Optimal mathematical programming and variable neighborhood search for *k*-modes categorical data clustering

Yiyong Xiao[a], Changhao Huang[a], Jiaoying Huang[a*], Ikou Kaku[b], Yuchun Xu[c]

[a]School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China

[b]Department of Environmental Management, Tokyo City University, Yokohama, 224-8551, Japan

[c]School of Engineering & Applied Science, Aston University, Birmingham, B4 7ET, United Kingdom

Abstract: The conventional *k*-modes algorithm and its variants have been extensively used for categorical data clustering. However, these algorithms have some drawbacks, e.g. they can be trapped into local optima and sensitive to initial clusters/modes. Our numerical experiments even showed that the *k*-modes algorithm cannot identify the optimal clustering results for some special datasets regardless the selection of the initial centers. In this paper, for small-sized datasets we developed an optimal programming approach based on integer linear programming (ILP) for *k*-modes clustering, which is independent to the initial solution; . For medium and large sized datasets, we developed a heuristic algorithm that implements iterative partial optimization in the ILP approach based on a framework of variable neighborhood search, known as IPO-ILP-VNS, to search for near-optimal results with controlled computing time. Experiments on 38 datasets from the UCI site, including 27 synthesized small datasets and 11 known benchmark datasets, were carried out to test the proposed ILP approach and the IPO-ILP-VNS algorithm. The experiment results outperformed the conventional and all other existing enhanced *k*-modes algorithms in literature.

Keywords: Categorical clustering; Variable neighborhood search; Data mining; Integer linear programming

1. Introduction

Cluster analysis is a popular data mining technique used in the field of knowledge discovery in databases (KDD). The *k*-means algorithm (Macqueen, 1967; Jain and Dubes, 1988) is a well-known clustering approach for its efficiency in clustering large datasets. However, the *k*-means algorithm cannot handle categorical datasets due to the lack of mean value measurement for categorical data. Huang (1997) presented the *k*-modes algorithm that extended the *k*-means algorithm to categorical data domains by using a simple measure of matching dissimilarity for categorical objects and by assigning the most frequent categorical values as the modes of the clusters. Mathematically, the *k*-modes algorithm can be formulated as an optimization model as following (Huang 1997):

[*] Corresponding author. Tel.: +86-010-82338294; fax: +86-010-82339519

E-mail addresses: xiaoyiyong@buaa.edu.cn (Yiyong Xiao, PhD), huangch1024@buaa.edu.cn (Changhao Huang, B.Eng.), huangjy@buaa.edu.cn (Jiaoying Huang, PhD), kakuikou@tcu.ac.jp (Ikou Kaku, PhD), y.xu16@Aston.ac.uk (Yuchun Xu, PhD)

$$\text{Min.}\quad F(W,Q)=\sum_{c\in C}\sum_{i\in N} w_{ic}\cdot d(X_i,Q_c) \qquad (1)$$

S.t.

$$(1)\quad \sum_{c\in C} w_{ic}=1 \qquad \forall i\in N$$

$$(2)\quad d(X_i,Q_c)=\sum_{j\in M}\delta(x_{ij},q_{cj}) \qquad \forall i\in N, c\in C$$

$$(3)\quad \delta(x_{ij},q_{cj})=\begin{cases}1, x_{ij}\neq q_{cj}\\ 0, x_{ij}=q_{cj}\end{cases} \qquad \forall i\in N, c\in C, j\in M$$

$$(4)\quad w_{ic}\in\{0,1\} \qquad \forall i\in N, c\in K$$

In the above formula, the objective function is to minimize the total inner-distances of all centers. Notations N, M, and C represent the set of data objects, the set of categorical attributes, and the set of objective clusters respectively. The numbers of elements in N, M, and C are denoted by n, m, and l ($l < n$) and indexed by i, j, and c respectively. Notation X_i represents the i^{th} data object in the dataset X described by m categorical attribute in accordance to $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$; The notation $Q_c \in Q$ denotes the center of the c^{th} cluster described by m categorical attributes in accordance to $Q_c = \{q_{c1}, q_{c2}, \ldots, q_{cm}\}$; The notation $W = \{w_{ic}\}$ denotes an $n \times l$ binary membership matrix indicating that when $w_{ic} = 1$, the data object X_i belongs to the cluster c, and when $w_{ic} = 1$ it does not. Function $d(X_i, Q_c)$ is the measurement of distance/dissimilarity between the object X_i and its cluster center Q_c, which is often referred to as the simple matching dissimilarity measure by Kaufman and Rousseeuw (1990).

However, the optimization model formulated by Equation (1) and Constraints (1)–(4) is nonlinear and cannot be solved with optimal solutions. For solving this problem, the *k*-modes algorithm adopts a partial optimization strategy to optimize one of the two variables (i.e. $\boldsymbol{W}$ and $\boldsymbol{Q}$) in turn until no further improvement can be made to the objective function. The steps are outlined in Fig. 1 below.

1. Find a set of l initial cluster centers
2. Fix $\boldsymbol{Q}$ and find $\boldsymbol{W}$ to minimize $F(\boldsymbol{W}, \boldsymbol{Q})$
3. Fix $\boldsymbol{W}$ and find $\boldsymbol{Q}$ to minimize $F(\boldsymbol{W}, \boldsymbol{Q})$
4. Repeat Steps 2 and 3, until $F(\boldsymbol{Q}, \boldsymbol{W})$ cannot be improved

Fig.1 Partial optimization of the *k*-modes algorithm

As pointed out by Huang (1997), the above partial optimization employed in the *k*-modes algorithm is not a global approach and the final clustering results are very sensitive to the initial centers/modes. Although many following-up studies (see the related work section) proposed enhanced/improved *k*-modes algorithms (or variants) in order to improve the qualities and stabilities of the clustering results, this known drawback with *k*-modes clustering algorithms still exists.

In this study, we formulated the *k*-modes clustering problem as an integer linear programming (ILP) model. This ILP model can be efficiently solved with optimal results for categorical clustering on small-sized datasets by using an MIP solver, without using initial centers/modes. We also proposed a heuristic algorithm, referred to as IPO-ILP-VNS, to find the near-optimal results for medium and large-sized datasets. Computational experiments on synthesized and benchmark datasets from the well-known UCI site were carried out to test the effectiveness and efficiency of the proposed ILP model and the IPO-ILP-VNS algorithm, and our new approach delivered better or equivalent clustering results than all existing algorithms in our experiments. Our numerical experiments also showed that the *k*-modes algorithm only has an average probability of 13.9% to identify the optimal results, with an average deviation of 16.2% from the optimal ones estimated by the ILP model.

The remaining part of this paper is organized as follows. In Section 2, we present current related studies. In Section 3, a domain-mapping method is introduced to represent the categorical objects with binary matrices, based on which an integer linear programming (ILP) model is presented for the *k*-modes clustering problem. In Section 4, we presents the proposed IPO-ILP-VNS algorithm, and an enhanced version of the ILP model for large-sized datasets. In Section 5, we carry out the first part of computational experiments to compare the ILP model with the conventional *k*-modes algorithm. In Section 6, we carry out the second part of the computational experiments to test the IPO-ILP-VNS algorithm and compare it with the existing algorithms on 11 benchmark datasets from the IUC site and three synthesized large-sized datasets. Finally, we conclude the study in Section 7.

2. Review of related studies

Since categorical data are ubiquitous in the real world, clustering data with categorical attributes has a broad range of practical applications. A number of clustering algorithms for categorical datasets have been studied and practiced in literature, such as the *k*-means-based algorithm (Ralambondrainy, 1995), the *k*-modes algorithms (Huang, 1998), the ROCK algorithm (Guha *et al.*, 2000), the CACTUS algorithm (Ganti *et al.*, 1999), the RST-based algorithms (Parmar *et al.*, 2007; Cao *et al.*, 2012), the k-populations algorithm (Kim *et al.*, 2005), the soft feature-selection scheme (Chen *et al.*, 2016), the Clustering ensemble selection algorithm (Zhao *et al.*, 2017), the k-multi-weighted-modes algorithm (Cao et al., 2017), and the clustering methods based on k-nearest-neighbor graph (Qin *et al.*, 2018, Myhre *et al.*, 2018). Among them, the *k*-modes categorical clustering algorithm is the most well-known algorithm that can cluster large-sized categorical datasets into a given number of clusters represented by the most-frequent attribute values (i.e., the modes) in a fast manner. However, the *k*-modes algorithm, as well as its extensions/variants, are quite sensitive to initial modes/centers and may not be able to identify the optimal clustering results even using different initializations.

A number of studies developed new initialization methods to improve the solution quality and stability of the *k*-modes algorithms, they can be largely categorized into four classes: (1) the refinement algorithms that refine the initial centers/modes iteratively (Bradley and Fayyad, 1998; Sun *et al.*, 2002), (2) the distance-based algorithms that identify the *k* most separated data objects from the dataset as initial seeds (Barbara *et al.*, 2002; He, 2006), (3) the multiple-clustering algorithms that utilize the results of multiple clusters on subsets of exemplars from the entire data object set (Frossyniotis *et al.*, 2002; Khan and Kant, 2007; Bai *et al.*, 2011; Khan and Ahmad, 2013), and (4) the density-based algorithms that consider the distributions/outlines of data objects to choose the initial centers/modes (Wu *et al.*, 2007; Cao *et al.*, 2009; Bai *et al.*, 2011,2012; Jiang *et al*., 2016; Chen *et al.*, 2017; Bai et al., 2017). The following section presents more details of these reviewed methods.

Sun *et al.* (2002) first applied the iterative refinement algorithm for the *k*-means clustering (Bradley and Fayyad, 1998) to the *k*-modes algorithm, which first drew an initial estimation of the clusters using the *k*-means or modes algorithm from a randomly selected subset of the data, and then subsequently extended the clusters to the entire data set by gradually adjusting the means or modes as more data are accumulated. The refinement algorithm reduced the sensitivity of the *k*-modes algorithms on their initialization, and was reported to yield better clustering results than the non-refinement initialization methods (Sun *et al.*, 2002).

Barbara *et al.* (2002) proposed the Coolcat algorithm which introduced the concept of entropy to measure the distance of data object pairs and used a max–min distance/entropy method to find the *k* most dissimilar data objects from the dataset as the initial seeds. He (2006) used a farthest-point heuristic approach as the initialization method of the *k*-modes algorithms, which takes an arbitrary point as the first initial center, and then picks iteratively the next point that has the maximum distance to the nearest point among all the points picked so far, until all initial centers were determined. However, the distance-based methods are apt to consider far away data objects, and outliers may thus be selected which may worsen the clustering results (Bai *et al.*, 2012).

Frossyniotis *et al.* (2002) presented the multiclustering fusion method which combines the results of multiple independent runs of clustering in order to obtain a stable dataset partition. Khan and Kant (2007) utilized the idea of evidence accumulation to combine the results of multiple clustering to determine the most diverse set of initial modes for the *k*-modes algorithm. Khan and Ahmad (2013) proposed a cluster center initialization algorithm that performs multiple dataset clusters, based on different selection of the attributes with respect to their significance ranking.

Wu *et al.* (2007) introduced the concept of data object density and proposed a density-based initialization method for the *k*-modes algorithm according to which the point with the highest density should be selected as the initial cluster centers. Cao *et al.* (2009) presented the concept of data object density, which was defined as the average frequency of one object's categorical attribute value that appeared in other objects, and combined it with a distance measure to determine a set of initial cluster centers. Bai *et al.* (2011, 2012) defined the data object's density as the number of other objects to which the object-of-interest was the nearest, and integrated the density with the distance measures to select *k* initial cluster centers from the potential exemplar set. Chen *et al.* (2017) proposed a fast clustering algorithm which determined the cluster center quickly by constructing the normal distribution function of the density and distance of the data objects. In contrast to the data object density, Jiang *et al.* (2016) defined the data objects' outlierness in terms of a weighted matching distance and thereby improved the initialization of the *k*-modes algorithm by avoiding to choose outliers as initial cluster centers. Chen *et al.* (2016) proposed an algorithm for clustering categorical data with a novel soft feature-selection scheme, where the dissimilarity between categorical data objects is measured using a probabilistic distance function. Zhao *et al.* (2017) presented a clustering ensemble selection algorithm for categorical data with five normalized validity indices to measure the quality and diversity of the clusters. Gupta et al. (2018) developed two leaping techniques to estimate the number of clusters that was needed to be given in advance for center-based clustering methods.

All the initialization algorithms/methods in the reviewed papers above reported some degrees of improvement on clustering qualities or stabilities of the *k*-modes algorithms. However, few of them discussed the optimality of the clustering results in terms of the objective function. Actually, our numeric experiments (see Section 5.1) showed that the *k*-modes algorithm has a very low (even zero for some tested datasets) probability to find out the optimal clustering results. The optimality of the clustering algorithm, especially for large-sized categorical datasets is an important issue, but has not received enough research attention.

In this study, we propose a new optimal approach based on integer linear programming (ILP) to solve the *k*-modes clustering problem toward the minimization of the total inner-distance function, based on the framework of variable neighborhood search (VNS), particularly for large-sized categorical dataset. Although ILP-based techniques were applied to other types of clustering problems in literature, such as the set-level constrained clustering problem by Mueller and Kramer (2010) and the hierarchical clustering problem by Gilpin *et al.* (2013), there is no work found on formalizing the *k*-modes clustering as an ILP model. Global meta-heuristics can also be found in literature for solving the clustering problems with near-optimal solutions, such as the Tabu search and genetic algorithms for *k*-modes clustering (Ng and Wong 2002; Gan *et al.*, 2009), the particle swarm optimization

(Kao *et al.*, 2008) for the *k*-means clustering, and the variable neighborhood search for the harmonic means clustering (Alguwaizani *et al.*, 2011).

3. An integer linear programming model for *k*-modes clustering

In this section, we first introduce a new categorical data representation method using a domain-mapping technique followed by presenting an integer linear programming (ILP) model for the *k*-modes clustering on categorical datasets.

3.1 A domain-mapping method for representing categorical objects

Suppose there is a set of data objects denoted as $X = \{X_1, X_2, \ldots, X_n\}$ and m associated categorical attributes denoted as $A_1, A_2..., A_m$. Each data object X_i can be expressed as $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ and each categorical attribute A_j corresponds to a domain with limited categorical values denoted as a vector $\overline{V}_j = \{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(p_j)}\}$, where p_j is the total number of categorical values in the domain A_j. We use the following two definitions to transfer the categorical values/objects into binary representations.

DEFINITION 1. *Domain-mapping of categorical values*: A categorical value of data object X_i on a limited domain A_j, i.e., x_{ij}, can be uniformly expressed as a mapping on the domain in the form of the point product $x_{ij} = \overline{\delta}_{ij} \cdot \overline{V}_j$, where $\overline{\delta}_{ij} = \{\delta_{ij}^1, \delta_{ij}^2, \ldots, \delta_{ij}^t, \ldots, \delta_{ij}^{p_j}\}$ is a mapping vector satisfying $\delta_{ij}^t \in \{0,1\}$ and $\sum_{t=1}^{p_j} \delta_{ij}^t = 1$, and $\overline{V}$ is the domain vector representing the ordered categorical values in domain A_j.

DEFINITION 2. *Domain-mapping of categorical object*: A data object X_i with m attribute values $x_{i1}, x_{i2}, \ldots, x_{im}$ on categorical domains $A_1, A_2, \ldots, A_m$ can be expressed in the form of the point product $X_i = \overline{\Delta}_i \cdot \overline{\Omega}$, where $\overline{\Delta}_i = \{\overline{\delta}_{i1}, \overline{\delta}_{i2}, \ldots, \overline{\delta}_{im}\}$ is a mapping matrix characterizing the exact position of the data object X_i in the domain matrix $\overline{\Omega} = \{\overline{V}_1, \overline{V}_2, \ldots, \overline{V}_m\}$.

Thus, based on Definition 1, we can express any categorical value as the product of an individualized mapping vector and a common domain vector. Furthermore, by Definition 2, a categorical data object with multiple categorical attributes can be expressed as the product of an individualized mapping matrix and a common domain matrix. Note that either the mapping vector, or the matrix, contain only binary elements, which allows us to build an integer programming model for data object clustering.

In the following, we use examples to illustrate how the categorical data objects are expressed by binary matrices. Suppose there are four objects in the dataset X, i.e., $X = \{X_1, X_2, X_3, X_4\}$ that have three categorical attributes on three domains, namely, SEX, EDUCATION, and JOB. Four data objects have specified attribute values as follows:

X_1 = {male, bachelor, engineer}

X_2 = {female, doctor, teacher}

X_3 = {female, master, manager}

X_4 = {male, bachelor, scientist}

The domain matrix on three attributes is given as follows.

$$\overline{\Omega} = \left\{ \begin{array}{ccc} male & bachelor & teacher \\ female & master & engineer \\ & doctor & manager \\ & & \mathrm{sci}entist \end{array} \right\}$$

Thus, according to Definitions 1 and 2, we can express the data objects X_1, X_2, X_3, and X_4, as $X_1 = \overline{\Delta}_1 \cdot \overline{\Omega}$, $X_2 = \overline{\Delta}_2 \cdot \overline{\Omega}$, $X_3 = \overline{\Delta}_3 \cdot \overline{\Omega}$, and $X_4 = \overline{\Delta}_4 \cdot \overline{\Omega}$, and each $\overline{\Delta}_i = \{\overline{\delta}_{i1}, \overline{\delta}_{i2}, \ldots, \overline{\delta}_{im}\}$, $\forall i = 1, 2, 3, 4$ corresponds to a binary matrix detailed as follows,

$$\overline{\Delta_1} = \left\{ \begin{array}{ccc} 1 & 1 & 0 \\ 0 & 0 & 1 \\ & 0 & 0 \\ & & 0 \end{array} \right\} \quad \overline{\Delta_2} = \left\{ \begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ & 1 & 0 \\ & & 0 \end{array} \right\} \quad \overline{\Delta_3} = \left\{ \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ & 0 & 1 \\ & & 0 \end{array} \right\} \quad \overline{\Delta_4} = \left\{ \begin{array}{ccc} 1 & 1 & 0 \\ 0 & 0 & 0 \\ & 0 & 0 \\ & & 1 \end{array} \right\}$$

3.2 Integer linear programming (ILP) model

Below, we present the ILP model for the *k*-modes clustering problem on categorical datasets.

Parameter notations:

N set of data objects to be clustered, n = card(N)

i index of data objects, $i \in N$

M set of attributes/domains to describe data objects, m = card(M)

j index of attributes/domains, $j \in M$

A_j value domain of attribute j

V_j set of all categorical values in domain A_j, p_j = card(V_j)

v index of values for domain A_j, $v \in V_j$

ω_{ijv} binary mapping matrix of data objects defined in Definitions 1 and 2, indicating that if object i takes v^{th} categorical values in domain A_j (by ω_{ijv} = 1) or not (by ω_{ijv} = 0), and satisfying $\sum_{v \in V_j} \omega_{ijv} = 1$ and $\omega_{ijv} \in \Omega$.

C set of clusters, l = card(C)

c index of clusters, c□C

Variables:

w_{ic} binary decision variable, indicating if a data object i belongs to cluster c (by $w_{ic} = 1$) or not (by $w_{ic} = 0$)

u_{cjv} binary decision variable, indicating if the center of cluster c takes the v^{th} categorical value in domain A_j (in accordance to $u_{cjv} = 1$) or not (in accordance to $u_{cjv} = 0$)

d_{icj} binary decision variable, indicating if data object i has a different value than the cluster center c on attribute j (by $d_{icj} = 1$) or not (by $d_{icj} = 0$)

Objective function

The objective function of the k-modes clustering is to minimize the total inner-distance of all clusters, which is expressed as follows:

$$\text{Min. } F(W,U) = \sum_{c \in C} \sum_{i \in N} \sum_{j \in M} d_{icj} \tag{2}$$

s. t.

$$(5)\quad d_{icj} \geq \left|\omega_{ijv} - u_{cjv}\right| - (1 - w_{ic}) \qquad \forall i \in N, c \in C, j \in M, v \in V_j$$

$$(6)\quad \sum_{v \in V_j} u_{cjv} = 1 \qquad \forall c \in C, j \in M$$

$$(7)\quad \sum_{c \in C} w_{ic} = 1 \qquad \forall i \in N$$

$$(8)\quad \sum_{i \in N} w_{ic} \geq 1 \qquad \forall c \in C$$

$$(9)\quad w_{ic}, u_{cjv}, d_{icj} \in \{0,1\} \qquad \forall i \in N, c \in C, j \in M, v \in V_j$$

In the above formula, Constraint (5) determines the binary distance of each data object to its cluster center on each attribute. The nonlinear absolute function is kept for the model simplification and can be linearized by dividing it into two linear constraints. Note that the term $(1-w_{ic})$ on the right side of the equation ensures that the variable d_{icj} must be greater than or equal to $|\omega_{ijt}-u_{cit}|$ only for $w_{ic} = 1$ (indicating that data object i belongs to cluster c). Constraint (6) indicates that each attribute of the cluster center must have only one categorical value from the attribute domain. Constraint (7) indicates that each object must belong to one cluster. Constraint (8) indicates that each cluster must have at least one data object/member. Constraint (9) indicates that the distances of a data object to other clusters (it does not belong to) must be zero. Constraint (10) defines the value domains of all variables. According to the theory of Huang (1998) that the optimal center of a given cluster always takes the most-frequent attribute values of its element members, the following Constraints (10) and (11) is not necessary for the integrity of the ILP model but always holds for optimal solutions.

$$(10) \quad d_{icj} \leq w_{ic} \qquad \forall i \in N, c \in C, j \in M$$

$$(11) \quad \sum_{i \in N} \omega_{ijv} w_{ic} - \sum_{i=1}^{n} \omega_{ijv'} w_{ic} \geq n(u_{cjv} - 1) \qquad \forall c \in C; j \in M; v, v' \in V_j; v \neq v'$$

Thus, based on the objective function Eq. (2) and Constraints (5)–(9), we have formulated the *k*-modes clustering problem as an integer linear programming (ILP) model. Note that since all expressions are linear, the ILP model can be directly solved optimally by an MIP solver that does not need any initial solution. In the experimental section, we use the ILP model to solve a set of synthesized datasets and several well-known benchmark datasets, and compare the results with those obtained by traditional *k*-modes-based algorithms.

4. VNS-based heuristic approach for medium and large-sized datasets

The ILP model formulated in Section 3 is practically unsolvable for medium and large-sized data set within reasonable CPU time. To deal with this, we develop a heuristic approach that implements iterative partial optimization (IPO) on the ILP model under the framework of variable neighborhood searches (VNS), known as IPO-ILP-VNS, to obtain a near-optimal solution whose quality can be controlled by the given CPU time. VNS is a new top-level meta-heuristics approach proposed by Mladenovic and Hansen (1997) and it has been successfully applied to combinatorial optimization problems in various fields, such as the travelling salesman problem, the P-median problem, the vehicle routing problem, and the multi-level lot-sizing problem (Hansen *et al.*, 2010; Xiao *et al.*, 2011a, 2014a). The mechanism of VNS is to perform local search with designed changes in multilevel neighborhoods, which gives many desirable properties of meta-heuristics such as simplicity, robustness, user-friendliness and generality (Hansen et al., 2008). This is a first try to implement the VNS scheme on the *k*-modes clustering problem. The IPO on an ILP model is also referred to as the ILP-based refinement or the “fix-and-optimize heuristics” in other fields of applied optimizations (Franceschi *et al.*, 2006; Helber and Sahling, 2010; Xiao *et al.*, 2011a, 2011b; Xiao and Konak, 2016).

4.1 ILP-based variable neighborhood search

The ILP model formulated in Eq. (2) and Constraints (5)–(9) has three groups of binary variables, which are w_{ic}, u_{cjt}, and d_{icj}. Since u_{cjt} is tightly bound to w_{ic} according to Constraint (11), and d_{icj} is dependent on w_{ic} and u_{cjt}, there is only one independent variable, namely w_{ic}. Thus, to determine an optimal solution, we just need to determine optimally $n \times l$ combinations of binary values for variable w_{ic}. However, for medium and large-sized problems, the binary variables in w_{ic} are too many for their optimization with the MIP solver at one time, to deal with this, we employ the IPO strategy to select only a small part of the binary variables in w_{ic} for optimization while fixing the rest with given values (which may not be optimal). Thus, an MIP solver can solve the “limited” problem very efficiently. Furthermore, by repeating the “select-optimize” IPO process iteratively based on the framework of the variable neighborhood search (VNS) (Mladenović and Hansen, 1997, Hansen and Mladenović, 2001; Mladenovic *et al.*, 2012; Xiao *et al.*, 2011a, 2014b), the original task to optimize the entire medium and large-sized problem at a single instant is decomposed to optimize multiple small-sized problems at multiple times.

To implement the IPO under a VNS framework, we first define a distance metric of solutions based on variable w_{ic}, and then define the neighborhood structure of solutions based on the distance metric.

DEFINITION 3. *The distance metric*: For any two solutions, say x and y, of the ILP model formulated in Eq. (2) and Constraints (5)–(9), the solution distance, noted as $dis(x, y)$, can be calculated by $dis(x,y)=\frac{1}{2}\sum_{i\in N,c\in C}|w_{ic}^{(x)}-w_{ic}^{(y)}|$, where $w_{ic}^{(x)}$ and $w_{ic}^{(y)}$ are membership matrices for solutions x and y respectively.

The distance metric reflects the exact number of data objects classified into different clusters in the two solutions. In Fig. 2, we provide two examples to show how the solution distances are calculated. The solutions x and y in Fig. 2A have only one data object (i.e., data object No. 2) that belongs to different clusters, so we have $dis(x, y) = 1$. The solutions x and y in Fig. 2B have two data objects (i.e., data objects No. 1 and 2) that belong to different clusters, and thus $dis(x, y) = 2$. The property $0 \le dis(x, y) \le n$ holds for any two solutions, where n is the total number of data objects in the dataset.

$$w_{ic}^{(x)}=\left\{\begin{matrix}0 & 1 & 0\\ \boxed{1 \quad 0} & & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{matrix}\right\},\ w_{ic}^{(y)}=\left\{\begin{matrix}0 & 1 & 0\\ \boxed{0 \quad 1} & & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{matrix}\right\} \qquad w_{ic}^{(x)}=\left\{\begin{matrix}0 & \boxed{1 \quad 0} & \\ \boxed{1 \quad 0} & & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{matrix}\right\},\ w_{ic}^{(y)}=\left\{\begin{matrix}0 & \boxed{0 \quad 1} & \\ \boxed{0 \quad 1} & & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{matrix}\right\}$$

(A) $dis(x, y) = 1$ (B) $dis(x, y) = 2$

Fig.2 Examples of solution distances

DEFINITION 4. *Neighborhood structures of incumbent solution*: For the incumbent solution x of the ILP model formulated in Eq. (2) and Constraints (5)–(9), a solution set $N_k(x)$ is the k^{th} neighborhood of x if it satisfies $dis(x,y)\le k$ for all $y\in N_k(x)$.

Based on Definitions 3 and 4, the algorithm of IPO-ILP-VNS that combines iterative partial optimization, the ILP model, and the VNS framework, is given in Fig. 3, as follows.

```
Algorithm IPO-ILP-VNS ($K_{\max}$, $L_{\max}$, $R$):
```

```
1)  Initialization: find a solution $x$ based on the following two steps:
      (a) Select randomly $l$ data objects as cluster centers
      (b) Assign all data objects to their nearest cluster center
2)  Set $k \leftarrow R$, $L \leftarrow 0$
3)  Until $k = k_{\max}$ repeat the following steps
    (A) Shaking: select randomly a set $K$ of $k$ data objects
    (B) Local search (the partial optimization):
        Fix variable instances $w_{ic}^{(x)}$ for all $i \in N, c \in C$
        Unfix variable instances $w_{ic}^{(x)}$ for all $i \in K, c \in C$
        Call the MIP solver to find a new solution $x'$ in the neighborhood $N_k(x)$
    (C) If $x'$ improves upon $x$ Then let $x \leftarrow x'$, $L \leftarrow 0$
        Else let $L \leftarrow L+1$
    (D) If $L > L_{\max}$ then let $k \leftarrow k + R$ and let $L \leftarrow 0$
4)  Return the best solution found
```

Fig.3 Framework of the IPO-ILP-VNS algorithm

The IPO-ILP-VNS algorithm has three parameters, namely $K_{\max}$, $L_{\max}$, and R, that represent the farthest neighborhood to-be-searched, the maximum tries between two improvements within a neighborhood, and the step needed to enlarge the neighborhood respectively. The algorithm first initializes a solution x from a set of randomly selected cluster centers, and then starts to search the first neighborhood (k←R) for improvement. The *Shaking* phase selects randomly k number of data objects that will be allowed (by an unfix operator) to be optimized for their clustering memberships. The *local search* phase implements partial optimization on the selected data objects by an MIP solver. In this phase, if a better solution x' is found (in Step 3C), then accept it by $x \leftarrow x'$, and reset the non-improvement counter by $L \leftarrow 0$. Otherwise, increase the non-improvement counter by $L \leftarrow L+1$. In the next step (step 3D), determine if the non-improvement counter is greater than the parameter $L_{\max}$. If it is, then search the next neighborhood by $k \leftarrow k + R$ and reset the non-improvement counter by $L \leftarrow 0$. The algorithm stops after the maximum neighborhood $K_{\max}$ has been searched without improvement after $L_{\max}$ continuous tries.

Note that in Step 3A) of the IPO-ILP-VNS algorithm, all the data objects have equal probability to be selected into the set of partial optimizations. However, two priority policies can be applied here in order to accelerate the converging process of the algorithm. The first one is the *Distance Priority Policy* (DPP), which assumes that data objects with larger distances to their cluster centers should have higher probabilities for being selected. The underlying philosophy is simple: since these data objects are far from the center they are more likely to switch their cluster memberships in the partial optimization. The other one is called *Time Priority Policy* (TPP), indicating that a data object waiting for longer time (to be selected) will have a higher probability to be selected. Under TPP, for each data object there is a time recorder to record the last time it had been selected into a partial optimization.

We have verified the effects of these two heuristic policies in our computational experiments.

The IPO-ILP-VNS algorithm adopts a general framework of the VNS algorithm with only a replacement of the local search by an IPO process, so it has the same level of computational complexity with the VNS which is measured by how many times of local search being implemented. We suppose the average CPU time used in each round of local search (i.e., the partial optimization by an MIP solver) is denoted by *T*, the times of local search is estimated by the product of the maximum number of neighborhoods to be searched (i.e., $K_{\max}$) and the maximum times of tries on each neighborhood (i.e., positively related to $L_{\max}$). Thus, the complexity level of the IPO-ILP-VNS algorithm can be estimated as O ($K_{\max} \times L_{\max} \times T$).

4.2 Improved expression of the ILP model for large size problems

When the problem size is large, such as up to 100,000 data objects, the ILP model formulated by Eq. (2) and Constraints (5)–(9) in Section 3 is actually unsolvable for an MIP solver, even for optimizing only a small part of the decision variables. In this subsection, we use a merging method to improve the expression of the ILP model so that it can be applied to solve problems with large sizes using the IPO-ILP-VNS algorithm. This merging method sums the attribute information of all the fixed data objects according to their cluster belongingness, and then uses the summed information in the ILP model. As shown next, three new parameters are introduced to describe the improved ILP model.

N' set of selected data objects to be re-clustered

z_{cjv} number/frequency of fixed data objects in cluster *c* that take the *t* value in domain A_j, $z_{cjv} = \sum_{i \in N \setminus N'} \omega_{ijv}$

r_c number of fixed data objects in cluster *c*, $r_c = \sum_{i \in N \setminus N'} w_{ic}$

The ILP model formulated in Eq. (2) and Constraints (5)–(9) in Section 3 can be improved as new formulas in Eq. (3) and Constraints (12)–(17) as follows.

$$\text{Min. } F(W,U) = \sum_{c \in C} \sum_{i \in N'} \sum_{j \in M} d_{icj} + \sum_{c \in C} \sum_{j \in M} \sum_{v \in V_j} z_{cjt} u_{cjv} \tag{3}$$

s.t.

$$\text{(12)} \quad d_{icj} \geq \left| \omega_{ijv} - u_{cjv} \right| - (1 - w_{ic}) \qquad \forall i \in N', c \in C, j \in M, v \in V_j$$

$$\text{(13)} \quad \sum_{v \in V_j} u_{cjv} = 1 \qquad \forall c \in C, j \in M$$

$$\text{(14)} \quad \sum_{c \in C} w_{ic} = 1 \qquad \forall i \in N'$$

$$(15)\quad r_c + \sum_{i \in N'} w_{ic} \geq 1 \qquad\qquad \forall c \in C$$

$$(16)\quad d_{icj} \leq w_{ic} \qquad\qquad \forall i \in N', c \in C, j \in M$$

$$(17)\quad w_{ic}, u_{cjv}, d_{icj} \in \{0,1\} \qquad\qquad \forall i \in N', c \in C, j \in M, v \in V_j$$

In the above formulas, the objective function has two terms, the first of which is from the selected data objects (will be re-optimized) and the second is from the unselected data objects (will be fixed). Note that z_{cjv} contains merged information of data objects with fixed values. It is a constant parameter so the objective function is still linear. Constraints (12)–(17) can be interpreted in a similar manner to Constraints (5)–(9) except that the data object set *N* is replaced by *N′*. Note that the replacement of *N* by *N′* has reduced the complexity of the ILP model significantly, because *N* can be a very large number but *N′* is a small number. This makes it possible to solve very large-sized categorical clustering problems using the IPO-ILP-VNS algorithm.

5. Computational experiment-part I

The experiments were conducted on a Mac book computer with a 2.9 Intel Core i7 CPU and a 4G RAM. The ILP model was coded using AMPL (see Fig. A1 in Appendix) and was solved with the well-known MIP solver CPLEX (Version 12.6.1.0).

5.1 Numeric experiments on synthesized datasets

Firstly, we generated a group of small-sized datasets of categorical data objects to test the ILP model and compared the clustering results with those of the *k*-modes algorithm. This group of datasets has five classes that contain 10, 12, 14, 16, 18, and 20 data objects, respectively, and each class is associated with four categorical attributes that have four categorical values. The data objects in each dataset will be clustered into 2, 3, 4, or 5 clusters, respectively, with an objective function for the minimization of the total inner-distance of all clusters. Thus, the combination produces 6×4 = 24 problem instances for testing.

We first applied the *k*-modes algorithm to solve the 24 problem instances. The first-match policy was applied in the algorithm when a data object had an equal distance to the multiple cluster centers. We executed the *k*-modes algorithm repeatedly using all the possible combinations of the initial cluster centers in order to obtain all the possible results. Subsequently, we used the AMPL/CPLEX to solve the ILP model formulated in Eq. (2) and Constraints (5)–(9) for each of the test problems to obtain the optimal solutions. The performance and comparison of the test are shown in Table 1, where column "*Optimal solutions*" indicates the optimal solutions obtained by solving the ILP model, column "*Number of initial combinations*", calculated as C_n^l, indicates the total number of initial center combinations, column "*Number of distinct solutions*" indicates the total number of final distinct solutions found by the *k*-modes algorithm, columns "*Best solution*" and "*AVG solution*" indicate the best and average solutions respectively, column "*Dev. (%)*" is the deviation of the average solution found by the *k*-modes algorithm from the optimal solution found by the ILP model, column "*Optimal solution found?*" indicates if the *k*-modes algorithm has found the optimal solution based on all the initial combinations, and column "*Optimality rate (%)*" indicates the statistical possibility of reaching the optimal solution if started from a random initialization.

Table 1 Experimental results and comparisons of 24 small-scale instances

				Execution of *k*-modes algorithm based on all initial combinations						
Problem ID	Number of objects	Number of clusters	Optimal solution	Number of initial combinations	Number of distinct solutions	Best solution	AVG solution	Dev. (%)	Optimal found?	Optimality rate (%)
1	10	2	**16**	45	3	**16**	16.6	3.6	Yes	64.4
2	10	3	**13**	120	5	**13**	13.8	6.2	Yes	37.5
3	10	4	**11**	210	5	**11**	12.3	12.1	Yes	4.8
4	10	5	**8**	252	6	**8**	10.3	28.8	Yes	0.4
5	12	2	**23**	66	5	24	24.8	7.8	No	0
6	12	3	**17**	220	6	**17**	19.1	12.4	Yes	7.3
7	12	4	**13**	495	8	**13**	15.2	17.0	Yes	8.3
8	12	5	**9**	792	7	**9**	11.3	25.1	Yes	3.9
9	14	2	**26**	91	6	**26**	27.0	4.0	Yes	37.4
10	14	3	**20**	364	7	**20**	22.2	10.8	Yes	8.0
11	14	4	**16**	1001	9	**16**	18.7	17.0	Yes	1.0
12	14	5	**12**	2002	9	**12**	14.4	19.7	Yes	1.8
13	16	2	**26**	120	7	**26**	28.6	9.8	Yes	18.3
14	16	3	**24**	560	9	25	27.4	14.2	No	0
15	16	4	**19**	1820	10	**19**	22.1	16.5	Yes	0.9
16	16	5	**17**	4368	9	**17**	18.8	10.5	Yes	12.5
17	18	2	**31**	153	9	**31**	32.5	4.8	Yes	58.2
18	18	3	**26**	816	12	**26**	28.5	9.5	Yes	19.5
19	18	4	**24**	3060	9	**24**	27.0	12.4	Yes	3.2
20	18	5	**21**	8568	9	**21**	24.0	14.3	Yes	1.8
21	20	2	**35**	190	9	**35**	37.9	8.4	Yes	14.2
22	20	3	**30**	1140	10	**30**	31.9	6.3	Yes	23.1
23	20	4	**24**	4845	11	**24**	26.3	9.5	Yes	6.2
24	20	5	**22**	15504	16	**22**	26.6	21.1	Yes	0.7
AVG								16.2		13.9

Note: numbers indicated in boldface denote optimal values

It can be observed from Table 1 that the ILP model has found optimal solutions for all tested problems. The *k*-modes algorithm may have different results if started from different initializations, with an average optimality rate of 13.9% for the 24 test problems. We note that the *k*-modes algorithm has a zero optimality rate for instances No. 5 and No. 14, thus indicating that irrespective of the combination of the selection of the initial centers, the optimal objective values, which are 23 and 24 for instances No. 5 and No. 14 according to the ILP model, respectively, will not be found by the *k*-modes algorithm. This is because the *k*-modes algorithm is a partial optimization process that performs iterative optimization on only one of the two variables $\boldsymbol{Q}$ and $\boldsymbol{W}$ at a time. So it is not a global strategy and does not guarantee the result optimality. In particular, when a data object has an equal distance to multiple clusters, the algorithm has to assign the data to one of the clusters under certain policies (e.g., first-match, last-match, at random), which may cause the result losing optimality. For other instances, such as No. 4, 11, 12, 15, 19, 20, and 24, the optimality rates are very low (approximately 1%), thus indicating that their optimal solutions are very difficult to find. The average deviation of the solutions by the *k*-modes algorithm is approximately 16.2% higher (or worse) than the optimal solutions found by the ILP model.

We list instance No. 5 in Table A1 in Appendix for interested readers, whose optimal clustering result is c_1 = {3,4,6,7,11,12} and c_2 = {1,2,5,8,9,10} with respect to the cluster centers [3, 3, 4, 3] and [4, 1, 3, 1] respectively. The detailed result distribution of instance No.14 is shown in Fig. A2 in Appendix. More distributions of distinct solutions can be found in Fig. A3 for the problem instances with Nos. 21, 22, 23 and 24. The relations between the number of local optima and the total number of data objects with respect to different cluster numbers can be found in Fig. A4. The trend of the average deviation of solutions against the total number of data objects and the number of clusters can be found in Fig. A5.

5.2 Numeric experiments on small-sized benchmark datasets

Next, we use a group of well-known benchmark datasets published in the UCI repository (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/) to test the performance of the ILP model and compare the results with the best ones provided in the literature. We select four larger datasets with object numbers 24, 132, 432, and 432, to test the ILP model. Detailed features of these datasets are listed in Table 2, including the number of data objects, the number of attributes, the number of original classes into which the objects were originally classified in reality, the attribute domain, and the outlines of the original classes.

Table 2 Benchmark datasets from UCI repository

Dataset name	Number of objects	Number of attributes	Number of classes	Number of values in each attribute domain	Number of objects in each class
Lenses	24	4	3	{3, 2, 2, 2}	{4, 5, 15}
Hayes–Roth	132	4	3	{3, 4, 4, 4}	{51, 51, 30}
Monks−1	432	6	2	{3, 3, 2, 3, 4, 2}	{216, 216}
Monks−2	432	6	2	{3, 3, 2, 3, 4, 2}	{216, 216}

We first used the *k*-modes algorithm to solve these benchmark datasets and repeated 30 runs for each dataset with random initializations to obtain 30 random results. Subsequently, we used AMPL/CPLEX to solve these datasets with the ILP model formulated in Eq. (2) and Constraints (5)–(9) to obtain the optimal results. The cluster qualities (in terms of the objective function value) and performances are shown and compared in Table 3. Subcolumns “*Best*”, “*AVG*”, “*Deviation (%)*”, and “*Time*” under column “*Thirty runs of k-modes algorithm*” represent the best solution of 30 runs, the average solution of 30 runs, the deviation of the average solution from the optimal solution found by the ILP model, and the average computational time used respectively. Subcolumns "*Optimum*" and "*Time*" under the column entitled "*ILP model*" represent the optimal solution obtained by the ILP model and the computational time used respectively.

Table 3 Experimental results and comparisons

Dataset	Thirty runs of k-modes algorithm				ILP model	
Name	Best	AVG	Deviation (%)	Time	Optimum	Time
Lenses	**27**	30.4	12.6	<1s	**27**	<1 s
Hayes–Roth	201	210.6	6.3	<1s	**198**	36.8 s
Monks-1	**1256**	1338.5	6.6	<1s	**1256**	27.5 s
Monks-2	**1256**	1333.5	5.8	<1s	**1256**	28.7 s
AVG			7.1			

Note: boldface numbers denote optimal values

It can be observed in Table 3 that the optimal results obtained by the ILP model for datasets "Lenses", "Hayes–Roth", "Monks–1", and "Monks–2" are 27, 198, 1256, 1256 respectively, they are always smaller than or equal to the best results obtained by the *k*-modes algorithm. Notably the *k*-modes algorithm found the optimal result of dataset "Lenses" (which is 27) only once in 30 runs, but could not find the optimal result for the dataset "Hayes–Roth" in 30 runs. The deviations of the average results the *k*-modes algorithm with respect to the optimal results are also quite large, which for example can be as high as 12.6% and 6.3% for datasets "Lenses" and "Hayes–Roth" respectively.

The benchmark datasets in the UCI repository provided their actual cases of class partitions in reality, which often serve as benchmarks for comparison in categorical clustering research studies (Saha and Das, 2008; Wu *et al.*, 2007). For examples, dataset "Lenses" has been partitioned into three classes in reality with the object numbers of 4, 5, and 15 respectively. However, our ILP model shows that these actual class partitions of the datasets may not be optimal in terms of their clustering quality evaluated by the total inner-distance objective function. In Table 4, we compare the cluster qualities of the actual cases with those of the optimized ones obtained using the ILP model. Column "*Known-best*" represents the actual class partitions provided in benchmark datasets, and Column "*ILP model*" represents the optimal results found by the ILP model. It can be observed that the objective values of the ILP model are much lower than those of the actual class partitions. We may thus say that the original classifications for these datasets are not optimal in terms of the objective function towards the minimization of the total inner-distance. Therefore, the clustering results of this study may serve as a new benchmark for future research studies in the field of categorical clustering.

Table 4 Experimental results and comparisons

	Known-best	*ILP model*	*Improved (%)*
Lenses	34	**27**	***−20.6***
Hayes–Roth	280	**198**	***−29.3***
Monks−2	1565	**1256**	***−19.7***
Monks−1	1500	**1256**	***−16.2***

Note: numbers in boldface indicate new, optimal values

6. Computational experiments--part II

In this section, we conducted computational experiments on small, medium, and large-sized datasets to test the performance of the IPO-ILP-VNS algorithm and compare the results with several enhanced *k*-modes algorithms in the literature. All experiments were run on a PC server with two 2.30 GHz Intel$^{@}$ Xeon(R) CPUs (32 cores) and 110 GB memory. The improved ILP model were coded by AMPL and solved with the well-known MIP solver CPLEX (Version 12.6.1.0). We first tested the optimality of the IPO-ILP-VNS algorithm using small-sized benchmark datasets in Section 6.1, and then tested the effects and efficiencies of the IPO-ILP-VNS with medium and large-sized benchmark datasets in Section 6.2.

6.1 Optimality test

We ran the IPO-ILP-VNS algorithm to solve the UCI benchmark datasets in Table 2. The calculations were repeated 30 times and started from a random *k*-modes initialization, such that 30 random results were obtained. The cluster qualities (in terms of the objective function) and performances are shown and compared in Table 5. Subcolumns "*Best*", "*AVG*", "*Deviation (%)*", and "*Time*" represent the best solutions for 30 runs, the average solutions for 30 runs, the deviation of the average solution from the optimal solution by the ILP model, and the average computational time used, respectively.

Table 5 Experimental results and comparisons

Dataset	Thirty runs of IPO-ILP-VNS				Thirty runs of *k*-modes				ILP model	
Name	Best	AVG	Deviation (%)	Time	Best	AVG	Deviation (%)	Time	Optimum	Time
Lenses	**27**	27	0	<1s	**27**	30.4	12.6	<1s	**27**	<1s
Hayes–Roth	**198**	200	1.01	26.9s	201	210.6	6.4	<1s	**198**	36.8s
Monks−1	**1256**	1257.1	0.09	15.1s	**1256**	1338.5	6.6	<1s	**1256**	27.5s
Monks−2	**1256**	1256.8	0.06	13.6s	**1256**	1333.5	6.2	<1s	**1256**	28.7s
AVG			0.29				7.9			

It can be observed in Table 5 that the IPO-ILP-VNS algorithm could find optimal solutions for all the test datasets, exhibiting good optimalities and high-computational efficiencies as well. The average deviation from the optimal solutions is as low as 0.29%, which is much lower than the average deviations of the k-modes algorithm, which is 7.9, and ranges from 6.2% to 12.6% in comparison to the IPO-ILP-VNS algorithm in 30 random runs.

6.2 Comparison of existing algorithms on medium and large-sized datasets

We used another group of medium and large-sized datasets in the UCI repository to compare the IPO-ILP-VNS algorithm with three improved *k*-modes algorithms in the literature (Cao *et al.*, 2009; and Wu, 2007; Khan, 2013). Detailed features of these datasets are listed in Table 6, including the number of objects, the number of attributes, the number of classes (i.e., the objects were originally classified in reality), the attribute domain, and class outlines. Note that although datasets “Zoo”, “Soybean”, and “Lung-cancer” have only small numbers of data objects, they may have large numbers of attributes, target classes, or attribute domains, so they cannot be solved directly using an MIP solver. Therefore, we categorized these problems into a medium-sized class. Datasets “L1”, “L2”, and “L3” are large-sized datasets randomly generated in this study that respectively contain 20000, 50000, and 100000 distinct data objects with 10 categorical attributes and 2–8 categorical values in each of them.

Table 6 Medium and large-sized benchmark datasets

Dataset name	Number of objects (*n*)	Number of attributes (*m*)	Number of classes (*l*)	Number of categorical values in each attribute domain	Number of objects in each class
Lung-cancer	32	56	3	2–3	{913, 10}
Tic-tac-toe	958	9	2	3	{626, 332}
Vote	435	16	2	2	{168, 267}
Soybean	47	21	4	2–7	{10, 10, 10, 17}
Car	1728	6	4	3–4	{1210, 384, 69, 65}
Zoo	101	16	7	2, 6	{41, 20, 5, 13, 4, 8, 10}
Mushroom	8124	22	2	1–10	{626, 332}
L1	20,000	10	3	2–8	--
L2	50,000	10	2	2–8	--
L3	100,000	10	2	2–8	--

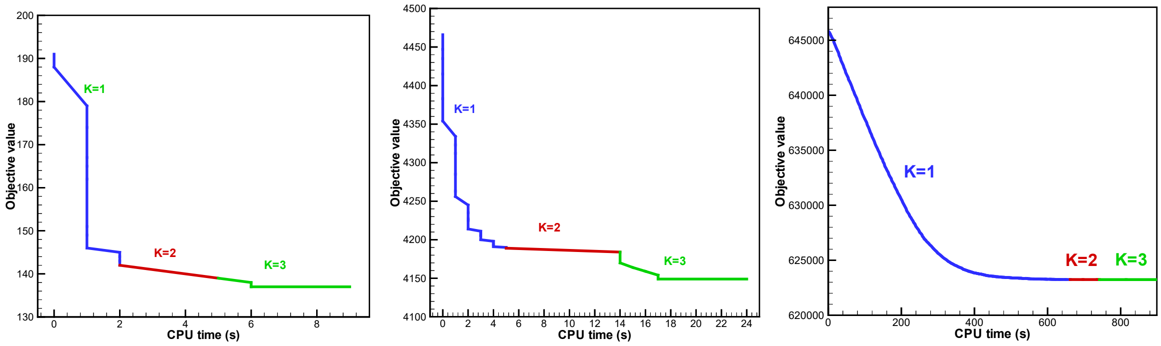
A preliminary analysis had been conducted to fine-tune the parameters $K_{\max}$, $L_{\max}$, and R, for the IPO-ILP-VNS algorithm with respect to different sizes of datasets. Our choice of parameter values may not be the best possible. However, the settings used generally worked well on our test datasets. For medium-sized datasets, including “Lung-cancer”, “Tic-tac-toe”, “Vote”, “Soybean”, “Car”, “Zoo”, and “Mushroom”, we applied the DPP policy in the random shaking of data objects for partial optimization, which helped us find higher quality solutions. For the large-sized datasets “L1”, “L2”, and “L3”, the TPP policy was applied to improve the computational efficiencies. The clustering results/solutions (in terms of the objective function value) are shown in Table 7 and are compared with the results of the traditional *k*-modes algorithm and with four existing results of *k*-modes-based algorithms in the literature (Wu *et al.*, 2007; Khan *et al.*, 2013; Jiang *et al.*, 2016). Column “*VNS parameters*” indicates the numbers of selected data objects in different neighborhoods. Subcolumns “*Best*,” “*AVG*,” “*Deviation (%)*,” and “*Time*” under column “*Ten runs of IPO-ILP-VNS*” represent the best results, the average results, the deviations, and the CPU time used of the IPO-ILP-VNS algorithm respectively. Column “*Ten runs of k-modes*” lists the results of the *k*-modes algorithm with randomly initialized centers. Note that the values shown in Columns “*Jiang et al. (2016)*” and “*Khan et al. (2013)*” are the best objective values re-calculated based on the confusion matrices provided in their publications. It can be observed that the IPO-ILP-VNS algorithm provided new results that are the best for all tested datasets and superior to all other methods, based on a comparison in terms of the cluster qualities evaluated by objective values. Five out of the seven tested datasets were updated with new, best-known results, which are “Lung-cancer”, “Tic-tac-toe”, “Car”, “Zoo”, and “Mushroom”, and the remaining two datasets were also solved with the existing best-known solution. For large-sized datasets “L1”, “L2”, and “L3”, it can be observed that the best/average results of 10 runs of the IPO-ILP-VNS are always better than those of the traditional *k*-modes algorithm. However, the *k*-modes algorithm has still efficiency advantages as it requires less CPU time. Note that both algorithms began their execution from randomly selected clusters.

Table 7 Comparison of existing *k*-modes algorithms

Dataset name	VNS Parameters (Number of selected objects)	Ten runs of IPO-ILP-VNS ($K_{max} = 3$, $L_{max} = 10$)				Ten runs of *k*-modes			Jiang *et al.* (2016)	Jiang *et al.* (2016)	Wu *et al.* (2007)	Khan *et al.* (2013)
		Best	AVG	D.%	T. (s)	Best	AVG	T. (s)	Distance	Entropy		
Zoo	0.1*n*, 0.2*n*, 0.3*n*	***132***	139.1	5.38	14	150	166.9	<1s	141	151	211	146
Soybean	0.1*n*, 0.2*n*, 0.3*n*	**199**	203.6	2.31	7	**199**	210.4	<1s	**199**	**199**	285	**199**
Lung-cancer	0.1*n*, 0.2*n*, 0.3*n*	***481***	483	0.42	19	484	490.7	<1s	486	495	623	498
Car	0.1*n*, 0.15*n*, 0.2*n*	***5134***	5180.3	0.90	74	5200	5319.9	<1s	**--**	--	*5725*	-
Tic-tac-toe	0.1*n*, 0.2*n*, 0.3*n*	**4149**	4176.6	0.67	27	**4149**	4227.5	<1s	-	-	-	-
Vote	0.1*n*, 0.2*n*, 0.3*n*	**1539**	**1539**	0.00	11	**1539**	1540.5	<1s	**1539**	***1539***	*1921*	***1539***
Mushroom	0.1*n*, 0.2*n*, 0.3*n*	***62106***	62439.9	0.54	2374	62107	63102	3.9	70372	**--**	68005	67712
L1	1000, 1500, 2000	***130835***	132971.7	1.63	619	132027	133079.6	2.7	--	--	--	--
L2	1000, 2000, 3000	***311355***	311813.5	0.15	516	311493	313869	7.2	--	**--**	--	--
L3	1000, 2000, 3000	***623238***	623681.9	0.07	937	623333	630787.6	16	--	--	--	--

Note: numbers in boldface indicate previous best values, while numbers in bold italics indicate new best values.

In Fig. 4, we show some detailed processes of the IPO-ILP-VNS algorithm on three selected datasets “Zoo”, “Tic-tac-toe”, and “L3”. The objective value of the incumbent solution and the elapsed time were recorded for every when a better solution is found during the entire process of the IPO-ILP-VNS algorithm. It can be observed that the objective values of the incumbent solution dropped very quickly at the beginning of the algorithm (e.g., in the first neighborhood by $K = 1$), and gradually slowed down after the algorithm moved to search farther/larger neighborhoods. The process stopped after the farthest/largest neighborhood (by $K = 3$) had been searched.



(A) Zoo (B) Tic-tac-toe (C) L3

Fig.4 Convergence trends of the incumbent solutions

7. Summary and conclusions

In this study, we built an integer linear programming (ILP) model for the *k*-modes clustering on categorical datasets based on which optimal results can be directly obtained by an MIP solver. Thus, there is no need to use initial centers/modes. We also developed a heuristic algorithm which conducts iterative partial optimizations of the ILP model under a VNS framework, known as IPO-ILP-VNS, to search for near-optimal results for medium and large-sized categorical datasets. Experiments on synthesized datasets and benchmark datasets showed that the ILP model solved with an MIP solver was able to deliver optimal clustering results for small-sized datasets, and the IPO-ILP-VNS algorithm could identify better clustering results for medium and large-sized datasets, thus outperforming all existing *k*-modes-based algorithms based on comparisons. We updated the best-known results for the benchmark datasets from the UCI repository, and provided new benchmark values for future research studies of categorical clustering. Notably, our numerical experiments also revealed that the conventional *k*-modes algorithm as well as those enhanced *k*-modes algorithms with novel initial methods might not be able to identify the optimal clustering results for some particular datasets. Future works can be on two aspects: (1) to apply the proposed approaches on clustering dataset with mixed data types, and (2) to improve the fuzzy *k*-model clustering algorithm with optimal mathematical programming.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. 71871003 and No. 61376042.

References

[1] Alguwaizani A., Hansen P., Mladenovic N., Ngai E. (2011). Variable neighborhood search for harmonic means clustering. Applied Mathematical Modelling 35, 2688–2694.

[2] Bai L., Liang J., Dang C. (2011). An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. Knowledge-Based Systems 24, 785-795.

[3] Bai L., Liang J., Dang C., Cao F. (2012). A cluster centers initialization method for clustering categorical data. Expert Systems with Applications 39, 8022–8029.

[4] Bai L., Cheng X., Liang J., Shen H., Guo Y., Fast density clustering strategies based on the k-means algorithm. Pattern Recognition 71, 375–386.

[5] Bradley P., Fayyad U. (1998). Refining initial points for k-means clustering. In: Proc. 15th Internat. Conf. on Machine Learning. Morgan Kaufmann, Los Altos, CA.

[6] Barbara, D., Couto, J., & Li, Y. (2002). COOLCAT: An entropy-based algorithm for categorical clustering. In Proceedings of the eleventh international conference on information and knowledge management (pp. 582–589).

[7] Cao F., Liang L., Liang B. (2009). A new initialization method for categorical data clustering. Expert Systems with Applications 36, 10223-10228.

[8] Cao F., Liang J., Li D., Bai L., Dang C. (2012). A dissimilarity measure for the k-Modes clustering algorithm. Knowledge-Based Systems 26, 120-127.

[9] Cao F., Yu L., Huang J. Z., Liang J., 2017. *k*-mw-modes: An algorithm for clustering categorical matrix-object data. Applied Soft Computing 57, 605–614.

[10] Chan E. Y., Ching W. K., Ng M. K., Huang J. Z. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. Pattern recognition 37(5), 943-952.

[11] Chen J., Lin X., Zheng H., Bao X. (2017). A novel cluster center fast determination clustering algorithm. Applied Soft Computing 57, 539–555.

[12] Chen L., Wang S., Wang K., Zhu J. (2016). Soft subspace clustering of categorical data with probabilistic distance. Pattern Recognition 51, 322-332.

[13] Franceschi R. D., Fischetti M., Toth P., (2006). A new ILP-based refinement heuristic for vehicle routing problems. Mathematical Programming 105, 471-499.

[14] Frossyniotis D., Pertselakis M., Stafylopatis A. (2002). A multi-clustering fusion algorithm, in: Proc. of the Second Hellenic Conference on AI, 2002, pp. 225–236.

[15] Gan G., Wu J., Yang Z. (2009). A genetic fuzzy k-Modes algorithm for clustering categorical data. Expert Systems with Applications 36(2), 1615-1620.

[16] Ganti V., Gehrke J., Ramakrishnan R. (1999). CACTUS – clustering categorical data using summaries, in: Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 73–83.

[17] Gilpin S., Nijssen S., Davidson I. (2013). Formalizing hierarchical clustering as integer linear programming. Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue Washington, July 2013, 372-378.

[18] Guha S., Rastogi R., Shim K. (2000). ROCK: a robust clustering algorithm for categorical attributes. Information Systems 25(5), 345–366.

[19] Gupta A., Datta S., Das S., 2018. Fast automatic estimation of the number of clusters from the minimum inter-center distance for k -means clustering. Pattern Recognition Letters 116, 72–79.

[20] Han J., Kamber M. (2006). Data mining: Concepts and Techniques. 2nd ed. California: Morgan Kaufmann.

[21] Hansen P., Mladenović N. (2001). Variable neighborhood search: Principles and applications. European Journal of Operational Research 130(3), 449-467.

[22] Hansen, P., Mladenovic N., Moreno-Perez J. A. (2008). Variable Neighborhood Search. European Journal of Operational Research 191, 593–595.

[23] Hansen, P., Mladenovic N., Moreno-Perez J. A. (2010). Variable Neighbourhood Search: Methods and Applications. Annals of Operations Research 175, 367–407.

[24] He Z. (2006). Farthest-point heuristic based initialization methods for k-modes clustering. arXiv:cs/0610043. https://arxiv.org/abs/cs/0610043

[25] Helber S., Sahling F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem.International Journal of Production Economics 123(2), 247-256.

[26] Huang Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery 2(3), 283-304.

[27] Jain A. K., Dubes R. C. (1988). Algorithms for clustering data. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

[28] Jiang F., Liu G., Du J., Sui Y. (2016). Initialization of K-modes clustering using outlier detection techniques. Information Sciences 332, 167-183.

[29] Kao Y. T., Zahara E., Kao I. W. (2008). A hybridized approach to data clustering. Expert Systems with Applications 34(3), 1754-1762.

[30] Kaufman L. R., Rousseeuw P. (1990). Finding groups in data: An introduction to cluster analysis. John Wiley.

[31] Khan S. S., Ahmad A. (2004). Cluster center initialization algorithm for K-means clustering. Pattern recognition letters 25(11), 1293-1302.

[32] Khan S. S., Kant S. (2007). Computation of initial modes for *k*-modes clustering algorithm using evidence accumulation. In Proceedings of the 20th international joint conference on artificial intelligence (IJCAI), pp. 2784-2789.

[33] Khan S. S., Ahmad A. (2013). Cluster center initialization algorithm for *k*-modes clustering. Expert Systems with Applications 40(18), 7444-7456.

[34] Kim D.-W., Lee K., Lee D., Lee H. K. (2005). A k-populations algorithm for clustering categorical data. Pattern Recognition 38 (7), 1131-1134.

[35] MacQueen J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability 1(14), 281-297. Berkeley: University of California Press.

[36] Mueller M., Kramer S. (2010). Integer linear programming models for constrained clustering. Proceedings of 13th international Conference, DS 2010, Canberra, Australia, October 2010.

[37] Mladenović, N., Hansen, P. (1997). Variable neighborhood search. Computers & operations research 24(11), 1097-1100.

[38] Mladenovic N., Urosevic D., Hanafi S., Ilic A. (2012). A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. European Journal of Operational Research 220, 270–285.

[39] Myhre J. N., Mikalsen K. Ø., Løkse S., Jenssen R., 2018. Robust clustering using a kNN mode seeking ensemble. Pattern Recognition 76, 491–505.

[40] Ng M. K., Wong J. C. (2002). Clustering categorical data sets using tabu search techniques. Pattern Recognition 35(12), 2783-2790.

[41] Parmar D., Wu T., Blackhurst J. (2007). MMR: An algorithm for clustering categorical data using Rough Set Theory.

Data & Knowledge Engineering 63, 879–893

[42] Qin Y., Yu Z. L., Wang C.-D., Gu Z., Li Y., 2018. A Novel clustering method based on hybrid K-nearest-neighbor graph. Pattern Recognition 74, 1–14.

[43] Ralambondrainy H. (1995). A conceptual version of the k-means algorithm. Pattern Recognition Letters 16, 1147–1157.

[44] Saha I., Mukhopadhyay A. (2008). Improved crisp and fuzzy clustering techniques for categorical data. International Journal of Computer Science 35(4), 438-450.

[45] Sun Y., Zhu Q., Chen Z. (2002). An iterative initial-points refinement algorithm for categorical data clustering. Pattern Recognition Letters 23, 875–884.

[46] Wu S., Jiang Q., Huang J. Z. (2007). A new initialization method for clustering categorical data. In Proceedings of the 11th Pacific-Asia Conference on advances in knowledge discovery and data mining PAKDD'07 (pp. 972-980). Berlin, Heidelberg: Springer-Verlag.

[47] Xiao Y., Kaku I., Zhao Q. (2011a). A variable neighborhood search based approach for uncapacitated multilevel lot-sizing problems. Computers & Industrial Engineering 60(2), 218-227.

[48] Xiao Y., Zhang R.-q., Kaku I. (2011b). A new approach of inventory classification based on loss profit. Expert Systems with Applications 38(8), 9382-9391.

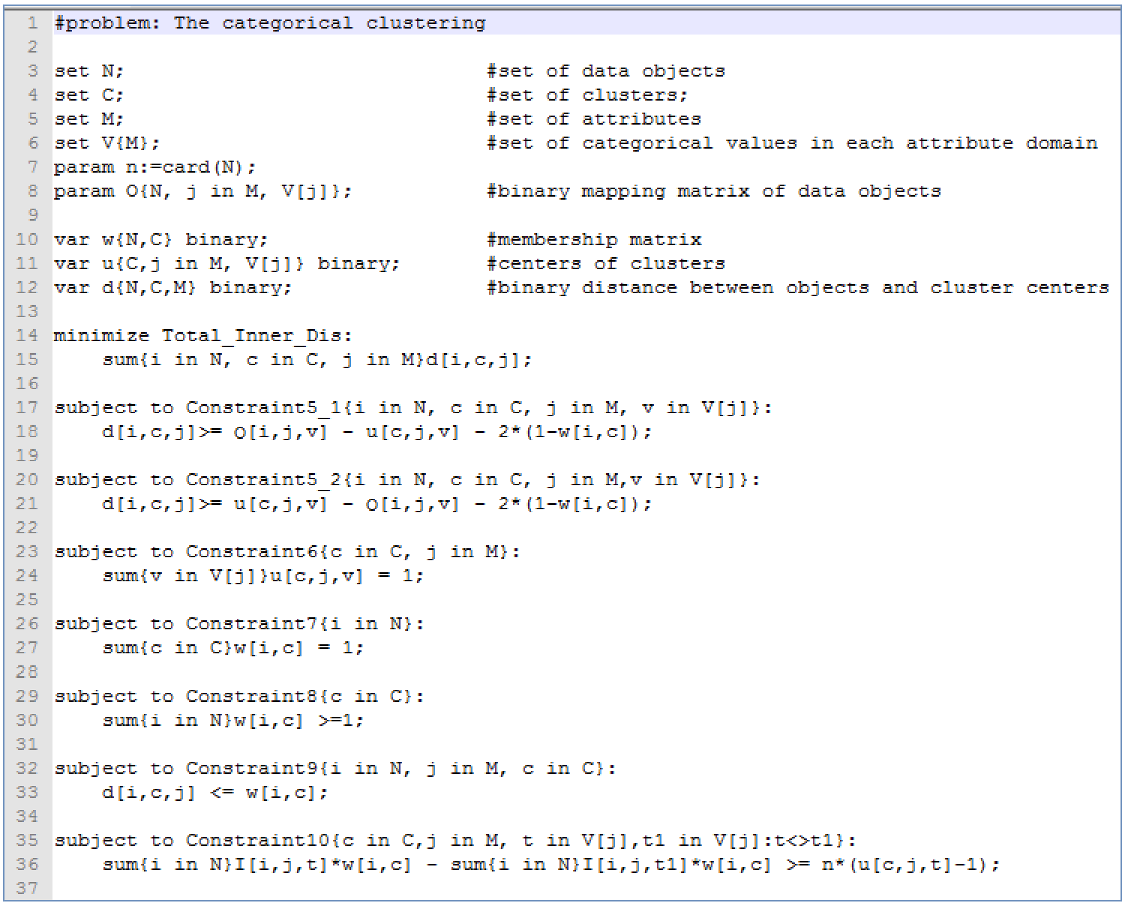
[49] Xiao Y., Zhao Q., Kaku I., Mladenovic N. (2014a). Variable Neighborhood Simulated Annealing Algorithm for Capacitated Vehicle Routing Problems. Engineering Optimization 46(4) , 562-579.

[50] Xiao Y., Zhang R., Zhao Q., Kaku I., Xu Y. (2014). A variable neighborhood search with an effective local search for uncapacitated multilevel lot-sizing problems. European Journal of Operational Research 235(1), 102–114.

[51] Xiao Y., Konak A. (2016). The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. Transportation Research Part E: Logistics and Transportation Review 88, 146-166.

[52] Zhao X., Liang J., Dang C. (2017). Clustering ensemble selection for categorical data based on internal validity indices. Pattern Recognition 69, 150-168.

Appendix



```
#problem: The categorical clustering

set N;                              #set of data objects
set C;                              #set of clusters;
set M;                              #set of attributes
set V{M};                           #set of categorical values in each attribute domain
param n:=card(N);
param O{N, j in M, V[j]};           #binary mapping matrix of data objects

var w{N,C} binary;                  #membership matrix
var u{C,j in M, V[j]} binary;       #centers of clusters
var d{N,C,M} binary;                #binary distance between objects and cluster centers

minimize Total_Inner_Dis:
    sum{i in N, c in C, j in M}d[i,c,j];

subject to Constraint5_1{i in N, c in C, j in M, v in V[j]}:
    d[i,c,j]>= O[i,j,v] - u[c,j,v] - 2*(1-w[i,c]);

subject to Constraint5_2{i in N, c in C, j in M,v in V[j]}:
    d[i,c,j]>= u[c,j,v] - O[i,j,v] - 2*(1-w[i,c]);

subject to Constraint6{c in C, j in M}:
    sum{v in V[j]}u[c,j,v] = 1;

subject to Constraint7{i in N}:
    sum{c in C}w[i,c] = 1;

subject to Constraint8{c in C}:
    sum{i in N}w[i,c] >=1;

subject to Constraint9{i in N, j in M, c in C}:
    d[i,c,j] <= w[i,c];

subject to Constraint10{c in C,j in M, t in V[j],t1 in V[j]:t<>t1}:
    sum{i in N}I[i,j,t]*w[i,c] - sum{i in N}I[i,j,t1]*w[i,c] >= n*(u[c,j,t]-1);

```

Fig. A1 AMPL/CPLEX code used for the ILP model.

Table A1 Data for instance No. 5

Object ID	Attributes 1	Attributes 2	Attributes 3	Attributes 4
1	1	2	2	1
2	4	1	3	4
3	1	3	4	3
4	3	3	4	2
5	4	1	3	3
6	3	4	4	4
7	3	3	2	2
8	4	2	2	1
9	1	1	3	4
10	2	4	1	1
11	3	1	1	3
12	2	2	1	3

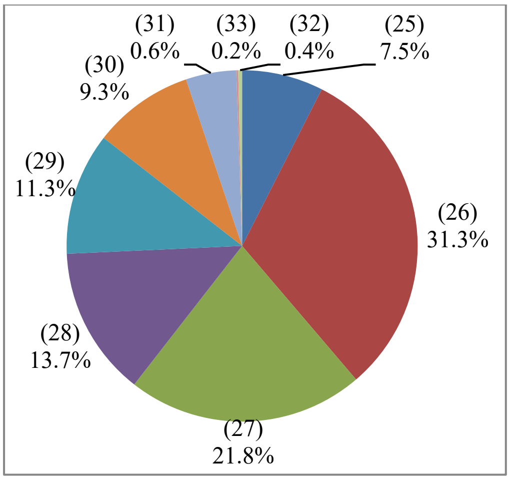
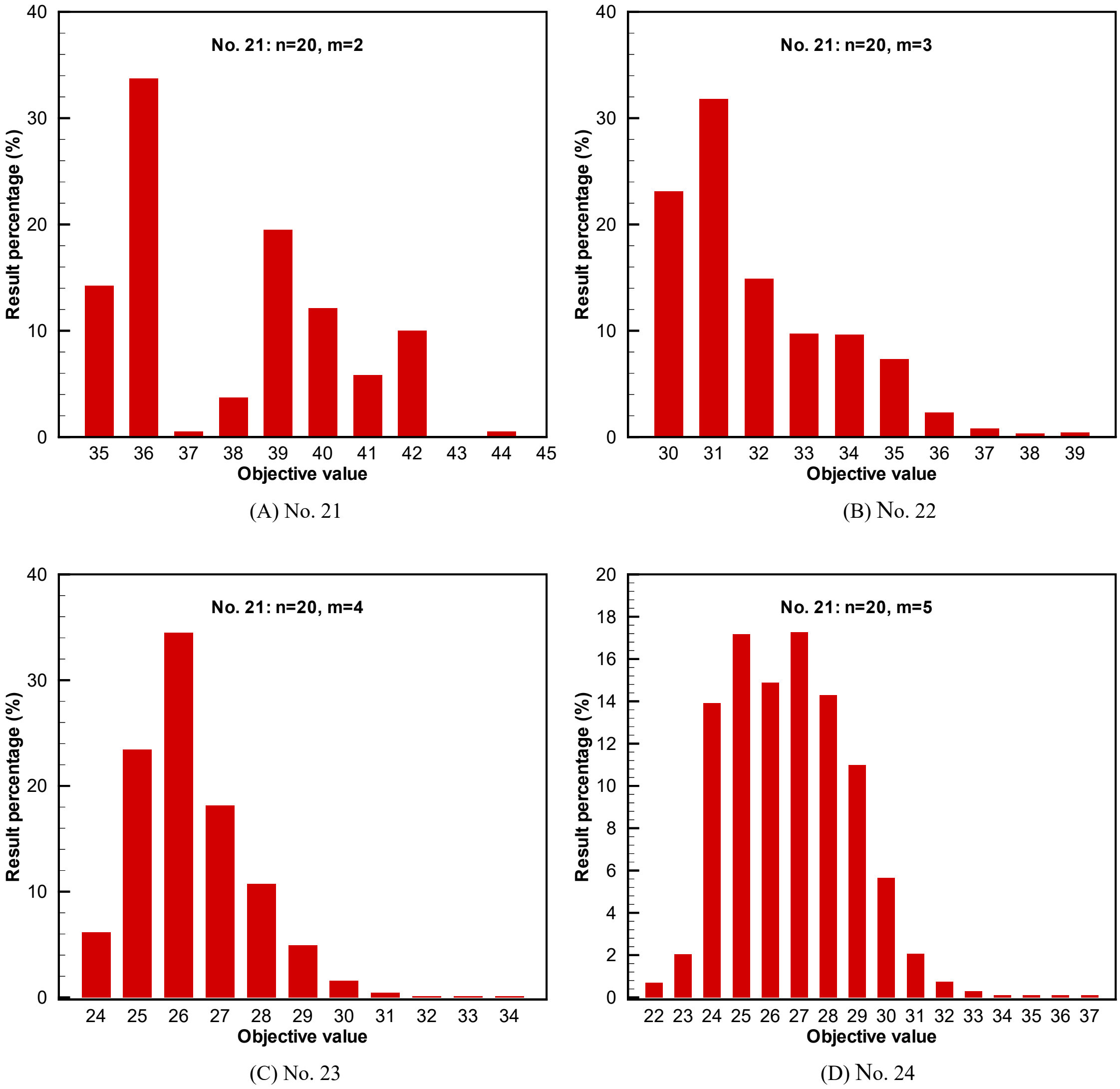


Fig.A2 Solution distribution of *k*-modes algorithm for instance No. 14



(A) No. 21

(B) No. 22

(C) No. 23

(D) No. 24

Fig.A3 Solution distributions of k-modes algorithm for instances No.21, 22, 23, and 24

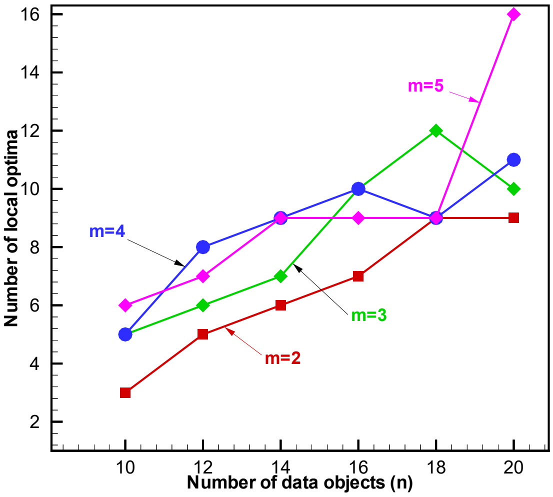


Fig.A4 Trends of local optima against object number

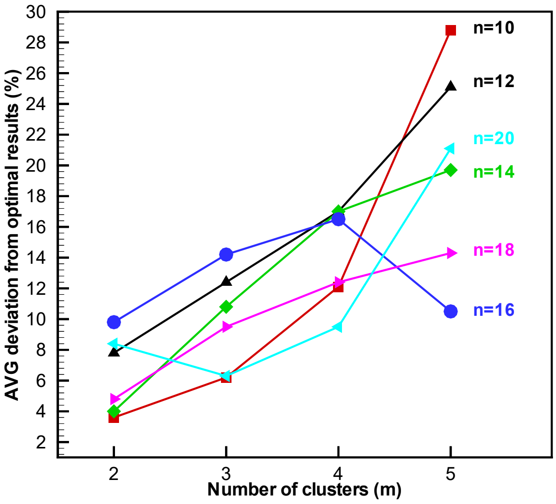


Fig.A5 Trends of average deviation against object number