Research Article

A New Binary Adaptive Elitist Differential Evolution Based Automatic k -Medoids Clustering for Probability Density Functions

D. Pham-Toan ¹, **T. Vo-Van**,¹ **A. T. Pham-Chau**,^{2,3}
T. Nguyen-Trang ^{2,3} and **D. Ho-Kieu** ^{2,3}

¹Natural Science College, Can Tho University, Can Tho, Vietnam

²Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam

³Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam

Correspondence should be addressed to D. Ho-Kieu; hokieudiem@tdtu.edu.vn

Received 22 December 2018; Accepted 24 March 2019; Published 22 April 2019

Academic Editor: Elio Masciari

Copyright © 2019 D. Pham-Toan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an evolutionary computing based automatic partitioned clustering of probability density function, the so-called binary adaptive elitist differential evolution for clustering of probability density functions (baeDE-CDFs). Herein, the k -medoids based representative probability density functions (PDFs) are preferred to the k -means one for their capability of avoiding outlier effectively. Moreover, addressing clustering problem in favor of an evolutionary optimization one permits determining number of clusters “on the run”. Notably, the application of adaptive elitist differential evolution (aeDE) algorithm with binary chromosome representation not only decreases the computational burden remarkably, but also increases the quality of solution significantly. Multiple numerical examples are designed and examined to verify the proposed algorithm’s performance, and the numerical results are evaluated using numerous criteria to give a comprehensive conclusion. After some comparisons with other algorithms in the literature, it is worth noticing that the proposed algorithm reveals an outstanding performance in both quality of solution and computational time in a statistically significant way.

1. Introduction

Clustering aims to divide input data into multiple groups such that elements in each group are similar and different from elements in other groups as much as possible. There are two primary objects of clustering, discrete element and probability density function (PDF). Over the past few years, discrete element is preferred in clustering with a lot of works such as [1, 2]. However, with the explosion of digital era, a massive amount of data is created each day [3, 4], and how to present such data well is a challenge task for discrete elements. The main reason for this is that the discrete element just presents whole data by a representative point so that it is unlikely to fully demonstrate characteristic of whole data, especially fluctuation data [5]. Meanwhile, the remaining object-PDF shows its advantage in such digital

era like capturing the distribution of whole data, giving a visible look of characteristic of the estimated objects [5]. However, regardless of the advantages of PDF, works related to clustering for this interesting object are still very limited. Therefore, this paper aims to contribute new approach for clustering of probability density functions (CDFs).

Concerning CDFs, two main approaches can be found, nonhierarchical and hierarchical [6]. With respect to nonhierarchical approach, k -means and k -medoids based algorithms can be seen as the typical ones [7]. However, k -means based approach shows its disadvantage when addressing outliers or noises [7]. In contrast, k -medoids based approach can handle outliers effectively as shown in the work of [8] since it directly employs objects in input data as the centers. The k -medoids is then applied to various fields. For example, in [9], Zhang B et al. presented an improved ranked k -medoids

algorithm by a specific cell-like P system and extended the application to membrane computing. In addition, in [10], Liu F et al. proposed an k -medoids clustering algorithm for testing software. In [11], Cahaya L et al. introduced an unsupervised learning algorithm, the so-called intelligent k -medoids, to predict the length of a study time of the students in universities. Moreover, in [12], Prihandoko et al. applied k -means and k -medoids algorithms to analyze the natural disaster data in Indonesia. By that way, they can reveal the hidden information so that the government could have appropriate solutions in advance. Nevertheless, these applications are still widely applied on discrete elements, not the PDFs. To the best of our knowledge, there are no works using k -medoids in clustering for probability density functions so far. As a result, it is necessary to develop a nonhierarchical crisp clustering algorithm for probability density functions (PDFs) such that the representative object is generated based on k -medoids scheme.

Over the past few years, tremendous research effort has been made to evolve the partitioned clustering algorithms in complex data sets through evolutionary computing techniques. However, not many studies concern how to determine the number of clusters at the same time, especially in the field of clustering for PDFs. For instance, just two recent evolutionary-based algorithms are proposed by author group of Vo-Van et al., but both accept a k input as the number of clusters rather than determining it “on the run” [13, 14]. Moreover, these works usually employ very classical and simple evolution methods to solve the problem, genetic algorithm (GA). Although GA is an effective method in solving heuristic optimized problems, there are still large amounts of modern and efficient evolution methods that need to be mentioned. Therefore, in this paper, one of state-of-the-art evolutionary algorithms will be employed and modified to address clustering of probability density functions automatically.

Regarding evolutionary-based algorithms, the adaptive elitist differential evolution (aeDE), firstly developed by Ho-Huu et al. [15], has proved to be one of the most effective and robust algorithms. Firstly, aeDE is a global searching algorithm inherited from its original version, Differential Evolution (DE) [15], which is an advantage compared with that of Genetic algorithm (GA), easily trapping into local solution [16]. Secondly, with two modifications suggested by Ho-Huu et al., the quality of solution as well as the computational cost has been enhanced significantly. On the one hand, the adaptive mutation scheme assists the algorithms searching more efficiently and decreasing computational burden; on the other hand, the elitist selection scheme supports algorithm in recording the best individuals for next generation. For these reasons, aeDE is being increasingly applied in numerous fields. For example, in [17], Ho-Huu V et al. presented an improved DE to solve the shape and size optimization problem of truss structure with frequency constraints. Also, in [18], Ho-Huu V et al. applied aeDE to optimize the truss structures with frequency constraints based on reliability under uncertainties of loadings and material properties. In [19], Vo-Duy T et al. suggested a global numerical approach for lightweight design

optimization of laminated composite plates subjected to frequency constraints by applying aeDE. Moreover, in [20], this author also proposed an aeDE based-effective approach to maximize the fundamental frequency of laminated functionally graded carbon nanotube-reinforced composite quadrilateral plates. Besides, in [21], Demertzis K and Iiadis L utilized aeDE to optimize extreme learning machine model, and then it is applied to create an advanced computer vision system for the phenotype based automatic recognition of invasive or other unknown species. Nevertheless, one noticeable point is that the application of such a good and robust algorithm as aeDE is being restricted in engineering problems as mentioned above, as none of the works related to aeDE have been found in clustering problem. Therefore, it would be emphasized that the aeDE is a fully expected candidate to be employed to address clustering problem regarding the evolutionary approach. However, to adapt this algorithm to the clustering problem, several works are required. For example, the variables in clustering problems are represented as discrete variables in the binary form. Therefore, we need to use a concept of binary chromosome inspired from the idea of [22] to deal with this problem. Moreover, employing binary presentation of chromosome in this paper provides advantages of coding and reducing complexity of the algorithm. Therefore, it would be more robust when dealing with automatic clustering problem. As a result, in this paper, we employ the evolutionary algorithm with binary chromosome representation to tackle the automatic partitioned clustering problem, called binary adaptive elitist differential evolution for clustering of probability density functions (baeDE-CDFs).

From what has been drawn, to fill the research gap and contribute more flexible algorithm for clustering of PDFs, this paper proposes a novel algorithm called baeDE for automatic k -medoids partitioned clustering of PDFs (baeDE-CDFs). More specifically, the clustering problem is restated and solved under the form of an optimal problem where the objective function is based on Silhouette index (SI-index), a modification from an internal validity measure index, Silhouette [23], and the optimal algorithm is baeDE derived from [15] with some modifications. Some advantages of the proposed method can be enumerated as follows. First, probability density function is considered as the main object in this paper instead of discrete element to overcome some shortages of discrete element as mentioned before. Second, by applying k -medoids scheme, the proposed method could face outlier or noise effectively. Third, the proposed algorithm can address the clustering problem without the number of clusters in advance, thus increasing adaptation in such fluctuation digital world. Finally, using new but successful population-based optimal algorithm, namely, aeDE, in clustering problem not only decreases computational burden but also enhances quality of solution. The superiority of the proposed algorithm has been shown through many numerical examples including simulated, Benchmark, and real data sets in a statistically significant manner.

Brief summarization of some discussed approaches for CDFs is illustrated in Table 1 along with the proposed method to clarify its contributions. The rest of the paper is divided into the following sections. Section 2.1 concerns distance measure

TABLE 1: The comparison of the properties of the proposed method and other existing nonhierarchical algorithms for CDFs.

Method	Define number of clusters	Solver	Address outliers/noise	Reach global solution	Computational time
MILXPM-CDF [13]	Given in advance	Classic GA	Not good	Medium	High
GA-CDF [14]	Given in advance	Modified GA	Not good	Medium	High
The method in [24]	Automatically defined	Based on data-driven learning mechanism	Not good if data is overlapping	Good	Low
The method in [25]	Given in advance	k -means	Not good	Medium	Low
The proposed method	Automatically defined	Binary aeDE	Good even for complex data	Good	Lower than [13, 14]

and states the clustering problem with respect to the evolutionary approach. Section 2.2 presents the Silhouette (SI) modified and the external validity measure index to evaluate the quality of produced partitioned clustering. Section 2.3 shows in detail the proposed algorithm. Section 2.4 then will prove the effectiveness of the proposed algorithm via multiple numerical examples in several aspects. Section 4 finally draws some conclusions of the whole work.

2. Materials and Methods

2.1. Distance Measure and Evolutionary Approach for Clustering of PDFs

2.1.1. L1-Distance for Evaluating Difference between PDFs. Usually, clustering multiple objects into the separate groups depends heavily on the type of measures or distances to assess similarity level between objects [6]. Overview of clustering of PDFs and advantages of L1-distance in computing similarity between PDFs have been shown in [26]. Moreover, recent works regularly employ L1-distance in clustering problem such as [5, 24, 27]. Therefore, in this paper, we also use L1-distance as a criterion to evaluate whether two or more objects are similar or dissimilar.

Definition 1. Given $F = \{f_1(x), f_2(x), \dots, f_n(x)\}$, $n > 2$ is the set of PDFs, where n is cardinality PDFs in set F , let $f_{\max} = \max\{f_1(x), f_2(x), \dots, f_n(x)\}$. Then the L1-distance of F is set as

$$\|f_1, f_2, \dots, f_n\|_1 = \int_{R^n} f_{\max}(x) dx - 1. \quad (1)$$

In case $n = 2$, we obtain

$$\|f_1, f_2\|_1 = \int_{R^n} |f_1(x) - f_2(x)| dx. \quad (2)$$

From (1), it is easy to show that $\|f_1, f_2, \dots, f_n\|_1$ is a nondecreasing function on n with $0 \leq \|f_1, f_2, \dots, f_n\|_1 \leq n - 1$.

Equation (2) implies that

$$\|f_1, f_2\|_1 = 2 \left(\int_{R^n} f_{\max}(x) dx - 1 \right). \quad (3)$$

Apparently, the L1-distance also satisfies normal criteria of a distance measure. First, it is always greater than zeros and it

should be symmetric. Moreover, the minimum value should be obtained in case of identical PDFs.

2.1.2. Clustering Problem as an Optimization Problem. Since the clustering problem can be stated in form of an optimization problem, the evolutionary approach can be applied to a clustering problem. The idea is originally derived from Darwin's evolutionary theory. The main point is to use the evolutionary operator and population of clustering structure to converge to a global solution. For more details, the reader can refer to [28]. This kind of clustering problem can be briefly presented as follows.

Input: n (number of PDFs), k (number of clusters), NP (population size).

Output: partitions.

1: Create randomly a *population* consisting of NP *chromosomes* where each *chromosome* corresponds to valid k -partitions of the data.

2: **Repeat.**

3: Evaluate fitness value for each *chromosome* belonging to *population*.

4: Regenerate a new generation of structures.

5: **Until:** some terminate conditions are satisfied.

The number of feasible partitions can be considered as

$$N(n, k) = \frac{1}{k!} \sum_{i=1}^k (-1)^i \binom{k}{i} (k-i)^i, \quad (4)$$

where n is number of PDFs and k is the number of clusters. Moreover, according to [29], the clustering problem is NP-hard when the number of clusters exceeds 3.

2.2. Internal and External Validity Measure Indexes. As discussed in the previous subsection, evaluating the quality of established partitions should be handled by validity measure indexes containing internal and external measures. The two following sub-subsections describe the measures that we use in this paper as fitness function and test criterion for produced partition, respectively.

2.2.1. Internal Measure: Modified Silhouette (SI). As mentioned in the above subsection, most internal validity measures that applied clustering for PDFs concentrate on building internal measures for the type of k -means clustering. Nevertheless, for kind of k -medoids clustering of PDFs, no internal

validity measure is taken into consideration. Therefore, this study proposes a modified Silhouette measure to tackle the problem of k -medoids clustering. Its formulation is shown below.

Notation

$F = \{f_1(x), \dots, f_n(x) \mid x \in \mathfrak{R}^p\}$, $n > 2$ is set of PDFs.

$C = \{C_1, \dots, C_k\}$, clustering data set into k disjoint sets in terms of F , such that $\bigcup_{i=1}^k C_i = F$.

$f_{v_j} = f_j$, $i, j \in C_j$, $j = \overline{1, k}$, $i = \overline{1, n}$, is the representing PDF or medoid of cluster C_j .

These following terms need to be computed:

(i) $a(f_i, C_j)$ is distance from PDF i to the medoid of the cluster it belongs to.

(ii) $d(f_i, C_j)$ is the distance from PDF i to medoids of other clusters.

(iii) $b(f_i)$ is minimum of $d(f_i, C_j)$.

All of them are presented as follows:

$$a(f_i, C_j) = d(f_i, f_{v_j}) \quad \text{with } f_i \in C_j, f_{v_j} \in C_j. \quad (5)$$

$$d(f_i, C_j) = d(f_i, f_{v_j}) \quad \text{with } f_i \notin C_j, f_{v_j} \in C_j. \quad (6)$$

$$b(f_i) = \min_{C_j \in C \setminus C_k} d(f_i, f_{v_j}) \quad (7)$$

with $f_i \in C_k, f_{v_j} \in C_j$.

The local SI value of a single PDF is defined by

$$SI(f_i) = \frac{b(f_i) - a(f_i, C_j)}{\max\{b(f_i), a(f_i, C_j)\}}. \quad (8)$$

The global SI value of a partition can be calculated as

$$SI(C) = \frac{1}{n} \sum_{i=1}^n SI(f_i). \quad (9)$$

All distances here are computed by $L1$ -distance. Further, instead of computing difference from one PDF to other PDFs in another cluster pair by pair, we calculate the distance from that PDF to the medoid of mentioned PDFs. Similarly, when assessing the distance from one PDF to another PDF in one cluster, we also calculate distance from this PDF to its medoid. This manner improves the computation cost significantly compared with the original formula. In case all PDFs f_i in a cluster converge into the medoid of that cluster, it means that the values $SI(f_i)$ of all these f_i are equal to 1, which reflects the compactness of the cluster. In contrast, if the distance from f_i to the medoid of cluster it belongs to is high, but the distance from f_i to the medoid of other clusters is tiny, this leads to $SI(f_i)$ receiving the negative value and, in the worst case, SI should equal -1. This case shows that the separation of the cluster is not good. Therefore, the value of SI is in range $[-1, 1]$ in general and the better value of SI indicates the better result. Nevertheless, in this paper, the algorithm is built to find the

minimum solution; a minus sign should be put ahead of SI to fit with the designed algorithm.

2.2.2. External Measure: Adjusted Rand Index (ARI). Regardless of advantages of the internal validity measure index, one should have another criterion to validate the produced partition and the goodness of the internal measure. The external measure could be a candidate to solve this problem. The usual strategy is making the comparison between the produced partition and the “solution partition” where “ground-truth” labeling is known. The higher the value of the external measure, the better the produced partition. Around a lot of external measures known to the users, the adjusted Rand index, a modified version of Rand index, is considered to be one of the most popular; more details can be found at [30]. If the P and Q are two given partitions of data set, the formulation of ARI is defined as follows:

$$ARI = \frac{a - (a+c)(a+b)/(a+b+c+d)}{((a+c) + (a+b))/2 - (a+c)(a+b)/(a+b+c+d)}, \quad (10)$$

where

a is the count of pairs of elements in the same cluster in P and Q ;

b is the count of pairs of elements in the same cluster in P , but in different clusters in Q ;

c is the count of pairs of elements in a different cluster in P , but in the same cluster in Q ;

d is the count of pairs of elements in a different cluster in both P and Q ;

P and Q are two partitions of set F : one is the produced partition and the other is solution partition, respectively.

2.3. The Proposed Algorithm

2.3.1. Chromosome Presentation. As presented in the previous subsection, clustering problem can apply the evolutionary algorithm to be solved under population structure. Therefore, the chromosome must be presented to build such a population. Because the proposed algorithm is automatic k -medoids clustering, the chromosome is encoded under a binary form. More specifically, at the position where label 1 appears, it means the PDF at that position is considered as a medoid. In addition, the minimal number of clusters must be at least two since it is meaningless to group all objects into one group. Furthermore, the maximum number of clusters is still a question; in this work, the square root of number of PDFs is considered to be the maximum number of clusters. Hence, depending on random value created initially and considered as the number of medoids in each chromosome (k_{random}), the number of medoids may vary leading to the different number of clusters. The following example gives a visible look for encoding the chromosome. Suppose having $n = 9$ PDFs, $F = \{f_1(x), f_2(x), \dots, f_9(x)\}$. Then, some chromosomes can be encoded as follows:

Chromosome 1. $k_{\text{random}} = 3$

1	0	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Chromosome 2. $k_{\text{random}} = 2$

0	1	0	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---

From the two above chromosomes, we see that for chromosome 1: the initial value is created randomly $k_{\text{random}} = 3$, having 3 PDFs encoded as medoids. The positions 1, 3, and 4 are places having label 1, so the corresponding PDFs f_1, f_3, f_4 are considered as the medoids for partitions. Based on this chromosome, the fitness function can be evaluated. The similar explanation is used for chromosome 2.

2.3.2. The Proposed Algorithm. The aeDE which is one of the state-of-the-art population-based optimization algorithms is firstly introduced by [15]. However, this standard aeDE only tackles the real code chromosomes. According to the best of our knowledge, there is no version of binary aeDE used for bit string chromosomes. In addition, since our problem faces automatic k -medoids clustering, a binary chromosome should be more convenient for computation than the real code one. That is the reason why the standard aeDE should be modified in binary form to match our scope. The detailed process is introduced as follows.

Initialization. The algorithm will commence with a randomly created population, including NP individuals. Each individual is a vector containing n design variables representing n input PDFs $\mathbf{px}_i = [x_1 \ x_2 \ \dots \ x_n]$, $x_i = \{0, 1\}$. The scheme to create initial individual is as below:

- (1) $\mathbf{px}_i = \text{zeros}(1, n)$;
- (2) $k_{\text{random}} = \text{randperm}(n, \text{randi}([k_{\min}, k_{\max}]));$
- (3) $px_{ik_{\text{random}}} = 1, i = 1, \dots, NP,$

where

- (i) n is the total number of PDFs;
- (ii) $k_{\min}$ is the minimum number of clusters, usually 2;
- (iii) $k_{\max}$ is the maximum number of clusters, usually square root of n ;
- (iv) NP is the population size.

First, a zero vector $\mathbf{px}_i$ is created. Next, the operator randi selects a random integer according to the chosen range. Subsequently, the operator randperm will randomly select $k = \text{randi}([k_{\min}, k_{\max}])$ unique integers from 1 to n inclusive. Then, the considered vector $\mathbf{x}_i$ at position k_{random} will be replaced by bit 1. After running to NP loops, a population including NP binary individuals will be available for next phases.

Mutation Phase. By suggestion of [15], the mutation process of aeDE algorithm is modified by two adaptive phases. In the first phase, the algorithm hopes to maintain the diversity of the population and tries to prevent the population from jumping into the optimal local solution. Therefore, the operator mutation “rand/1” is employed since it is good at global search [15]. Nevertheless, this operator has its main drawbacks such as slow convergence to optimal global solution and being bad at searching the local optimal solution [31]. That is the reason why the second adaptive phase appears. This phase intends to accelerate convergence speed of the population toward the best individual after the global solution region already located. Thus, the mutation operators, which prefer local solution searching ability, are employed in this phase. They would be “current-to-best/1” or “best/1” due to their effective searching capacity for local solution. After some experiments, we see that the operator “best/1” may be more suitable for the type of our problem. Hence, this mutation operator is employed for the second phase in the mutation scheme of our algorithm. However, there must be a value called *threshold* to know when the first phase terminates or the second phase starts. When the absolute value of the deviation of the objective function between the best individual and the entire population in the previous generation (denoted *delta*) is smaller than the *threshold*, the transfer will be taken. By this effective modification, the ability to search the global and the local optimal solutions as well as convergence rate is enhanced significantly in comparison with the original DE [15].

Here are some mutation operators already discussed above.

$$(i) \text{ rand/1: } \mathbf{v}_i = \mathbf{px}_{r_1} + F \times (\mathbf{px}_{r_2} - \mathbf{px}_{r_3}). \quad (11)$$

$$(ii) \text{ current-to-best/1: } \mathbf{v}_i = \mathbf{px}_i + F \times (\mathbf{px}_{\text{best}} - \mathbf{px}_i) + F \times (\mathbf{px}_{r_1} - \mathbf{px}_{r_2}). \quad (12)$$

$$(iii) \text{ best/1: } \mathbf{v}_i = \mathbf{px}_{\text{best}} + F \times (\mathbf{px}_{r_1} - \mathbf{px}_{r_2}). \quad (13)$$

Probability Estimation Operator. However, it seems like when employing the modification scheme from aeDE, the new mutant individual is still the real-coded one. Therefore, inspired by the idea of [22], in this paper, we employ the scheme of the probability estimation operator to generate the binary coded mutant individual. Here, in mutation process, the author creates multiple probability models at each iteration from information gained from parent individuals. For

more details, the probability estimation operator is defined as follows:

$$P(v_{ij}) = \frac{1}{\{1 + \exp[-2b * (MO - 0.5) / (1 + 2F)]\}} \quad (14)$$

$$MO = px_{r_1j} + F * (px_{r_2j} - px_{r_3j}),$$

where

- (i) F is the scale factor;
- (ii) $px_{r_1j}, px_{r_2j}, px_{r_3j}$ are the j th-bits of three randomly selected individuals;
- (iii) b is denoted as bandwidth factor and receives a positive real constant.

By this scheme, three parent individuals will be taken into consideration to capture differential information aiming to establish the probability distribution model. The bit of the mutant individual will then be decided, based on this model, whether “1” or “0”. The appearance of parameter b plays a role in tuning the range and shape of the probability distribution model. One suitable value of b will efficiently improve searching capability as well as retaining the diversity of population simultaneously [32].

Once probability estimation vector is determined, the corresponding mutant individual pv_i is deduced by the following equation:

$$pv_{ij} = \begin{cases} 1, & \text{if rand}() \leq \Pr(v_{ij}) \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

where

rand is a uniform distributed random number in range $[0, 1]$;

$\Pr(v_{ij})$ is the j th component of probability vector of the i th target individual.

Crossover Phase. Subsequently, the crossover phase aims to produce trial individual pu_i by employing the binomial crossover operation. This operation is performed by exchanging some elements between the target individual and its mutant individual. It can be illustrated by the equation below:

$$pu_{ij} = \begin{cases} pv_{ij}, & \text{if rand}() \leq CR \text{ or } j = j_{\text{rand}} \\ px_{ij}, & \text{otherwise,} \end{cases} \quad (16)$$

where $i = 1, 2, \dots, NP$, $j = 1, 2, \dots, n$; j_{rand} is an integer number selected from 1 to n ; and CR is the crossover rate ranging from $[0.7, 1]$.

Selection Phase. In traditional DE, each trial individual pu_i generated after crossover phase will be compared with the target vector px_i to select a better individual for the next generation. However, by this way, some good information of ignored individuals can be forgotten. Somehow, this ignored individual is still better than other ones in the entire population although it is worse than its target individual in

comparison with pair-by-pair one. Therefore, the authors Ho-Huu et al. [15] have proposed the elitist selection scheme original from [33] to maintain the good individual in the current population. This scheme will firstly combine both trial individuals in children population C and target individuals in parent population P into one population, called Q . Then, NP best individuals will be selected from this combined population Q to establish population for the next generation. By this way, all the best individuals are always stored and maintained for the next generation. As a result, the convergence rate of the algorithm will be improved, too.

Stopping Condition. As discussed in [15], the baeDE-CDFs will be stopped either when value of *delta* is greater than that of *tolerance* or when the maximum number of iteration (*Maxiter*) is reached. Here, the values of *tolerance* and *Maxiter* are, respectively, set to $1e-6$ and 2000 for all examples.

By the improvement in aeDE algorithm combined with the probability density estimation, the baeDE-CDFs produces binary chromosome which increases the flexibility of such fast and robust optimization algorithm as aeDE in various circumstances. The flowchart in Figure 1 shows general process of the proposed algorithm baeDE-CDFs.

2.4. Numerical Examples

2.4.1. Simulation Strategy. In this subsection, the proposed baeDE-CDFs in Section 2.3 is employed to solve numerous clustering problems. The outline of simulation strategy is briefly presented as follows. Firstly, the details of each data set will be presented. Secondly, the influence of the mutant factor F , crossover control parameter CR , population size NP , threshold *thres*, and bandwidth factor b of baeDE-CDFs on the optimal solution is examined for all examples. Nevertheless, one of these investigations is demonstrated only to avoid prolixity. Subsequently, based on the obtained results of investigations, some adequate parameters are recommended later. Thirdly, a comparison between the k -means based algorithms and k -medoids based baeDE-CDFs in terms of ARI (adjusted Rand index) is designed to measure the effectiveness of baeDE-CDFs. It would be noticed that all algorithms are performed over 50 independent loops to guarantee the stability of algorithm and the result is taken as average value plus the standard deviation. Further, statistics tests are also performed to validate whether the difference is significant or not. Next, another comparison between the evolutionary algorithms in terms of both accuracy and computational time is performed to evaluate the impact of applying the aeDE technique to clustering problem. The statistical tests are also employed there with the same purpose. Finally, some primary conclusions are drawn based on the deduced results. Besides, it is worth noticing that all the problems are implemented on 2015-version Matlab software on an Intel® Core™ i5-2400 CPU @ 3.10 GHz with 4GB main memory in Windows Server 2010 environment.

2.4.2. Data Sets Used. In this paper, we approach the problem from the point of view of academic study. At the beginning, some Benchmark examples are performed to investigate

FIGURE 1: Flowchart of baeDE-ACDF algorithm.

some properties as well as effectiveness of the proposed algorithm. Then, it is extended to real application with more complex problem. It is also noticed that the size and complexity of data sets of examples gradually increase from Benchmark examples to real application. Herein, a summary introduction of each data set is presented, but more characteristics of each data set such as total of objects, number of clusters, nominal partition, or complexity level can be found in Table 2. Moreover, the data set has the explicit distribution; the PDF of that data set is estimated based on the explicit formula (example 1 for instance). Nevertheless, if the data set has not known the distribution yet, such as data sets in examples 2, 3, 4, and 5, the kernel method is employed to estimate the PDFs as follows. Let $x_1, x_2, \dots, x_n$ be N discrete elements including n dimensions which are employed to estimate the PDFs. The kernel formula is shown as follows:

$$\hat{f}(x) = \frac{1}{Nh_1 h_2 \dots h_n} \sum_{i=1}^N \prod_{j=1}^n K_j \left(\frac{x - x_j}{h_j} \right), \quad (17)$$

where $h_j = (4/N(n+2))^{1/(n+4)} \times \sigma_j$ is a bandwidth parameter for the j th variable; $K_j(\cdot)$ is a kernel function of the j th variable which is usually Gaussian, Epanechnikov, Biweight, and so on.

In the above formula, there are two important parameters that need to be estimated, the bandwidth h and the kernel

function K . Recently, there are various ways to choose these parameters. In this paper, we take the estimation of h according to suggestion of Scott [34] and the kernel function is the Gaussian one.

(i) *Example 1.* Seven univariate normal probability density functions (PDFs) are considered. This data firstly proposed by [25] is one of the simulated databases with “well-separated” class structure. The details of estimated parameters can be found in [25]. Seven PDFs estimated from these parameters can be seen in Figure 2.

(ii) *Example 2.* Some samples of Brodatz data set are taken into account [35], which are D4, D10, D12, and D34, respectively, as shown in Figure 3. Each of the samples is cropped randomly into ten pieces with uniform size of 256×256 pixels so that there are forty images in total. The PDFs estimated from these images are demonstrated in Figure 4.

(iii) *Example 3.* Thirty texture images from UIUC data set are involved in this case, separated into three categories: Floor, Wall, and Upholstery as illustrated in Figure 5 [36]. PDFs estimated from all images are demonstrated in Figure 6. It is obvious that the complexity of this data set is higher than the two previous ones because of more remarkable overlapping classes.

TABLE 2: Characteristics of data sets surveyed in section of numerical example.

Data set	Object	Number of objects	Exact number of clusters	Nominal partition	Complexity
Benchmark	Univariate normal distribution	7	3	$C_1 = \{f_1, f_4\}, C_2 = \{f_2, f_5, f_7\},$ $C_3 = \{f_3, f_6\}.$	simple
Brodatz	Image 256 x 256 pixel	40	4	$C_1 = \{f_1, \dots, f_{10}\}, C_2 = \{f_{11}, \dots, f_{20}\},$ $C_3 = \{f_{21}, \dots, f_{30}\}, C_4 = \{f_{31}, \dots, f_{40}\}.$	pretty complex
UIUC	Image 640 x 480 pixel	30	3	$C_1 = \{f_1, \dots, f_{10}\}, C_2 = \{f_{11}, \dots, f_{20}\},$ $C_3 = \{f_{21}, \dots, f_{30}\}.$	complex
CUReT	Material Image 640 x 480 pixel	30	3	$C_1 = \{f_1, \dots, f_{10}\}, C_2 = \{f_{11}, \dots, f_{20}\},$ $C_3 = \{f_{21}, \dots, f_{30}\}.$	hardly complex
Traffic data	Digital image 1920 x 1080 pixel	100	2	$C_1 = \{f_1, \dots, f_{50}\}, C_2 = \{f_{21}, \dots, f_{100}\}.$	significantly complex

FIGURE 2: Seven PDFs estimated from univariate normal distributions in example 1.

(iv) *Example 4.* Thirty images derived from the Columbia-Utrecht Reflectance and Texture Database (CURET) are considered in the example, clustered into three groups: Human Skin, Ribbed Paper, and Insulation as shown in Figure 7. The details of CURET data set can be found according to the following link <http://www1.cs.columbia.edu/CAVE/software/curet/>. Estimation of PDFs is depicted in Figure 8. In this case, it can be seen that the complexity is a little more remarkable than the previous cases due to significant overlapping area.

(v) *Example 5.* In this example, a real application is extended to deal with the traffic problem. Commonly, traffic congestion is always a headache problem in Asian countries, and so is Vietnam. Specially, in rush hour, this problem is likely more complicated. Concerning this problem, this paper

TABLE 3: Influence of mutant factor F on the optimal solution in example 5.

F	Fitness value min SI	Iteration	Time (seconds)
0.4	-0.637	54	10.888
0.6	-0.628	187	71.137
0.8	-0.622	234	88.815
1.0	-0.622	252	103.140
[0.4, 1.0]	-0.635	182	63.106

contributes one solution by applying the proposed algorithm to determine which photos present traffic jam and which present non-traffic jam. A total of 100 digital images extracted from the short video in front of Ton Duc Thang University are considered. These are divided into two main groups: traffic jam and non-traffic jam, respectively. Some of such photos are presented in Figure 9 and distribution of PDFs estimated from the photos is shown in Figure 10.

3. Results and Discussion

3.1. Influence of the Parameters F , CR , NP , $Thres$, and b on the Optimal Solution. One of the major problems of evolutionary problem is parameter setting. As a result, a survey of influence of parameter setting on algorithm performance in each example will be presented first. This aims to get satisfactory parameters of F , CR , NP , $thres$, and b of baeDE-CDFs for each problem. However, to avoid wordiness, only typical example is demonstrated in detail. Based on the obtained results, an adequate parameter set is recommended.

In this paper, an investigation of how parameters in baeDE-CDFs influence the optimal solution in case of example 5 is presented carefully. In detail, the impact of the mutant factor is described in Table 3 and Figure 11, and the influence of the crossover control parameter is illustrated in

FIGURE 3: Four samples consisting of D4, D10, D12, and D34 derived from Brodatz data set.

FIGURE 4: Forty PDFs estimated from 40 images of 4 types including D4, D10, D12, and D34 in Brodatz data set.

TABLE 4: Influence of crossover control parameter CR on the optimal solution in example 5.

CR	Fitness value min SI	Iteration	Time (seconds)
0.7	-0.634	133	42.782
0.8	-0.635	151	51.031
0.9	-0.636	209	73.674
1	-0.635	204	74.662
[0.7, 1.0]	-0.634	140	48.936

TABLE 5: Influence of population size NP on the optimal solution in example 5.

NP	Fitness value min SI	Iteration	Time (seconds)
30	-0.635	176	60.697
35	-0.635	213	88.760
40	-0.636	263	125.818
45	-0.634	284	157.953
50	-0.636	335	204.885

Table 4 and Figure 12. Subsequently, a conclusion that can be drawn is that it would be better if values of F and CR lie in intervals $[0.4, 1.0]$ and $[0.7, 1.0]$, respectively, and then a balance between the solution quality and computational cost

TABLE 6: Influence of threshold $thres$ on the optimal solution in example 5.

$thres$	Fitness value min SI	Iteration	Time (seconds)
1e_1	-0.629	34	12.330
1e_2	-0.632	132	47.281
1e_3	-0.635	162	55.781
1e_4	-0.634	166	56.890
1e_5	-0.636	205	72.003

of baeDE-CDFs would be better than those obtained by other values.

Table 5 and Figure 13 depict the impact of population size NP on the optimal solution. It can be found that there is an upward trend in computational cost as the population size is increasing. However, no significant change is found in the value of fitness function, just a minor difference. Therefore, $NP = 30$ would be a proper choice that can satisfy both the quality solution and computational cost.

The influence of threshold $thres$ on the optimal result is indicated in Table 6 and Figure 14. It can be realized that there is a significant increase in computational cost when the threshold is stricter, in contrast to the trend of fitness value. However, there is just a tiny fluctuation of fitness value when $thres \in [1e_3, 1e_5]$. Therefore, it can be seen that, with a larger value of $thres$, the algorithm converges rapidly with less iterations, but the result is unstable. In contrast, with a stricter value of $thres$, the algorithm converges more slowly, but the result is more stable. Hence, it would be appropriate to choose $thres = 1e_3$ as the threshold for this example.

The impact of bandwidth on the solution quality is shown in Table 7 and Figure 15. It can be observed that both the computational cost and fitness value are higher than expected when value of bandwidth ranges from 0 to 10. From bandwidth of 20 or greater, the computational cost as well as the solution quality is more stable. However, to achieve a balance between computational cost and solution quality, a bandwidth of 50 is suggested in this case.

For the remaining examples, parameter setting is performed in a similar way. In detail, for examples 1, 2, and 3, the population size is set to 25, and for example 4, it is 30. The bandwidth of examples 1, 3, and 4 is 20 and for example 2 is 50. The values of F , CR are set within the ranges $[0.4, 1.0]$ and $[0.7, 1.0]$, respectively, for all examples. Similarly, threshold of $1e_3$ is set for all examples.

FIGURE 5: Three samples including Floor, Wall, and Upholstery in UIUC data set.

FIGURE 6: Thirty PDFs estimated from thirty images of 3 types including Floor, Wall, and Upholstery in UIUC data set.

TABLE 7: Influence of bandwidth b on the optimal solution in example 5.

b	Fitness value min SI	Iteration	Time (seconds)
0	-0.593	2000	991.394
2	-0.6	2000	981.214
4	-0.605	2000	974.628
6	-0.622	2000	958.310
8	-0.636	1870	814.326
10	-0.637	767	312.356
20	-0.636	170	59.523
50	-0.635	137	45.888
100	-0.635	138	46.111
200	-0.635	146	49.634
500	-0.636	155	52.326
1000	-0.636	131	43.955

3.2. Evaluation of Scheme to Select the Representative PDF. In this subsection, an evaluation regarding the way to choose the representative PDF is presented. In detail, there are two main schemes to select the representative PDF as mentioned before. The first is based on the k -means method, which

selects the average of PDFs as the representative PDF; the second relies on the k -medoids method which picks up one of PDFs as the representative PDF. In this paper, the representative PDF is chosen with regard to second way. Therefore, we would like to measure whether the k -medoids based representative PDF is superior to the k -means based one or not. Herein, three k -means based clustering algorithms are considered to have a comparison with the proposed algorithm, baeDE-CDFs in terms of ARI. The first is an automatic clustering algorithm proposed by [24]; the second is a state-of-the-art crisp clustering algorithm suggested by Thao Trang et al., namely, GA-CDF [14]; the third is a nonhierarchical clustering algorithm derived from [25].

The evaluation strategy is outlined as follows. Firstly, the Shapiro-Wilk test is employed to test the normality of ARI derived from algorithms. The result confirms that all ARI of algorithms do not follow a normal distribution due to $\text{sig.} < 0.05$ as illustrated in Table 8 and Figure 16. As a result, some nonparametric tests are used in this case. Specifically, for comparison of mean difference between ARI of baeDE-CDFs and that of remaining algorithms, the Mann-Whitney test is employed; for rank of performance of algorithms, the Kruskal-Wallis test is used. The detailed result of all descriptive statistics, test, and ranking is demonstrated in Table 9. However, it should be noticed that in the case ranks of algorithms' performance are not statistically significantly different, the ranks are not presented in the table. Besides, a sign "+" indicates that there is a statistically significant difference between baeDE-CDFs and compared algorithm and vice versa for a sign "-".

From the table, one common point that can be drawn is that the proposed algorithm is always superior to the remaining algorithms in terms of accuracy. For example, for the first three examples, the baeDE-CDFs maintain the absolute value of ARI, along with the standard deviation being 0. In addition, it is also ranked the first by result of Kruskal-Wallis test and the difference between baeDE-CDFs is statistically significant. These findings confirm that baeDE-CDFs really works well in the first three examples as proved by its accuracy and stability.

In last two examples, there is a downward trend of ARI of most algorithms, including baeDE-CDFs. It is worth noticing that most k -means based algorithms do not perform well in this case. For instance, algorithm of Chen and Hung constantly produces ARI 0 in both last cases; GA-CFD is a bit better but still achieves a low ARI, just around 0.5. In contrast, baeDE-CDFs is likely to be more superior by its ARI

FIGURE 7: Three samples including Human Skin, Ribbed Paper, and Insulation in CURET data set.

TABLE 8: Test of normality with significant level 0.05 for ARI value derived from baeDE-CDF for CURET data set.

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
CURET data set	.523	50	.000	.380	50	.000

a. Lilliefors Significance Correction

FIGURE 8: Thirty PDFs estimated from thirty images of 3 types including Human Skin, Ribbed Paper, and Insulation in CURET data set.

persistently greater than 0.9 in the last two cases. Further, the mean test by Mann-Whitney and ranking by Kruskal-Wallis authenticate the dominance of baeDE-CDFs over the remaining algorithms in last cases.

In general, from performance of algorithms in this part, it is obvious that the k -medoids based algorithm baeDE-CDFs has proved its superiority over the compared k -means algorithms in terms of ARI. In particular, in complex problem like examples 4 and 5, this superiority is more apparent. Therefore, it would be noticed that the way to choose a representative PDF in clustering really plays a pivotal role in algorithm's performance. In particular, through all examples investigated, selecting representative PDF based on k -medoids shows an outstanding performance compared with one based on k -means.

3.3. Evaluation within Evolutionary Algorithms Only. In this subsection, one more survey is executed to measure the performance of evolutionary algorithms applied to clustering

problem. To the best of our knowledge, there are two available algorithms using population-based techniques to perform the clustering problem: one is GA-CDF of [14] and the other is MI-LXPM-CDF of [13]. In this circumstance, these algorithms are considered to have a comparison with baeDE-CDFs in five examples. Moreover, all these algorithms will be performed with the same objective function: modified SI. In detail, performance of all algorithms would be measured on both aspects: computational time and accuracy. Herein, computational time is computed based on average time of 50 running loops and so does ARI. Besides, some similar tests as in previous part are also performed to confirm statistical significance of difference in ARI between algorithms.

The detailed result of all algorithms is depicted in Table 10 and Figures 17 and 18. For the first example, we witness a good performance of all algorithms by ARI values approximating 1. Therefore, a statistical nonsignificant mean difference of ARI values between baeDE-CDFs has been found in this case. Similarly, ranking by Kruskal-Wallis is also useless so that ranks are not presented here. However, regarding computational time, it is noticed that baeDE-CDFs consumes the least time and with MILXPM-CDF the opposite is true.

For image data sets, we record the worse performance of two algorithms, namely, GA-CDF and MILXPM-CDF, compared with baeDE-CDFs with regard to accuracy. Both GA-CDF and MILXPM-CDF achieve ARI value lower than 0.5, with even negative value for MILXPM-CDF in case of CURET data set. Moreover, results from Mann-Whitney test have confirmed the superiority of baeDE-CDFs over the remaining genetic algorithms with statistically significant level of 0.05. Further, ranks derived from Kruskal-Wallis test also show that baeDE-CDFs is always ranked first in four later data sets.

In connection with speed of algorithms' performance in four later data sets, due to the image object presented by a lot of pixel points, the computational time also increases. More specifically, computation speed of GA-CDF and MILXPM-CDF is far more inferior to that of baeDE-CDFs. The baeDE-CDFs is persistently of the first rank, while second and third positions are occupied by GA-CDF and MILXPM-CDF. In particular, in real application, the computational time is

FIGURE 9: Two pictures represent traffic state: non-traffic jam and traffic jam, respectively, in front of Ton Duc Thang University.

TABLE 9: Comparison of performance of k -means based algorithms and k -medoids based baeDE-CDF in terms of mean of ARI, standard deviation, result of Mann-Whitney test, and ranking by Kruskal-Wallis test.

Data set	Algorithm	Mean	Std.	Result of Mann-Whitney test	Rank by Kruskal-Wallis test
7 normal PDFs	<i>baeDE-CDF</i>	1.000	0.000		142.50
	Chen and Hung	0.442	0.000	+	50.50
	GA-CDF	0.945	0.152	+	135.18
	Vovan and PhamGia	0.469	0.493	+	73.82
Brodatz images	<i>baeDE-CDF</i>	1.000	0.000		162.5
	Chen and Hung	0.316	0.000	+	49.00
	GA-CDF	0.324	0.073	+	56.44
	Vovan and PhamGia	0.772	0.255	+	134.06
UIUC images	<i>baeDE-CDF</i>	1.000	0.000		158.50
	Chen and Hung	0.554	0.000	+	68.50
	GA-CDF	0.541	0.083	+	62.22
	Vovan and PhamGia	0.759	0.368	+	112.78
CURET images	<i>baeDE-CDF</i>	0.981	0.056		172.7
	Chen and Hung	0.000	0.000	+	26.50
	GA-CDF	0.571	0.084	+	118.41
	Vovan and PhamGia	0.366	0.255	+	84.39
Traffic images	<i>baeDE-CDF</i>	0.909	0.131		173.66
	Chen and Hung	0.000	0.000	+	43.50
	GA-CDF	0.55	0.075	+	127.17
	Vovan and PhamGia	0.044	0.093	+	57.67

TABLE 10: Comparison of performance of evolutionary algorithms only in terms of mean, standard deviation, result of Mann-Whitney test, and ranking by Kruskal-Wallis test.

Data set	Algorithm	Mean	Std.	Result of Mann-Whitney test	Rank by Kruskal-Wallis test	Time (Seconds)
7 normal PDFs	<i>baeDE-CDF</i>	1.000	0.000		-	0.340
	GA-CDF	0.991	0.065	-	-	7.054
	MILXPM-CDF	1.000	0.000	-	-	7.832
	<i>baeDE-CDF</i>	1.000	0.000		125.50	13.699
Brodatz images	GA	0.324	0.073	+	35.50	114.885
	MILXPM-CDF	0.132	0.000	+	65.50	126.987
	<i>baeDE-CDF</i>	1.000	0.000		125.50	4.677
UIUC images	GA-CDF	0.094	0.083	+	50.50	40.45
	MILXPM-CDF	0.094	0.083	+	50.50	37.72
	<i>baeDE-CDF</i>	0.982	0.056		125.50	3.973
CURET images	GA-CDF	0.049	0.093	+	70.50	36.07
	MILXPM-CDF	-0.041	0.000	+	30.50	41.52
	<i>baeDE-CDF</i>	0.909	0.131		114.06	62.958
Traffic images	GA-CDF	0.087	0.051	+	26.940	551.573
	MILXPM-CDF	0.908	0.131	+	85.50	408.621

FIGURE 10: One hundred PDFs estimated from images in real traffic data set derived from a traffic video taken in front of Ton Duc Thang University.

FIGURE 11: Influence of the mutant factor F on the optimal solution.

a burden for evolutionary algorithms in such a problem. Particularly for cases of GA-CDF and MILXPM-CDF, the computational time is overestimated. However, looking back into the proposed method baeDE-CDFs, although the computational time may be remarkable, it seems nothing to that of the two remaining algorithms. Therefore, one would state that aeDE, being modified in binary chromosome and combined with estimation of distribution algorithm and the fitness function Silhouette used for k -medoids, produces more valid results than other genetic algorithms.

4. Conclusions

In this work, two primary modifications of aeDE are proposed to give the so-called binary adaptive elitist differential evolution for clustering of probability density functions (baeDE-CDFs) for solving clustering problem. Firstly, a k -medoids based scheme is suggested instead of k -means based one to be a representative PDF. This replacement helps the algorithm address outliers well leading to expecting a more accurate result. Secondly, a binary form of aeDE is suggested for the first time, which is inspired by the concept of estimation of distribution algorithm. This helps original aeDE to be more convenient for computation problem as well as taking advantage of the probability distribution model. In addition, an internal validity measure, namely, SI, is modified to be appropriate for k -medoids based automatic clustering and

FIGURE 12: Influence of the crossover control parameter CR on the optimal solution.

FIGURE 13: Influence of the population size Pop on the optimal solution.

FIGURE 14: Influence of the threshold $thres$ on the optimal solution.

also to contribute a measure index in field of clustering for PDFs.

The superiority of baeDE-CDFs has been already proved in the numerical example subsection. In comparison with k -means based clustering algorithm, baeDE-CDFs defeats convincingly the other algorithms in the literature in a statistically significant way in terms of quality of solution. As a result, it is always of the first rank according to Kruskal-Wallis nonparametric test. In comparison with evolutionary algorithms only, baeDE-CDFs is still far superior to GA-CDF and MILXPM-CDF both in quality of solution and computational time. In particular, the accuracy and speed of baeDE-CDFs are still maintained in more complex problem as example 5; meanwhile a poor quality of solution and overestimated computational time are witnessed in the remaining algorithms.

These illustrate that the baeDE-CDFs can offer a robust, effective, and reliable optimization method for solving clustering of PDFs, even in complicated cases. In addition, baeDE-CDFs is just quite similar to original aeDE. For that reason, it is simple to understand and implement. Moreover, with fast convergence rate, useful global search ability, and automatic clustering, the baeDE-CDFs can be an attractive tool for dynamic clustering of entirely unknown data sets.

FIGURE 15: Influence of the bandwidth b on the optimal solution. FIGURE 16: Performance comparison of k -means based algorithms and k -medoids based baeDE over five examples in terms of ARI value.

FIGURE 17: Performance comparison of evolutionary algorithms only over five examples in terms of ARI value.

Therefore, from what has been concluded, one can extend the application of the proposed algorithm in numerous fields. For example, it is possible to reduce traffic congestion

FIGURE 18: Performance comparison of evolutionary algorithms only over five examples in terms of computational time.

in Vietnam by applying the proposed algorithm. Further, attenuating such traffic congestion could bring benefit to the citizens because they will spend less time in transit and have more time for their business. In addition, applying the proposed algorithm in recognizing materials in factories is also achievable. Hence, these extensions would be considered in our upcoming research.

Data Availability

For examples 1, 2, 3 and 4, the data used to support the findings of this study are included within the article. For example 5, the traffic data used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] N. Bouguila and W. ElGuebaly, "Discrete data clustering using finite mixture models," *Pattern Recognition*, vol. 42, no. 1, pp. 33–42, 2009.
- [2] Z. Izakian, M. Saadi Mesgari, and A. Abraham, "Automated clustering of trajectory data using a particle swarm optimization," *Computers, Environment and Urban Systems*, vol. 55, pp. 55–65, 2016.
- [3] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.
- [4] G. George, M. R. Haas, and A. Pentland, "Big data and management," *Academy of Management Journal (AMJ)*, vol. 57, no. 2, pp. 321–326, 2014.
- [5] T. Nguyentrang and T. Vovan, "Fuzzy clustering of probability density functions," *Journal of Applied Statistics*, vol. 44, no. 4, pp. 583–601, 2017.
- [6] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, NY, USA, 2009.
- [7] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [8] L. Kaufman and P. Rousseeuw, *Clustering by Means of Medoids*, North-Holland, 1987.
- [9] B. Zhang, L. Xiang, and X. Liu, *An Improved Ranked K-medoids Clustering Algorithm Based on a P System BT - Human Centered Computing*, Q. Zu and B. Hu, Eds., Springer International Publishing, Cham, Switzerland, 2018.
- [10] F. Liu, J. Zhang, and E. Zhu, "Test-suite reduction based on k-medoids clustering algorithm," in *Proceedings of the 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 186–192, Nanjing, China, 2017.
- [11] L. Cahaya, L. Hiryanto, and T. Handhayani, "Student graduation time prediction using intelligent K-Medoids Algorithm," in *Proceedings of the 2017 3rd International Conference on Science in Information Technology (ICSITech)*, pp. 263–266, Bandung, Indonesia, October 2017.
- [12] P. Prihandoko, Bertalya, and M. I. Ramadhan, "An analysis of natural disaster data by using K-means and K-medoids algorithm of data mining techniques," in *Proceedings of the 2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering*, pp. 221–225, Nusa Dua, July 2017.
- [13] V. Tai, N. T. Thao, and C. N. Ha, "Clustering for probability density functions based on genetic algorithm," in *Proceedings of the 1st International Conference on Applied Mathematics in Engineering and Reliability (ICAMER 2016)*, (Ho Chi Minh City, Vietnam, 4–6 May 2016), p. 51, 2016.
- [14] T. Vo-Van, T. Nguyen-Thoi, T. Vo-Duy, V. Ho-Huu, and T. Nguyen-Trang, "Modified genetic algorithm-based clustering for probability density functions," *Journal of Statistical Computation and Simulation*, vol. 87, no. 10, pp. 1964–1979, 2017.
- [15] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang, "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables," *Computers & Structures*, vol. 165, pp. 59–75, 2016.
- [16] B. Hegerty, C. Hung, and K. Kasprak, "A comparative study on differential evolution and genetic algorithms for some combinatorial problems," in *Proceedings of the Mexican International Conference on Artificial Intelligence*, 2009.
- [17] V. Ho-Huu, T. Vo-Duy, T. Luu-Van, L. Le-Anh, and T. Nguyen-Thoi, "Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme," *Automation in Construction*, vol. 68, pp. 81–94, 2016.
- [18] V. Ho-Huu, T. Vo-Duy, T. Nguyen-Thoi, and L. Ho-Nhat, "Optimization of truss structures with reliability-based frequency constraints under uncertainties of loadings and material properties," in *Proceedings of the 1st International Conference on Applied Mathematics in Engineering and Reliability*, pp. 59–65, CRC Press, 2016.
- [19] T. Vo-Duy, V. Ho-Huu, T. Do-Thi, H. Dang-Trung, and T. Nguyen-Thoi, "A global numerical approach for lightweight design optimization of laminated composite plates subjected to frequency constraints," *Composite Structures*, vol. 159, pp. 646–655, 2017.
- [20] T. Vo-Duy, T. Truong-Thi, V. Ho-Huu, and T. Nguyen-Thoi, "Frequency optimization of laminated functionally graded carbon nanotube reinforced composite quadrilateral plates using smoothed FEM and evolution algorithm," *Journal of Composite Materials*, vol. 52, no. 14, pp. 1971–1986, 2017.
- [21] K. Demertzis and L. Iliadis, "Adaptive elitist differential evolution extreme learning machines on big data: intelligent recognition of invasive species," in *Advances in Big Data: Proceedings of the 2nd INNS Conference on Big Data*, P. Angelov, Y. Manolopoulos, and L. Iliadis, Eds., vol. 529 of *Advances in Intelligent Systems and Computing*, pp. 333–345, Springer International Publishing, Cham, Switzerland, 2016.
- [22] S. Baluja, *Population-Based Incremental Learning. A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, Carnegie-Mellon Univ Pittsburgh Pa Department of Computer Science, 1994.
- [23] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [24] J. H. Chen and W.-L. Hung, "An automatic clustering algorithm for probability density functions," *Journal of Statistical Computation and Simulation*, vol. 85, no. 15, pp. 3047–3063, 2015.
- [25] T. V. Van and T. Pham-Gia, "Clustering probability distributions," *Journal of Applied Statistics*, vol. 37, no. 11, pp. 1891–1910, 2010.
- [26] T. Pham-Gia, N. Turkkan, and T. Vovan, "Statistical discrimination analysis using the maximum function," *Communications in Statistics—Simulation and Computation*, vol. 37, no. 1–2, pp. 320–336, 2008.
- [27] T. Vovan, " L^1 -distance and classification problem by Bayesian method," *Journal of Applied Statistics*, vol. 44, no. 3, pp. 385–401, 2017.
- [28] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 39, no. 2, pp. 133–155, 2009.
- [29] P. Brucker, "On the complexity of clustering problems," in *Optimization and Operations Research*, R. Henn, B. Korte, and W. Oettli, Eds., vol. 157 of *Lecture Notes in Economics and Mathematical Systems*, pp. 45–54, Springer, Berlin, Germany, 1978.
- [30] L. M. Collins and C. W. Dent, "Omega: a general formulation of the rand index of cluster recovery suitable for non-disjoint

- solutions,” *Multivariate Behavioral Research*, vol. 23, no. 2, pp. 231–242, 1988.
- [31] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
 - [32] L. Wang, X. Fu, Y. Mao, M. Ilyas Menhas, and M. Fei, “A novel modified binary differential evolution algorithm and its applications,” *Neurocomputing*, vol. 98, pp. 55–75, 2012.
 - [33] A. W. Mohamed and H. Z. Sabry, “Constrained optimization based on modified differential evolution algorithm,” *Information Sciences*, vol. 194, pp. 171–208, 2012.
 - [34] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2015.
 - [35] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Pubns, 1966.
 - [36] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

Submit your manuscripts at
www.hindawi.com