Modified global k-means algorithm for minimum sum-of-squares clustering problems

Adil M. Bagirov

*Centre for Informatics and Applied Optimization, School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria, 3353, Australia,
E-mail: a.bagirov@ballarat.edu.au, Tel.: +61 3 5327 9330, Fax: +61 3 5327 9289*

Abstract

k-means algorithm and its variations are known to be fast clustering algorithms. However, they are sensitive to the choice of starting points and inefficient for solving clustering problems in large data sets. Recently, a new version of the k-means algorithm, the global k-means algorithm has been developed. It is an incremental algorithm that dynamically adds one cluster center at a time and uses each data point as a candidate for the k-th cluster center. Results of numerical experiments show that the global k-means algorithm considerably outperforms the k-means algorithms. In this paper, a new version of the global k-means algorithm is proposed. A starting point for the k-th cluster center in this algorithm is computed by minimizing an auxiliary cluster function. Results of numerical experiments on 14 data sets demonstrate the superiority of the new algorithm, however, it requires more computational time than the global k-means algorithm.

Keywords: minimum sum-of-squares clustering, nonsmooth optimization, k-means algorithm, global k-means algorithm.

1 Introduction

The cluster analysis deals with the problems of organization of a collection of patterns into clusters based on similarity. It is also known as the *unsupervised* classification of patterns and has found many applications in different areas.

In cluster analysis we assume that we have been given a finite set of points A in the n-dimensional space $\mathbb{R}^n$, that is

$$A = \{a^1, \dots, a^m\}, \text{ where } a^i \in \mathbb{R}^n, \ i = 1, \dots, m.$$

There are different types of clustering. In this paper, we consider the hard unconstrained partition clustering problem, that is the distribution of the points of the set A into a given number k of disjoint subsets $A^j, \ j = 1, \dots, k$ with respect to predefined criteria such that:

1) $A^j \neq \emptyset, \ j = 1, \dots, k;$

2) $A^j \bigcap A^l = \emptyset, \ j, l = 1, \dots, k, \ j \neq l;$

3) $A = \bigcup\limits_{j=1}^{k} A^j$;

4) no constraints are imposed on the clusters $A^j, \ j = 1, \dots, k$.

The sets $A^j, \ j = 1, \dots, k$ are called clusters. We assume that each cluster A^j can be identified by its center (or centroid) $x^j \in \mathbb{R}^n, \ j = 1, \dots, k$. Then the clustering problem can be reduced to the following optimization problem (see [5, 22]):

$$\text{minimize} \ \ \psi_k(x, w) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} \|x^j - a^i\|^2 \tag{1}$$

$$\text{subject to} \ \ x = (x^1, \dots, x^k) \in \mathbb{R}^{n \times k}, \tag{2}$$

$$\sum_{j=1}^{k} w_{ij} = 1, \ i = 1, \dots, m, \tag{3}$$

and

$$w_{ij} = 0 \text{ or } 1, \ i = 1, \dots, m, \ j = 1, \dots, k \tag{4}$$

where w_{ij} is the association weight of pattern a^i with the cluster j, given by

$$w_{ij} = \begin{cases} 1 & \text{if pattern } a^i \text{ is allocated to the cluster } j, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and

$$x^j = \frac{\sum_{i=1}^{m} w_{ij} a^i}{\sum_{i=1}^{m} w_{ij}}, \ \ j = 1, \dots, k. \tag{6}$$

Here $\| \cdot \|$ is an Euclidean norm and w is an $m \times k$ matrix. The problem (1) is also known as minimum sum-of-squares clustering problem.

Different algorithms have been proposed to solve the clustering problem. The paper [16] provides survey of most of existing algorithms. We mention among them heuristics like k-means algorithms and their variations (h-means, j-means etc.), mathematical programming techniques including dynamic programming, branch and bound, cutting plane, interior point methods, the variable neighborhood search algorithm and metaheuristics like simulated annealing, tabu search, genetic algorithms (see [1, 6, 7, 8, 9, 11, 12, 13, 14, 17, 21, 22, 23]).

The objective function ψ_k in (1) has many local minimizers (local solutions of problem (1)-(6)). Local minimizers are points, where the function ψ_k achieves its smallest value in some feasible neighborhood of these points. Global minimizers (or global solutions of problem (1)-(6)) of ψ_k are points where the function attains its least value over the feasible set. It is expected that global minimzers provide better cluster structure of a data set. However, the most of clustering algorithms can locate only local minimizers of the function ψ_k and these local minimizers may differ from global ones significantly as the number of clusters increases. Global optimization algorithms, mentioned above, are not applicable to even relatively large data sets. Another difficulty is that the number of clusters is not known a priori.

Over the last several years different incremental algorithms have been proposed to address these difficulties. Incremental clustering algorithms attempt to optimally add one new cluster center at each stage. In order to compute k-partition of the set A these algorithms start from an initial state with the $k-1$ centers for the $(k-1)$-clustering problem and the remaining k-th center is placed in an appropriate position. Results of numerical experiments show that these algorithms are able to locate either a global minimizer or a local minimizer close to global one. The paper [4] develops an incremental algorithm based on nonsmooth optimization approach to clustering. The incremental approach is also discussed in [15].

The global k-means algorithm, introduced in [18], is a significant improvement of the k-means algorithm. It is an incremental algorithm. In this algorithm each data point is used as a starting point for the k-th cluster center. Such an approach leads at least to a near global minimizer. However this approach is not efficient since it is very time consuming, as m applications of k-means algorithm are made. Instead the authors suggest two procedures to reduce computational load.

The first algorithm is called the fast global k-means algorithm. Given the solution $x^1, \ldots, x^{k-1}$ of the $(k-1)$-clustering problem and the corresponding value $\psi^*_{k-1} = \psi_{k-1}(x^1, \ldots, x^{k-1})$ of the function ψ_k in (1) this algorithm does not execute the k-means algorithm for each data point. Instead it computes an upper bound $\psi^*_k \leq \psi^*_{k-1} - b_j$ on the ψ^*_k, where

$$b_j = \sum_{i=1}^{m} \max\{0, d^i_{k-1} - \|a^j - a^i\|^2\}, \; j = 1, \ldots, m. \tag{7}$$

Here d^i_{k-1} is the squared distance between a^i and the closest center among $k-1$ cluster centers $x^1, \ldots, x^{k-1}$:

$$d^i_{k-1} = \min\left\{\|x^1 - a^i\|^2, \ldots, \|x^{k-1} - a^i\|^2\right\}. \tag{8}$$

A data point $a^j \in A$ with the maximum value of b_j is chosen as a starting point for the k-th cluster center.

In the second procedure a $k-d$ tree is used to partition A into $m' \ll m$ subsets; their centroids are used as starting points in the global k-means scheme. The second procedure can be applied to low dimensional data sets.

In this paper, we propose a new version of the global k-means algorithm. The difference between the new version and the fast global k-means algorithm lies in the way a starting point for the k-th cluster center is obtained. Given the solution $x^1, \ldots, x^{k-1}$ of the $(k-1)$-clustering problem, we formulate the so-called auxiliary cluster function:

$$\bar{f}_k(y) = \frac{1}{m} \sum_{i=1}^{m} \min\left\{d^i_{k-1}, \|y - a^i\|^2\right\}. \tag{9}$$

We apply the k-means algorithm to minimize this function. A local minimizer found is selected as a starting point for the k-th cluster center. We present the results of numerical experiments on 14 data sets. These results demonstrate that the superiority of the proposed algorithm over the global k-means algorithm, however, it is less computationally efficient.

The rest part of the paper is organized as follows: Section 2 gives a brief description of k-means and the global k-means algorithms. The nonsmooth optimization approach to clustering and an algorithm for the computation of a starting point is described in Section 3. Section 4 presents an algorithm for solving clustering problems. The results of numerical experiments are given in Section 5. Section 6 concludes the paper.

2 k-means and the global k-means algorithms

In this section we give a brief description of the k-means and the global k-means algorithms.

The k-means algorithm proceeds as follows.

Algorithm 1 The k-means algorithm

Step 1. Choose a seed solution consisting of k centers (not necessarily belonging to A).

Step 2. Allocate data points $a \in A$ to its closest center and obtain k-partition of A.

Step 3. Recompute centers for this new partition and go to Step 2 until no more data points change their clusters.

This algorithm is very sensitive to the choice of a starting point. It converges to a local solution which can significantly differ from the global solution in many large data sets.

The global k-means algorithm proposed in [18] is an incremental clustering algorithm. To compute $k \leq m$ clusters this algorithm proceeds as follows.

Algorithm 2 The global k-means algorithm.

Step 1. (Initialization) Compute the centroid x^1 of the set A:

$$x^1 = \frac{1}{m} \sum_{i=1}^{m} a^i, \quad a^i \in A, \ i = 1, \ldots, m \tag{10}$$

and set $q = 1$.

Step 2. (Stopping criterion) Set $q = q + 1$. If $q > k$, then stop.

Step 3. Take the centers $x^1, x^2, \ldots, x^{q-1}$ from the previous iteration and consider each point a of A as a starting point for the q-th cluster center, thus obtaining m initial solutions with q points $(x^1, \ldots, x^{q-1}, a)$; apply the k-means algorithm to each of them; keep the best q-partition obtained and its centers $y^1, y^2, \ldots, y^q$.

Step 4. Set $x^i = y^i, \ i = 1, \ldots, q$ and go to Step 2.

This version of the algorithm is not applicable for clustering on middle sized and large data sets. Two procedures were introduced to reduce its complexity (see [18]). We mention here only one of them, because the second procedure is applicable only to low dimensional data sets. Let d^i_{k-1} be a squared distance between $a^i \in A,\ i = 1, \ldots, m$ and the closest cluster center among the $k-1$ cluster centers obtained so far. In order to find the starting point for the k-th cluster center, for each $a^j \in A,\ j = 1, \ldots, m$ we compute b_j using (7).

$b_j,\ j = 1, ..., m$ shows how much one can decrease the value of the function ψ_k from (1) if the data point a^j is chosen as the k-th cluster center. Obviously, if $a^j \in A,\ j = 1, \ldots, m$ is not among the cluster centers $x^1, \ldots, x^{k-1}$, then $b_j > 0$. This means that by selecting any such data point as a starting point for the k-th cluster center one can decrease the value of the function ψ_k at least by b_j. It is clear that a data point $a^j \in A$ with the largest value of the b_j is the best candidate to be a starting point for the k-th cluster center. Therefore, first we compute

$$\bar{b} = \max_{j=1,\ldots,m} b_j \tag{11}$$

and find the data point $a^j \in A$ such that $b_j = \bar{b}$. This data point is selected as a starting point for the k-th cluster center. In our numerical experiments we use this procedure.

3 Computation of starting points

The clustering problem (1) can be reformulated in terms of nonsmooth, nonconvex optimization as follows (see [2, 3, 5]):

$$\text{minimize} \quad f_k(x) \quad \text{subject to } x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \tag{12}$$

where

$$f_k(x^1, \ldots, x^k) = \frac{1}{m} \sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|^2. \tag{13}$$

We call f_k a *cluster function.* Comparing two different formulations (1) and (12) of the hard clustering problem one can note that:

1. The objective function ψ_k depends on variables $w_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, k$ (coefficients, which are integers) and $x^1, x^2, \ldots, x^k,\ x^j \in \mathbb{R}^n,\ j = 1, \ldots, k$ (cluster centers, which are continuous variables). However, the function f_k depends only on continuous variables $x^1, \ldots, x^k$.

2. The number of variables in problem (1) is $(m + n) \times k$ whereas in problem (12) this number is only $n \times k$ and the number of variables does not depend on the number of instances. It should be noted that in many real-world data sets the number of instances m is substantially greater than the number of features n.

3. The function ψ_k is continuously differentiable with respect to both variables w and x. Since the function f_k is represented as a sum of minima functions it is nonsmooth for $k > 1$, that is it is not differentiable everywhere.

4. Both functions ψ_k and f_k are nonconvex.

5. Problem (1) is mixed integer nonlinear programming problem and problem (12) is nonsmooth global optimization problem. However, they are equivalent in the sense that their global minimizers coincide (see [5]).

Circumstances mentioned in Items 1 and 2 can be considered as advantages of the nonsmooth optimization formulation (12) of the clustering problem.

Assume that $k > 1$ and the cluster centers $x^1, \dots, x^{k-1}$ for $(k-1)$-partition problem are known. Considering k-partition problem we introduce the following function:

$$\bar{f}_k(y) = \frac{1}{m} \sum_{i=1}^{m} \min \left\{ d_{k-1}^i, \|y - a^i\|^2 \right\}, \tag{14}$$

where $y \in \mathbb{R}^n$ stands for k-th cluster center and d_{k-1}^i is defined in (8). The function $\bar{f}_k$ is called an *auxiliary cluster function*. It depends on n variables only. It is clear that

$$\bar{f}_k(y) = f_k(x^1, \dots, x^{k-1}, y) \tag{15}$$

for all $y \in \mathbb{R}^n$. This means that the auxiliary cluster function $\bar{f}_k$ coincides with the cluster function f_k with fixed $k-1$ cluster centers $x^1, \dots, x^{k-1}$. For each data point $a^i \in A$ consider also the following function:

$$\varphi_{ik}(y) = \min \left\{ d_{k-1}^i, \|y - a^i\|^2 \right\}. \tag{16}$$

This function is represented as a minimum of constant and very simple quadratic function. If the data point a^i is a cluster center then $d_{k-1}^i = 0$ and $\varphi_{ik}(y) \equiv 0$. Otherwise

$$\varphi_{ik}(y) = \begin{cases} \|y - a^i\|^2 & \text{if } \|y - a^i\|^2 < d_{k-1}^i, \\ \\ d_{k-1}^i & \text{if } \|y - a^i\|^2 \geq d_{k-1}^i. \end{cases} \tag{17}$$

Since it is natural to assume that $k < m$, it is obvious that $\varphi_{ik}(y) > 0$ for some $a^i \in A$ and $y \in \mathbb{R}^n$ and $\bar{f}_k(y) > 0$ for all $y \in \mathbb{R}^n$. As a minimum function, φ_{ik} is nonsmooth and nonconvex. It is nondifferentiable at points $y \in \mathbb{R}^n$, where $\|y - a^i\|^2 = d_{k-1}^i$. Therefore, the function $\bar{f}_k$ is also nonsmooth and nonconvex. The set where this function is nondifferentiable can be represented as a union of sets, where functions φ_{ik} are nondifferentiable.

Minimum value f_{k-1}^* of the function f_{k-1} is

$$f_{k-1}^* = \frac{1}{m} \sum_{i=1}^{m} d_{k-1}^i. \tag{18}$$

If the data point $a^j \in A$ is not a cluster center, then

$$f_k(x^1, \dots, x^{k-1}, a^j) = \frac{1}{m} \sum_{i=1}^{m} \min \left\{ d_{k-1}^i, \|a^j - a^i\|^2 \right\}. \tag{19}$$

Given $a^j \in A$ consider the following two index sets:

$$I_1 = \left\{ i \in \{1, \dots, m\} : \ \|a^i - a^j\|^2 \geq d^i_{k-1} \right\}, \tag{20}$$

$$I_2 = \left\{ i \in \{1, \dots, m\} : \ \|a^i - a^j\|^2 < d^i_{k-1} \right\}. \tag{21}$$

Using this notation one can rewrite a formulae for b_j from (7) as follows

$$b_j = \sum_{i \in I_2} \left(d^i_{k-1} - \|a^j - a^i\|^2 \right). \tag{22}$$

Then

$$\begin{aligned} \bar{f}_k(a^j) &= f_k(x^1, \dots, x^{k-1}, a^j) \\ &= \frac{1}{m} \left(\sum_{i \in I_1} d^i_{k-1} + \sum_{i \in I_2} \|a^j - a^i\|^2 \right) \end{aligned} \tag{23}$$

and therefore,

$$\begin{aligned} f_{k-1}(x^1, \dots, x^{k-1}) - f_k(x^1, \dots, x^{k-1}, a^j) &= \sum_{i \in I_2} \left(d^i_{k-1} - \|a^j - a^i\|^2 \right) \\ &= b_j. \end{aligned} \tag{24}$$

This means that if one selects a^j as a starting point for the k-th cluster center then the optimal value of the function f_{k-1} can be decreased by $b_j \geq 0$. Therefore it is natural to select a data point with the largest value of b_j as a starting point for the k-th cluster center, which is done in one of the versions of the global k-means algorithm. In this paper, we suggest to minimize the auxiliary cluster function $\bar{f}_k$ to find a starting point for the k-th cluster center. Since the auxiliary cluster function coincides with the cluster function f_k when previous $k-1$ cluster centers $x^1, \dots, x^{k-1}$ are fixed, the minimization of the auxiliary cluster function is equivalent to the minimization of the cluster function f_k with fixed $k-1$ cluster centers $x^1, \dots, x^{k-1}$. The k-means algorithm is applied to find a local minimizer of $\bar{f}_k$.

Now consider the set

$$\overline{D} = \left\{ y \in {\rm I\!R}^n : \|y - a^i\|^2 \geq d^i_{k-1} \right\}. \tag{25}$$

$\bar{D}$ is the set where the distance between any its point y and any data point $a^i \in A$ is no less than the distance between this data point and its cluster center. We also consider the following set

$$D_0 = {\rm I\!R}^n \setminus \overline{D} \equiv \left\{ y \in {\rm I\!R}^n : \exists I \subset \{1, \dots, m\}, \ I \neq \emptyset : \|y - a^i\| < d^i_{k-1} \ \ \forall i \in I \right\}. \tag{26}$$

The function $\bar{f}_k$ is a constant on the set $\overline{D}$ and its value is

$$\bar{f}_k(y) = d_0 \equiv \frac{1}{m} \sum_{i=1}^{m} d^i_{k-1}, \quad \forall y \in \overline{D}. \tag{27}$$

It is clear that $x^j \in \overline{D}$ for all $j = 1, \ldots, k-1$ and $a^i \in D_0$ for all $a^i \in A, \;\; a^i \neq x^j, \; j = 1, \ldots, k-1$. It is also clear that $\bar{f}_k(y) < d_0$ for all $y \in D_0$.

Any $y \in D_0$ can be selected as a starting point for the k-th cluster center. The function $\bar{f}_k$ is nonconvex function with many local minima and the global minimizer of this function can be the best candidate to be starting point for the k-th cluster center. However, it is not always possible to find the global minimizer of $\bar{f}_k$ in a reasonable time. Therefore, we propose an algorithm for finding a local minimizer of the function $\bar{f}_k$.

For any $y \in D_0$ consider the following sets:

$$S_1(y) = \left\{ a^i \in A : \|y - a^i\|^2 = d^i_{k-1} \right\}, \tag{28}$$

$$S_2(y) = \left\{ a^i \in A : \|y - a^i\|^2 < d^i_{k-1} \right\}, \tag{29}$$

$$S_3(y) = \left\{ a^i \in A : \|y - a^i\|^2 > d^i_{k-1} \right\}. \tag{30}$$

The set $S_2(y) \neq \emptyset$ for any $y \in D_0$.

The following algorithm is proposed to find a starting point for the k-th cluster center.

Algorithm 3 An algorithm for finding a starting point.

Step 1. For each $a^i \in D_0 \bigcap A$ compute the set $S_2(a^i)$, its center c^i and the value $\bar{f}_{k,a^i} = \bar{f}_k(c^i)$ of the function $\bar{f}_k$ at the point c^i.

Step 2. Compute

$$\bar{f}_{k,min} = \min_{a^i \in D_0 \bigcap A} \bar{f}_{k,a^i}, \tag{31}$$

$$a^j = \text{arg min}_{a^i \in D_0 \bigcap A} \bar{f}_{k,a^i}, \tag{32}$$

the corresponding center c^j and the set $S_2(c^j)$.

Step 3. Recompute the set $S_2(c^j)$ and its center until no more data points escape or return to this cluster.

Let $\bar{x}$ be a cluster center generated by Algorithm 3. Since we consider the hard clustering problem, that is each data point belongs to only one cluster, one can assume that $S_1(\bar{x}) = \emptyset$.

Proposition 1 *The point $\bar{x}$ is a local minimizer of the function $\bar{f}_k$.*

The proof can be found in Appendix.

4 An incremental clustering algorithm

In this section we describe an incremental algorithm for solving cluster analysis problems.

Algorithm 4 An incremental algorithm for clustering problems.

Step 1. (Initialization). Select a tolerance $\varepsilon > 0$. Compute the center $x^1 \in \mathbb{R}^n$ of the set A. Let f^1 be the corresponding value of the objective function (13). Set $k = 1$.

Step 2. (Computation of the next cluster center). Set $k = k + 1$. Let $x^1, \dots, x^{k-1}$ be the cluster centers for $(k-1)$-partition problem. Apply Algorithm 3 to find a starting point $\bar{y} \in \mathbb{R}^n$ for the k-th cluster center.

Step 3. (Refinement of all cluster centers). Select $(x^1, \dots, x^{k-1}, \bar{y})$ as a new starting point, apply k-means algorithm to solve k-partition problem. Let $y^1, \dots, y^k$ be a solution to this problem and f^k be the corresponding value of the objective function (13).

Step 4. (Stopping criterion). If

$$\frac{f^{k-1} - f^k}{f^1} < \varepsilon \tag{33}$$

then stop, otherwise set $x^i = y^i, \ i = 1, \dots, k$ and go to Step 2.

It is clear that $f^k \geq 0$ for all $k \geq 1$ and the sequence $\{f^k\}$ is decreasing, that is,

$$f^{k+1} \leq f^k \ \text{ for all } \ k \geq 1.$$

This means that the stopping criterion in Step 4 will be satisfied after finite many iterations. Thus Algorithm 4 computes as many clusters as the data set A contains with respect to the tolerance $\varepsilon > 0$.

The choice of the tolerance $\varepsilon > 0$ is crucial for Algorithm 4. Large values of ε can result in the appearance of large clusters whereas small values can produce artificial clusters. The recommended values for ε are $\varepsilon \in [0.01, 0.1]$.

5 Results of numerical experiments

To verify the efficiency of the proposed algorithm numerical experiments with a number of real-world data sets have been carried out on a PC Pentium-4 with CPU 2.4 GHz and RAM 512 MB. 14 data sets have been used in numerical experiments. The brief description of the data sets is given in Table 1. The detailed description of German towns, Bavaria postal data sets can be found in [22], Fisher's Iris Plant data set in [10], the traveling salesman problems TSPLIB1060 and TSPLIB3038 in [20] and all other data sets in [19].

We computed up to 10 clusters in data sets with no more than 150 instances, up to 50 clusters in data sets with the number of instances between 150 and 1000 and up to 100 clusters in data sets with more than 1000 instances. The multi-start k-means (MS k-means) and the global k-means algorithms (GKM) have been used in numerical experiments for comparison purpose. To find k clusters, 100 times k starting points were randomly chosen in the MS k-means algorithm for all data sets and starting points were data points. In the GKM and the modified global k-means (MGKM) algorithms a distance matrix $D = (d_{ij})_{i,j=1}^m$ of a data set was computed before the start of the algorithms. Here $d_{ij} = \|a^i - a^j\|^2$. This matrix was used by both algorithms to find starting points.

Table 1: The brief description of data sets

Data sets	Number of instances	Number of attributes
German towns	59	2
Bavaria postal 1	89	3
Bavaria postal 2	89	4
Fisher's Iris Plant	150	4
Heart Disease	297	13
Liver Disorders	345	6
Ionosphere	351	34
Congressional Voting Records	435	16
Breast Cancer	683	9
Pima Indians Diabetes	768	8
TSPLIB1060	1060	2
Image Segmentation	2310	19
TSPLIB3038	3038	2
Page Blocks	5473	10

Results of numerical experiments are presented in Tables 2-8. In these tables we use the following notation:

- k is the number of clusters;
- f_{opt} is the best known value of the cluster function (13) (multiplied by m) for the corresponding number of clusters. For German towns, Bavaria Postal 1 and 2, Iris Plant data sets f_{opt} is the value of the cluster function at the known global minimizer (see [15]);
- E is the error in %;
- N is the number of Euclidean norm evaluations for the computation of the corresponding number of clusters. To avoid big numbers in tables we use its expression in the form $N = \alpha \times 10^l$ and present the values of α in tables. $l = 4$ for German towns, Bavaria Postal 1 and 2, Iris Plant data sets, $l = 5$ for Heart Disease, Liver Disorders, Ionosphere, Congressional Voting Records data sets, $l = 6$ for Breast Cancer, Pima Indians Diabetes, TSPLIB1060, Image Segmentation data sets and $l = 7$ for TSPLIB3038, Page Blocks data sets.
- t is the CPU time (in seconds).

The values of f_{opt} for German towns, Bavaria postal , Iris Plant, Image Segmentation ($k \leq 50$), TSPLIB1060 ($k \leq 50$) and TSPLIB3038 ($k \leq 50$) data sets are available, for example, in [4, 15]. In all other cases we take as f_{opt} the best value obtained by the MS k-means, GKM and MGKM algorithms.

The error E is computed as

$$E = \frac{(\bar{f} - f_{opt})}{f_{opt}} \cdot 100, \tag{34}$$

where $\bar{f}$ is the best value (multiplied by m) of the objective function (13) obtained by an algorithm. $E = 0$ implies that an algorithm finds the best known solution. We say that an algorithm finds a near global (or best known) solution if $0 < E < 1$.

Table 2: Results for German towns and Bavaria postal 1 data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
German towns										
2	$0.12143 \cdot 10^6$	0.00	6.80	0.00	0.00	0.230	0.00	0.00	0.590	0.00
3	$0.77009 \cdot 10^5$	0.00	13.6	0.00	1.45	0.289	0.00	1.45	0.991	0.00
4	$0.49601 \cdot 10^5$	0.00	23.6	0.00	0.72	0.366	0.00	0.72	1.430	0.00
5	$0.38716 \cdot 10^5$	0.00	31.2	0.00	0.00	0.490	0.00	0.00	1.910	0.00
6	$0.30536 \cdot 10^5$	0.00	38.3	0.00	0.00	0.602	0.00	0.27	2.350	0.00
7	$0.24433 \cdot 10^5$	5.35	44.9	0.00	0.09	0.732	0.00	0.00	2.800	0.00
8	$0.21748 \cdot 10^5$	0.33	46.1	0.00	0.10	0.832	0.00	0.00	3.260	0.00
9	$0.18946 \cdot 10^5$	4.14	57.8	0.00	0.00	0.997	0.00	2.28	3.730	0.00
10	$0.16555 \cdot 10^5$	13.98	61.4	0.02	0.28	1.120	0.00	0.00	4.270	0.00
Bavaria postal 1										
2	$0.60255 \cdot 10^{12}$	0.00	11.7	0.00	7.75	0.445	0.00	0.00	1.26	0.00
3	$0.29451 \cdot 10^{12}$	0.00	30.5	0.00	0.00	0.507	0.00	0.00	2.13	0.00
4	$0.10447 \cdot 10^{12}$	0.00	43.0	0.00	0.00	0.730	0.00	0.00	3.16	0.00
5	$0.59762 \cdot 10^{11}$	0.00	67.6	0.00	0.00	1.050	0.00	0.00	4.29	0.00
6	$0.35909 \cdot 10^{11}$	27.65	76.0	0.00	0.00	1.170	0.00	0.00	5.22	0.00
7	$0.21983 \cdot 10^{11}$	0.61	107	0.02	1.50	1.550	0.00	1.50	6.41	0.02
8	$0.13385 \cdot 10^{11}$	0.00	124	0.03	0.00	1.980	0.00	0.00	7.65	0.02
9	$0.84237 \cdot 10^{10}$	35.81	135	0.03	0.00	2.150	0.00	0.00	8.71	0.02
10	$0.64465 \cdot 10^{10}$	30.67	160	0.03	0.00	2.870	0.00	0.00	10.2	0.02

The results presented in Table 2 show that the MS k-means algorithm can locate global solutions when the number of clusters $k \leq 6$ for German towns and $k \leq 5$ for Bavaria postal 1 data sets. However, the results also show that this algorithm is not effective at computing more than 5 clusters even for small data sets. For German towns data set the GKM algorithm does as same as the MGKM algorithm four times, it does two times better and three times worse than the MGKM algorithm. For Bavaria postal 1 data set the GKM algorithm does as same as the MGKM algorithm eight times and it does once worse than the MGKM algorithm. The MS k-means algorithm is better than two other algorithms when the number of clusters $k \leq 5$. The GKM algorithm requires less computational efforts than other two algorithms.

In these data sets both the GKM and MGKM algorithms in most of cases could locate either global or near global solutions. For German towns data set the MS k-means algorithm could find global or near global solutions six times, the GKM algorithm eight times and the MGKM algorithm seven times. On Bavaria Postal 1 data set the MS k-means algorithm finds global or near global solutions six times, the GKM algorithm seven times and the MGKM algorithm eight times.

Table 3: Results for Bavaria postal 2 and Iris Plant data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
Bavaria postal 2										
2	$0.19908 \cdot 10^{11}$	144.28	13.3	0.00	162.17	0.445	0.00	162.17	1.25	0.00
3	$0.17399 \cdot 10^{11}$	0.00	23.9	0.00	0.00	0.507	0.00	0.00	2.13	0.00
4	$0.75591 \cdot 10^{10}$	0.00	40.9	0.00	0.00	0.659	0.00	0.00	3.12	0.00
5	$0.53429 \cdot 10^{10}$	0.00	53.5	0.00	1.86	0.801	0.00	1.86	4.08	0.00
6	$0.32263 \cdot 10^{10}$	37.37	69.4	0.00	0.00	0.917	0.00	0.00	5.00	0.02
7	$0.22271 \cdot 10^{10}$	10.75	91.6	0.00	0.00	1.49	0.00	0.00	6.32	0.02
8	$0.17170 \cdot 10^{10}$	12.31	106	0.03	0.00	1.71	0.00	0.00	7.35	0.02
9	$0.14030 \cdot 10^{10}$	9.50	126	0.03	0.00	2.12	0.00	0.00	8.41	0.02
10	$0.11928 \cdot 10^{10}$	18.88	132	0.05	0.00	2.31	0.00	0.00	9.41	0.02
Iris Plant										
2	152.348	0.00	17.8	0.00	0.00	1.26	0.00	0.00	3.55	0.00
3	78.851	0.00	41.9	0.00	0.01	1.78	0.00	0.01	6.34	0.00
4	57.228	0.00	81.4	0.03	0.05	2.21	0.00	0.05	9.11	0.00
5	46.446	0.00	105	0.05	0.54	2.53	0.02	0.54	11.7	0.02
6	39.040	0.00	121	0.05	1.44	2.81	0.02	1.44	14.4	0.02
7	34.298	4.20	157	0.05	3.17	3.14	0.02	3.17	17.0	0.02
8	29.989	10.69	171	0.05	1.71	3.88	0.02	1.71	19.9	0.02
9	27.786	2.31	184	0.06	2.85	4.16	0.02	2.85	22.4	0.02
10	25.834	8.27	212	0.08	3.55	4.48	0.02	3.55	25.0	0.02

As one can see from Table 3, the GKM and MGKM algorithms find the same solutions for both Bavaria postal 2 and Iris Plant data sets. However, the GKM algorithm requires less computational efforts than the MGKM algorithm.

All algorithms failed to find the global solution for $k = 2$ in Bavaria postal 2 data set. The MS k-means algorithm fails to find the global solution when the number of clusters $k > 5$ for Bavaria postal 2 data set and $k > 6$ for Iris Plant data set. For Bavaria postal 2 data set the MS k-means algorithm finds global or near global solutions three times, the GKM and MGKM algorithms seven times. For Iris Plant data set the MS k-means algorithm finds such solutions five times, the GKM and MGKM algorithms four times.

The results from Table 4 demonstrate that the MS k-means algorithm cannot locate the global solution for Heart Disease data set when $k > 5$ and for Liver Disorders data

Table 4: Results for Heart Disease and Liver Disorders data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
Heart Disease										
2	$0.59890 \cdot 10^6$	0.00	7.86	0.16	0.00	0.505	0.02	0.00	1.40	0.02
5	$0.32797 \cdot 10^6$	0.00	29.7	0.47	0.52	0.722	0.02	0.52	4.25	0.05
10	$0.20222 \cdot 10^6$	2.76	80.1	0.84	0.00	1.57	0.03	1.93	9.31	0.09
15	$0.14771 \cdot 10^6$	8.79	113	1.14	0.00	2.68	0.06	0.68	14.6	0.17
20	$0.11778 \cdot 10^6$	7.46	130	1.19	0.00	3.98	0.09	1.34	20.4	0.23
25	$0.10213 \cdot 10^6$	5.16	151	1.31	0.48	5.46	0.11	0.00	25.9	0.33
30	$0.88795 \cdot 10^5$	18.66	180	1.64	0.00	6.80	0.14	0.31	31.5	0.44
40	$0.68645 \cdot 10^5$	28.65	213	1.67	1.71	9.71	0.20	0.00	43.5	0.69
50	$0.55894 \cdot 10^5$	33.68	250	1.88	2.06	13.2	0.27	0.00	55.4	1.03
Liver Disorders										
2	$0.42398 \cdot 10^6$	0.00	6.91	0.09	93.96	0.600	0.00	93.96	0.600	0.00
5	$0.21826 \cdot 10^6$	0.00	41.7	0.42	0.08	0.990	0.03	0.08	5.75	0.03
10	$0.12768 \cdot 10^6$	0.09	87.5	0.67	0.00	2.00	0.05	0.02	12.7	0.08
15	$0.97474 \cdot 10^5$	6.53	147	0.92	1.62	3.41	0.08	0.00	20.3	0.13
20	$0.81820 \cdot 10^5$	9.05	184	1.11	0.29	5.12	0.11	0.00	27.5	0.19
25	$0.70419 \cdot 10^5$	16.64	208	1.17	0.23	6.99	0.13	0.00	35.1	0.28
30	$0.61143 \cdot 10^5$	24.33	229	1.31	0.21	8.75	0.16	0.00	43.0	0.39
40	$0.47832 \cdot 10^5$	37.83	290	1.61	3.59	14.6	0.23	0.00	60.4	0.66
50	$0.39581 \cdot 10^5$	50.64	337	1.88	5.50	19.9	0.28	0.00	78.0	0.97

set when $k > 10$. For Heart Disease data set the GKM algorithm does as same as the MGKM algorithm two times, it does four times better and three times worse than the MGKM algorithm. For Liver Disorder data set the GKM algorithm does as same as the MGKM algorithm two times and it does once better and six times worse than the MGKM algorithm. Again the GKM algorithm requires less computational efforts than other two algorithms.

For Heart Disease data set the MS k-means algorithm finds the best known or near best known solutions two times, the GKM and MGKM algorithms find those solutions seven times. For Liver Disorder data set the MS k-means algorithm finds the best known or near best known solutions three times, the GKM algorithm five times and the MGKM algorithm eight times. The MGKM algorithm outperforms two other algorithms as the number of clusters increases.

In Ionosphere and Congressional Voting Records data sets the MS k-means algorithm again cannot find the global solution when the number of clusters $k > 5$ (see Table 5). For Ionosphere data set the GKM algorithm does as same as the MGKM algorithm once, it does once better and seven times worse than the MGKM algorithm. For Congressional Voting Records data set the GKM algorithm does as same as the MGKM algorithm two

Table 5: Results for Ionosphere and Congressional Voting Records data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
Ionosphere										
2	$0.24194 \cdot 10^4$	0.00	5.75	0.45	0.00	0.663	0.03	0.00	1.90	0.05
5	$0.18915 \cdot 10^4$	0.00	26.2	0.70	0.07	0.899	0.05	0.18	5.85	0.13
10	$0.15694 \cdot 10^4$	1.02	67.3	1.88	1.73	1.40	0.08	0.00	12.5	0.27
15	$0.14014 \cdot 10^4$	3.72	104	2.47	4.31	1.88	0.11	0.00	19.3	0.42
20	$0.12714 \cdot 10^4$	2.62	136	3.05	5.73	2.53	0.13	0.00	26.1	0.77
25	$0.11486 \cdot 10^4$	11.95	182	4.02	6.76	3.35	0.16	0.00	33.3	1.39
30	$0.10469 \cdot 10^4$	13.99	200	4.19	7.37	4.35	0.20	0.00	40.6	2.20
40	$0.85658 \cdot 10^3$	30.35	273	5.59	7.82	7.88	0.30	0.00	55.8	4.77
50	$0.70258 \cdot 10^3$	45.90	352	6.72	6.63	11.1	0.38	0.00	71.6	8.39
Congressional Voting Records										
2	$0.16409 \cdot 10^4$	0.00	7.77	0.28	0.12	1.00	0.02	0.12	2.91	0.03
5	$0.13371 \cdot 10^4$	0.00	37.5	0.39	1.02	1.60	0.05	1.02	9.15	0.11
10	$0.11312 \cdot 10^4$	1.12	95.8	1.48	1.33	2.84	0.08	0.00	19.7	0.20
15	$0.10089 \cdot 10^4$	1.42	134	1.73	0.00	4.72	0.13	0.17	30.7	0.31
20	$0.91445 \cdot 10^3$	6.11	174	2.30	1.40	6.25	0.17	0.00	41.9	0.44
25	$0.85032 \cdot 10^3$	5.87	209	2.38	2.03	7.55	0.22	0.00	53.0	0.58
30	$0.78216 \cdot 10^3$	12.31	238	2.73	2.73	10.1	0.27	0.00	64.8	0.73
40	$0.69412 \cdot 10^3$	18.36	291	3.20	3.32	15.2	0.38	0.00	87.1	1.16
50	$0.62451 \cdot 10^3$	25.72	351	3.69	4.35	19.9	0.48	0.00	111	1.84

times and it does once better and six times worse than the MGKM algorithm. Again the GKM algorithm requires less computational efforts than other two algorithms.

For Ionosphere data set the MS k-means and GKM algorithms find the best known (or near best known) solutions two times and MGKM algorithms finds those solutions nine times. For Congressional Voting Records data set the MS k-means and GKM algorithms find such solutions two times and the MGKM algorithm eight times. The MGKM algorithm significantly outperforms two other algorithms as the number of clusters increases.

Results from Table 6 show that the MS k-means algorithm cannot find the global solution when the number of clusters $k > 5$ in Breast Cancer data set and when $k > 10$ in Pima Indians Diabetes data set. For Breast Cancer data set the GKM algorithm does as same as the MGKM algorithm once, it does two times better and six times worse than the MGKM algorithm. For Pima Indians Diabetes data set the GKM algorithm does as same as the MGKM algorithm three times and it does four times better and two times worse than the MGKM algorithm. Again the GKM algorithm requires less computational efforts than other two algorithms.

For Breast Cancer data set the MS k-means and GKM algorithms find the best known or near best known solutions three times and the MGKM algorithm finds such solutions

Table 6: Results for Breast Cancer and Pima Indians Diabetes data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
Breast Cancer										
2	$0.19323 \cdot 10^5$	0.00	0.891	0.38	0.00	0.242	0.05	0.00	0.709	0.06
5	$0.13705 \cdot 10^5$	0.00	8.50	1.30	2.28	0.306	0.09	1.86	2.17	0.17
10	$0.10216 \cdot 10^5$	4.40	15.8	1.47	0.00	0.559	0.17	0.02	4.60	0.33
15	$0.87813 \cdot 10^4$	0.20	24.2	1.91	0.00	0.803	0.23	0.04	7.14	0.48
20	$0.77855 \cdot 10^4$	5.99	34.0	2.45	1.80	1.06	0.31	0.00	9.65	0.66
25	$0.69682 \cdot 10^4$	9.87	40.6	2.66	4.12	1.27	0.38	0.00	12.4	0.83
30	$0.64415 \cdot 10^4$	10.44	49.3	3.23	3.43	1.63	0.45	0.00	15.0	0.98
40	$0.56171 \cdot 10^4$	15.99	61.7	3.77	3.70	2.22	0.61	0.00	20.2	1.39
50	$0.49896 \cdot 10^4$	22.37	74.2	4.27	4.21	3.03	0.77	0.00	25.6	1.83
Pima Indians Diabetes										
2	$0.51424 \cdot 10^7$	0.00	2.30	1.13	0.00	0.318	0.06	0.00	0.909	0.09
5	$0.17370 \cdot 10^7$	0.00	10.6	1.58	0.14	0.440	0.13	0.14	2.81	0.22
10	$0.94436 \cdot 10^6$	0.00	30.4	2.75	0.36	0.646	0.20	0.36	5.98	0.41
15	$0.69725 \cdot 10^6$	2.30	46.5	3.73	0.00	1.06	0.30	0.03	9.36	0.59
20	$0.57438 \cdot 10^6$	3.50	56.1	3.94	0.00	1.53	0.39	0.36	12.8	0.80
25	$0.49058 \cdot 10^6$	5.75	66.5	4.61	0.00	2.20	0.52	0.53	16.3	0.98
30	$0.43641 \cdot 10^6$	10.65	77.2	5.28	1.84	2.53	0.59	0.00	19.9	1.22
40	$0.36116 \cdot 10^6$	13.77	106	6.61	0.00	4.02	0.83	0.51	27.0	1.70
50	$0.31439 \cdot 10^6$	20.16	120	7.09	0.24	5.31	1.06	0.00	34.1	2.28

eight times. For Pima Indians Diabetes data set the MS k-means algorithm finds such solutions three times, the GKM algorithm eight times and the MGKM algorithm nine times.

The MS k-means algorithm cannot find the global solution when the number of clusters $k > 10$ for TSPLIB1060 data set and $k > 2$ for Image Segmentation data set (Table 7). For TSPLIB1060 data set the GKM algorithm does as same as the MGKM algorithm two times, it does two times better and five times worse than the MGKM algorithm. For Image Segmentation data set the GKM algorithm does as same as the MGKM algorithm five times and it does two times better and two times worse than the MGKM algorithm. Again the GKM algorithm requires less computational efforts than two other algorithms.

For TSPLIB1060 data set the MS k-means algorithm finds the best known (or near best known) solutions two times, the GKM algorithm five times and the MGKM algorithm six times. For Image Segmentation data set the MS k-means algorithm finds such solutions only once, the GKM algorithm six times and the MGKM algorithm five times.

The MS k-means algorithm again cannot find the global solution when the number of clusters $k > 10$ for TSPLIB3038 data set and $k \geq 2$ for Page Blocks data set (Table 8). For TSPLIB3038 data set the GKM algorithm does as same as the MGKM algorithm

Table 7: Results for TSPLIB1060 and Image Segmentation data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
TSPLIB1060										
2	$0.98319 \cdot 10^{10}$	0.00	1.71	0.75	0.00	0.580	0.08	0.00	1.71	0.08
10	$0.17548 \cdot 10^{10}$	0.05	53.1	2.36	0.23	1.45	0.36	0.05	11.6	0.34
20	$0.79179 \cdot 10^{9}$	8.74	93.9	2.78	1.88	2.96	0.69	1.88	24.3	0.66
30	$0.48125 \cdot 10^{9}$	4.91	123	3.14	3.34	4.70	1.03	3.37	37.3	0.97
40	$0.35312 \cdot 10^{9}$	8.23	141	3.48	1.14	6.45	1.38	0.00	50.3	1.30
50	$0.25551 \cdot 10^{9}$	21.17	167	3.95	3.10	8.92	1.73	2.53	64.2	1.69
60	$0.20443 \cdot 10^{9}$	22.11	199	4.58	0.72	11.3	2.08	0.00	78.0	2.06
80	$0.13535 \cdot 10^{9}$	33.51	251	5.47	0.00	17.2	2.80	0.06	107	2.89
100	$0.10041 \cdot 10^{9}$	52.12	281	5.94	0.10	22.7	3.53	0.00	135	3.75
Image Segmentation										
2	$0.35606 \cdot 10^{8}$	0.00	6.49	11.59	0.00	2.71	1.06	0.00	8.04	1.39
10	$0.97952 \cdot 10^{7}$	2.25	80.3	15.95	1.76	3.67	3.97	1.76	51.6	6.75
20	$0.51283 \cdot 10^{7}$	14.06	188	20.58	0.09	6.36	7.58	1.49	108	13.11
30	$0.35076 \cdot 10^{7}$	14.52	270	23.83	0.06	12.5	11.36	0.06	167	20.89
40	$0.27398 \cdot 10^{7}$	21.56	339	26.59	1.25	17.1	16.67	1.24	225	28.92
50	$0.22249 \cdot 10^{7}$	27.33	423	30.55	2.41	22.8	18.73	2.41	283	37.72
60	$0.19095 \cdot 10^{7}$	35.21	493	33.33	0.00	29.7	22.50	0.86	343	46.91
80	$0.14440 \cdot 10^{7}$	45.87	659	39.47	0.93	45.9	30.19	0.00	466	68.81
100	$0.11512 \cdot 10^{7}$	50.03	805	45.17	0.92	63.8	38.00	0.00	589	93.69

two times, it does three times better and four times worse than the MGKM algorithm. For Page Blocks data set the GKM algorithm does three times better and six times worse than the MGKM algorithm. The MGKM algorithm requires less CPU time than other two algorithms for both data sets.

For TSPLIB3038 data set the MS k-means algorithm finds the best known (or near best known) solutions three times, the GKM algorithm four times and the MGKM algorithm seven times. For Block Pages data set the MS k-means algorithm finds such solutions only once, the GKM algorithm eight times and the MGKM algorithm nine times.

Overall on 14 data sets, the GKM algorithm does as same as the MGKM algorithm 50 (39.7 %) times, it does 25 (19.8 %) times better and 51 (40.5 %) times worse than the MGKM algorithm. The MS k-means algorithm finds the best known (or near best known) solutions 42 (33.3 %) times, the GKM algorithm 76 (60.3 %) times and the MGKM algorithm 102 (81.0 %) times.

The following results clearly demonstrate that the MGKM algorithm is better than two other algorithms at computing large number of clusters ($k \geq 25$) in larger data sets ($m > 150$). Indeed, in this case the GKM algorithm does as same as the MGKM algorithm 3 (6.3 %) times, it does 12 (25.0 %) times better and 33 (68.7 %) times worse than the

Table 8: Results for TSPLIB3038 and Page Blocks data sets

k	f_{opt}	MS k-means			GKM			MGKM		
		E	α	t	E	α	t	E	α	t
TSPLIB3038										
2	$0.31688 \cdot 10^{10}$	0.00	0.860	12.97	0.00	0.469	1.38	0.00	1.39	0.86
10	$0.56025 \cdot 10^{9}$	0.00	14.2	11.52	2.78	0.857	8.41	0.58	9.16	3.30
20	$0.26681 \cdot 10^{9}$	0.42	37.1	14.53	2.00	1.60	16.63	0.48	19.2	5.77
30	$0.17557 \cdot 10^{9}$	1.16	57.8	19.09	1.45	2.97	25.00	0.67	29.5	8.25
40	$0.12548 \cdot 10^{9}$	2.24	74.6	22.28	1.35	3.98	33.23	1.35	39.9	10.70
50	$0.98400 \cdot 10^{8}$	2.60	84.5	23.55	1.19	5.26	41.52	1.41	50.5	13.23
60	$0.82006 \cdot 10^{8}$	5.56	103	27.64	0.00	6.39	49.75	0.98	61.0	15.75
80	$0.61217 \cdot 10^{8}$	4.84	119	30.02	0.00	9.56	66.42	0.63	82.9	20.94
100	$0.48912 \cdot 10^{8}$	5.99	138	33.59	0.59	12.9	83.16	0.00	105	26.11
Page Blocks										
2	$0.57937 \cdot 10^{11}$	0.24	1.82	577.05	0.24	1.50	8.19	0.00	4.50	6.92
10	$0.45662 \cdot 10^{10}$	206.38	42.3	168.45	0.80	1.66	49.62	0.00	28.6	34.09
20	$0.17139 \cdot 10^{10}$	70.44	259	367.39	0.00	2.30	92.30	0.19	59.3	62.09
30	$0.94106 \cdot 10^{9}$	399.77	452	417.28	0.75	3.15	132.41	0.00	90.1	89.42
40	$0.62570 \cdot 10^{9}$	485.89	641	477.88	0.17	4.22	172.13	0.00	121	118.55
50	$0.42937 \cdot 10^{9}$	725.19	760	503.03	0.04	5.86	212.27	0.00	152	149.77
60	$0.31185 \cdot 10^{9}$	1057.99	920	571.77	0.00	10.1	254.88	0.33	185	184.06
80	$0.20576 \cdot 10^{9}$	1647.96	889	513.25	1.46	14.2	334.36	0.00	250	258.69
100	$0.14545 \cdot 10^{9}$	998.80	796	443.64	0.00	20.5	415.19	0.10	316	346.94

MGKM algorithm. The MS k-means algorithm failed to find the best known (or near best known) solutions, the GKM algorithm finds such solutions 22 (45.8 %) times and the MGKM algorithm 42 (87.5 %) times.

Thus, these results allow us to draw the following conclusions:

1. The MS k-means algorithm is not effective at computing even moderately large number of clusters in large data sets.

2. Three algorithms, considered in this paper, are different versions of the k-means algorithm. Their main difference is in the way they compute starting points. In the MS k-means algorithm starting points are chosen randomly, however in two other algorithms special schemes are applied to find them. Results of numerical experiments show that the MGKM algorithm is more effective than two other algorithms at finding good starting points.

3. There is no any significant difference between the results of the GKM and MGKM algorithms on small data sets. However, the GKM requires significantly less compu-

tational efforts.

4. The MGKM algorithm works better than the GKM algorithm for large data sets and for large number of clusters ($k \geq 25$). The MGKM algorithm is especially effective for data sets such as Ionosphere, Congressional Voting Records, Liver Disorders data sets, which do not have well separated clusters.

6 Conclusions

In this paper, we have developed the new version of the global k-means algorithm, the modified global k-means algorithm. This algorithm computes clusters incrementally and to compute k-partition of a data set it uses $k-1$ cluster centers from the previous iteration. An important step in this algorithm is the computation of a starting point for the k-th cluster center. This starting point is computed by minimizing the so-called auxiliary cluster function. The proposed algorithm computes as many clusters as a data set contains with respect to a given tolerance.

We have presented the results of numerical experiments on 14 data sets. These results clearly demonstrate that the multi-start k-means algorithm cannot be alternative to both the global k-means and the modified global k-means algorithms when the number of clusters $k > 5$. The results presented also demonstrate that the modified global k-means algorithm is more effective than the global k-means algorithm at computing of large number of clusters in large data sets. However, the former algorithm requires more CPU time than the latter one. Results presented in this paper again confirms that the choice of starting points in k-means algorithms is crucial.

Acknowledgements

Dr. Adil Bagirov is the recipient of an Australian Research Council Australian Research Fellowship (Project number: DP 0666061).

The author thanks an anonymous referee for comments and criticism that significantly improved the quality of the paper.

References

[1] K.S. Al-Sultan, A tabu search approach to the clustering problem, *Pattern Recognition*, 28(9)(1995) 1443-1451.

[2] A.M. Bagirov, A.M. Rubinov, J. Yearwood, A global optimisation approach to classification, *Optimization and Engineering,* 3(2)(2002) 129-155.

[3] A.M. Bagirov, A.M. Rubinov, N.V. Soukhoroukova, J. Yearwood, Supervised and unsupervised data classification via nonsmooth and global optimization, *TOP: Spanish Operations Research Journal,* 11(1)(2003) 1-93.

[4] A.M. Bagirov, J. Yearwood, A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, *European Journal of Operational Research,* 170(2006) 578-596.

[5] H.H. Bock, Clustering and neural networks, in: A. Rizzi, M. Vichi, H.H. Bock (eds), *Advances in Data Science and Classification*, Springer-Verlag, Berlin, 1998, pp. 265-277.

[6] D.E. Brown, C.L. Entail, A practical application of simulated annealing to the clustering problem, *Pattern Recognition*, 25(1992) 401-412.

[7] O. du Merle, P. Hansen, B. Jaumard, N. Mladenovic, An interior point method for minimum sum-of-squares clustering, *SIAM J. on Scientific Computing,* 21(2001) 1485-1505.

[8] G. Diehr, Evaluation of a branch and bound algorithm for clustering, *SIAM J. Scientific and Statistical Computing*, 6(1985) 268-284.

[9] R. Dubes, A.K. Jain, Clustering techniques: the user's dilemma, *Pattern Recognition*, 8(1976) 247-260.

[10] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugenics,* VII part II (1936) 179-188. Reprinted in: Fisher R.A. *Contributions to Mathematical Statistics,* Wiley, 1950.

[11] P. Hanjoul, D. Peeters, A comparison of two dual-based procedures for solving the p-median problem, *European Journal of Operational Research,* 20(1985) 387-396.

[12] P. Hansen, B. Jaumard, Cluster analysis and mathematical programming, *Mathematical Programming,* 79(1-3)(1997) 191-215.

[13] P. Hansen, N. Mladenovic, J-means: a new heuristic for minimum sum-of-squares clustering, *Pattern Recognition*, 4(2001) 405-413.

[14] P. Hansen, N. Mladenovic, Variable neighborhood decomposition search, *Journal of Heuristic,* 7(2001) 335-350.

[15] Hansen P., Ngai E., Cheung B.K., Mladenovic N. (2002) Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering, *Les Cahiers du GERAD, G-2002-43*, 2002. (to appear in: *J. of Classification*).

[16] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, *ACM Computing Surveys,* 31(3)(1999) 264-323.

[17] W.L.G. Koontz, P.M. Narendra, K. Fukunaga, A branch and bound clustering algorithm, *IEEE Transactions on Computers*, 24(1975) 908-915.

[18] A. Likas, M. Vlassis, J. Verbeek, The global k-means clustering algorithm, *Pattern Recognition*, 36(2003) 451-461.

[19] UCI repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.html.

[20] G. Reinelt, TSP-LIB-A Traveling Salesman Library, *ORSA J. Comput.* 3(1991), 319-350.

[21] S.Z. Selim, K.S. Al-Sultan, A simulated annealing algorithm for the clustering, *Pattern Recognition*, 24(10)(1991) 1003-1008.

[22] H. Spath, *Cluster Analysis Algorithms*, Ellis Horwood Limited, Chichester, 1980.

[23] L.X. Sun, Y.L. Xie, X.H. Song, J.H. Wang, R.Q. Yu, Cluster analysis by simulated annealing, *Computers and Chemistry,* 18(1994) 103-108.

7 Appendix

Proof of Proposition 1: Since $S_1(\bar{x}) = \emptyset$ we get that

$$\bar{f}_k(\bar{x}) = \frac{1}{m} \sum_{a^i \in S_2(\bar{x})} \|\bar{x} - a^i\|^2 + \frac{1}{m} \sum_{a^i \in S_3(\bar{x})} d^i_{k-1}. \tag{35}$$

It is clear that $\bar{x}$ is a global minimizer of the convex function

$$\Phi(x) = \frac{1}{m} \sum_{a^i \in S_2(\bar{x})} \|x - a^i\|^2 \tag{36}$$

that is $\Phi(\bar{x}) \leq \Phi(x)$ for all $x \in \mathbb{R}^n$.

Let $B_\varepsilon(\bar{x}) = \{y \in \mathbb{R}^n : \ \|y - \bar{x}\| < \varepsilon\}$. There exists $\varepsilon > 0$ such that

$$\|x - a^i\|^2 < d^i_{k-1} \quad \forall\ a^i \in S_2(\bar{x}) \ \text{ and } \ \forall\ x \in B_\varepsilon(\bar{x}), \tag{37}$$

$$\|x - a^i\|^2 > d^i_{k-1} \quad \forall\ a^i \in S_3(\bar{x}) \ \text{ and } \ \forall\ x \in B_\varepsilon(\bar{x}). \tag{38}$$

Then for any $x \in B_\varepsilon(\bar{x})$ we have

$$\begin{aligned} \bar{f}_k(x) &= \frac{1}{m} \sum_{a^i \in S_2(\bar{x})} \|x - a^i\|^2 + \frac{1}{m} \sum_{a^i \in S_3(\bar{x})} d^i_{k-1} \\ &= \Phi(x) + \frac{1}{m} \sum_{a^i \in S_3(\bar{x})} d^i_{k-1} \\ &\geq \Phi(\bar{x}) + \frac{1}{m} \sum_{a^i \in S_3(\bar{x})} d^i_{k-1} \\ &= \bar{f}_k(\bar{x}). \end{aligned} \tag{39}$$

Thus $\bar{f}_k(x) \geq \bar{f}_k(x)$ for all $x \in B_\varepsilon(\bar{x})$. △