Neurocomputing 205 (2016) 116–129

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An application of a metaheuristic algorithm-based clustering ensemble method to APP customer segmentation

R.J. Kuo [a,*], C.H. Mei [a], F.E. Zulvia [a], C.Y. Tsai [b]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei, Taiwan 106, ROC
[b] Department of Industrial Engineering and Management, Yuan Ze University, No. 135, Yuan-Tung Road, Chungli, Taoyuan, Taiwan, ROC

ARTICLE INFO

Article history:
Received 26 July 2015
Received in revised form
3 January 2016
Accepted 12 April 2016
Communicated by: Dr. Swagatam Das
Available online 7 May 2016

Keywords:
Cluster analysis
Clustering ensembles
Genetic algorithm
Particle swarm optimization
Artificial bee colony
Customer segmentation

ABSTRACT

This study proposes a metaheuristic-based clustering ensemble method. It integrates the clustering ensembles algorithm with the metaheuristic-based clustering algorithm. In the clustering ensembles, this study performs an improved generation mechanism and a co-association matrix in the co-occurrence approach. In order to improve the efficiency, a principle component analysis is employed. Furthermore, three metaheuristic-based clustering algorithms are proposed. This paper uses a real-coded genetic algorithm, a particle swarm optimization and an artificial bee colony optimization to combine with clustering ensembles algorithm. The experimental results indicate that the proposed metaheuristic-based clustering ensembles algorithms have better performance than metaheuristic-based clustering without clustering ensembles method. Furthermore, the proposed algorithms are applied to solve a customer segmentation problem. The real problem is come from a mobile application. Among all of the proposed algorithms, the artificial bee colony optimization-based clustering ensembles algorithm outperforms other algorithms. Therefore, the marketing strategy for the real application is made based on the best result.

1. Introduction

Cluster analysis has been applied in many different fields including data mining, image segmentation and document exploration. It aims to identify the data structure by organizing the data into several groups. A large variety of clustering algorithms have been proposed. They are K-means, fuzzy c-means, single linkage agglomerative hierarchical clustering (SLHC) and etc. However, there is no one algorithm which is suitable for all types of datasets [1–3]. Therefore, this paper proposes a combination of some clustering algorithms using a clustering ensemble method in order to improve the clustering methods and obtain better clustering results. Clustering ensemble gives a better solution than a single clustering algorithm in term of robustness, novelty and stability [1,4,5]. The proposed clustering ensemble method comprises of two steps. They are generation mechanism and consensus function. The generation mechanism generates the data partition. Then the process is continued by consensus function to integrate the results of the generation mechanism. Furthermore, three metaheuristic-based clustering algorithms are combined with the clustering ensembles. This paper employs three single clustering algorithms namely, the real-coded GA-based K-means clustering ensembles (GKCE), PSO-based K-means clustering ensembles (PSOKCE) and ABC-based K-means clustering ensembles (ABCKCE).

The validation is conducted using iris, wine, tae, flame, bank authentication and D31 datasets. Furthermore, the proposed method is applied to solve a customer segmentation problem. This real application is taken from a weight control APP.

The remainder of this paper is organized as follows. Section 2 presents the relevant background of this paper. Section 3 presents a novel clustering ensembles method. Furthermore, the validation is given in Section 4. In Section 5, application of the proposed method to the weight loss APP is discussed. Finally, concluding remarks are made in Section 6.

2. Literature review

This section briefly recalls some theories applied in this paper, including clustering methods, meta-heuristic algorithm-based clustering methods and clustering ensembles.

2.1. Cluster analysis

Clustering is an unsupervised data processing which classifies instances into some groups. Herein, a cluster should comprise of

* Corresponding author. Tel.: +886-2-27376328, fax: +886-2-27376344 .
E-mail address: rjkuo@mail.ntust.edu.tw (R.J. Kuo).

http://dx.doi.org/10.1016/j.neucom.2016.04.017
0925-2312/

similar data and should be different with other clusters [6]. Cluster analysis has been applied in many applications. Therefore, many clustering methods have been proposed. In general, clustering methods can be divided in two categories, hierarchical and partition clustering methods. Hierarchical clustering constructs the clusters step-by-step based on the similarity of two data points. There are some similarity measurements such as single linkage, complete linkage, average linkage, etc [7]. Partition clustering methods generate a number of clusters simultaneously. Basically, they use a clusters center and assign each data into the nearest cluster. K-means is the most popular partition clustering method. Many papers have employed K-means algorithm since it can generate a quite good clustering result with a relatively simple algorithm [8–10].

2.2. Meta-heuristic algorithm-based clustering methods

Recently, many researches have applied a metaheuristic algorithm to improve the clustering methods. Genetic algorithm is one of metaheuristic algorithm which has widely combined with clustering methods [11–13]. Furthermore, application of particle swarm optimization (PSO) in clustering problem can be founded in [14–17]. In addition to the GA and PSO, a combination of artificial bee colony algorithm (ABC) with clustering methods also has been introduced in the previous paper [18]. Basically, the metaheuristic algorithm represents the clustering result in their solution representation. They start from any random initial clustering results and then iteratively improve their clustering results. GA updates its solutions using selection, crossover and mutation operators. PSO explores the search space based on a certain direction considering the particle and social best solutions. On the other hand, the updating rules in ABC are inspired from bee behavior in finding the nectar source.

2.3. Clustering ensembles

Clustering ensembles is a clustering method which combines several clustering methods to improve the stability of the result [19]. It comprises of two steps, generation mechanism and the consensus function (see Fig. 1). The generation mechanism generates the data partitions. It then projects all data points onto one dimension. Furthermore, any clustering methods with different parameter setting can be applied to generate the initial clusters. The second step combines the clustering results from the previous step using a consensus function. There are two main consensus methods, object co-occurrence and a median partition. This study applies an object co-occurrence approach using a co-association matrix since it is easier to be understood [20]. The co-association matrix determines the similarity between objects.

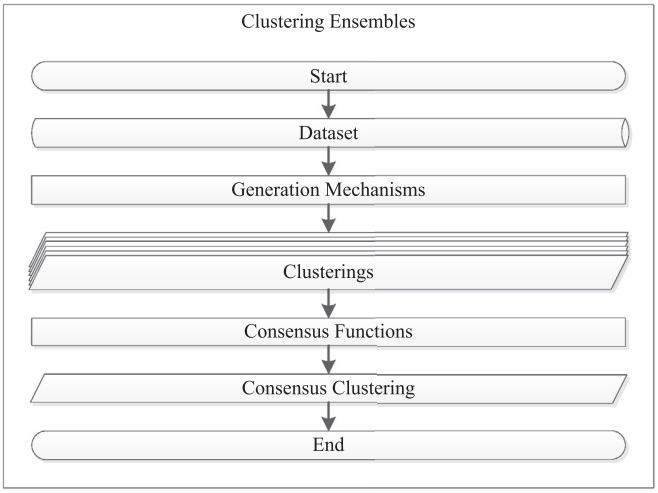

Fig. 1. The flow chart of clustering ensembles.

The most challenging problem in clustering ensembles is finding the best combination of the employed clustering methods. Azimi et al. [4] proposed an improvement of clustering ensembles algorithm with GA. Their proposed method obtains the best combination using a co-association function as the consensus functions. Yang et al. [21] proposed an innovative weighted combination model with multiple parts. It also applies a PSO algorithm to optimize the parameter used in the combining process. Furthermore, Li et al. [15] introduced an clustering ensembles algorithm using K-means and agglomerative hierarchical clustering algorithms. Herein, they applied single linkage, complete linkage and average linkage in the agglomerative hierarchical. The experimental results reveal that average linkage gives better results than single linkage or complete linkage. Clustering ensembles using a combination of partition and hierarchical clustering algorithms is also proposed by Zheng et al. [22].

3. Methodology

This paper proposes three metaheuristics-based clustering ensembles algorithms, which are a real-coded genetic algorithm-based K-means clustering ensembles (GKCE), a particle swarm optimization-based K-means clustering ensembles (PSOKCE) and an artificial bee colony-based K-means clustering ensembles (ABCKCE). This section discusses these proposed algorithms as well as the research methodology. The methodology proposed in this paper comprises of several steps. The first step applies a clustering ensembles algorithm. In order to improve the efficiency, a principle component analysis (PCA) is performed to reduce the problem complexity. Furthermore, the three metaheuristics-based K-means clustering algorithms are applied independently. Fig. 2 illustrates the framework of the proposed method.

3.1. The proposed clustering ensemble methods

The proposed clustering ensemble method comprises of two steps: generation mechanism and consensus function using a co-association matrix (see Fig. 3).

The generation mechanism divides the dataset into several subsets. In the proposed algorithm, this mechanism employs different algorithm to avoid similar subsets. There are four basic clustering algorithms applied, namely, K-means, single linkage agglomerative hierarchical clustering (SLHC), complete linkage agglomerative hierarchical clustering (CLHC) and average linkage agglomerative hierarchical clustering (ALHC). From these four algorithms, three different combinations are proposed. They are: (1) K-means algorithm, (2) SLHC, CLHC and ALHC, and (3) K-means, SLHC, CLHC and ALHC. The subsets are also generated from different attributes, $A_a, 1 \le a \le m$, and instances $I_i, 1 \le i \le n$. The combinations of attributes and instances are generated randomly. Fig. 4 illustrates the subsets generation.

The co-association matrix is a polymerization step. It combines different clustering results obtained by the generation mechanism (see Fig. 5). The elements of the co-association matrix are either 0 or 1. It is determined using Eq. (1).

$$c_{ij} = \frac{\sum_{p=1}^{q} \delta(x_{pi}, x_{pj})}{q} \tag{1}$$

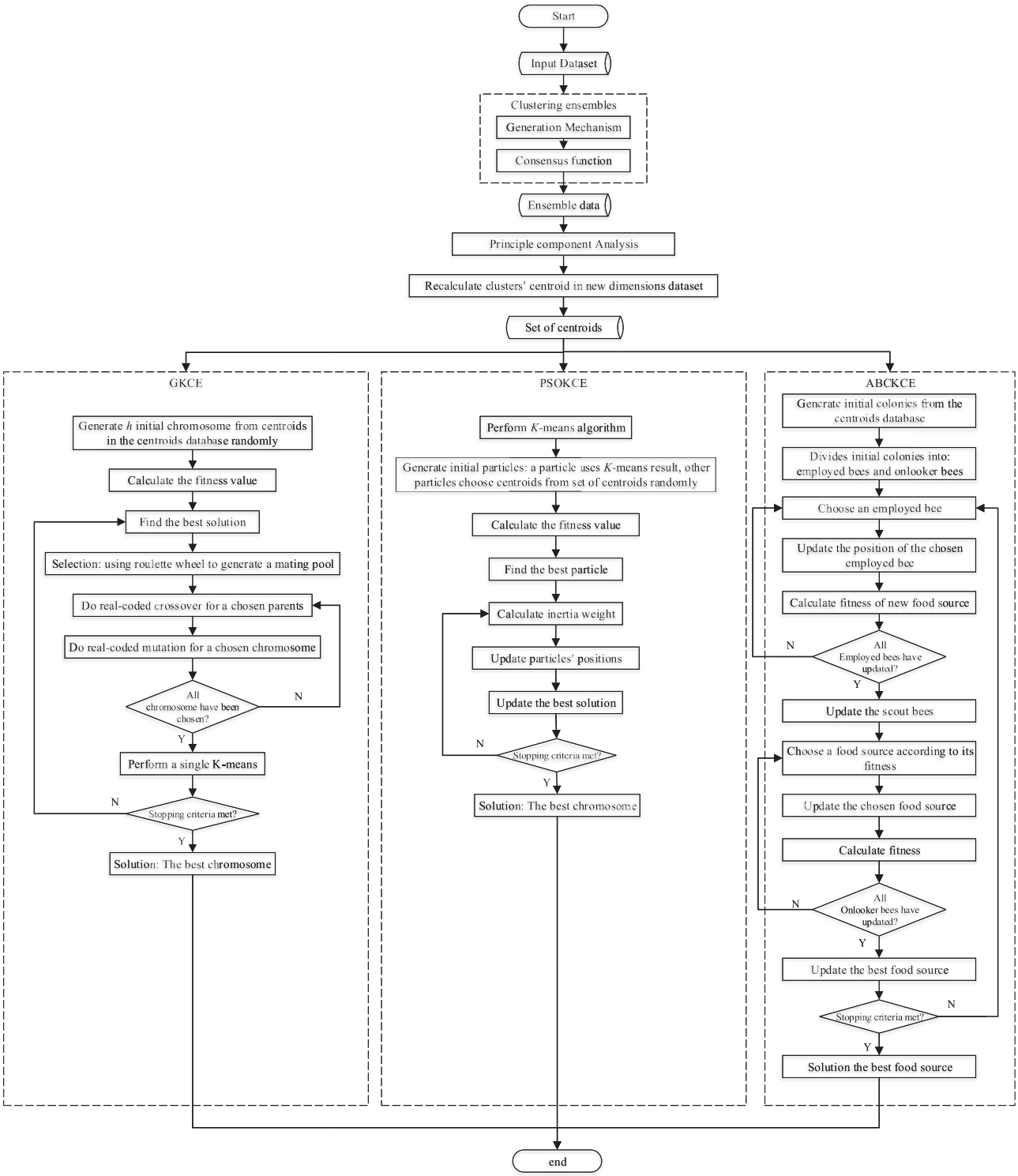

Fig. 2. The flow chart of methodology framework.

$$\delta(x_{pi}, x_{pj}) \equiv \begin{cases} 1, & \text{if} x_{pi} = x_{pj} \\ 0, & \text{if} x_{pi} \neq x_{pj} \end{cases} \quad (2)$$

where c_{ij} is the element of co-association matrix consisted of all co-associations between object $i, 1 \le i \le n$, and object $j, 1 \le j \le m$. The x_{pi} and x_{pj} are object i and j in part p, respectively. The co-association matrix informs whether two instances belongs to the same cluster or not. Although this method is very easy to be implemented, it still requires high computational especially for higher dimensional dataset. Therefore, this study performs a PCA to reduce the dimensions. PCA is a dimensional reduction techniques which represents the original data features in a lower dimensional dataset. Instead of choosing a subset of features, PCA projects the entire data features into lower dimensional space while preserving as much as information as possible. It represents the projection in a covariance matrix and let us choose the projection that minimizes the sum of squared error in reconstructing the original data. Through this concept, although PCA represent

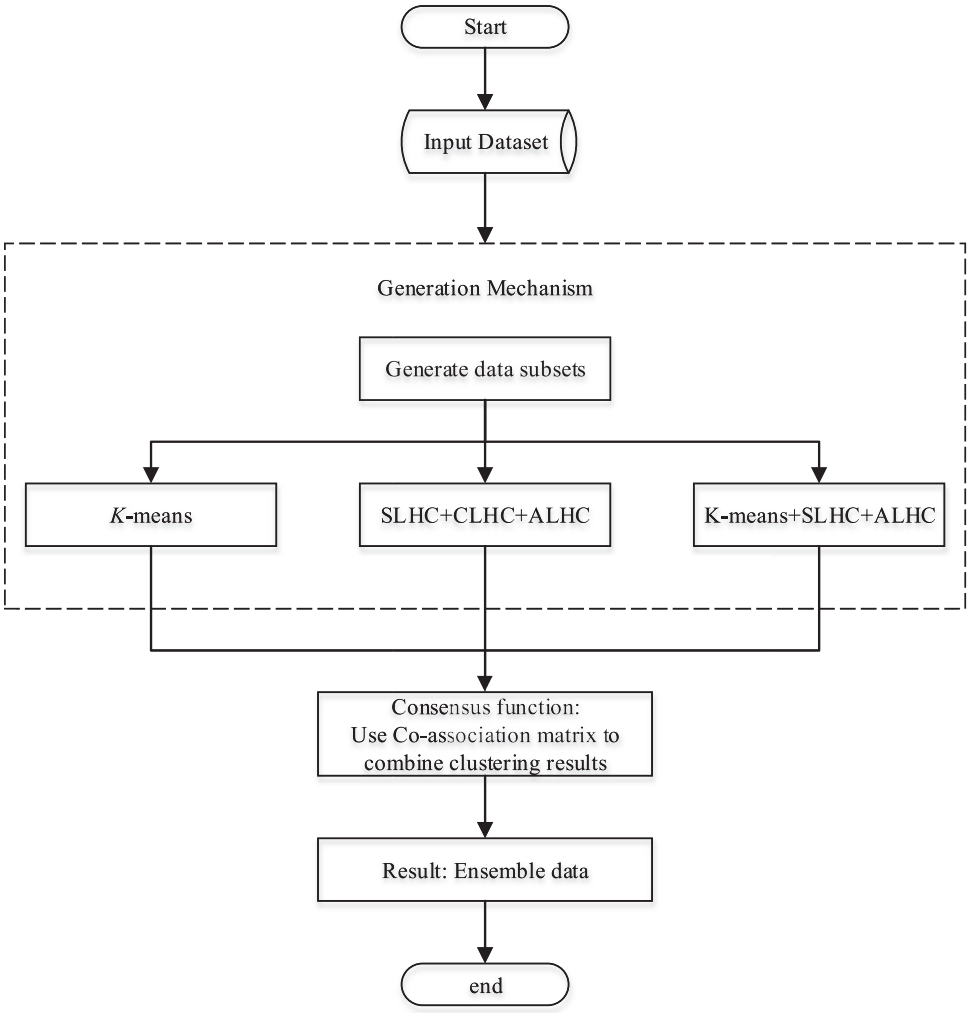

Fig. 3. The flow chart of the proposed clustering ensembles.

the data features in smaller dimensions, it does not delete the data pattern. Thus, the cluster analysis still can be conducted by using any cluster validity measurement such as the mean squared error.

3.2. Meta-heuristic algorithm-based K-means clustering methods

The result form clustering ensembles is further processed by a metaheuristic based *K*-means clustering algorithm. This study applies three different metaheuristic algorithms, namely GA, PSO and ABC algorithms. Basically, the metaheuristic algorithm is combined with *K*-means algorithm to improve its result.

1) Real-Coded Genetic Algorithm-Based K-means Clustering (GKC)
The proposed GKC algorithm performs one step K-means at the end of each iteration. The proposed GKC algorithm is as the following.
Step 1: Generate h initial chromosome from centroids stored the database randomly
Step 2: Calculate the fitness (sum of square error) of each chromosome.
Step 3: Update the best solution using greedy selection.
Step 4: Do roulette wheel to select h chromosomes and record them in the mating pool.
Step 5: Do real-coded crossover towards the selected parents from the mating pool. This proposed algorithm set the crossover probability is within [0.85, 0.95]. The real-coded crossover is given in Eqs. (3) and (4).

$$x_g^{new} = rx_g + (1-r)y_g \tag{3}$$

$$y_g^{new} = (1-r)x_g + ry_g \tag{4}$$

where x_g^{new} and y_g^{new} are the offspring from parents x_g and y_g and r is random number within $[0, 1]$.
Step 6: Do real-coded mutation for the selected chromosomes. The probability of mutation is within range [0.0001, 0.1]. The real-coded mutation is given as follows,

$$x_g^{new} = x_g + \phi_1 |x_g - y_g| \tag{5}$$

$$y_g^{new} = y_g + \phi_2 |x_g - y_g| \tag{6}$$

where x_g^{new} and y_g^{new} are the offspring from chromosomes x_g and y_g and ϕ_i is random number within $[-1, 1]$.
Step 7: Repeat Step 5 and 6 until all chromosomes are selected.
Step 8: Perform a single step *K*-means.
Step 9: Back to Step 2 until the stopping criteria met.

2) Particle Swarm Optimization-Based K-means Clustering (PSOKC)
The first particle in PSOKC applies *K*-means algorithm to obtain a better solution. The algorithm of PSOKC is as follows.
Step 1: Particle initialization. Perform *K*-means algorithm and record the result in a particle. Other particles choose any centroid from the database randomly.

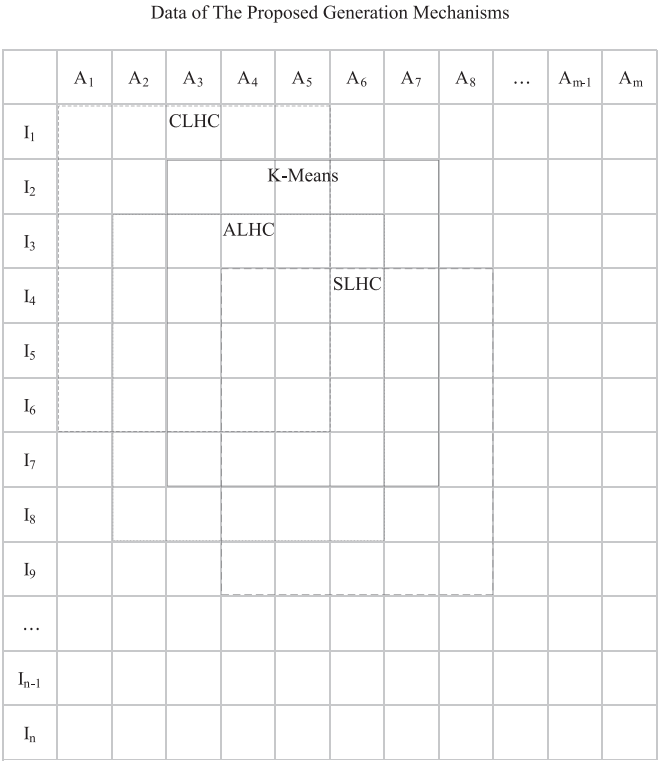

Fig. 4. The diagram of the proposed generation mechanisms.

Table 1
Initial parameter setting.

Method	Parameter	Initial value
All methods	Number of iterations [27]	100
	Population size	50; 80; 100
Real-coded GA [28]	Crossover rate	0.085; 0.9; 0.95
	Mutation rate	0.001; 0.01; 0.1
PSO [29–31]	c_1, c_2	0.5; 1.49; 2
	w^{max}	0.9
	w^{max}	0.4
ABC [32,33]	Limit of search and scouts	5; 10; 20

Table 2
The best parameter setting.

Method	Parameter	Initial value
Real-coded GA	Population size	80
	Crossover rate	0.95
	Mutation rate	0.1
PSO	Population size	80
	c_1, c_2	0.5
	w^{max}	0.9
	w^{max}	0.4
ABC	Population size	100
	Limit of search	5
	Limit of scouts	20

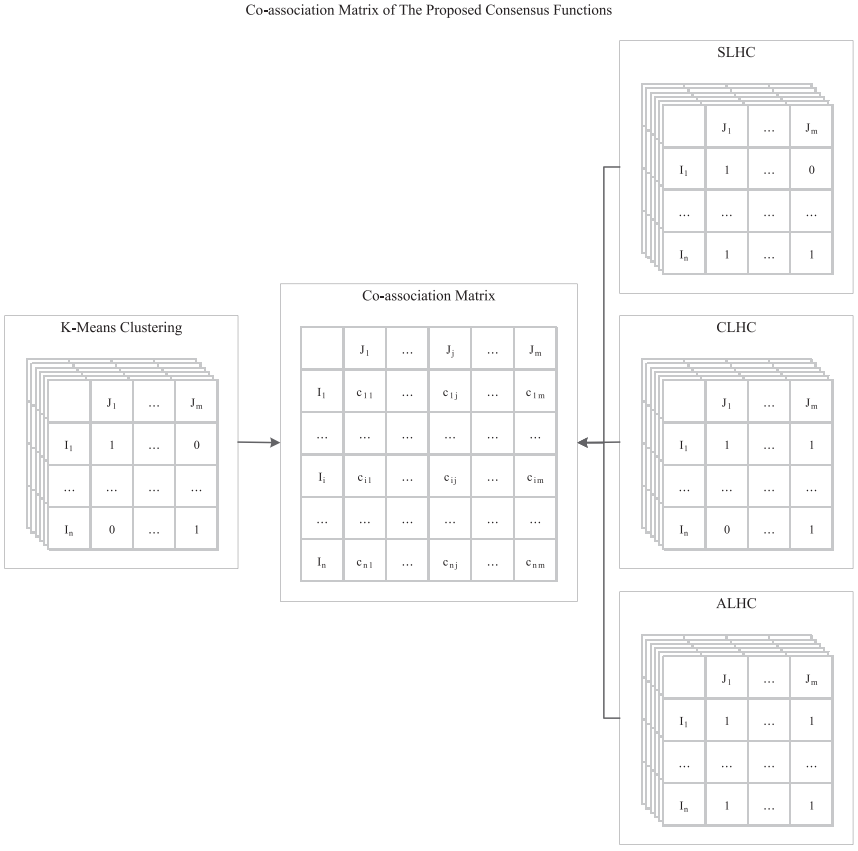

Fig. 5. The diagram of the proposed consensus function.

Table 3
Clustering ensembles parameter setting.

Dataset	Tested setting		The best setting		Best feature
	Attributes partition	Instance partition	Attributes partition	Instance partition	
Iris	2; 3; 4	50; 100; 150	2	150	HC (SLHC, CLHC and ALHC)
Wine	4; 8; 13	60; 120; 178	4	178	*K*-means combined with HC (*K*-means, SLHC, CLHC and ALHC)
Tae	3; 4; 5	50; 100; 151	4	151	*K*-means combined with HC (*K*-means, SLHC, CLHC and ALHC)
Flame	2	80; 160; 240	2	80	*K*-means combined with HC (*K*-means, SLHC, CLHC and ALHC)
Banknote authentication	2; 3; 4	457; 915; 1372	3	1372	*K*-means combined with HC (*K*-means, SLHC, CLHC and ALHC)
D31	2	1033; 2066; 3100	2	3100	*K*-means combined with HC (*K*-means, SLHC, CLHC and ALHC)

Table 4
The mean and standard deviation of accuracy for benchmark datasets.

Datasets algorithms	Iris		Wine		Tae		Flame		Banknote authentication		D31	
K-means	78.04%	0.15	94.87%	0.01	46.47%	0.05	84.83%	0.01	57.58%	0.00	98.26%	0.00
SLHC	66.00%	0.00	38.76%	0.00	42.38%	0.00	64.58%	0.00	55.47%	0.00	77.89%	0.00
CLHC	88.00%	0.00	93.26%	0.00	44.37%	0.00	51.67%	0.00	66.84%	0.00	99.04%	0.00
ALHC	88.67%	0.00	38.76%	0.00	42.38%	0.00	83.33%	0.00	64.50%	0.00	99.41%	0.00
GC	89.20%	0.03	87.90%	0.07	46.00%	0.04	84.78%	0.01	65.70%	0.04	97.28%	0.04
PSOC	88.18%	0.07	94.87%	0.01	46.25%	0.04	84.17%	0.00	58.24%	0.00	97.91%	0.00
ABCC	90.13%	0.02	91.97%	0.04	44.19%	0.03	84.18%	0.00	59.11%	0.01	98.14%	0.01
GKC	88.69%	0.00	96.31%	0.01	42.52%	0.01	84.19%	0.00	59.18%	0.00	98.24%	0.00
PSOKC	88.78%	0.00	94.08%	0.07	47.44%	0.04	84.17%	0.00	58.24%	0.00	98.49%	0.00
ABCKC	88.67%	0.00	96.63%	0.00	42.67%	0.01	84.32%	0.00	57.73%	0.01	99.15%	0.01
GKCE	92.73%	0.03	96.69%	0.01	55.12%	0.02	97.57%	0.01	67.13%	0.05	98.95%	0.05
PSOKCE	92.87%	0.03	95.51%	0.05	49.47%	0.06	94.15%	0.06	68.73%	0.08	95.28%	0.08
ABCKCE	92.91%*	0.03	96.70%*	0.01	55.30%*	0.01	97.75%*	0.01	67.93%	0.06	96.58%	0.06

* The best accuracy for the dataset.

Table 5
The MSE for benchmark datasets.

	Iris	Wine	Tae	Flame	Banknote authentication	D31
K-means	0.1952	0.4987	0.4500*	0.2418	0.5749	1.2281
SLHC	0.2422	0.7043	0.4500*	0.3079	0.5547*	2.2532
CLHC	0.1961	0.5017	0.4502	0.2773	0.6684	1.0071
ALHC	0.1964	0.7040	0.4500	0.2427	0.6450	0.9791*
GC	0.1961	0.5024	0.4606	0.2392	0.6222	1.3327
PSOC	0.1952	0.4988	0.4497	0.2368*	0.5824	1.2237
ABCC	0.1953	0.5009	0.4611	0.2368*	0.5805	1.1789
GKC	0.1952	0.4983	0.4500	0.2368*	0.5868	1.1851
PSOKC	0.1952	0.4984	0.4606	0.2368*	0.5823	1.1113
ABCKC	0.1952	0.4983	0.4497	0.2368*	0.5789	1.0172
GKCE	0.1951*	0.4992*	0.4682	0.2418	0.6160	1.1329
PSOKCE	0.1951*	0.5004	0.4682	0.2398	0.6283	1.6564
ABCKCE	0.1951*	0.4992*	0.4682	0.2414	0.6042	1.6844

* The best MSE for the dataset.

Table 6
Comparison with AIS-based clustering algorithm (accuracy).

	AIS	AISK	aiNet	aiNetK	GKCE	PSOKCE	ABCKCE
Iris	70.67%	86.00%	86.00%	89.33%	92.73%	92.87%	92.91%
Wine	61.10%	94.94%	60.11%	95.51%	96.69%	95.51%	96.70%

Step 2: Calculate the fitness (sum of square error) of each particle.
Step 3: Update the best solution using a greedy selection.
Step 4. Calculate the inertia weight for iteration t, w_t, using

Table 7
Average confusion matrix obtained by GKCE, PSOKCE, and ABCKCE for Iris dataset.

Method	Actual class	Clustered class		
		0	1	2
GKCE	0	50	0	0
	1	0	47.25	2.75
	2	0	13	37
PSOKCE	0	50	0	0
	1	0	47.25	2.75
	2	0	13	37
ABCKCE	0	50	0	0
	1	0	47.25	2.75
	2	0	13	37

Table 8
Average confusion matrix obtained by GKCE, PSOKCE, and ABCKCE for Wine dataset.

Method	Actual class	Clustered class		
		0	1	2
GKCE	0	56	3	0
	1	0	68	3
	2	0	1	47
PSOKCE	0	56.2	2.8	0
	1	0	67.6	3.4
	2	0	1	47
ABCKCE	0	56	3	0
	1	0	68	3
	2	0	9.6	38.4

Table 9
Average confusion matrix obtained by GKCE, PSOKCE, and ABCKCE for Tae dataset.

Method	Actual class	Clustered class		
		0	1	2
GKCE	0	31	6.4	10.4
	1	19.6	22.4	15.4
	2	11.2	12.4	23.2
PSOKCE	0	34.8	6.4	6
	1	28.6	18.6	11.6
	2	18.8	8.4	18.8
ABCKCE	0	28.5	7.25	11.25
	1	11.6	21	14.4
	2	7.75	15.25	26.5

Table 10
Average confusion matrix obtained by GKCE, PSOKCE, and ABCKCE for Flame dataset.

Method	Actual class	Clustered class	
		0	1
GKCE	0	83	4
	1	31.4	121.6
PSOKCE	0	83.4	3.6
	1	31.4	121.6
ABCKCE	0	83.4	3.6
	1	31.4	121.6

Table 11
Average confusion matrix obtained by GKCE, PSOKCE, and ABCKCE for Bank Authentication dataset.

Method	Actual class	Clustered class	
		0	1
GKCE	0	424.2	337.8
	1	135.6	474.4
PSOKCE	0	470.4	291.6
	1	155.4	454.6
ABCKCE	0	408.8	353.2
	1	141.6	468.4

Eq. (7).

$$w_t = \frac{t^{max} - t}{t^{max}}\left(w^{max} - w^{min}\right) + w^{min} \tag{7}$$

where t^{max} is the maximum iteration, w^{max} and w^{min} are the maximum and minimum inertia weight, respectively.

Step 5: Update the position of each particle using Eqs. (8) and (9).

$$v_{pd}^{new} = w_t v_{pd} + c_1 r_1 \left(x_{pd}^{local} - x_{pd}\right) + c_2 r_2 \left(x_{pd}^{global} - x_{pd}\right) \tag{8}$$

$$x_{pd}^{new} = x_{pd} + v_{pd}^{new} \tag{9}$$

where v_{pd}^{new} is the velocity for particle p in dimension d. The c_1 and c_2 are learning rates controlling the influence of the local best, x_{pd}^{local}, and global best, x_{pd}^{global}, particles, respectively. The r_1 and r_2 are random number within [0, 1].

Step 6: Back to Step 2 until the stopping criteria met.

3) Artificial Bee Colony-Based-K-means Clustering (ABCKC).

The ABC algorithm employs K-means algorithm to accelerate its convergence. It improves the scout bees using K-means algorithm. The proposed ABCKC algorithm is as the following.

Step 1: Generate s initial colonies from the centroids stored in the database randomly. Divide the s initial colonies into employed bees and onlooker bees. If number of scout bees is higher than the limits of the scout bees, set the bees with better solution as the employed bees and the remaining as the scout bees.

Step 2: Do a greedy selection to choose an employed bee. Update solution of the chosen employed bee using Eq. (10).

$$b_{sd}^{new} = b_{sd} + \phi(b_{sd} - b_{rd}), \tag{10}$$

where b_{rd} is any random neighbor of b_{sd} and ϕ is a random number within $[-1,1]$. Calculate the fitness value using the sum of square error. Repeat for all employed bees.

Step 3: Update the scout bees. In this paper, there are two types of ABC-based clustering algorithm proposed. They are ABCC and ABCKC. For the ABCC, the scout bees are updated using Eq. (10) while ABCKC using K-means algorithm and compute their fitness.

Step 4: For onlooker bees, select a solutions from the new updated solutions according to their fitness. And update the chosen food source using Eq. (10). Calculate its fitness value. Repeat until all onlooker bees update their food source.

Step 5: Update the best solution.

Step 6: Back to Step 2 until the stopping criteria met.

4. Experimental results

This section presents the computational results of the proposed clustering ensembles. All of the proposed clustering algorithms are coded using C# programming language and implemented on 2.5 GHz CPU and 4 GB of RAM computer. The experiments use six datasets to verify the performance of the proposed algorithms [23]. They are iris, wine, tae, flame, banknote authentication and D31 datasets. The performance evaluation uses the accuracy and the MSE value as the parameters.

4.1. Parameter setting

Metaheuristic algorithms involve some parameter setting. In most of the cases, these parameters have significant influence to the results. Therefore, determining the best parameter setting is also an important issues. This paper uses a Taguchi method to determine the best parameter setting for each metaheuristics-based clustering algorithms proposed. The Taguchi method requires a set of initial parameters and gives the best combination of the parameters [24]. In this paper, the initial parameters are taken from some previous papers. Table 1 lists all the combination of parameter setting tested in this experiments while the best combination of parameter setting is given in Table 2.

The statistic tests for Taguchi method are executed in Minitab. According to the rule of thumb, the minimum sample size in statistic test is 25 or 30 therefore, for each combination of parameters, 30 independent run are performed. Table 3 summaries the results.

In addition to the metaheuristic-based algorithm, clustering ensembles also requires a predetermine parameter setting. The best combination of clustering ensembles parameter setting is also determined using Taguchi method. Table 3 lists the parameter setting for each dataset.

4.2. Computational results

The performance evaluation is performed using the best parameter setting. Herein, the proposed algorithms are compared with other metaheuristic-based without clustering ensembles algorithms. They are GA clustering (GC), PSO clustering (PSOC), ABC clustering (ABCC), GA K-means clustering (GKC), PSO K-means clustering (PSOKC) and ABC K-means clustering (ABCKC). For each algorithm, 30 independent run are performed. Table 4 summaries

Table 12
Average confusion matrix obtained by GKCE for D31 dataset.

Actual class	Clustered class																														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	98.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	96.7	0.0	4.3	1.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	95.3	0.0	0.0	0.0	2.0	2.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	67.0	0.0	0.0	0.0	0.0	0.0	2.7	1.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	94.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.3	3.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.0	0.0	0.0	0.0	0.0	64.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	97.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.3	0.7	0.0	0.0	0.0	99.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	95.3	0.0	0.0	0.3	0.3	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1.0	33.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	67.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.7	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.7	1.0	1.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	33.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	32.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.3	0.0	0.0	0.0	0.7	0.0	0.0	94.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	95.0	0.0	0.0	0.0	0.0	0.0	1.7	1.7	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2.3	0.0	0.0	0.0	0.0	0.0	30.3	0.0	99.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0
22	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.7	2.3	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	66.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	66.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24	0.7	33.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	94.3	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.3	98.7	0.7	0.0	0.0	0.3	0.0
26	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.0	0.0	0.0	0.0	0.0	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.3	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.3	0.0	0.0
29	0.0	0.0	0.0	0.0	0.0	0.3	0.3	33.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	66.7	32.7
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	65.0

Table 13
Average confusion matrix obtained by PSOKCE for D31 dataset.

Actual class	Clustered class																													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	98.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	79.3	6.0	28.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	100.0	0.0	1.3	0.0	1.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.3	98.3	0.0	0.0	2.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9.3	0.0	0.0
4	0.0	0.0	0.0	0.0	97.3	0.0	0.0	0.0	2.3	1.7	0.0	1.0	1.0	0.0	0.0	0.0	0.0	10.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.0	0.0	0.0	0.0	0.0	99.0	0.0	0.0	0.3	0.0	0.3	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.7	0.0	0.3	0.0	0.0	95.3	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	1.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.7	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.7	0.0	0.0	0.3	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.7	0.0	0.7	0.0	0.0	0.0	0.0	97.3	0.0	0.3	0.3	0.0	0.0	0.7	0.3	0.3	0.0	0.3	0.3	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1.0	32.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.7	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.0	0.3	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	97.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	1.3	2.0	1.3	0.0	0.0	0.0	0.0	4.3	0.0	0.0
12	0.0	0.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.3	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.3	0.0	0.0
15	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.3	0.3	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.3
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	21.0	0.0	0.0	1.7	0.0	0.0	0.0	100.0	0.0	0.0	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.3	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	75.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.3	0.0	1.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	96.3	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	33.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.3	0.3	0.3	74.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	1.0	0.0	0.7	1.3	0.3	0.0	0.7	0.0	0.0	1.3	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	6.7	26.7	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	1.0	0.0	0.0	60.3	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	33.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.3	0.0	0.0	76.0	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	1.3	0.0	0.0	0.0	75.3	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	3.0	0.0	0.0	8.0	1.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	80.7	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	83.0	0.0	0.0
29	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	80.0	0.7
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	95.3

Table 14
Average confusion matrix obtained by ABCKCE for D31 dataset.

Actual class	Clustered class																														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	98.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	99.0	5.0	31.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	98.7	0.0	0.0	4.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.7	2.3	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	95.0	0.0	0.0	0.0	0.0	0.0	1.3	3.7	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.0	0.0	0.0	0.0	0.0	99.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	98.3	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.3	0.3	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.3	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	95.7	0.0	1.3	0.0	0.0	0.0	0.0	0.7	0.3	0.3	0.7	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1.0	32.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	67.7	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.7	2.3	0.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.3	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	2.7	0.0	0.0	0.0	99.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	4.3	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.0	0.7	95.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.7	0.3	0.0	0.0	68.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.7	0.0	0.0	0.0	0.7	1.3	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	99.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	99.3	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	96.3	0.0	0.0	0.0	0.0	0.0
26	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.7	0.0	0.0	0.0	0.0	98.3	0.0	0.0	0.0	0.0
27	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.3	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	97.3	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.3	0.0	0.0
29	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.3	0.0
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0

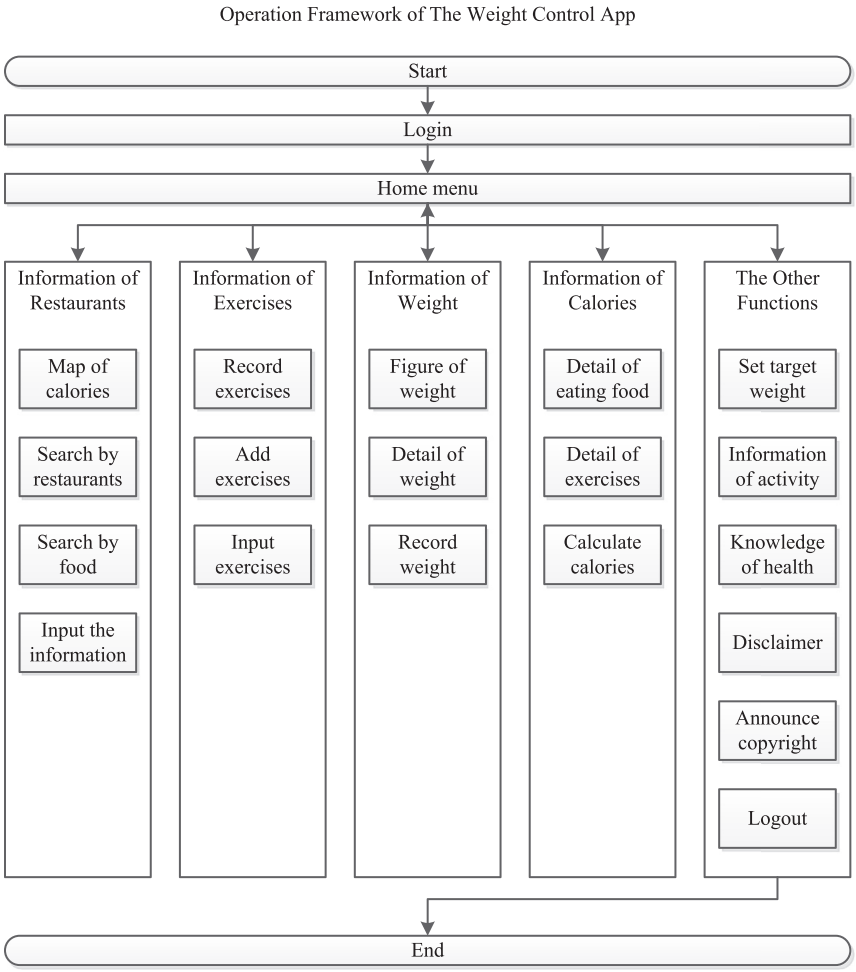

Fig. 6. The diagram of operation framework of the weight control App.

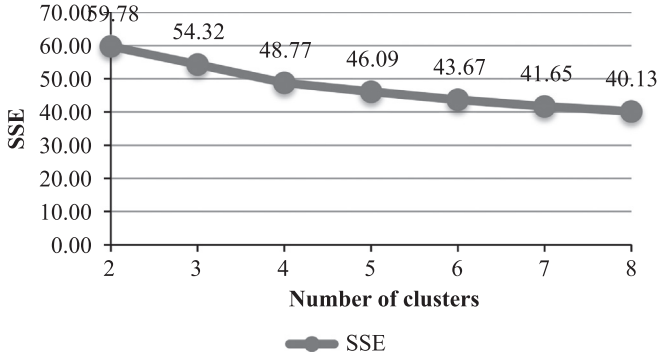

Fig. 7. The chart of number of clusters determination.

Table 15
The best parameter setting for the study case.

Method	Parameter	Initial value
Real-coded GA	Population size	80
	Crossover rate	0.9
	Mutation rate	0.1
PSO	Population size	80
	c_1	1.495
	c_2	0.5
	w^{max}	0.9
	w^{max}	0.4
ABC	Population size	100
	Limit of search	20
	Limit of scouts	5

the computational results. This table shows the mean and standard deviation values of accuracy.

For the iris dataset, the clustering ensemble methods are better than other single clustering algorithms. The best accuracy is 92.91% obtained by ABCKCE. For the wine dataset, PSOC, GKC, PSOKC, ABCKC, GKCE, PSOKCE and ABCKCE produce similar accuracy. Among them, the best accuracy is obtained by ABCKCE, which is 96.70%. For the tae dataset, the GKCE result is as good as ABCKCE result. In flame dataset, ABCKCE also has the best performance than other algorithms. Its best accuracy is 97.75%. For banknote authentication, the clustering algorithm with clustering ensembles also show better results than clustering without ensembles. Unfortunately, for D31 dataset, the proposed algorithms cannot perform better than metaheuristic-based clustering algorithms. In addition, MSE values are also illustrated in Table 5.

Besides, a comparison with artificial immune system (AIS) based clustering methods is also presented. The result of the AIS-based clustering results are taken from a previous paper [25]. Table 6 shows the comparison. It shows that the proposed metaheuristic-based clustering ensembles algorithms obtain better accuracy than AIS-based clustering algorithms. These results indicate that the ABCKCE can perform better than other tested clustering algorithms. The proposed clustering ensembles

algorithms employ our basic clustering algorithms: *K*-means, SLHC, CLHC and ALHC. In the generation mechanism, these algorithms are combined. The proposed clustering ensembles will give a better result if the results from individual clustering algorithm are better. A more detailed result obtained by the proposed GKCE, PSOKCE, and ABCKCE are given in Tables 7–14. These tables show the average of confusion matrix.

A further verification is performed using statistic test. A statistic *t*-test is applied to investigate if the proposed algorithms have significantly better results than the others. The statistic test results reveals that the proposed metaheuristic-based using clustering ensembles always have significantly better result than clustering algorithm without clustering ensembles. However, the differences between GKCE, PSOKCE and ABCKCE are not significant.

5. Model evaluation results

In recent years, smart phones have had a profound effect on people' lives. Therefore, application developers must understand and respond to changes of user's behavior. Customer segmentation is one of a strategy to understand the user's behaviors. The application developer collects the log data using Smartphone measurement software, which runs in the background of the mobile phone, and regularly transmits log files concerning user activities to the server [26]. This study uses the log files from a mobile app to perform customer segmentation, using the proposed clustering methods. The clustering result shows the characteristics of customers who use the mobile app, so the application developer can improve the mobile app depending on the clustering result.

5.1. Problem overview

This paper uses a primary data collected from a weight control mobile app. The mobile app has five main functions: searching for restaurants, recording the information from exercise, recording weight information, managing the information concerning calories and other functions. The operational framework for the weight control app is shown in Fig. 6.

The weight control data records the log data from the browsing conditions. It consists of 204 customers with 5 attributes. The number of clusters is determined by the sum of the square error (SSE) obtained by the *K*-means algorithm, as shown in Fig. 7. It shows that when the number of clusters is 4, the SEE reduce insignificantly. Therefore, the number of clusters is set as four.

5.2. Computational preparation

In this study case, the real class of each instances is unknown. Therefore, the performance of each algorithm in solving the real problem is evaluated using the mean square error (MSE). Before applying the proposed algorithms, the raw data collected from the mobile app is prepared by deleting instances with missing value and performing a normalization.

Furthermore, the best parameter setting for each algorithm is also determined using a Taguchi method. With the same initial parameters as tested in Section 4, the best parameter setting for the real problem is shown in Table 15. In addition, the best feature in generation mechanism is *K*-means, the best number of attribute and instance partition are 4 and 204, respectively.

5.3. Computational results

Table 16 shows the results the weight control dataset. It shows that the clustering ensemble methods are better than other single clustering algorithms, in terms of the MSE value. The best result is 0.23621, provided by ABCKCE. Therefore, the further analysis for the App customer segmentation uses the ABCKCE solution.

Table 16
The mean and standard deviation of MSE for weight control dataset.

Dataset Algorithms	Weight control	
K-means	0.24104	0.008
SLHC	0.31173	0.000
CLHC	0.29009	0.000
ALHC	0.30851	0.000
GC	0.24335	0.006
PSOC	0.23660	0.000
ABCC	0.23802	0.001
GKC	0.23642	0.000
PSOKC	0.23643	0.000
ABCKC	0.23640	0.000
GKCE	0.23623	0.000
PSOKCE	0.23767	0.005
ABCKCE	0.23621*	0.000

* The best MSE for weight control dataset.

Table 18
The characteristic of each cluster.

Cluster	Characteristics
Cluster 1	The users only use a little time or already abandon the weight control App.
Cluster 2	The users focus on using the function of exercise. They are the main users for weight control App.
Cluster 3	The users usually use the function of finding restaurants which provide low calories dish. These users is the main users for weight control App.
Cluster 4	The users who are the new members try to use all kinds of the functions.

Table 17
The details of each cluster.

Cluster	Features						
	The mean using time	The mean using functions	Info of restaurants	Info of exercises	Info of weight	Info of calories	The other function
1	423.2	12.09	0.2640	0.2228	0.1559	0.0415	0.0624
2	549.4	16.67	0.2940	0.5955	0.1138	0.0148	0.0898
3	587.1	17.38	0.6380	0.2095	0.1223	0.0581	0.0968
4	327.3	17.72	0.3985	0.2264	0.1180	0.0357	0.4251

Table 19
The strategy for each cluster.

Cluster	Strategy
Cluster 1	The application developer can provide new function to attract the customers who continue use the weight control App.
Cluster 2	The application developer should put large developmental source in the users of this cluster, and improve the function of weight continually by user's feedback.
Cluster 3	The users of this cluster are the main customers, so the functions of finding restaurants which provide low calories dish need to continue improving. The application developer should collect the feedback to maintain the competitiveness.
Cluster 4	The application developer can add a simple introduction about the App by the website or App.

Moreover, a statistic test is also conducted to evaluate the differences among the tested algorithms. The statistic *t*-test test with 95% confidence interval reveals that GKCE and ABCKCE are better than the other clustering algorithms.

This result reveals that the ABCKCE algorithm has more stable the optimal solution than GKCE. The mean of ABCKCE results is given in Table 17. The sizes of each cluster are 76, 37, 47 and 44. Table 18 summaries the user's characteristic of each clusters. Users in cluster 1 is the users who seldom use the weight control app. Cluster 2 and the cluster 3 contains of the frequent users. Totally, they have 84 users. Cluster 4 is new users. As the new users, they use many different functions of the app to explore the app functions. Through this result, a marketing strategy can be derived. This paper suggests some further strategy for the mobile app developer. Since different user's characteristic need different strategy, the suggestions given in Table 19 are made for each clusters.

6. Conclusions

This study proposes three novel clustering ensembles algorithm which integrate meta-heuristic algorithms and clustering ensemble. The proposed clustering ensemble algorithm comprises of two main steps. The first step is generation mechanism. It aims to partition the data using some clustering algorithms. The data subsets from the generation mechanism are then integrates using a co-association matrix. Furthermore, the proposed algorithms applies three metaheuristic-based clustering algorithms to construct the final clusters. This paper uses GA-based, PSO-based and ABC-based clustering algorithms.

The experimental results show that GKCE and ABCKCE perform better than or equal to the other clustering algorithms. The GKCE, PSOKCE and ABCKCE are further applied to analyze the log files for a weight control App. The computational results also indicate that the proposed ensemble methods outperform the other clustering algorithms. Among them, the best algorithm is ABCKCE. The mobile app developer can use this result to improve the app's features.

In the future research, some other single clustering algorithms like fuzzy *c*-means should be included. In addition, the consensus function proposed in this study is a co-association matrix. This method is very easy to implement and understand. However, it requires high computational time, especially for a large dataset. Therefore, other techniques should be applied in the consensus function.

References

[1] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, Int. J. Pattern Recognit. Artif. Intell. 25 (2011) 337–372.
[2] J. Kleinberg, An impossibility theorem for clustering, NIPS (2002) 463–470.
[3] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (1999) 264–323.
[4] J. Azimi, M. Mohammadi, A. Movaghar, M. Analoui, Clustering ensembles using genetic algorithm, Comput. Arch. Mach. Percept. Sens. (2006) 119–123.
[5] A. Topchy, A.K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, Pattern Anal. Mach. Intell. IEEE Trans. 27 (2005) 1866–1881.
[6] R. Patel, M. Raghuwanshi, A.N. Jaiswal. Modifying genetic algorithm with species and sexual selection by using K-means algorithm, in: Proceedings of Advanced Computing Conference, 2009, pp. 114–119.
[7] D. Tamasauskas, V. Sakalauskas, D. Kriksciuniene, Evaluation framework of hierarchical clustering methods for binary data, Hybrid. Intell. Syst. (HIS) (2012) 421–426.
[8] J. MacQueen Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (California, USA . 1967, pp. 14.
[9] Q. Chen, J. Mo, Optimizing the ant clustering model based on K-Means algorithm, Comput. Sci. Inf. Eng. (2009) 699–702.
[10] G. Sun, Y.-h Feng, X.-j Guo, C.-p Zhang, Research on K-means clustering algorithm, J. Chang. Norm. Univ. 2 (2011) 001.
[11] K. Krishna, M.N. Murty, Genetic K-means algorithm, IEEE Trans. Syst., Man, Cybern., Part B: Cybern. 29 (1999) 433–439.
[12] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognit. 33 (2000) 1455–1465.
[13] S.-S. Cheng, Y.-H. Chao, H.-M. Wang, H.-C. Fu, A prototypes-embedded genetic K-means algorithm, Pattern Recognit. (2006) 724–727.
[14] A. Rani, L. Parthipan, Clustering analysis by improved particle swarm optimization and K-means algorithm, Sustain. Energy Intell. Syst. (SEISCON) (2012) 1–6.
[15] Z. Li, Y. Li, L. Xu, Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization, in: Proceedings of the International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), 2011 , (IEEE'2011), pp. 157–161.
[16] J. Dong, M. Qi, A new algorithm for clustering based on particle swarm optimization and K-means, Artif. Intell. Comput. Intell. (2009) 264–268.
[17] D. Van der Merwe, A.P. Engelbrecht, Data clustering using particle swarm optimization, Evolut. Comput. (2003) 215–220.
[18] C. Zhang, D. Ouyang, J. Ning, An artificial bee colony approach for clustering, Exp. Syst. Appl. 37 (2010) 4761–4767.
[19] A.L. Fred, A.K. Jain, Data clustering using evidence accumulation, in: Proceedings of the 16th International Conference on Pattern Recognition, 2002. (IEEE'2002), pp. 276–280.
[20] A.P. Topchy, A.K. Jain, W.F. Punch, A Mixture Model for Clustering Ensembles, SDM, (SIAM'2004).
[21] L.-y Yang, J.-y Zhang, W.-j Wang, Cluster ensemble based on particle swarm optimization, Intell. Syst. (2009) 519–523.
[22] L. Zheng, T. Li, C. Ding, Hierarchical ensemble clustering, Data Min. (ICDM) (2010) 1199–1204.
[23] L. Fu, E. Medico, FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data, BMC Bioinforma 8 (2007) 3.
[24] R.N. Kackar, Off-line quality control, parameter design, and the Taguchi method, J. Qual. Technol. 17 (1985) 176–188.
[25] R.J. Kuo, S.S. Chen, W.C. Cheng, C.Y. Tsai, Integration of artificial immune network and K-means for cluster analysis, Knowl. Inf. Syst. 40 (2013) 541–557.
[26] F. Hamka, H. Bouwman, M. De Reuver, M. Kroesen, Mobile customer segmentation based on smartphone measurement, Telemat. Inform. 31 (2014) 220–227.
[27] R.J. Kuo, Y.J. Syu, Z.-Y. Chen, F.C. Tien, Integration of particle swarm optimization and genetic algorithm for dynamic clustering, Inf. Sci. 195 (2012) 124–140.
[28] E. Michielssen, S. Ranjithan, R. Mittra, Optimal multilayer filter design using real coded genetic algorithms, IEEE Proc. J. (Optoelectron.) 139 (1992) 413–420.
[29] Y. Shi, R.C. Eberhart, Parameter Selection in Particle Swarm Optimization, Evolutionary Programming VII, Springer (1998), p. 591–600.
[30] B. Jiang, N. Wang, L. Wang, Parameter identification for solid oxide fuel cells using cooperative barebone particle swarm optimization with hybrid learning, Int. J. Hydrog. Energy 39 (2014) 532–542.
[31] R. Kuo, F.E. Zulvia, K. Suryadi, 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 (2012) 2574–2588.

[32] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142.
[33] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (2008) 687–697.

R. J. Kuo received the MS degree in Industrial and Manufacturing Systems Engineering from Iowa State University, Ames, IA, in 1990 and the Ph.D. degree in Industrial and Management Systems Engineering from the Pennsylvania State University, University Park, PA, in 1994. Currently, he is the Distinguished Professor in the Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan. His research interests include data mining and metaheuristics, and their applications in decision support systems, forecasting, supply chain management, and customer relationship management.

C. H. Mei received the MS degree in Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan in 2014. His research interests include data mining and metaheuristics, and their applications in customer relationship management.

F. E. Zulvia received the MS degree and Ph.D. in Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan. Currently, she is under postdoctoral research in Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan. Her research interests include metaheuristics and their applications in data mining and logistic distribution.

C. Y. Tsai is a professor in the Department of Industrial Engineering and Management at Yuan Ze University, Taiwan. He received his M.S. and Ph.D. degrees in Department of Industrial and Manufacturing Systems Engineering from the University of Missouri-Columbia, USA. His research activities include data mining, customer relationship management (CRM), RFID technology and applications, and product data management.