | 1 | 2 |

clustering performance. Thus, choosing an easier metaheuristic is a viable alternative.

### 4.5 Statistical hypothesis

In this study, to prove whether the proposed clustering algorithms are significantly superior to other clustering algorithms or not, a statistical hypothesis was conducted. The $t$ test was, and the significance level $\alpha$ was set at 0.05. All statistical hypotheses were implemented on Minitab 17. The procedure of the statistic test is as follows. First, the two-sides test, in which the null hypothesis is "A = B", is conducted. If there is no evidence to reject the null hypothesis in the first test, the testing procedure is terminated. Then, it can be concluded that there is no significant difference between the two compared algorithms. If the null hypothesis in the first test is rejected, the second test is performed to determine whether "A < B" or "A > B". This procedure is implemented for all the tested datasets.

Table 10 shows the comparison results of single objectives and multiple objectives for all datasets. The symbol + means that the proposed algorithm performs better. Similarly, the symbol = means that there is no significant difference between the two compared algorithms, while the symbol – signifies that the proposed algorithm performs worse.

According to Table 10, the MOGA–PIFCM is significantly better than GA-PIFCM(π) for seven datasets, and there is no difference between MOGA–PIFCM and GA–PIFCM(π) for eight datasets. Compared with GA–PIFCM(Sep), MOGA–PIFCM obtains better results for three datasets, and worse results for three datasets, while there is no difference for the remaining datasets. Thus, it can be concluded that MOGA–PIFCM has better performance than GA–PIFCM(π). However, there is no significant difference between MOGA–PIFCM and GA–PIFCM(Sep).

The proposed MOPSO–PIFCM yields better results for seven datasets compared with PSO–PIFCM(π), and ten datasets compared with PSO–PIFCM(*Sep*). There are two datasets for which MOPSO–PIFCM obtains worse results comparedg with PSO–PIFCM(π). For the remaining datasets, there is no difference between MOPSO–PIFCM and other algorithms. Thus, it can be concluded that MOPSO–PIFCM has a better performance than the single-objective PSO algorithm.

Compared with GE–PIFCM(π), MOGE–PIFCM has better performance for six datasets, and only one dataset yields a worse result. For the remaining datasets, there is no difference between MOGE–PIFCM and GE–PIFCM(π) peromance. GE–PIFCM(*Sep*) achieves better results for ten datasets, equal results for three datasets, and worse results for two datasets. This is sufficent evidence that

MOGE–PIFCM is better than a single-objective GE algorithm. The three proposed multi-objective algorithms, therefore, all demonstrate better performance than the other compared algorithms.

## 5 Conclusions

This study proposed a PIFCM algorithm, which combines IFS and PFCM algorithms. Meta-heuristic algorithms were employed to improve clustering performance and avoid local optimal solutions of the PIFCM algorithm. Three multi-objective meta-heuristic algorithms, namely MOGA, MOPSO, and MOGE, were integrated with the PIFCM algorithm. For multi-objective optimization problems, selecting an optimal solution in the Pareto front is also important. Thus, this study also employed a new technique to select the clustering result for the three proposed multi-objective-based algorithms. The clustering performance of the proposed algorithms was verified using fifteen datasets from the UCI machine learning repository. To implement the proposed algorithms, parameters were determined by Taguchi method. This study applied the Taguchi method to find the optimal combination of parameters for each algorithm on each tested dataset. To evaluate the clustering results, accuracy and ARI were selected as two cluster validation indices. In addition, the results of the proposed multi-objective meta-heuristic-based PIFCM algorithms were compared with those of a conventional IFCM algorithm, a PIFCM algorithm, and its single-objective clustering algorithms. The experiment results showed that the proposed multi-objective-based PIFCM algorithms achieved better performance than the single-objective-based PIFCM algorithms in terms of both accuracy and ARI. In addition, a comparison between the three proposed multi-objective algorithms was made. The clustering result of the MOGE–PIFCM algorithm was better than those of the MOGA–PIFCM and MOPSO–PIFCM algorithms in most of the tested datasets.

However, in this study, the parameter $\alpha$ in the Yager-generating functions was applied based on previous studies. In the future research, this parameter will be optimized by the meta-heuristic algorithms to improve the accuracy of the clustering results. In addition, all problems in this study were continuous. Thus, future work will extend the results of this study for binary and discrete problems, since many real problems involve discrete and binary variables. Finally, the proposed algorithms will also be tested for high dimensional data in the future research.

**Author contributions** RJK and CYT: Supervision, Conceptualization, Writing—Reviewing and Editing. CCH: Software, Formal analysis, Validation, Writing—Original draft. TPQN: Conceptualization, Methodology, Formal analysis, Validation, Writing—Editing.

**Funding** This work was supported by The University of Danang, University of Science and Technology, code number of Project: T2023-02–16. This support is really appreciated.

**Data availability** All dataset are collected from UCI machine learning repository (https://archive.ics.uci.edu/ml/index.php).

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

Agard B, Penz B (2009) A simulated annealing method based on a clustering approach to determine bills of materials for a large product family. Int J Prod Econ 117:389–401

Antoine V, Guerrero JA, Romero G (2022) Possibilistic fuzzy c-means with partial supervision. Fuzzy Sets Syst 449:162–186. https://doi.org/10.1016/j.fss.2022.08.003

Atanassov KT (1986) Intuitionistic fuzzy sets. Fuzzy Sets Syst 20:87–96

Atanassov KT (1989) More on intuitionistic fuzzy sets. Fuzzy Sets Syst 33:37–45

Belhor M, El-Amraoui A, Jemai A, Delmotte F (2023) Multi-objective evolutionary approach based on K-means clustering for home health care routing and scheduling problem. Expert Syst Appl 213:119035. https://doi.org/10.1016/j.eswa.2022.119035

Bezdek JC (1981) Objective function clustering. In: Pattern recognition with fuzzy objective function algorithms. pp 43–93

Burillo P, Bustince H (1996) Entropy on intuitionistic fuzzy sets and on interval-valued fuzzy sets. Fuzzy Sets Syst 78:305–316

Cao F, Liang J, Li D, Zhao X (2013) A weighting k-modes algorithm for subspace clustering of categorical data. Neurocomputing 108:23–30

Cao F, Huang JZ, Liang J (2017) A fuzzy SV-k-modes algorithm for clustering categorical data with set-valued attributes. Appl Math Comput 295:1–15

Chaira T (2011) A novel intuitionistic fuzzy C means clustering algorithm and its application to medical images. Appl Soft Comput 11:1711–1717

Chen P-Y, Huang J-J (2019) A hybrid autoencoder network for unsupervised image clustering. Algorithms 12:122

Chen L, Wang S, Wang K, Zhu J (2016) Soft subspace clustering of categorical data with probabilistic distance. Pattern Recogn 51:322–332

Chen J, Zhang H, Pi D, Kantardzic M, Yin Q, Liu X (2021) A weight possibilistic fuzzy c-means clustering algorithm. Sci Program. https://doi.org/10.1155/2021/9965813

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197

Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybern 3:32–57

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Micro machine and human science, 1995. MHS'95. Proceedings of the sixth international symposium. IEEE, pp 39–43

Emary E, Zawbaa HM, Hassanien AE, Schaefer G, Azar AT (2014) Retinal vessel segmentation based on possibilistic fuzzy c-means clustering optimised with cuckoo search. In: 2014 International joint conference on neural networks (IJCNN). IEEE. pp 1792–1796

Gan G, Wu J, Yang Z (2009) A genetic fuzzy k-Modes algorithm for clustering categorical data. Expert Syst Appl 36:1615–1620

Holland JH (1975) Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. University of Michigan Press, Ann Arbor, pp 439–444

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2:193–218

Ji Z-X, Sun Q-S, Xia D-S (2011) A modified possibilistic fuzzy c-means clustering algorithm for bias field estimation and segmentation of brain MR image. Comput Med Imaging Graph 35:383–397

Ji J, Pang W, Zhou C, Han X, Wang Z (2012) A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. Knowl-Based Syst 30:129–135

Jiang B, Wang N, Wang L (2014) Parameter identification for solid oxide fuel cells using cooperative barebone particle swarm optimization with hybrid learning. Int J Hydrogen Energy 39:532–542. https://doi.org/10.1016/j.ijhydene.2013.09.072

Kackar RN (1985) Off-line quality control, parameter design, and the Taguchi method. J Qual Technol 17:176–188

Kahali S, Sing JK, Saha PK (2019) A new entropy-based approach for fuzzy c-means clustering and its application to brain MR image segmentation. Soft Comput 23:10407–10414. https://doi.org/10.1007/s00500-018-3594-y

Kannan SR, Devi R, Ramathilagam S, Hong TP (2017) Effective fuzzy possibilistic c-means: an analyzing cancer medical database. Soft Comput 21:2835–2845. https://doi.org/10.1007/s00500-016-2198-7

Krishna K, Murty MN (1999) Genetic K-means algorithm. IEEE Trans Syst Man Cybern Part B (cybernetics) 29:433–439

Kuo RJ, Zulvia FE (2015) The gradient evolution algorithm: a new metaheuristic. Inf Sci 316:246–265

Kuo RJ, Zulvia FE (2020) Multi-objective cluster analysis using a gradient evolution algorithm. Soft Comput 24:11545–11559

Kuo RJ, Ho LM, Hu CM (2002) Integration of self-organizing feature map and K-means algorithm for market segmentation. Comput Oper Res 29:1475–1493

Kuo RJ, Zulvia FE, Suryadi K (2012) Hybrid particle swarm optimization with genetic algorithm for solving capacitated vehicle routing problem with fuzzy demand – a case study on garbage collection system. Appl Math Comput 219:2574–2588. https://doi.org/10.1016/j.amc.2012.08.092

Kuo R, Setiawan MR, Nguyen TPQ (2022) Sequential clustering and classification using deep learning technique and multi-objective sine-cosine algorithm. Comput Ind Eng 173:108695

Li XA (2003) non-dominated sorting particle swarm optimizer for multiobjective optimization. Genetic and evolutionary computation conference. Springer, pp 37–48

Li Z, Li Y, Xu L (2011) Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization. In: Information technology, computer engineering and management sciences (ICM), 2011 International conference. IEEE, pp 157–161

Lin K-P (2014) A novel evolutionary kernel intuitionistic fuzzy c - means clustering algorithm. IEEE Trans Fuzzy Syst 22:1074–1087

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol 14. Oakland, CA, USA, pp 281–297

Memon KH, Memon S, Qureshi MA, Alvi MB, Kumar D, Shah RA (2019) Kernel possibilistic fuzzy c-means clustering with local information for image segmentation. Int J Fuzzy Syst 21:321–332

Van der Merwe D, Engelbrecht AP (2003) Data clustering using particle swarm optimization. In: Evolutionary computation, 2003. CEC'03. The 2003 Congress. IEEE, pp 215–220

Michielssen E, Ranjithan S, Mittra R (1992) Optimal multilayer filter design using real coded genetic algorithms. IEE Proc J (optoelectronics) 139:413–420

Milligan GW, Sokol LM (1980) A two-stage clustering algorithm with robust recovery characteristics. Educ Psychol Measur 40:755–759

Mostaghim S, Teich J (2003) Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: Swarm intelligence symposium, 2003. SIS'03. Proceedings of the 2003 IEEE. IEEE, pp 26–33

Mukhopadhyay A, Maulik U, Bandyopadhyay S (2009) Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. IEEE Trans Evol Comput 13:991–1005

Nguyen TPQ, Kuo R-J (2019) Automatic fuzzy clustering using non-dominated sorting particle swarm optimization algorithm for categorical data. IEEE Access 7:99721–99734

Pal NR, Pal K, Keller JM, Bezdek JC (2005) A possibilistic fuzzy c-means clustering algorithm. IEEE Trans Fuzzy Syst 13:517–530

Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66:846–850

Sarkar JP, Saha I, Maulik U (2016) Rough possibilistic type-2 fuzzy c-means clustering for MR brain image segmentation. Appl Soft Comput 46:527–536

Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization. International conference on evolutionary programming. Springer, pp 591–600

Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: evolutionary computation. Proceedings of the 2001 congress 2001. IEEE, pp 81–86

Singh P, Muchahari MK (2023) Solving multi-objective optimization problem of convolutional neural network using fast forward quantum optimization algorithm: application in digital image classification. Adv Eng Softw 176:103370. https://doi.org/10.1016/j.advengsoft.2022.103370

Singh M, Asha C, Sharma N (2022) Multi-objective particle swarm optimization based enhanced fuzzy c-means algorithm for the segmentation of MRI data. Recent trends in electronics and communication. Springer, pp 1031–1041

Sumathi S, Hamsapriya T, Surekha P (2008) Evolutionary intelligence: an introduction to theory and applications with Matlab. Springer Science & Business Media

Taguchi G (1986) Introduction to quality engineering: designing quality into products and processes.

Tan P-N (2006) Introduction to data mining. Pearson Education India

Verma H, Agrawal R (2015) Possibilistic intuitionistic fuzzy c-means clustering algorithm for MRI brain image segmentation. Int J Artif Intell Tools 24:1550016

Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854

Wang N, Zhao W-j, Wu N, Wu D (2017) Multi-objective optimization: a method for selecting the optimal solution from Pareto non-inferior solutions. Expert Syst Appl 74:96–104. https://doi.org/10.1016/j.eswa.2017.01.004

Wu C, Peng S (2023) Robust interval type-2 kernel-based possibilistic fuzzy clustering algorithm incorporating local and non-local information. Adv Eng Softw 176:103377. https://doi.org/10.1016/j.advengsoft.2022.103377

Wu C, Zhang X (2022) A self-learning iterative weighted possibilistic fuzzy c-means clustering via adaptive fusion. Expert Syst Appl 209:118280. https://doi.org/10.1016/j.eswa.2022.118280

Xu Z, Wu J (2010) Intuitionistic fuzzy C-means clustering algorithms. J Syst Eng Electron 21:580–590

Yang C-L, Kuo R, Chien C-H, Quyen NTP (2015) Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering. Appl Soft Comput 30:113–122

Zadeh LA (1965) Information and Control. Fuzzy Sets 8:338–353

Zhao F, Cao L, Liu H, Tang Z, Fan J (2022a) Particle competitive mechanism-based multi-objective rough clustering algorithm for image segmentation. IEEE Trans Fuzzy Syst. https://doi.org/10.1109/TFUZZ.2022.3141752

Zhao F, Liu Y, Liu H, Fan J (2022b) Broad learning approach to surrogate-assisted multi-objective evolutionary fuzzy clustering algorithm based on reference points for color image segmentation. Expert Syst Appl 200:117015. https://doi.org/10.1016/j.eswa.2022.117015

Zhao L, Tsujimura Y (1996) Gen M Genetic algorithm for fuzzy clustering. In: Evolutionary computation. Proceedings of IEEE international conference on. IEEE, pp 716–719

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.