Received December 12, 2019, accepted December 22, 2019, date of publication December 25, 2019,
date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962191

Size Constrained Clustering With MILP Formulation

WEI TANG, YANG YANG, LANLING ZENG, AND YONGZHAO ZHAN
School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China

Corresponding author: Yang Yang (yyoung@ujs.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402205 and Grant 61672268, in part by the China Postdoctoral Science Foundation under Grant 2015M571688, and in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX18_2258.

ABSTRACT Clustering is one of the essential tools for data mining since it reveals the natural structures of the unlabeled data. Many clustering algorithms have been proposed in the last decades. However, few of them are designed to adapt prior knowledge that is available in many real applications, such as the sizes of clusters. In this paper, we propose a novel iterative clustering algorithm that can impose the constraints on the sizes of clusters. Given an unordered set of cluster size constraints, the proposed method minimizes the mean squared error (MSE) while simultaneously considers the size constraints. Each iteration of the proposed method consists of two steps, namely an assignment step and an update step. In the assignment step, the observations are assigned into clusters under the size constraints. The assignment task is modeled as an integer linear programming (ILP) problem. We prove that part of the constraint matrix of this ILP problem is total unimodular. Therefore, the integer constraints on most of the variables can be omitted so that the problem would become a mixed integer programming (MILP) problem which is much easier to solve. In the update step, new cluster centroids will be updated as the centers of the observations in the corresponding clusters. Experiments on UCI data sets indicate that (1) imposing the size constraints as proposed could improve the clustering performance; (2) compared with the state-of-the-art size constrained clustering methods, the proposed method could efficiently derive better clustering results.

INDEX TERMS Clustering, mean squared error, size constraints, linear program.

I. INTRODUCTION

Clustering is one of the most fundamental unsupervised learning methods that has been employed in many disciplines [1]–[4]. Many clustering algorithms have been proposed [5], such as k-means [6], spectral clustering [7], hierarchical clustering [8], fuzzy c-means [9], clustering ensemble methods [10]–[12], etc. The algorithms are intended to partition observations into k homogeneous and well-separated clusters so that observations in a cluster are similar to one another, yet dissimilar to observations in other clusters. Although traditional clustering algorithms have achieved decent performance in wide applications, the solution naturally found from a set of data by using a fully unsupervised clustering algorithm may not always be close to the one that users seek.

Fortunately, in many real applications such as gene clustering [13], [14], face clustering in videos [15], facility location problem [16], automatic lane detection problem [17] and customer segmentation problem [18], there exists some background knowledge about the data which can be obtained beforehand. Such background knowledge usually reflects itself as the user specified constraints which can be classified into two types [19], namely cluster-level constraints [16], [18], specifying requirements on the clusters, and instance-level constraints [13]–[15], [17], specifying requirements on pairs of observations. In the last decades, many studies have been done in the field of constrained clustering [20]. However, most of them focus on the instance-level constraints. Little attention is drawn to the cluster-level constraints. In this paper, our focus is on the cluster-level constraints, specifically, cluster size constraints.

One of the motivations for introducing the size constraints into clustering is to improve the clustering results. As shown

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

in [21]–[24], introducing the size constraints could prevent the formation of tiny or even empty clusters. Moreover, the studies in [25]–[27] indicate that imposing the actual sizes of the clusters as constraints could improve the clustering performance. The size constraints driven by the needs of improving results are mostly soft, i.e., the size constraints do not have to be strictly satisfied. Another motivation for the research of size constrained clustering is the application requirements. For example, [28] proposes to impose upper bounds on the sizes of clusters so as to maximize the lifetime of the wireless sensor network. In the article clustering problem [29], the articles are clustered under the equality constraints that there are a specific number of articles in each session. The authors in [27] believe that the size constrained clustering in equality form could be applied to job scheduling, where the jobs are assigned to machines with different capacities. [30] claims that the task of resource allocation can be modeled as the size constrained clustering problem, where the sizes of clusters equal the fixed resource capacities. More examples can be found in image searching [31], customer segmentation [18], where the balanced size constraints are imposed on the clustering tasks. The size constraints driven by the application requirements are mostly hard, i.e., the size constraints have to be strictly met.

A common strategy in the field of clustering is to choose k centroids and then minimize the averaged squared distance between the observations and the corresponding cluster centroids. There is a proportional choice to measure the distance aforementioned, i.e., the mean squared error (MSE) [32], which is one of the most popular cost functions used in clustering [24], [33]–[35]. By optimizing the MSE, similar observations are put into the same cluster, yet dissimilar observations are arranged in different clusters. Furthermore, the MSE can be optimized by a mature iterative solution (k-means) that converges rapidly. In this paper, we adopt a similar iterative strategy in our size constrained clustering method. The high efficiency of the iterative strategy makes it possible for us to evaluate the proposed method on large data sets.

In this paper, we propose a novel clustering algorithm that optimizes the MSE under the hard size constraints. Given an unordered set of cluster size constraints as prior knowledge, the proposed method minimizes the MSE while simultaneously ensures that each cluster chooses the optimum size. The proposed algorithm runs in an iterative manner. There are two steps in each iteration, namely an assignment step and an update step. In the assignment step, the assignments between the observations and the clusters are established. The assignment task is formulated as an integer linear programming (ILP) problem [36]. There are two types of variables in the constraints of this ILP problem, which we define as the observation partition decision variables (OPDVs) and the cluster size decision variables (CSDVs). We prove that the integer constraints on the OPDVs can be directly removed. The ILP problem is then simplified as a mixed integer linear programming (MILP) problem [37]. In the update step, the observations in each cluster are averaged to derive the corresponding cluster center. We have conducted experiments to evaluate the proposed method. The data sets involved in our experiments are taken from the UCI machine learning repository [38]. Various external validity indices including the Entropy (ENT) [39], Accuracy (ACC) [40], Fowlkes and Mallows Index (FMI) [41] and Jaccard Index (JCI) [42] are explored. Besides, we evaluate the methods regarding objective function values and efficiency. The results indicate that the proposed method could efficiently leverage the size constraints to improve the clustering performance.

The rest of this paper is organized as follows. The literature review is provided in Section II. Section III covers the details about our proposed size constrained clustering algorithm. In Section IV, the experimental settings and results are given. The discussion is presented in Section V. Finally, Section VI concludes the paper and presents possible directions for further investigation.

II. RELATED WORK

In this section, we briefly review some of the methods that are proposed for clustering with size constraints. Typically, the size constraints can be roughly separated into two categories [43], namely soft size constraints and hard size constraints. Soft size constraints are usually added to the objective functions as regulation terms, so they may not be strictly satisfied. On the other hand, hard size constraints are the conditions that must be met.

A. SOFT SIZE CONSTRAINED CLUSTERING

The soft size constrained clustering methods are more often used to improve the clustering results. For example, the ratio cut [22] and normalized cut [21] introduces direct and indirect equipartition constraints to the objective function of min-cut [44] to prevent the formation of tiny or even empty clusters. [31], [45] present frequency sensitive competitive learning (FSCL) with the multiplicative or additive bias to penalize large clusters so that large clusters are less likely to win observations. [46] proposes a scalable framework to keep the balance of clusters, which applies to a wide range of clustering algorithms. The method first performs clustering on the downsampled data set and then populates the clusters with the remaining data. It is reported that the method is very efficient in practice ($O(kn\log(n))$). In [47], an extension of k-means algorithm was presented. It introduces the size constraints by adding three punishment terms, which includes the overall size divergence cost, oversize cost, and undersize cost. Although this approach achieves certain improvement, it has too many parameters that need to be set, and there is no guidance for the setting of these parameters. [25] presents a size regularized cut (SRcut) by exploring the sizes of the clusters as prior knowledge to guide the clustering process.

As a result, the method improves the clustering performance over traditional methods. [26] proposes to regularize the size constraints with submodular functions. An algorithm based on submodular optimization techniques is presented to solve the size constrained clustering problem. In [23], the authors exploit the exclusive lasso to exert the balanced size constraints, and they apply the idea into the min-cut and k-means algorithm. Their experiments indicate improved results. Although there has been significant progress in the soft size constrained clustering algorithms, they may not cater to a large number of real applications since they only treat constraints as guidance rather than requirements that must be met. In this paper, we mainly focus on clustering with hard size constraints.

B. HARD SIZE CONSTRAINED CLUSTERING

Despite the great application requirements, few studies have been done in the field of hard size constrained clustering. [33] proposes an iterative method for clustering with hard balanced size constraints. The method transforms the k-means assignment step into a balanced assignment problem that can be solved by the Hungarian algorithm [48]. It works fine when the number of observations n is divisible by the number of clusters k. However, when n is not divisible by k, it has trouble in deciding the optimal size for each cluster. [49] imposes size constraints to an adapted neural gas algorithm. The method ensures that the constraints on the cluster sizes are satisfied. However, the greedy strategy cannot guarantee the optimality of the clustering algorithm. A variant of fuzzy c-means (FCM) clustering algorithm is proposed in [30]. The size constraints are integrated into the objective function via Lagrange multipliers. The experimental results show that it outperforms traditional clustering algorithms. The authors in [50] propose a deterministic clustering approach based on the deterministic annealing (DA) algorithm to address the capacitated resource allocation problem with several forms of size constraints. [27] proposed a method for clustering with hard size constraints. A clustering algorithm without considering any size constraints is applied to get the initial partition, and the final result is derived by finding a constraint-satisfying partition that maximizes its agreement with the initial partition. The method is very efficient in practice. However, it fails to consider the similarity between the observations during the reassigning process. [29] presented a hard size constrained clustering methods based on ILP. The method works if we assume that the correspondences between the initial centers and the cluster sizes are known. Nevertheless, the assumption is rarely practical in real applications. There are also studies which put hard lower bounds [24], [51] and upper bounds [52] on the sizes of clusters. In Section V, we show that the proposed method could also be adapted to facilitate the inequality constraints.

TABLE 1. Meanings of the notations.

Notation	Meaning
o	set of observations
s	set of cluster size constraints
c	set of cluster centers
k	number of clusters
n	number of observations
f	clusters derived from the clustering algorithm
p	partition matrix consists of observation partition decision variables
q	auxiliary matrix consists of cluster size decision variables
A	constraint matrix on p
α	number of integer constraints

III. SIZE CONSTRAINED CLUSTERING

A. NOTATIONS

The key notations involved in our method are shown in Table 1.

B. PROBLEM FOMULATION

In this paper, our objective is to minimize the MSE under the given size constraints of clusters. Given a data set $o = \{o_1, o_2, \ldots, o_n\}$, where o_i denotes the i-th observation, and a set of cluster size constraints $s = \{s_1, s_2, \ldots, s_k\}$, $\sum_{j=1}^{k} s_j = n$, where s_j denotes the j-th size constraint and it is integral. Our purpose is to partition the n observations into k clusters $f_1, f_2, \ldots, f_k$ ($c_1, c_2, \ldots, c_k$ are the cluster centers), such that the set of cluster sizes $\{|f_1|, |f_2|, \ldots, |f_k|\}$ equals the set of size constraints s.

$$\begin{aligned} &Minimize\ E = (1/n)\sum_{j=1}^{k}\sum_{o_i\in f_j} \left\|o_i - c_j\right\|^2 \\ &s.t.\ \{|f_1|, |f_2|, \ldots, |f_k|\} = s \end{aligned} \tag{1}$$

where $\left\|o_i - c_j\right\|^2$ denotes the squared Euclidean distance between the i-th observation and the j-th cluster center.

Notice that the size of the j-th cluster $|f_j|$ does not have to be the j-th size constraint s_j. Because at the beginning of the clustering process, the correspondences between the cluster sizes and the size constraints are unclear. We can not specify which cluster is the j-th cluster f_j to have s_j observations. The number of possible correspondences between the cluster sizes and the size constraints is $k!$, and we want the algorithm to automatically choose the optimum one from the $k!$ possible correspondences.

Let p be the partition matrix of size $n \times k$, where each row of p represents an observation, and each column represents a cluster. $p_{i,j} = 1$ indicates that observation x_i belongs to cluster j, while $p_{i,j} = 0$ means otherwise. It is clear that summing each row of p equals 1 because each observation can only be assigned to one cluster, i.e., $\sum_{j=1}^{k} p_{i,j} = 1, i = 1, 2, \ldots, n$. On the other hand, summing each column of p equals the size of the corresponding cluster $|f_j|$. Thus we reformulate the problem as:

$$\begin{aligned} & \mathit{Minimize}\ E(c,p) = (1/n)\sum_{j=1}^{k}\sum_{i=1}^{n} p_{i,j}\left\|o_i - c_j\right\|^2 \\ & s.t.\ \sum_{i=1}^{n} p_{i,j} = |f_j|, \quad j = 1, 2, \ldots, k \\ & \quad \sum_{j=1}^{k} p_{i,j} = 1, \quad i = 1, 2, \ldots, n \\ & \quad \{|f_1|, |f_2|, \ldots, |f_k|\} = s \\ & \quad p_{i,j} \in \{0, 1\}, \quad i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, k \end{aligned} \tag{2}$$

Let q be an auxiliary matrix of size $k \times k$, where $q_{j,l} = 1$ indicates that j-th cluster chooses l-th size constraint, $q_{j,l} = 0$ means otherwise. It is obvious that summing each row or column of q equals 1 as the correspondences between cluster sizes and size constraints are one-to-one. The problem can be further reformulated as:

$$\begin{aligned} & \mathit{Minimize}\ E(c,p) = (1/n)\sum_{j=1}^{k}\sum_{i=1}^{n} p_{i,j}\left\|o_i - c_j\right\|^2 \\ & s.t.\ \sum_{i=1}^{n} p_{i,j} = \sum_{l=1}^{k} q_{j,l}s_l, \quad j = 1, 2, \ldots, k \\ & \quad \sum_{j=1}^{k} p_{i,j} = 1, \quad i = 1, 2, \ldots, n \\ & \quad \sum_{j=1}^{k} q_{j,l} = 1, \quad l = 1, 2, \ldots, k \\ & \quad \sum_{l=1}^{k} q_{j,l} = 1, \quad j = 1, 2, \ldots, k \\ & \quad p_{i,j} \in \{0, 1\}, \quad i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, k \\ & \quad q_{j,l} \in \{0, 1\}, \quad j = 1, 2, \ldots, k,\ l = 1, 2, \ldots, k \end{aligned} \tag{3}$$

To simplify the description in Section III-D, we define three kinds of variables for the above problem, the cluster centers c, the variables in the partition matrix p which we call observation partition decision variables (OPDVs), and the variables in the auxiliary matrix q which we call cluster size decision variables (CSDVs). The OPDVs are used to indicate the relations between the observations and the clusters. The CSDVs have no explicit impact on the objective function E. They are used to indicate the correspondences between the cluster sizes and the size constraints.

It should be noted that once the sizes of the $k-1$ clusters are set, the sizes of all k clusters can be determined, so one of the cluster size constraints in Equation (3) is redundant. However, we have found that the redundant size constraint has no significant impacts on the results. For ease of description and comprehension, we intend to keep the redundant constraint for the rest of this paper.

C. PROPOSED SOLUTION

1) ASSIGNMENT STEP

In the assignment step, we try to solve Equation (3) with respect to p while holding c fixed, i.e., we try to assign the observations to the cluster centers so as to optimize the MSE under the given set of size constraints. The problem here is an ILP problem and the integer constraints on the decision variables make it difficult to solve. A typical solution to the ILP problem is to repeatedly solve the LP relaxations with the simplex algorithm in a branch-and-bound way [53]. In the worst case, the simplex algorithm needs to be executed exponential times which is computationally expensive.

If the constraint matrix of the ILP is totally unimodular, then the integer constraints can be removed [54]. In our case, the constraint matrix on all the decision variables is not totally unimodular. However, as indicated by Theorem 1 in Section III-D, the constraint matrix on the OPDVs is totally unimodular, so most of the integer constraints can be removed from our ILP problem. Specifically, the integer constraints on the $n \times k$ OPDVs can be removed, which leaves us only the integer constraints on the $k \times k$ CSDVs. The ILP problem would then become an MILP problem that can be solved in much less running time.

2) UPDATE STEP

In the update step, we try to solve Equation (3) with respect to c while holding p fixed. Actually, once the observations are assigned, new centroids should be updated so that the MSE is minimized. Since p is fixed, Equation (3) can then be relaxed to an unconstrained optimization problem as shown below.

$$\begin{aligned} \mathit{Minimize}\ E(c,p) &= (1/n)\sum_{j=1}^{k}\sum_{i=1}^{n} p_{i,j}\left\|o_i - c_j\right\|^2 \\ &= (1/n)\sum_{j=1}^{k}\sum_{i=1}^{n} p_{i,j}(o_i - c_j)(o_i - c_j)^{\mathrm{T}} \end{aligned} \tag{4}$$

Obviously, the optimal E can be achieved when

$$\begin{aligned} \partial(E(c,p))/\partial(c_j) = 0 &\Rightarrow \sum_{i=1}^{n} p_{i,j}c_j - \sum_{i=1}^{n} p_{i,j}o_i = 0 \\ &\Rightarrow c_j = (1/\sum_{i=1}^{n} p_{i,j})\sum_{i=1}^{n} p_{i,j}o_i \end{aligned} \tag{5}$$

With the description above, the proposed size constrained clustering can be detailed as Algorithm 1.

The proposed algorithm is guaranteed to converge, but not always to the global optimum due to the non-convexity of the objective function. In each iteration, the value of the objective function monotonically decreases. Assuming that $p^{(t)}$ and $c^{(t)}$ are the OPDVs and centroids at the end of the t-th iteration respectively. In the assignment step of the $(t+1)$-th

TABLE 2. A brief summary of experimental data sets.

Data	Size	Dimension	Size constraints (ground truth)
Iris	150	4	50, 50, 50
Wine	178	14	59, 71, 48
Data User Modeling	285	5	24, 88, 83, 63
Movement Libras	360	91	24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24
Ecoli	366	4	143, 77, 2, 2, 35, 20, 5, 52
Wine Quality Red	1599	11	10, 53, 681, 638, 199, 18
EMPGA1	98886	8	9830, 10000, 9611, 9756, 10000, 9964, 10000, 9725, 10000, 10000
EMPGA2	198816	8	19830, 20000, 19611, 19756, 20000, 19964, 20000, 19725, 20000, 20000

```
Algorithm 1 Clustering With Size Constraints
Input: Data set $o = \{o_1, o_2, ..., o_n\}$;
    Size constraints $s = \{s_1, s_2, ..., s_k\}$;
Output:
    OPDVs $p$; CSDVs $q$;
    Cluster centers $c = \{c_1, c_2, ..., c_k\}$;
Initialize centroids $c^{(0)} = \{c_1^{(0)}, c_2^{(0)}, ..., c_k^{(0)}\}$ with k-means++ algorithm [55];
Initialize $i = 0$;
repeat
    $i = i + 1$;
    Assignment step:
        solve the MILP problem stated in Section III-C.1 for $p^{(i)}$ and $q^{(i)}$;
    Update step:
        update the centroids as $c^{(i)}$ according to Equation (5);
until The centroids no longer change
return $p^{(i)}, q^{(i)}, c^{(i)}$;
```

iteration, we optimize Equation (3) with respect to p while holding c fixed, thus we have $E(c^{(t)}, p^{(t+1)}) \leq E(c^{(t)}, p^{(t)})$. In the update step, we optimize Equation (3) with respect to c while holding p fixed, thus, there must be $E(c^{(t+1)}, p^{(t+1)}) \leq E(c^{(t)}, p^{(t+1)})$.

D. EFFICIENCY ANALYSIS

Typically, the ILP problem in standard form can be formulated as follows.

$$\begin{aligned} &\textit{Minimize } \lambda x^{\mathrm{T}}\\ &s.t.\ Mx^{\mathrm{T}} = b\\ &\qquad x \geq 0\\ &\qquad x \in \mathbb{Z}^n \end{aligned} \tag{6}$$

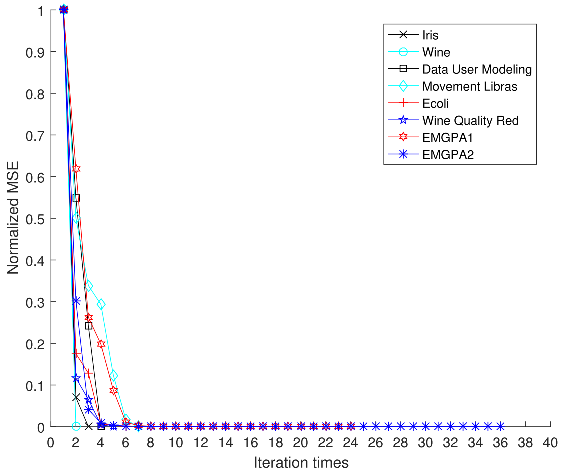


FIGURE 1. The convergence process of the proposed algorithm.

An interesting property of the ILP problem is that if the constraint matrix M is totally unimodular (a matrix is totally unimodular if and only if the determinant of every square submatrix is 0,1,or -1) and the vector b is integral, the integer constraints can be removed so that the ILP problem can be relaxed to an LP problem that still guarantees the integral solution [56]. Although the constraint matrix in our case is not totally unimodular, it is special in that its submatrix on OPDVs is totally unimodular according to Theorem 1 (the proof can be found below). Therefore, we could remove the integer constraints on the OPDVs. It will greatly reduce the complexity during the course of branch-and-bound.

Before the proof of Theorem 1, we first introduce Lemma 1 and Lemma 2. The two Lemmas can also be found in [54].

Lemma 1: The constraint matrix M remains totally unimodular if multiplying a column (row) with -1.

For example, if

$$M_1 = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

TABLE 3. The means and standard deviations of the MSE and running time.

Data	Algorithm	MSE	MSE S.D.	Time	Time S.D.
Iris	KM	9.3645 (10^{11})	3.2478 (10^{9})	0.0021	1.3853 (10^{-4})
	SCK1	2.1751 (10^{12})	1.7329 (10^{12})	0.0205	0.0012
	SCK2	9.3505 (10^{11})	1.0037 (10^{8})	0.0776	0.0056
	MILP-KM	9.3505 (10^{11})	1.0037 (10^{8})	0.0465	0.0119
Wine	KM	1.2395 (10^{9})	6.8694 (10^{5})	0.0023	4.4505 (10^{-4})
	SCK1	1.6023 (10^{9})	1.6177 (10^{8})	0.0224	0.0025
	SCK2	1.2937 (10^{9})	2.1483 (10^{7})	0.2306	0.0922
	MILP-KM	1.2508 (10^{9})	8.5844 (10^{4})	0.2152	0.1988
Data User Modeling	KM	0.1613	0.0028	0.0022	3.5834 (10^{-4})
	SCK1	0.2339	0.0246	0.0305	0.0046
	SCK2	0.1706	0.0043	3.3566	1.8176
	MILP-KM	0.1665	0.0043	0.2537	0.1004
Movement Libras	KM	1.6948	0.1116	0.0028	4.3360 (10^{-4})
	SCK1	11.0019	3.6047	0.0463	0.0039
	SCK2	1.7209	0.0664	4.7269	1.3667
	MILP-KM	1.7209	0.0664	0.4407	0.1387
Ecoli	KM	0.0445	0.0012	0.0043	0.0041
	SCK1	0.1998	0.0621	0.0650	0.0367
	SCK2	0.0650	0.0077	9.8950	5.5808
	MILP-KM	0.0525	0.0015	0.5029	0.1599
Wine Quality Red	KM	0.1002	0.0020	0.0041	0.0011
	SCK1	0.2202	0.0466	0.1815	0.1426
	SCK2	0.1226	0.0032	6.222 (10^{3})	4.5337 (10^{3})
	MILP-KM	0.1191	0.0012	5.2584	1.9082
EMPGA1	KM	3.4618 (10^{6})	9.3382 (10^{4})	0.0330	1.5872
	SCK1	1.9447 (10^{7})	1.3817 (10^{6})	2.4501 (10^{3})	669.8757
	SCK2	-	-	-	-
	MILP-KM	5.2408 (10^{6})	505.5101	2.0697 (10^{3})	720.6405
EMPGA2	KM	4.1569 (10^{6})	3.1058 (10^{4})	0.0894	0.0878
	SCK1	-	-	-	-
	SCK2	-	-	-	-
	MILP-KM	6.6271 (10^{6})	61.9008	1.1163 (10^{4})	4.6754 (10^{3})

is totally unimodular, then we multiply the first row with −1, we can get

$$M_1' = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

According to Lemma 1, the total unimodularity is preserved.

Lemma 2: The constraint matrix M is totally unimodular if it has at most two non-zero entries being ±1 in each column (row), and, for every column (row) with two non-zero entries, the sum of the column (row) is 0.

For instance, given a matrix

$$M_2 = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}$$

According to Lemma 2, the matrix is totally unimodular since it only contains two non-zero entries with ±1 in each column, and the sum of each column equals 0.

Theorem 1: The constraint matrix on the OPDVs in the ILP problem described in Section III-C.1 is totally unimodular.

TABLE 4. Games-Howell test and student's *t*-test results on the MSE and running time.

Data	Group	$\text{MSE}_{p-value}$	$\text{Time}_{p-value}$
Iris	MILP-KM vs. KM	0.5487	$4.1993\ (10^{-6})$
	MILP-KM vs. SCK1	0.1783	0.0003
	MILP-KM vs. SCK2	1.0000	$2.5864\ (10^{-5})$
Wine	MILP-KM vs. KM	$6.5220\ (10^{-12})$	0.0335
	MILP-KM vs. SCK1	0.0003	0.0543
	MILP-KM vs. SCK2	0.0006	0.9960
Data User Modeling	MILP-KM vs. KM	0.0849	0.0001
	MILP-KM vs. SCK1	$3.9368\ (10^{-5})$	0.0003
	MILP-KM vs. SCK2	0.0710	0.0020
Movement Libras	MILP-KM vs. KM	0.9187	$1.7518\ (10^{-5})$
	MILP-KM vs. SCK1	$9.1926\ (10^{-5})$	$4.1085\ (10^{-5})$
	MILP-KM vs. SCK2	1.0000	$1.6753\ (10^{-5})$
Ecoli	MILP-KM vs. KM	$2.0074\ (10^{-9})$	$1.9293\ (10^{-5})$
	MILP-KM vs. SCK1	0.0002	$3.7518\ (10^{-5})$
	MILP-KM vs. SCK2	0.0026	0.0022
Wine Quality Red	MILP-KM vs. KM	$1.4305\ (10^{-12})$	$5.3795\ (10^{-5})$
	MILP-KM vs. SCK1	0.0003	$6.7898\ (10^{-5})$
	MILP-KM vs. SCK2	0.0346	0.0084
EMPGA1	MILP-KM vs. KM	$2.5414\ (10^{-8})$	$2.1222\ (10^{-5})$
	MILP-KM vs. SCK1	$2.5491\ (10^{-8})$	0.4555
	MILP-KM vs. SCK2	-	-
EMPGA2	MILP-KM vs. KM	$1.2630\ (10^{-18})$	$3.5035\ (10^{-5})$
	MILP-KM vs. SCK1	-	-
	MILP-KM vs. SCK2	-	-

Proof: Putting all the OPDVs in a vector y, we have:

$$y = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{n,k} \end{pmatrix}$$

then the constraint matrix on the OPDVs can be derived as:

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

where A_1 consists of the coefficients of the first constraint set $\sum_{i=1}^{n} p_{i,j} = b = \sum_{l=1}^{k} q_{j,l}s_l,\ j = 1, 2, ..., k$ and A_2 contains the coefficients of the second constraints set $\sum_{j=1}^{k} p_{i,j} = 1,\ i = 1, 2, ..., n$.

A concrete form for A_1 and A_2 can be formulated as following.

$$A_1 = \begin{pmatrix} I_{k\times k} & I_{k\times k} & \cdots & I_{k\times k} \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 1 & 1 & \cdots & 1 \end{pmatrix}$$

where, $I_{k\times k}$ is the identity matrix of size $k \times k$.

According to Lemma 1, we multiply every row in A_2 with -1 to get A_2'. If $A' = \begin{pmatrix} A_1 \\ A_2' \end{pmatrix}$ is totally unimodular, so does A.

According to Lemma 2, it is clear that A' is totally unimodular as every column has only two non-zero elements being ± 1, and the sum of of each column equals 0.

Therefore, the constraint matrix on the OPDVs is totally unimodular.

Due to the total unimodularity of the constraint matrix on the OPDVs in the ILP problem described at the beginning of Section III-C.1, we only need to keep the integer constraints on the CSDVs thus leading us to the MILP problem described in Section III-C.1. Therefore, the MILP problem is equivalent to the ILP problem. Solving the MILP problem still results in integral solutions on both OPDVs and CSDVs.

The ILP and MILP problems can be addressed by repeatedly solving the LP relaxations with the simplex algorithm in a branch-and-bound way. In the worst case, it needs to solve $O(2^{\alpha})$ LP problems, where α is the number of

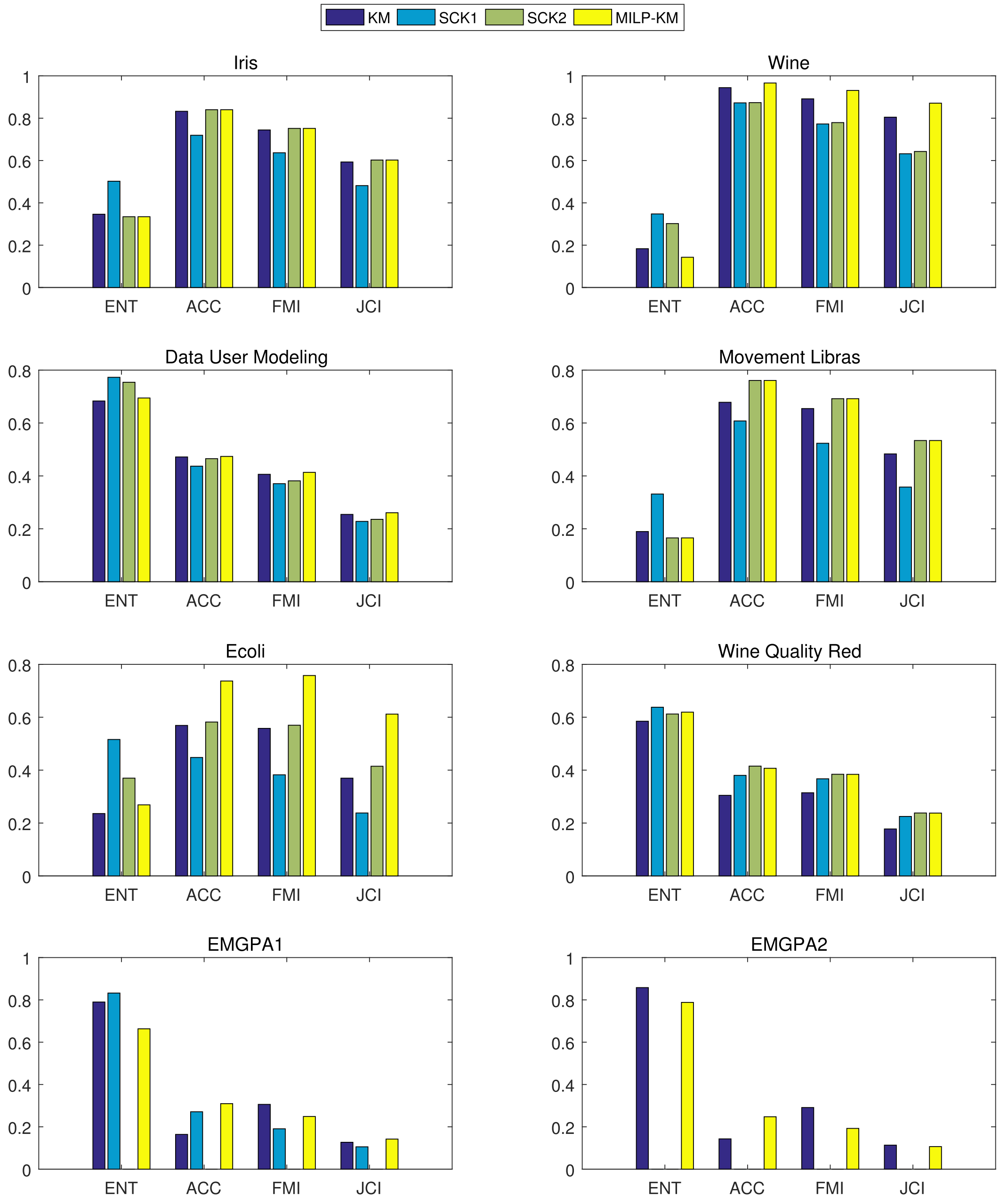


FIGURE 2. The external indexes of the *k*-means based algorithms.

integer constraints. Accordingly, to address the ILP problem described at the beginning of Section III-C.1, we need to solve $O(2^{n\times k+k^2})$ LP problems, where n is the number of observations and k is the number of clusters. According to Theorem 1, we can remove the integer constraints on the $n \times k$ OPDVs, so that we only need to solve $O(2^{k^2})$ LP problems. Under certain circumstances when $n \gg k$, the strategy would make a great contribution to decreasing the total time complexity.

IV. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

In this section, we present the experiments conducted on UCI machine learning data sets to evaluate the performance of the proposed algorithm. Table 2 shows the details of the data sets that are used in our experiments.

We integrate the size constraints into the k-means algorithm as described in Section III. We compare the proposed algorithm with the k-means algorithm and other size constrained k-means algorithms which include the algorithm presented in [27] and the algorithm in [33]. For simplicity, we refer to the k-means algorithm as KM, the algorithm in [27] as SCK1, the algorithm in [33] as SCK2, and the proposed algorithm as MILP-KM. To further study the performance of the proposed algorithm, all the k-means based algorithms mentioned above are adapted to the normalized cut based algorithms as the normalized cut clustering

TABLE 5. The means and standard deviations of the NCUT and running time.

Data	Algorithm	NCUT	NCUT S.D.	Time	Time S.D.
Iris	NC	1.6060	0.0084	0.0168	0.0017
	SCN1	1.7812	0.1775	0.0355	0.0022
	SCN2	1.5492	$2.0935\ (10^{-16})$	0.4848	0.0386
	MILP-NC	1.5492	0	0.1034	0.0356
Wine	NC	0.7737	0.1318	0.0149	0.0013
	SCN1	1.3863	0.1792	0.0351	0.0025
	SCN2	0.7743	0.0519	0.4385	0.1182
	MILP-NC	0.6796	0.0035	0.0669	0.0137
Data User Modeling	NC	$2.8308\ (10^{-11})$	$2.1168\ (10^{-11})$	0.0157	0.0014
	SCN1	0.1110	0.0074	0.0551	0.0065
	SCN2	$5.4973\ (10^{-4})$	0.0011	5.2673	0.4565
	MILP-NC	0.0336	0.0253	1.4223	0.4104
Movement Libras	NC	0.0124	0.0109	0.0381	0.0025
	SCN1	5.4506	0.2402	0.0976	0.0060
	SCN2	0.7701	0.1248	9.6795	2.8257
	MILP-NC	0.7414	0.0982	0.3705	0.1082
Ecoli	NC	6.1122	0.2291	0.0317	0.0051
	SCN1	6.9205	0.0033	0.0722	0.0087
	SCN2	5.9505	0.0810	11.6999	3.9234
	MILP-NC	5.7669	0.0338	0.8038	0.4983
Wine Quality Red	NC	$1.7515\ (10^{-4})$	$2.0203\ (10^{-20})$	0.4847	0.1019
	SCN1	4.8672	0.0043	0.9015	0.2504
	SCN2	0.8043	0.4263	$1.0201\ (10^{3})$	$4.4459\ (10^{3})$
	MILP-NC	2.6064	0.0565	18.1378	6.4826

algorithm can be seen as applying the k-means algorithm on the data set with reduced dimensions. The normalized graph Laplacian matrix is implemented as a fully connected graph according to the research in [21]. For simplicity, we refer to the normalized cut algorithm as NC, the algorithm adapted from [27] as SCN1, the algorithm adapted from [33] as SCN2, and the proposed algorithm as MILP-NC.

For the evaluation criterion, we consider four external indexes, including the Clustering Accuracy (ACC) [40], Entropy (ENT) [39], Fowlkes and Mallows Index (FMI) [41] and Jaccard Index (JCI) [42]. Apart from these measures, the MSE for the algorithms based on the k-means, the NCUT for the algorithms based on the normalized cut and running time (measured in seconds) are explored. All these measures are recorded and averaged over 10 runs. Besides, the statistical tests are applied to further validate the results on the MSE, NCUT, and running time. For all the data sets except EMPGA2, we apply the Games-Howell test [57], as there are more than two methods to compare. For EMPGA2, both SCK1 and SCK2 can not derive a result within an acceptable time, which leaves us with only two methods to compare, so we make use of the Student's t-test [57].

All the algorithms involved in this paper are implemented in MATLAB, which run on an Intel i7-7700HQ 2.8 GHz processor with 16 GB memory. We explore the build-in MILP solver of MATLAB with default parameters except for "RootLPMaxIterations", "LPMaxIterations", and "MaxTime", which are set as 1000000, 100000, and 18000, respectively. In addition, all clustering algorithms get the initial centroids by k-means++ algorithm [55]. They share the same stopping criterion $||c^{(t+1)} - c^{(t)}||_2 < 0.0001$, i.e., the centroids barely change between adjacent iterations. The code and data sets can be downloaded from https://github.com/IGGIUJS/SizeConstrainedClustering.

Notice that we do not conduct the experiments on EMPGA1 and EMPGA2 for the normalized cut based algorithms, because these methods need to construct a normalized graph Laplacian matrix that enormously beyond the memory capacity of our experimental equipment (16 GB RAM).

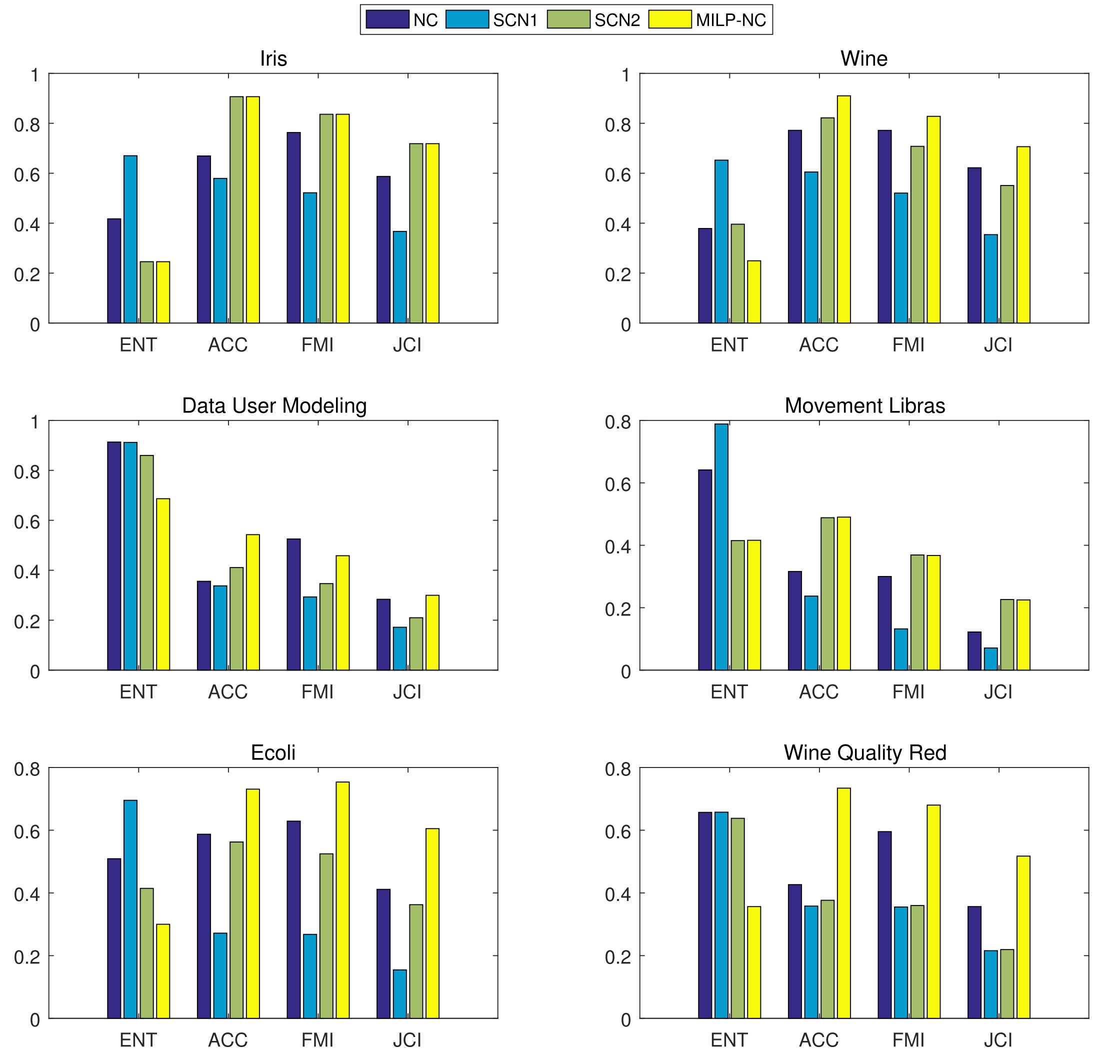


FIGURE 3. The external indexes of the normalized cut based algorithms.

B. CONVERGENCE

The proposed method is guaranteed to converge according to the analysis in Section III-C.2. The process of convergence can be found in Fig. 1. Since the MSE varies greatly across the data sets, we normalized it to the range [0, 1]. We can observe that for the eight data sets involved, our algorithm converges. For EMPGA1 and EMPGA2, the proposed algorithm takes dozens of iterations, while for other data sets, it converges in less than ten iterations.

C. COMPARISON AMONG K-MEANS BASED ALGORITHMS

In this section, we report the performance comparison among KM, SCK1, SCK2 and MILP-KM in terms of the ENT, ACC, FMI, JCI, MSE and running time.

It can be observed from Table 3 that mostly KM achieves the best MSE and efficiency. It is natural because KM optimizes the MSE without any constraints. As far as the size constrained k-means algorithms are concerned, MILP-KM outperforms SCK1 and SCK2 in the resultant MSE. The statistical test results shown in Table 4 indicate that the differences on the MSE are mostly significant ($p <= 0.05$). In addition, MILP-KM is significantly faster than SCK2 on most of the data sets. Although MILP-KM takes more time than SCK1 on small data sets, it outputs results with far better MSE. Moreover, for large data sets, such as EMPGA1 and EMPGA2, MILP-KM is even faster than SCK1. Especially for EMPGA2, SCK1 is no longer able to produce a result within an acceptable time.

From the external indexes shown in Fig. 2, we can see that mostly MILP-KM outperforms KM, SCK1, and SCK2. This indicates that incorporating the size constraints as proposed could better improve the performance on the external indexes for the k-means algorithm. Notice that the size constrained method SCK1 performs even worse than KM, this is because that SCK1 is a heuristic method with strong randomness when adjusting the results given by an initial clustering to adapt the size constraints.

In most cases, MILP-KM outperforms SCK2 on the four external indexes. However, there are cases when SCK2 performs better, such as the external indexes on Wine Quality

TABLE 6. Games-Howell test results on the NCUT and running time.

Data	Group	NCUT$_{p-value}$	Time$_{p-value}$
Iris	MILP-NC vs. NC	2.4184 (10^{-8})	0.0001
	MILP-NC vs. SCN1	0.0112	0.0009
	MILP-NC vs. SCN2	0.0030	1.0739 (10^{-12})
Wine	MILP-NC vs. NC	0.1797	3.3248 (10^{-6})
	MILP-NC vs. SCN1	2.6994 (10^{-6})	0.0002
	MILP-NC vs. SCN2	0.0012	1.5913 (10^{-5})
Data User Modeling	MILP-NC vs. NC	0.0103	8.8741 (10^{-6})
	MILP-NC vs. SCN1	1.0951 (10^{-5})	1.1221 (10^{-5})
	MILP-NC vs. SCN2	0.0113	1.7945 (10^{-12})
Movement Libras	MILP-NC vs. NC	8.0778 (10^{-9})	2.1957 (10^{-5})
	MILP-NC vs. SCN1	1.3309 (10^{-12})	0.0001
	MILP-NC vs. SCN2	0.9391	1.2147 (10^{-5})
Ecoli	MILP-NC vs. NC	0.0045	0.0038
	MILP-NC vs. SCN1	1.6920 (10^{-12})	0.0055
	MILP-NC vs. SCN2	0.0001	4.3950 (10^{-5})
Wine Quality Red	MILP-NC vs. NC	1.6637 (10^{-12})	5.8717 (10^{-5})
	MILP-NC vs. SCN1	1.6231 (10^{-12})	7.0460 (10^{-5})
	MILP-NC vs. SCN2	1.1585 (10^{-6})	0.0002

Red shown in Fig. 2. The reason for this is over-fitting, as we can see in Table 3, the MSE of MILP-KM is lower than that of SCK2, yet the SCK2 performs better than MILP-KM on the four external indexes.

D. COMPARISON AMONG NORMALIZED CUT BASED ALGORITHMS

In this section, we report the performance comparison among NC, SCN1, SCN2 and MILP-NC in terms of the ENT, ACC, FMI, JCI, NCUT, and running time.

It can be observed from Table 5 that among the size constrained normalized cut algorithms, MILP-NC mostly outputs results with the optimum NCUT. The Games-Howell test results shown in Table 6 indicate that the differences on the NCUT are significant ($p <= 0.05$). In addition, MILP-NC runs significantly faster than SCN2 on all the data sets. Despite that the MILP-NC is less efficient than SCN1, it is signifcantly more accurate.

From the result shown in Fig. 3, we can see that MILP-NC outperforms SCN1, SCN2, and NC on the four external indexes. Thus, the proposed method could adapt the size constraints to better improve the clustering performance on the external indexes for the normalized cut algorithm.

V. DISCUSSION

This paper tackles the problem of incorporating equality constraints in the clustering task, i.e., the sizes of the clusters equal a set of constraints. The proposed method could be extended to a general framework that adapts to any kinds of size constraints. The generalized framework requires the user-specified lower bounds $s' = \{s'_1, s'_2, ..., s'_k\}$ and upper bounds $s'' = \{s''_1, s''_2, ..., s''_k\}$ on the sizes of the clusters. If $0 < s'_j \leqslant n, s''_j \geq n$, then there is only a lower bound constraint on the size of the j-th cluster (the upper bound constraint does not affect the results). If $s'_j \leqslant 0, 0 \leqslant s''_j < n$, then there is only a upper bound constraint. If $s'_j \leqslant 0, s''_j \geq n$, then there is no constraint on the size of the j-th cluster. If $0 < s'_j < s''_j < n$, then there are both lower and upper bounds. If $0 \leqslant s'_j = s''_j \leqslant n$, then there is an equality constraint.

The framework works in a similar way as the proposed method, i.e., iterating between the assignment step and the update step. The update step is the same as the one in the proposed method. To solve the problem in the assignment step, firstly, we transform the lower bound constraints into upper bound constraints by placing negative signs on both sides of the inequations. Then, we change the inequality constraints into equality constraints by adding slack variables. Finally, we have an ILP problem with a set of equality constraints. The constraint matrix on the OPDVs and slack variables is totally unimodular so that we can remove the integer constraints on these variables. The original ILP problem would become an MILP problem that can be efficiently solved. Our trials show that the solution of the framework converges as fast as the proposed method. We intend to skip the proof of the total unimodularity here, as it is beyond the scope of this paper.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel iterative approach to address the issue of size constrained clustering, which consists of an assignment step and an update step. In the assignment step, the prior knowledge about the size constraints specified by users are modeled into an ILP problem. We show that the integer constraints on the OPDVs can be removed due to the total unimodularity. Thus, the ILP problem is equivalent to an MILP problem, which can be much more efficiently solved. In the update step, new centers are updated as the centroids of the clusters. We have conducted extensive experiments on common data sets to evaluate the performance of the proposed method in terms of recognized benchmarks. The experimental results show that the clustering performance could be better improved by leveraging the cluster size constraints as proposed.

Several issues remain to be investigated in the future work. For example, we can explore more real applications where the sizes of clusters need to be restricted, such as the capacitated resource allocation problem. In addition, it is a challenging work to introduce other types of constraints into clustering, such as instance-level constraints. Last but not least, it is interesting to adapt the size constraints into other types of clustering algorithms.

REFERENCES

[1] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[2] N. Binkiewicz, J. T. Vogelstein, and K. Rohe, "Covariate-assisted spectral clustering," *Biometrika*, vol. 104, no. 2, pp. 361–377, Jun. 2017.

[3] J. Liu, J. Lee, L. Li, Z.-Q. Luo, and K. Wong, "Online clustering algorithms for radar emitter classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1185–1196, Aug. 2005.

[4] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sep. 1997.

[5] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.

[6] D. Steinley, "K-means clustering: A half-century synthesis," *Brit. J. Math. Stat. Psychol.*, vol. 59, no. 1, pp. 1–34, May 2006.

[7] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 176–190, Jan. 2008.

[8] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Comput. J.*, vol. 26, no. 4, pp. 354–359, Nov. 1983.

[9] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Hoboken, NJ, USA: Wiley, 1999.

[10] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm," *Pattern Anal. Appl.*, vol. 18, no. 1, pp. 87–112, Feb. 2015.

[11] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, "Optimizing fuzzy cluster ensemble in string representation," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 27, no. 2, Mar. 2013, Art. no. 1350005.

[12] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on elite selection of weighted clusters," *Adv. Data Anal. Classification*, vol. 7, no. 2, pp. 181–208, Jun. 2013.

[13] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, Dec. 1998.

[14] E. Segal, H. Wang, and D. Koller, "Discovering molecular pathways from protein interaction and gene expression data," *Bioinformatics*, vol. 19, pp. i264–i272, Jul. 2003.

[15] B. Wu, Y. Zhang, B.-G. Hu, and Q. Ji, "Constrained clustering and its application to face clustering in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3507–3514.

[16] O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, and R. Poliah, "A survey on clustering algorithms for wireless sensor networks," in *Proc. IEEE Int. Conf. Netw.-Based Inf. Syst.*, Sep. 2010, pp. 358–364.

[17] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained *k*-means clustering with background knowledge," in *Proc. 18th Int. Conf. Mach. Learn.*, vol. 1, 2001, pp. 577–584.

[18] Y. Yang and B. Padmanabhan, "Segmenting customer transactions using a pattern-based clustering approach," in *Proc. IEEE Int. Conf. Data Mining*, Apr. 2003, pp. 411–418.

[19] T.-B.-H. Dao, K.-C. Duong, and C. Vrain, "Constrained clustering by constraint programming," *Artif. Intell.*, vol. 244, pp. 70–94, Mar. 2017.

[20] M. E. Celebi and K. Aydin, *Unsupervised Learning Algorithms*. Cham, Switzerland: Springer, 2016.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[22] U. Von Luxburg, "A tutorial on spectral clustering," *Stat Comput*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[23] X. Chang, F. Nie, Z. Ma, and Y. Yang, "Balanced *k*-means and min-cut clustering," 2014, *arXiv:1411.6235*. [Online]. Available: https://arxiv.org/abs/1411.6235

[24] K. P. Bennett, P. S. Bradley, and A. Demiriz, "Constrained *k*-means clustering," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2000-65, May 2000. [Online]. Available: https://www.microsoft.com/en-us/research/publication/constrained-k-means-clustering/

[25] Y. Chen, Y. Zhang, and X. Ji, "Size regularized cut for data clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 211–218.

[26] Y. Kawahara, K. Nagano, and Y. Okamoto, "Submodular fractional programming for balanced clustering," *Pattern Recognit. Lett.*, vol. 32, no. 2, pp. 235–243, Jan. 2011.

[27] S. Zhu, D. Wang, and T. Li, "Data clustering with size constraints," *Knowl.-Based Syst.*, vol. 23, no. 8, pp. 883–889, Dec. 2010.

[28] B. Aoun and R. Boutaba, "Clustering in WSN with latency and energy consumption constraints," *J. Netw. Syst. Manage.*, vol. 14, no. 3, pp. 415–439, Sep. 2006.

[29] D. Vallejo-Huanga, P. Morillo, and C. Ferri, "Semi-supervised clustering algorithms for grouping scientific articles," *Procedia Comput. Sci.*, vol. 108, pp. 325–334, 2017.

[30] F. Höppner and F. Klawonn, "Clustering with size constraints," in *Computational Intelligence Paradigms*. Berlin, Germany: Springer, 2008, pp. 167–180.

[31] T. Althoff, A. Ulges, and A. Dengel, "Balanced clustering for content-based image browsing," in *Proc. Gesellschaft Informatik*, vol. 1, Mar. 2011, pp. 27–30.

[32] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics With Applications*. Boston, MA, USA: Cengage Learning, 2014.

[33] M. I. Malinen and P. Fränti, "Balanced *k*-means for clustering," in *Proc. Joint Int. Workshop Struct., Syntactic, Stat. Pattern Recognit.*, 2014, pp. 32–41.

[34] W. Tang, Y. Yang, L. Zeng, and Y. Zhan, "Optimizing MSE for clustering with balanced size constraints," *Symmetry*, vol. 11, no. 3, p. 338, Mar. 2019.

[35] N. S. Madiraju, S. M. Sadat, D. Fisher, and H. Karimabadi, "Deep temporal clustering: Fully unsupervised learning of time-domain features," 2018, *arXiv:1802.01059*. [Online]. Available: https://arxiv.org/abs/1802.01059

[36] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. Hoboken, NJ, USA: Wiley, 2014.

[37] M. W. P. Savelsbergh, "Preprocessing and probing techniques for mixed integer programming problems," *ORSA J. Comput.*, vol. 6, no. 4, pp. 445–454, Nov. 1994.

[38] D. Dheeru and E. K. Taniskidou. (2019). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[39] Y. Zhao and G. Karypis, "Soft clustering criterion functions for partitional document clustering: A summary of results," in *Proc. 13th ACM Int. Conf. Inf. Knowl. Manage.*, 2004, pp. 246–247.

[40] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Proc. IEEE Int. Conf. Data Mining*, Oct. 2007, pp. 577–582.

[41] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Stat. Assoc.*, vol. 78, no. 383, pp. 553–569, Sep. 1983.

[42] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bull. Soc Vaudoise Sci. Naturelles*, vol. 37, pp. 547–579, 1901.

[43] V. Grossi, A. Romei, and F. Turini, "Survey on using constraints in data mining," *Data Mining Knowl. Discovery*, vol. 31, no. 2, pp. 424–464, Mar. 2017.

[44] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1101–1113, Nov. 1993.

[45] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high–dimensional hyperspheres," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 702–719, May 2004.

[46] A. Banerjee and J. Ghosh, "Scalable clustering algorithms with balancing constraints," *Data Mining Knowl. Discovery*, vol. 13, no. 3, pp. 365–395, Sep. 2006.

[47] S. Zhang, H.-S. Wong, and D. Xie, "Semi-supervised clustering with pairwise and size constraints," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2014, pp. 2450–2457.

[48] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 52, nos. 1–2, pp. 7–21, Feb. 2005.

[49] I. D. Luptáková, M. Šimon, L. Huraj, and J. Pospíchal, "Neural gas clustering adapted for given size of clusters," *Math. Problems Eng.*, vol. 2016, Oct. 2016, Art. no. 9324793.

[50] M. Baranwal and S. M. Salapaka, "Clustering with capacity and size constraints: A deterministic approach," in *Proc. IEEE Indian Control Conf.*, Jan. 2017, pp. 251–256.

[51] S. Yuepeng, L. Min, and W. Cheng, "A modified k-means algorithm for clustering problem with balancing constraints," in *Proc. IEEE Int. Conf. Measuring Technol. Mechatronics Autom.*, vol. 1, Jan. 2011, pp. 127–130.

[52] N. Ganganath, C. T. Cheng, and K. T. Chi, "Data clustering with cluster size constraints using a modified k-means algorithm," in *Proc. IEEE Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, Oct. 2014, pp. 158–161.

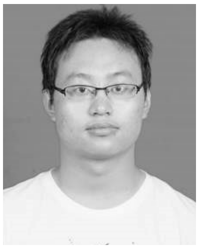
[53] A. Koberstein, "Progress in the dual simplex algorithm for solving large scale LP problems: Techniques for a fast and stable implementation," *Comput. Optim. Appl.*, vol. 41, no. 2, pp. 185–204, Nov. 2008.

[54] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, 1986.

[55] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*. Philadelphia, PA, USA: SIAM, 2007, pp. 1027–1035.

[56] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.

[57] D. C. Montgomery, G. C. Runger, and N. F. Hubele, *Engineering Statistics*. Hoboken, NJ, USA: Wiley, 2009.



WEI TANG is currently pursuing the master's degree with the Department of Computer Science, Jiangsu University. His research interests include cluster analysis and motion analysis.

YANG YANG received the Ph.D. degree in engineering from the University of Science and Technology of China. He is currently an Associate Professor with the Department of Computer Science, Jiangsu University. His research interests include cluster analysis and motion analysis.

LANLING ZENG received the Ph.D. degree in mathematics from Zhejiang University. She is currently an Associate Professor with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. Her research interests mainly include computer graphics and plant modeling.

YONGZHAO ZHAN received the Ph.D. degree in computer science and technology from Nanjing University. He is currently a Professor with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. His research interests mainly include sparse representation and video analysis.