A Novel Cluster Prediction Approach Based on Locality-Sensitive Hashing for Fuzzy Clustering of Categorical Data

TOAN NGUYEN MAU1, YASUSHI INOGUCHI 1, (Member, IEEE), and VAN-NAM HUYNH1, (Member, IEEE)
1School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology, Ishikawa, Japan (e-mail: {nmtoan91,inoguchi,huynh}@jaist.ac.jp)
Corresponding authors: Toan Nguyen Mau and Van-Nam Huynh (e-mail: huynh@jaist.ac.jp).

ABSTRACT This paper addresses the problem of fuzzy clustering for categorical data. During the last two decades, many attempts have been made to extend the $k$-means algorithm, making it applicable to clustering for categorical data, due to its simplicity and efficiency. However, as $k$-means-like algorithms are local optimization methods, their clustering results are highly sensitive to initialization. In this paper, we propose to use Locality-Sensitive Hashing (LSH) to reduce the categorical data dimensions and predict the initial fuzzy clusters in low-dimensional space. Different from the existing cluster initialization optimization methods that aim to create only crisp initial clusters, the proposed method aims at predicting ‘high quality’ fuzzy clusters at the initialization step before proceeding in the $k$-means-like fashion. The numerical results show that the proposed method yields relatively accurate results on 16 UCI datasets and outperforms all other related approaches in terms of both crisp and fuzzy clustering effectiveness.

INDEX TERMS Cluster analysis, fuzzy clustering, categorical data, locality-sensitive hashing, initial cluster prediction.

I. INTRODUCTION
Unsupervised learning is an important research branch in the machine learning field, especially in the context of big data. While supervised learning requires labeling the data for training learning models, unsupervised learning aims at directly exploring the correlations or hidden information from the unlabeled data [1]. In the era of big data, the data labeling task is extremely costly and time-consuming, and unsupervised learning has recently gained increasing attention [2, 3].

Clustering is one of the most important techniques in unsupervised learning [4]. Basically, the aim of clustering is to find the autocreated groups in the data so that similar objects are placed in the same cluster while dissimilar objects are placed in different clusters [5]. The cluster analysis approaches can be classified into two categories: Hierarchical clustering seeks to build the hierarchy of clusters, while flat clustering seeks to find the distinct clusters in the data [4]. In practice, flat clustering techniques are more prominent for their intelligibility with lower complexity. There are several families of algorithms that can handle the flat clustering problem, such as Expectation-Maximization (EM) [6], Genetic Algorithms (GA) [7], Self-Organizing Map (SOM) [8], Cuckoo search [9], and $k$-means-like algorithms [4, 10, 11, 12]. Moreover, $k$-means-like algorithms are a class of algorithms that use a representation to present the cluster center and seek to minimize the total distances between objects and their nearest cluster representations [4]. However, for special data types such as categorical data, the cluster representation must be defined differently. Note that the categorical data type is the kind of data that cannot be represented by numerical values such as shape or color of an object. For that reason, the center of gravity of a categorical cluster cannot be calculated directly. In that case, using medoids [13], modes [14], representatives [15, 16], or centers [17] can be useful instead of using means as for the case of numerical data. However, with different kinds of cluster representations, the distance measure between objects and representations must be defined appropriately [18].

Because the clusters of $k$-means-like algorithms are crisp, the labels of objects laying near the boundaries might be inaccurately represented. This weakness can be simply solved by fuzzy clustering approaches [19, 20]. In the context of fuzzy clustering, all objects share their membership to all clusters
in the way that the membership degrees are inversely proportional to the distances between objects and corresponding cluster representations. Several $k$-means-like algorithms can tackle the fuzzy clustering problem of categorical effectively, namely Fuzzy-$k$-modes [21], Fuzzy-$k$-representatives [16]. However, their results are still not stable because $k$-means-like algorithms are locally optimal methods [4].

It is obvious that a good initial state can lead to a better optimization for $k$-means-like algorithms [4, 22, 23]. For crisp categorical data clustering, several attempts have been made to predict the initial clusters at the initialization stage to improve the efficiency of clustering algorithms [14, 22, 23, 24]. Recently, in [24], we proposed using the Locality-Sensitive Hashing (LSH) technique to predict the initial clusters for categorical clustering, which yields better results than other initialization methods without consuming considerable computations. In this paper, we further extend the LSH-based approach for predicting the initial fuzzy clusters for fuzzy clustering of categorical data, and apply it to a newly developed fuzzy clustering algorithm called F$k$-centers [17]. Note that most previously proposed initialization methods only aimed to generate crisp clusters at the initial stage for fuzzy clustering.

Briefly, the contributions of this paper can be summarized as follows:

- We first develop a new LSH-based initialization technique to predict initial fuzzy clusters for categorical data and then incorporate it into the F$k$-centers algorithm resulting in the so-called LSHF$k$-centers algorithm.
- We design a series of comprehensive experiments for testing the clustering effectiveness of the proposed method compared with state-of-the-art fuzzy categorical data clustering methods.

The rest of the paper is organized as follows: Section II states the categorical data fuzzy clustering problem, the literature reviews, and research backgrounds. The principle of the proposed method is then shown in section III. After that, section IV shows the experimental designs and results. Finally, section V yields the conclusion and possible future works.

II. PRELIMINARIES
A. PROBLEM STATEMENT

The important notations of the fuzzy clustering problem are defined as follows. Denote $X = \{x_1, \ldots, x_N\}$ as the dataset comprising $N$ categorical objects to be fuzzily clustered into $k$ fuzzy clusters. Each categorical object $x_i$ in the dataset $X$ is a vector of $D$ categorical values $x_i = [x_{i1}, \ldots, x_{iD}]$ $(i = 1, \ldots, N)$, and the number $D$ is the number of attributes or dimensions of dataset $X$. The unique categorical values on different attributes are different and independent from each other, where $A_d$ $(d = 1, \ldots, D)$ is the domain of the $d$-th attribute, which contains all unique categorical values of the $d$-th attribute.

Note that $k$ is the target number of fuzzy clusters in the dataset $X$, and a fuzzy cluster is a group of objects each of which shares its membership degrees to all fuzzy clusters. Let $u_{ij}$ be a shared membership degree of object $x_i$ $(i = 1, \ldots, N)$ to the $j$-th cluster $(j = 1, \ldots, k)$ such that the greater the value of $u_{ij}$, the higher the linkage of object $x_i$ and the $j$-th cluster, and the membership degrees satisfy the following constraints:

$$0 \leq u_{ij} \leq 1 \quad \text{and} \quad \sum_{j=1}^{k} u_{ij} = 1 \quad (1)$$

Then, the membership matrix $U = [u_{ij}]_{N \times k}$ can entirely show the clustering statuses of all objects to all clusters.

To find the optimal values of the membership matrix $U$, different evaluation functions can be applied, such as minimizing the total distance between objects within the same clusters and maximizing the total distance between objects in different clusters. For $k$-means-like algorithms, the clusters are represented by representations, and $k$-means-like algorithms seek to minimize the total distances from objects to their nearest representations. Therefore, the set of representations is also one of the outputs of fuzzy clustering algorithms.

B. DISSIMILARITY MEASURES FOR CLUSTER ANALYSIS

Dissimilarity measures support the formation of the cluster for measuring the similarity/dissimilarity between objects in the same or different cluster(s). For metric spaces, the basic Euclidean metric can be simply applied with high efficiency. However, the situation becomes slightly more complex when dealing with categorical data because categorical values are discrete and unordered. The simplest technique to measure the dissimilarity between two categorical values is the overlap function, which counts the number of mismatch values in all attributes and uses it as the distance:

$$\text{Dis}(x_i, x_j) = \sum_{d=1}^{D} \delta(x_{id}, x_{jd}), \quad \text{and}$$
$$\delta(x_{id}, x_{jd}) = \begin{cases} 1, & \text{if } x_{id} \neq x_{jd} \\ 0, & \text{Otherwise} \end{cases} \quad (2)$$

It is clear that the overlap function is simple and easy to deploy. However, it cannot work effectively in some cases, such as subdividing a domain into multiple clusters of similar categorical values [17]. In that case, a context-based dissimilarity measure can show better performance by providing a richer value range. Distance Learning Dissimilarity for Categorical Data (DILCA) is the first context-based dissimilarity measure [25]. More specifically, DILCA defines the distance between two values $a_{di}, a_{dj}$ of categorical attribute $A_d$, denoted by $\delta(a_{di}, a_{dj})$, depending on how the values of the other attributes $A_{d'}$ within the context of $A_d$ are distributed in the data objects, where the context of $A_d$ is defined as consisting of the attributes $A_{d'}$ that are highly correlated to $A_d$. Formally,
\[
\delta(a_{di}, a_{dj}) = \sqrt{\sum_{A_{d'} \in \text{context}(A_{d})} \left( \frac{\Pr(a_{di}|a_{d'}) - \Pr(a_{dj}|a_{d'})}{\text{context}(A_{d})} \right)^2} \\
\sum_{A_{d'} \in \text{context}(A_{d})} N_{d'}
\]

where \(\Pr(a_{di}|a_{d'})\) is the conditional probability of \(a_{di}\) given \(a_{d'}\), and \(N_{d'}\) is the number of categorical values of \(A_{d'}\).

### C. FUZZY K-MEANS

Fuzzy k-means (FK-means for short) is the basic approach for conducting fuzzy cluster analysis for numeric data. Because all k-means-like algorithms for fuzzy clustering are based on FK-means, we briefly remark on the FK-means as follows.

The center of gravity (centroid) is used to represent the cluster center. Denote \(C = \{c_1, \ldots, c_k\}\) as the set of \(k\) centroids, and each centroid \(c_j\) is a vector in the same metric space as the data objects. The membership degrees of each object are calculated by the reciprocal value of the distance of that object to all cluster centroids:

\[
u_{ij} = \left( \sum_{j=1}^{k} \left( \frac{\text{Dis}(x_i, c_j)}{\text{Dis}(x_i, c_{j'})} \right) \right) ^{-1}, 1 \leq i \leq N, 1 \leq j \leq k
\]

where \(\alpha\) is the parameter controlling the degree of fuzziness such that \(\alpha \in [1, +\infty)\). With \(\alpha = 1\), the fuzzy cluster analysis model is considered back as crisp cluster analysis.

FK-means uses the centers of gravity to recalculate the centroids of all clusters when the degrees of membership matrix are updated:

\[
c_j = \frac{\sum_{i=1}^{N} u_{ij}^{\alpha} x_i}{\sum_{i=1}^{N} u_{ij}^{\alpha}}, j = 1, \ldots, k
\]

Note that the fuzziness parameter \(\alpha\) is also used to update the centroids, which determines the different magnitudes of the objects near and far from the cluster centroids.

To evaluate the convergence of the FK-means algorithm, the total fuzzy distance from all objects to all cluster centroids is used:

Minimize: \(P(U, C) = \sum_{j=1}^{k} \sum_{i=1}^{N} u_{ij}^{\alpha} \text{Dis}(x_i, c_j)\)

Technically, a lower value of \(P(U, C)\) means a better clustering outcome.

The above three equations are the foundation principles of FK-means, and the algorithm of FK-means can be briefly described in the following four steps:

- **Step 1:** Randomly select \(k\) objects and let them centroids for \(k\) clusters.
- **Step 2:** Calculate/recalculate the membership matrix \(U\) following equation (4).
- **Step 3:** Recalculate the centroids \(C\) following equation (5).
- **Step 4:** Repeat Step 2 and Step 3 until value \(P(U, C)\) (see equation (6)) converges or the number of maximum iterations is reached.

### D. K-MODES AND FUZZY K-MODES

k-modes is the first attempt to handle crisp clustering of categorical data, and fuzzy k-modes (FK-modes) is an extension for fuzzy clustering. k-modes and FK-modes use the categorical object with the highest frequencies on all attributes as the cluster’s mode. Let \(m_j = [m_{j1}, \ldots, m_{jd}]\) be the mode of the \(j\)-th cluster; the formulation of \(m_j\) is:

\[
m_{jd} = \arg \max \sum_{v=1}^{N} u_{ij}^{\alpha}, \forall x_i \in A_d
\]

Therefore, the modes of k-modes and FK-modes are categorical vectors, which implies that the membership degrees formula in equation (4) can be used.

### E. K-REPRESENTATIVES AND FUZZY K-REPRESENTATIVES

When using modes, there is a high probability that the information of values with high frequencies will be lost, especially with domains with high numbers of unique categorical values [15, 16]. San et al. [15] proposed \(k\)-representatives for dealing with such problems by capturing all the probabilities of all unique categorical values into the so-called representatives. Let \(r_j = [r_{j1}, \ldots, r_{jD}]\) be the representative of the \(j\)-th cluster. Then a representative is defined as:

\[
r_{jd} = \{v, \text{Pro}_j(v)\}) |\forall v \in A_d, 1 \leq d \leq D, 1 \leq j \leq k
\]

where \(\text{Pro}_j(v)\) is the probability of categorical value \(v\) appearing in the \(j\)-th cluster.

It is clear that representatives are not categorical vectors, so equation (2) cannot be used for calculating the distance from an object to a representative. San et al. [15] also introduced a simple function for this task:

\[
\text{Dis}(x_i, r_j) = \sum_{d=1}^{D} \sum_{v \in A_d} \text{Pro}_j(v) = D - \sum_{d=1}^{D} \text{Pro}(x_{id})
\]

In terms of extension of \(k\)-representatives for fuzzy clustering of categorical data, Kim et al. [16] proposed using fuzzy centroids, which are expansions of representatives as follows:

\[
r_{jd} = \{v, \text{FPro}_{jd}(v)\}) |\forall v \in A_d, 1 \leq d \leq D, 1 \leq j \leq k
\]

where \(\text{FPro}_{jd}(v)\) is the probability of categorical value \(v\) in the \(j\)-th cluster on the \(d\)-th attribute in terms of fuzzy membership:

\[
\text{FPro}_{j,d}(v) = \frac{\sum_{x_i \in X, x_{id} = v} u_{ij}^{\alpha}}{\sum_{i=1}^{N} u_{ij}^{\alpha}}
\]
F. \( \kappa \)-CENTERS

Continuing the work of \( k \)-representatives, Chen et al. [26] introduced using a kernel combination of uniform distribution and observed distribution to better represent cluster. In detail, the concept of Pro\(_j\)(\( v \)) in equation (8) is replaced by the kernel probability estimate KPro\(_j\)(\( v \)), which is defined as:

\[
KPro_j(v) = \lambda_j \frac{1}{|A_j|} + (1 - \lambda_j)Pro_j(v), \forall v \in G_j
\]

where \( \lambda_j \) is the smoothing parameter for adjusting the uniform distribution contribution to the \( j \)-th center. With Least Squares Cross-Validation (LSCV) [27], the optimal value of \( \lambda_j \) can be learned statistically based on the distribution of categorical values on that attribute:

\[
\lambda_j = \frac{1}{|G_j| - 1} \sum_{d=1}^{D_j} (1 - \sum_{v \in A_d} Frequency_j(v)^2) \sum_{d=1}^{D_j} (\sum_{v \in A_d} Frequency_j(v)^2 - \frac{1}{|A_d|})^{-1}
\]

where Frequency\(_j\)(\( v \)) is the frequency of appearance of categorical value \( v \) in the \( j \)-th attribute.

G. PREVIOUS WORK: FUZZY \( \kappa \)-CENTERS

Recently, Nguyen et al. [17] proposed a fuzzy clustering algorithm so-called Fk-centers that extends the center to fcenter to make it workable with fuzzy clustering. Particularly, the smoothing parameter equation (13) is modified to match the concept of fuzzy clusters:

\[
\lambda_j = \frac{1}{\sum_{i=1}^{N} u_{ij}} \sum_{d=1}^{D_j} \frac{(1 - \sum_{v \in A_d} Frequency_j,d(v)^2)}{(\sum_{v \in A_d} Frequency_j,d(v)^2 - \frac{1}{|A_d|})^{-1}}
\]

where FFrequency\(_j,d\)(\( v \)) is the frequency of the categorical value \( v \) in the \( j \)-th fuzzy cluster:

\[
FFrequency_j,d(v) = \sum_{x_i \in X, x_i,d = v} u_{ij}^d, 1 \leq d \leq D, 1 \leq j \leq k
\]

Because the structures of fcenters are the same as centers, the dissimilarity measures can be reused. As such, in this paper fcenter is selected for representing the fuzzy clusters. Moreover, we further focus on enhancing the performance of Fk-centers algorithm by optimizing its initialization process.

H. LOCALITY-SENSITIVE HASHING (LSH)

LSH is a well-known dimension reduction method that is widely used for approximate nearest neighbor search in high-dimensional big data [28, 29, 30]. Technically, LSH uses a family of locality-sensitive hash functions to project high-dimensional data into lower-dimensional data, which seeks the consistency of similarity across the new space.

For the principle of LSH, a family of \( l \) hash functions \( H = [h_1, \ldots, h_l] \) is called \((R, cR, p_1, p_2)\)-sensitive if for any objects \( x_i \) and \( x_j \):

- If \( \text{Dis}(x_i, x_j) \leq R \) then \( \Pr[H(x_i) = H(x_j)] \geq p_1 \)
- If \( \text{Dis}(x_i, x_j) \geq cR \) then \( \Pr[H(x_i) = H(x_j)] \leq p_2 \)

Several kinds of hash functions can have good \((R, cR, p_1, p_2)\)-sensitivity such as random projection functions, random hyperplane functions, or threshold functions.

For fast traversing and hash value storage, we need to build a hash table to store the hash values of all objects in the data. In detail, each locality-sensitive hash function gives a binary value for an input object vector. Therefore, \( l \) hash functions can give a maximum of \( 2^l \) different hash values in total. When the number of objects \( N \) is greater than \( 2^l \) (with \( N \gg 2^l \)), numerous objects have the same hash value. The objects having the same hash value are stored in the same bucket in the hash table. Therefore, the key to that bucket is the hash value of all objects in it. Interestingly, the keys can be used to determine the similarity of objects in different buckets by taking the Hamming distances.

Certainly, building the hash table is fast because the hash functions can be conducted independently. In this research, we take advantage of the similarity of buckets of LSH to predict the initial fuzzy clusters for the fuzzy clustering problem.

III. THE PROPOSED METHOD

The outline of the proposed method is shown in Figure 1, which includes multiple processes grouped into two main stages: clustering initialization and clustering iteration. In this section, these two stages are detailed following the flows in Figure 1.

A. CLUSTERING INITIALIZATION

1) Dissimilarity matrices creating

The first process is building the dissimilarity matrix for each categorical attribute. Particularly, a dissimilarity matrix of a domain is formed by the pairwise distances of all unique categorical values in that domain:

| TABLE 1. | Semantic dissimilarity matrix for attribute \( A_d \) with \( N_d \) categorical values |
|--------------------------|-------------------------------------------------|
| \( a_{d1} \) | \( a_{d2} \) | \( a_{d3} \) | \( \ldots \) | \( a_{dN_d} \) |
| \( a_{d1} \) | 0 | \( \delta(a_{d1}, a_{d2}) \) | \( \delta(a_{d1}, a_{d3}) \) | \( \ldots \) | \( \delta(a_{d1}, a_{dN_d}) \) |
| \( a_{d2} \) | \( \delta(a_{d2}, a_{d1}) \) | 0 | \( \delta(a_{d2}, a_{d3}) \) | \( \ldots \) | \( \delta(a_{d2}, a_{dN_d}) \) |
| \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) |
| \( a_{dN_d} \) | \( \delta(a_{dN_d}, a_{d1}) \) | \( \delta(a_{dN_d}, a_{d2}) \) | \( \ldots \) | 0 |

Table 1 indicates an example of the semantic dissimilarity matrix for the \( d \)-th categorical attribute with \( N_d \) different categorical values. Because the distance function is semantic and the distance of a categorical value to itself is zero, we just need to calculate \( N_d(N_d-1)/2 \) values in total.

In this study, we recommend using the context-sensitive dissimilarity measures such as Association based similarity measure [31], distance learning for categorical attribute based on context information [32], and Distance Learning Dissimilarity for Categorical Data (DILCA) [25]. Namely, the DILCA is used in our implementation.
These matrices not only support predicting the clusters but also make the dictionaries for fast access during the clustering process.

2) LSH hash functions creating

We are inspired by the threshold hash function for selecting thresholds in selected dimensions to subdivide each dimension into two domains with high inner locality sensitivity. In this research, we propose to subdivide the categorical attributes into multiple two subdomains.

First, the dissimilarity matrices in the previous step are converted into full undirected graphs so that we can subdivide them easier. An example of an undirected graph for the $d$th attribute is shown in Figure 2.

![Example of undirected graph for the $d$th attribute with four categorical values](image)

Second, the Stoer-Wagner algorithm [33] is utilized to find the maximum cuts on the graphs, which aims to separate the categorical values in a domain into two locality-sensitive groups.

Third, let $A_{d0}$ and $A_{d1}$ be two subsets of $A_d$, which are created from the maximum cut in the previous step ($A_{d0} \cup A_{d1} = A_d$ and $A_{d0} \cap A_{d1} = \emptyset, 1 \leq d \leq D$). Thus, the hash function $h_d$ for the $d$th attribute can be formed as:

$$h_d(x_i) = \begin{cases} 0, & \text{if } x_{id} \in A_{d0} \\ 1, & \text{if } x_{id} \in A_{d1} \end{cases}, \quad 1 \leq d \leq D$$ (16)

Fourth, because the attributes affect the clustering performance differently [26, 34], we propose selecting the $l$ hash functions that have the highest inner-separation scores to be used for building the hash table. Because the values in the dissimilarity matrices are normalized by the DILCA principle, a conventional method for selecting the well-cut maximum cuts is to evaluate the average weights of edges in the cuts.

Denote $MC_d$ as the set of edges in the maximum cut of the $d$th attribute. The average weight can be calculated as:

$$\hat{\gamma}_d = \frac{\sum_{\{a_{dj}, a_{dj'}\} \in MC_d} \text{Weight}(\{a_{dj}, a_{dj'}\})}{|MC_d|}$$ (17)

where $\text{Weight}(\{a_{dj}, a_{dj'}\}) = \delta(a_{dj}, a_{dj'})$ is the weight of edge $\{a_{dj}, a_{dj'}\}$ in the graph of the $d$th dissimilarity matrix. Then, the $l$ hash functions with the highest average maximum cut weights are selected for building the hash table:

$$H = [h_1, \ldots, h_l]$$ where $h_j \in \{h_d|1 \leq d \leq D\}, \quad (1 \leq j \leq l)$ (18)

subject to:

$$\forall 1 \leq i, j \leq l \text{ if } j > i \text{ then } \gamma_i \geq \gamma_j$$ (19)

where $\gamma_i$ is the corresponding average maximum weight value of hash function $h_i$. 

![Example of undirected graph for the $d$th attribute with four categorical values](image)
3) Hash table generating

After the hash functions are generated, we can simply create the hash table of multiple buckets so that the objects with the same hash value are put into the same bucket. The singleton hash value of objects in a bucket is also used as the key value of that bucket. Because there is a hash function, there are a maximum of $2^l$ different hash values that the objects can have. In our proposed method, the buckets are used to predict the initial clusters. Therefore, the number of buckets $2^l$ should be greater than the number of clusters $k$.

![Diagram showing the hash table generation](image)

Figure 3 represents an example of hashing three different categorical objects with two attributes. In detail, $A_1 = \{a_{11}, a_{12}, a_{13}\}$, $A_2 = \{a_{21}, a_{22}, a_{23}, a_{24}\}$, suppose that $A_{10} = \{a_{11}, a_{13}\}$, $A_{12} = \{a_{12}\}$, $A_{20} = \{a_{22}, a_{23}\}$, and $A_{21} = \{a_{21}, a_{24}\}$. Based on equation (16), we can obtain two hash functions: $h_1(a) = 1$ if $a \in A_{10}$ else $h_1(a) = 0$ and $h_2(a) = 1$ if $a \in A_{20}$ else $h_2(a) = 0$. After applying the family of hash functions $H = [h_1, h_2]$ to the three objects $x_1 = [a_{11}, a_{21}]$, $x_2 = [a_{12}, a_{22}]$, $x_3 = [a_{12}, a_{23}]$, we can obtain their hash values as [01], [10], and [10], respectively.

Then, we can create the hash table with four buckets as $H = [[00] : \emptyset], [[01] : \{x_1\}], [[10] : \{x_2, x_3\}], [[11] : \emptyset]$. To archive fast traversing the hash table, we use an inverted file structure that stores the objects indices of a bucket in the same segment in ascending order, which supports fast merging of a bucket to another bucket [35, 36].

4) Fuzzy initial clusters prediction

It is clear that the largest buckets are potentially located near the natural clusters in the dataset because they have dense objects with the same locality-sensitive scores. For that reason, we propose using the $k$ largest buckets as the core buckets for $k$ initial clusters. Denote $B_1^k, \ldots, B_k^k$ as the $k$ largest buckets in hash table $H$; the objects belonging to these buckets fully belong to the corresponding clusters:

$$\forall x_i \in \{B_1^k, \ldots, B_k^k\}, u_{ij} = \begin{cases} 1, & \text{if } x_i \in B_j^* \\ 0, & \text{Otherwise} \end{cases}, 1 \leq j \leq k$$

(20)

The objects in smaller and remaining buckets share their membership degrees with $k$ initial clusters based on the distances of the buckets they belong to the core buckets. At this time, cluster representations have not yet been formed, but we can approximate the distance from object to core buckets by using Hamming distances between corresponding bucket keys:

$$\text{Dis}(x_i, r_j) \approx \hat{\text{Dis}}(x_i, B_j^*) = \text{Hamming}(\text{Key}(B_{i'}), \text{Key}(B_j^*))$$

(21)

where Key($B_{i'}$) is the bucket that holds categorical object $x_i$.

Next, if we consider the core buckets to be representations, the equation (4) can be applied to calculate the degrees of membership of objects outside core buckets.

5) LSH-based initial cluster prediction algorithm

**Algorithm 1** LSH-based initial cluster prediction algorithm

**Require:** $X, k, D, A_d(1 \leq d \leq D), l$

**Ensure:** $k$ initial clusters

1. Create dissimilarity matrix for each categorical attribute $A_d(1 \leq d \leq D)$ using the DILCA dissimilarity measure.
2. Convert dissimilarity matrices into dissimilarity graphs.
3. Find the maximum cut for each dissimilarity graph using Stoer-Wagner algorithm.
4. Calculate the average weight of edges in each cut using equation (17) and select $l$ attribute with the greatest values.
5. Generate the hash function via equation (18). The hash values of all objects in $X$ then can be obtained. Correspondingly, the hash table $H = [B_1, \ldots, B_2]$ also can be established.
6. Select $k$ largest buckets $\{B_1^k, \ldots, B_k^k\}$.
7. for $x_i \in X$ do
8. if $x_i \in \{B_1^k, \ldots, B_k^k\}$ then
9. Assign degrees of membership $u_{ij}(1 \leq j \leq k)$ by equation (20).
10. else
11. Estimate distance from object $x_i$ to the all core buckets by equation (21).
12. Assign degrees of membership $u_{ij}(1 \leq j \leq k)$ by equation (4).
13. end if
14. end for
15. return $U = [u_{ij}]_{N \times k}$

We summarize the processes of the proposed initial cluster prediction method in Algorithm 1. However, the initial clusters are represented by the membership matrix $U$ instead of the set of cluster representations $C$.
Note that it is necessary to have a hash table with the number of nonempty buckets greater than the number cluster $k$ for our method to work effectively. However, because the bucket sizes are unpredictable, the selection of a suitable number of hash functions $l$ is tricky. Ideally, our method can work effectively when the number of non-empty buckets is the same as the number of clusters $k$, but in practice, we suggest selecting the number of hash functions $l$ so that the number of maximum buckets is twice the number of clusters $k$, thereby $l \approx \log_2(k) + 1$. When the number of nonempty buckets is smaller than $k$, we can increase the number of hash functions $l$ to 1 or just proceed with empty core bucket(s).

### B. CLUSTERING ALGORITHM

In this section, we summarize our proposed method (LSH$k$-centers) in Algorithm 2, with $\text{iter}_{\text{max}}$ and $\epsilon$ being the parameters controlling the convergence of the algorithm.

#### Algorithm 2 LSH-based fuzzy $k$-centers clustering algorithm (LSH$k$-centers)

**Require:** $X$, $k$, $D$, $A_d (1 \leq d \leq D)$, $l$, $\text{iter}_{\text{max}}$, $\epsilon$

**Ensure:** $k$ centers of $k$ clusters $R$ and membership matrix $U$ that can locality minimize the value $P(U, C)$ (equation (6)).

1. Find the initial membership matrix $U$ by Algorithm 1.
2. Set $\lambda_j \leftarrow 0$ for $1 \leq j \leq k$.
3. for $\text{iter} \leftarrow 1$ : $\text{iter} \leq \text{iter}_{\text{max}}$; $\text{iter}++$ do
4. Update the centers $R$ following equation (8) with the fuzzy probability in equation (12).
5. Update the membership matrix $U$ following equation (4).
6. Update the smoothing parameter $\lambda_j$ for $1 \leq j \leq k$ following equation (13).
7. if $\Delta P(U, C) \leq \epsilon$ then
8. Break the loop.
9. end if
10. end for
11. return $C, U$

### IV. EXPERIMENTS AND RESULTS

In this section, we describe the details and results of designed experiments that aim to show the clustering performance of our proposed method compared to its original method and other state-of-the-art approaches.

#### A. EXPERIMENTAL SETUP

1) Competitors

First, these crisp clustering algorithms for categorical data are included: $k$-medoids [13], $k$-modes [14], $k$-representatives ($k$-reps for short) [15], $k$-centers [16], and LSH$k$-reps are also included [17].

Second, encoding-based fuzzy clustering algorithms such as F$k$-means, FE$k$ [37], and Fuzzy Space-Based Clustering (FSBC) [38] are also selected as our competitors.

Third, recent state-of-the-art works for fuzzy clustering of categorical data are also added. For example, single-objective genetic algorithms such as SGA-Dist, SGA-Sep, and SGA-SepDist [39]; multiobjective genetic algorithm methods such as Multi-Objective Genetic Algorithm (MOGA) [40], NSGA-FMC (NSGA for short) [41], and Many Objectives Fuzzy Centroids (MaOFCentroids) [42].

2) Testing computer and datasets

Our method and all related works are developed by Python programming language, and the experiments are carried out by a high-end computer cluster with Intel Xeon G-6240M 2.6GHz (16 cores x 4) CPUs. Therefore, each algorithm can run 64 times on 64 cores in parallel in a node.

Next, we select 16 common UCI categorical datasets that are historically widely used in categorical data clustering, classification, and mixed data [43]. The details of these categorical datasets are shown in Table 2.

#### TABLE 2. Common categorical datasets.

| Name               | #Items | #Attributes | #Classes |
|--------------------|--------|-------------|----------|
| Soybean-small      | 47     | 21          | 4        |
| Audiology          | 226    | 21          | 4        |
| Balance-scale      | 625    | 4           | 3        |
| Zoo                | 101    | 17          | 7        |
| Hayes-roth         | 132    | 5           | 3        |
| Dermatology        | 366    | 34          | 6        |
| Soybean            | 683    | 35          | 19       |
| Connect            | 10,000 | 42          | 3        |
| Chess              | 3,196  | 36          | 2        |
| Breast             | 699    | 9           | 2        |
| Car                | 1,728  | 6           | 4        |
| Mushroom           | 8,124  | 22          | 2        |
| Tic-tactoe         | 958    | 9           | 2        |
| Vote               | 435    | 16          | 2        |
| Flare              | 1,066  | 12          | 3        |
| Lung               | 32     | 56          | 3        |

3) Evaluation metrics

First, to justify the ability of our method to crisp clustering, we select to use the purity score, which is the most commonly used clustering evaluation metric [44]. Denote $\mathcal{G} = \{G_1, \ldots, G_k\}$ as the set of $k$ output clusters of a clustering algorithm and $\mathcal{G}^* = \{G_1, \ldots, G^*_k\}$ as the set of ground-truth clusters of the corresponding dataset. The purity metric seeks the best match of these two clusters and then count the number of correct assignments:

$$\text{Purity}(\mathcal{G}, \mathcal{G}^*) = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |G_i \cap \hat{G}_j| \quad (22)$$

Note that we can simply convert the fuzzy membership matrix to a crisp membership matrix as:

$$u_{ij} = \begin{cases} 1, & \text{if } j = \arg \max_{1 \leq j' \leq k} u_{ij'} \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

Second, for the fuzzy clustering effectiveness evaluation, the fuzzy silhouette (FSilhouette) scoring is utilized [45], and FSilhouette seeks for the average normalization of the
silhouette scores product with the amplitude of degrees of membership of all objects:

$$FSilhouette(U, X) = \frac{\sum_{i=1}^{N}(\mu_{p,i} - \mu_{q,i})^2 s_i}{\sum_{i=1}^{N}(\mu_{p,i} - \mu_{q,i})^2}$$

(24)

where $\mu_{p,i}$ and $\mu_{q,i}$ are the first and second largest values of $u_{ij}$ for $(1 \leq j \leq k)$, respectively, and $s_i$ is the crisp silhouette score of $x_i$ [45].

Finally, to estimate the complexities of all compared methods, the average total clustering time (including initialization time) of each method is measured.

### B. CLUSTER PREDICTION ANALYSIS

![Accuracy of LSH-based fuzzy cluster prediction versus ground-truth labels on soybean-small dataset](image)

**FIGURE 4.** Accuracy of LSH-based fuzzy cluster prediction versus ground-truth labels on soybean-small dataset

First, the prediction accuracy of the proposed method is visualized when applied to the UCI soybean-small dataset. This dataset is the most commonly used categorical dataset for demonstration because of its simplicity. In detail, Figure 4.a shows the ground-truth labels\(^1\) as $G_1 = \{x_{21}, \ldots, x_{30}\}$, $G_2 = \{x_{31}, \ldots, x_{47}\}$, $G_3 = \{x_{1}, \ldots, x_{10}\}$, and $G_4 = \{x_{11}, \ldots, x_{20}\}$. In this experiment, our LSH-based prediction model uses $l = 4$ hash functions, and with that number of hash functions, we can achieve a maximum of $2^l = 16$ different hash values. However, after conducting hash table construction, only 5 buckets have objects, namely, $B_{12} = \{x_{1}, \ldots, x_{10}\}$, $B_{11} = \{x_{11}, \ldots, x_{20}\}$, $B_6 = \{x_{21}, x_{23}, x_{26}, x_{27}, x_{29}\}$, $B_{14} = \{x_{22}, x_{24}, x_{25}, x_{28}, x_{30}, x_{31}, x_{34}, x_{41}, x_{42}, x_{44}, x_{46}\}$, and $B_{15} = \{x_{32}, x_{33}, x_{35}, \ldots, x_{40}, x_{43}, x_{45}, x_{47}\}$. Therefore, $B_{14}, B_{15}, B_{12},$ and $B_{11}$ are selected as core buckets, while objects in $B_6$ must share their membership degrees to the core buckets. As a result, in Figure 4.b, because the Hamming distances between buckets $B_6$ and $B_{14}$ are 1 and are the smallest, the membership degrees of objects in $B_6$ to the core bucket (cluster) $B_{14}$ are highest. Intuitively, our prediction model gives relatively accurate results compared to the ground-truth labels, which can predict absolutely accurate results for the cluster 3 and cluster 4, with an accuracy of 87.2% (41/47 objects).

### C. CRISP CLUSTERING PERFORMANCE

![Purity scores on UCI datasets](image)

**FIGURE 5.** Purity scores on UCI datasets

Second, because the fuzzy clustering algorithms can solve the crisp clustering problem by assigning the objects to their nearest clusters, we can compare the crisp clustering performance of our proposed method with that of other competitors. Figure 5 shows such comparison results as the average purity scores and standard deviations of all compared methods on 16 testing datasets with $\alpha = 1.1$. In this experiment, all the methods are tested with singleton initialization 64 times with different random seed numbers. As a result, our method and FSBC are the methods with the best average purity score of 0.582. However, because our method gives consistent results over different runs, LSHF$k$-centers become the most stable method with a standard deviation of 0. F$k$-means can also achieve a standard deviation of 0 but its average purity score is less than that of LSHF$k$-center.
Next, we conduct a comprehensive evaluation of the fuzzy cluster of compared methods for 16 UCI testing datasets. Addition, similar to crisp clustering methods, the proposed method can handle datasets with high balance such as Car and Balance-scale while the proposed method can handle them accurately. Compared to the original method (Fk-centers), the average Silhouette score of the proposed method is higher than Fk-centers by 8% (0.349 vs. 0.323).

D. FUZZY CLUSTERING PERFORMANCE ANALYSIS

Finally, the complexity comparison is shown in Figure 6. Note that those clustering time metrics are extracted from the
same experiment that also provides the Fsilhouette scores in Table 3; hence, we can clearly show the trade-off between speed and accuracy. It is clear that genetic algorithms take much more clustering time than the $k$-means-like algorithms because they have up to $4 \times k$ chromosomes. Moreover, the membership chromosome-based methods such as NSGA, MOFCentroids, and MaOCentroids are even more time-consuming for taking extra processes to convert the membership chromosomes into cluster representations for evaluation. Additionally, FSBC has comparable crisp clustering accuracy with our proposed method, but it takes 3.59 times more computation time than LSHF$k$-centers (115.43 (seconds)/32.59 (seconds)). However, because of the tradeoff between accuracy and complexity, LSHF-$k$-centers is nearly 2 times slower than its original method $k$-centers, and the average clustering times are 32.59 (seconds) and 16.83 (seconds), respectively. In the same manner, simple methods such as F$k$-means, FE-$k$-means, $k$-modes, $k$-reps, and F$k$-modes can run fast, but their clustering accuracy scores are relatively low.

V. CONCLUSION

In this paper, we introduced LSHF$k$-centers as the first novel approach to predict the fuzzy categorical clusters of categorical data. Particularly, in the initialization stage, we create hash functions based on the separations of categorical values in each attribute. After that, the LSH hash table can be established from multiple buckets of similar objects by using selected hash functions. LSHF$k$-centers selects the $k$ largest buckets as the core buckets of fuzzy clusters, and the objects of the remaining buckets must share their membership degrees with the core buckets. As a piece of evidence, our fuzzy clustering prediction model itself can predict up to 87.2% accuracy compared to the ground-truth labels on the soybean-small dataset. Moreover, the proposed method is 8% more accurate than its original method (F$k$-centers) in terms of fuzzy clustering effectiveness.

Combined with the advantages of centers, our clustering algorithm outperforms all other related works in terms of clustering effectiveness in both crisp and fuzzy clustering evaluations. However, due to the extra process for building the LSH hash table, our method takes a considerable amount of time compared to its original method $k$-centers.

We recommend that future research focus on two following approaches to increase the performance of the LSHF$k$-centers algorithm:

- Investigate the usefulness of different measures to the construction of the hash table. Perhaps a different dissimilar measure can give better accuracy than DILCA while taking fewer computations.
- In this study, the LSH hash function is based on properties of a single attribute only. Future work can use multiple-attribute hash functions to achieve higher locality-sensitive factors.

For the sake of reproducing and creating open sources for future research, we published the source code of LSHF$k$-centers into the PyPI repository as https://pypi.org/project/lsfhkcenters/.

REFERENCES

[1] M. W. Berry, A. Mohamed, and B. W. Yap, Supervised and unsupervised learning for data science. Springer, 2019.
[2] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, “Information security in big data: privacy and data mining,” IEEE Access, vol. 2, pp. 1149–1176, 2014.
[3] Y. Roh, G. Heo, and S. E. Whang, “A survey on data collection for machine learning: a big data-ai integration perspective,” IEEE Transactions on Knowledge and Data Engineering, 2019.
[4] R. Xu and D. Wunsch, “Survey of Clustering Algorithms,” IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645–678, 2005.
[5] J. Huang, Y. Liang, H. Bian, and X. Wang, “Using cluster analysis and least square support vector machine to predicting power demand for the next-day,” IEEE Access, vol. 7, pp. 82 681–82 692, 2019.
[6] T. K. Moon, “The expectation-maximization algorithm,” IEEE Signal processing magazine, vol. 13, no. 6, pp. 47–60, 1996.
[7] R. Krovi, “Genetic algorithms for clustering: a preliminary investigation,” in Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, vol. 4. IEEE, 1992, pp. 540–544.
[8] J. Vesanto and E. Alhoniemi, “Clustering of the self-organizing map,” IEEE Transactions on neural networks, vol. 11, no. 3, pp. 586–600, 2000.
[9] J. Ji, W. Pang, Z. Li, F. He, G. Feng, and X. Zhao, “Clustering mixed numeric and categorical data with cuckoo search,” IEEE Access, vol. 8, pp. 30 988–31 003, 2020.
[10] M. A. Mahdi, K. M. Hosny, and I. Elhenawy, “Scalable clustering algorithms for big data: A review,” IEEE Access, 2021.
[11] K. Chowdhury, D. Chaudhuri, and A. K. Pal, “An entropy-based initialization method of $k$-means clustering on the optimal number of clusters,” Neural Computing and Applications, vol. 33, no. 12, pp. 6965–6982, 2021.
[12] P. Kumar and A. Kanavalli, “A similarity based $k$-means clustering technique for categorical data in data mining application,” International Journal of Intelligent Engineering and Systems, vol. 14, no. 2, pp. 43–51, 2021.
[13] C. B. Lucasius, A. D. Dane, and G. Kateman, “On $k$-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison,” Analytica Chimica Acta, vol. 282, no. 3, pp. 647–669, 1993.
[14] Z. Huang, “Extensions to the $k$-means Algorithm for Clustering Large Data Sets with Categorical Values,” Data Mining and Knowledge Discovery, vol. 2, no. 3, pp. 283–304, 1998.
[15] O. M. San, V.-N. Huynh, and Y. Nakamori, “An Alternative Extension of the k-means Algorithm for Clustering Categorical Data,” International Journal of Applied Mathematics and Computer Science, vol. 14, pp. 241–247, 2004.

[16] D.-W. Kim, K. H. Lee, and D. Lee, “Fuzzy Clustering of Categorical Data Using Fuzzy Centroids,” Pattern recognition letters, vol. 25, no. 11, pp. 1263–1271, 2004.

[17] T. N. Mau and V.-N. Huynh, “Kernel-Based k-Representatives Algorithm for Fuzzy Clustering of Categorical Data,” in 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2021, pp. 1–6.

[18] A. Ahmad and S. S. Khan, “Survey of state-of-the-art mixed data clustering algorithms,” Ieee Access, vol. 7, pp. 31 883–31 902, 2019.

[19] M.-S. Yang, “A survey of fuzzy clustering,” Mathematical and Computer modelling, vol. 18, no. 11, pp. 1–16, 1993.

[20] W. Li, K. Zhang, Y. Chen, C. Tang, X. Ma, and Y. Luo, “Random fuzzy clustering granular hyperplane classifier,” IEEE Access, vol. 8, pp. 228 767–228 778, 2020.

[21] Z. Huang and M. K. Ng, “A Fuzzy k-modes Algorithm for Clustering Categorical Data,” IEEE transactions on Fuzzy Systems, vol. 7, no. 4, pp. 446–452, 1999.

[22] F. Cao, J. Liang, D. Li, L. Bai, and C. Dang, “A Dissimilarity Measure for the k-modes Clustering Algorithm,” Knowledge-Based Systems, vol. 26, pp. 120–127, 2012.

[23] D.-T. Dinh and V.-N. Huynh, “k-pbC: an Improved Cluster Center Initialization for Categorical Data Clustering,” Applied Intelligence, pp. 1–23, 2020.

[24] T. N. Mau and V.-N. Huynh, “An LSH-Based k-Representatives Clustering Method for Large Categorical Data,” Neurocomputing, vol. 463, pp. 29–44, 2021.

[25] D. Ienco, R. G. Pensa, and R. Meo, “From Context to Distance: Learning Dissimilarity for Categorical Data Clustering,” ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 6, no. 1, pp. 1–25, 2012.

[26] L. Chen and S. Wang, “Central Clustering of Categorical Data with Automated Feature Weighting,” in IJCAI, 2013, pp. 1260–1266.

[27] Q. Li and J. S. Racine, Nonparametric Econometrics: Theory and Practice. Princeton University Press, 2007.

[28] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive Hashing Scheme Based on p-stable Distributions,” in Proceedings of the twentieth Annual Symposium on Computational Geometry, 2004, pp. 253–262.

[29] Y. Cao, H. Qi, W. Zhou, J. Kato, K. Li, X. Liu, and J. Gui, “Binary hashing for approximate nearest neighbor search on big data: A survey,” IEEE Access, vol. 6, pp. 2039–2054, 2017.

[30] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” in 2006 47th annual IEEE symposium on foundations of computer science (FOCS’06). IEEE, 2006, pp. 459–468.

[31] S. Q. Le and T. B. Ho, “An association-based dissimilarity measure for categorical data,” Pattern Recognition Letters, vol. 26, no. 16, pp. 2549–2557, 2005.

[32] Z. Khorshidpour, S. Hashemi, and A. Hamzeh, “Distance learning for categorical attribute based on context information,” in 2010 2nd International Conference on Software Technology and Engineering, vol. 2. IEEE, 2010, pp. V2–296.

[33] M. Stoor and F. Wagner, “A Simple Min-cut Algorithm,” Journal of the ACM (JACM), vol. 44, no. 4, pp. 585–591, 1997.

[34] T. N. Mau and V.-N. Huynh, “Automated Attribute Weighting Fuzzy k-Centers Algorithm for Categorical Data Clustering,” in International Conference on Modeling Decisions for Artificial Intelligence. Springer, 2021, pp. 205–217.

[35] F. Yang, Y. Sato, Y. Tan, and Y. Inoguchi, “Searching Acceleration for Audio Fingerprinting System,” in Joint conference of Hokuriku chapters of electrical societies, 2012.

[36] T. Nguyen Mau and Y. Inoguchi, “Locality-Sensitive Hashing for Information Retrieval System on Multiple GPGPU Devices,” Applied Sciences, vol. 10, no. 7, p. 2539, 2020.

[37] K. Potdar, T. S. Pardawala, and C. D. Pai, “A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers,” International Journal of Computer applications, vol. 175, no. 4, pp. 7–9, 2017.

[38] Y. Qian, F. Li, J. Liang, B. Liu, and C. Dang, “Space Structure and Clustering of Categorical Data,” IEEE transactions on neural networks and learning systems, vol. 27, no. 10, pp. 2047–2059, 2015.

[39] G. Gan, J. Wu, and Z. Yang, “A Genetic Fuzzy k-modes Algorithm for Clustering Categorical Data,” Expert Systems with Applications, vol. 36, no. 2, pp. 1615–1620, 2009.

[40] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, “Multiobjective Genetic Algorithm-based Fuzzy Clustering of Categorical Attributes,” IEEE transactions on evolutionary computation, vol. 13, no. 5, pp. 991–1005, 2009.

[41] C.-L. Yang, R. Kuo, C.-H. Chien, and N. T. P. Quyen, “Non-dominated Sorting Genetic Algorithm using Fuzzy Membership Chromosome for Categorical Data Clustering,” Applied Soft Computing, vol. 30, pp. 113–122, 2015.

[42] S. Zhu and L. Xu, “Many-objective fuzzy centroids clustering algorithm for categorical data,” Expert Systems with Applications, vol. 96, pp. 230–248, 2018.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
TOAN NGUYEN MAU received the bachelor’s degree (Hons.) in information technology from the Ho Chi Minh City University of Science, Vietnam National University, Ho Chi Minh City, Vietnam, in 2013, and the master’s and Ph.D. degrees in information science from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2016 and 2021, respectively. His research interests include data mining, machine learning, and parallel processing using GPGPU.

YASUSHI INOGUCHI received his B.E. degree from Department of Mechanical Engineering, Tohoku University in 1991, and received MS degree and Ph.D from JAIST (Japan Advanced Institute of Science and Technology) in 1994 and 1997, respectively. He is a Professor of the Research Center for Advanced Computing Infrastructure, JAIST. He was a research fellow of the Japan Society for the Promotion of Science from 1994 to 1997. From 2002 to 2006 he was a researcher of the PRESTO program of Japan Science and Technology Agency. From 2008 to 2009 he was a courtesy senior research scholar at the University of South Florida. His research interest has been mainly concerned with parallel computer architecture, interconnection networks, Cloud architecture, and high performance computing on parallel machines. Dr. Inoguchi is a member of IEEE, the virtual reality society of Japan, and a senior member of IEICE and IPS of Japan.

VAN-NAM HUYNH received the Ph.D. degree in Mathematics from Vietnam Academy of Science and Technology in 1999. He is currently a Professor of the School of Knowledge Science, Japan Advanced Institute of Science and Technology (JAIST). He was also a Visiting Professor at National Electronics and Computer Technology Center, Thailand (2019), an Adjunct Professor at Chiang Mai University, Thailand (2015-2017), and Part-time Lecturer at Tsukuba University, Japan (2011-2015). His current research interests include decision analysis and management science, AI and machine learning, modelling and reasoning with uncertain knowledge, argumentation, multi-agent systems, kansei information processing and applications. He has (co-)edited over fifteen Springer volumes and seven special issues for International Journal of Approximate Reasoning, Annals of Operations Research, and Data and Knowledge Engineering. Dr. Huynh currently serves as an Area Editor of International Journal of Approximate Reasoning, Editor-in-Chief of International Journal of Knowledge and Systems Science, and an editorial board member of the journal ARRAY.

***