J Glob Optim (2015) 63:427–443
DOI 10.1007/s10898-014-0175-1

New heuristic for harmonic means clustering

Emilio Carrizosa · Abdulrahman Alguwaizani · Pierre Hansen · Nenad Mladenović

Received: 11 December 2012 / Accepted: 17 March 2014 / Published online: 6 May 2014

Abstract It is well known that some local search heuristics for K-clustering problems, such as k-means heuristic for minimum sum-of-squares clustering occasionally stop at a solution with a smaller number of clusters than the desired number K. Such solutions are called degenerate. In this paper, we reveal that the degeneracy also exists in K-harmonic means (KHM) method, proposed as an alternative to K-means heuristic, but which is less sensitive to the initial solution. In addition, we discover two types of degenerate solutions and provide examples for both. Based on these findings, we give a simple method to remove degeneracy during the execution of the KHM heuristic; it can be used as a part of any other heuristic for KHM clustering problem. We use KHM heuristic within a recent variant of variable neighborhood search (VNS) based heuristic. Extensive computational analysis, performed on test instances usually used in the literature, shows that significant improvements are obtained if our simple degeneracy correcting method is used within both KHM and VNS. Moreover, our VNS based heuristic suggested here may be considered as a new state-of-the-art heuristic for solving KHM clustering problem.

E. Carrizosa
Faculdad de Matematicas, Universidad de Seville, Seville, Spain
e-mail: ecarrizosa@us.es

A. Alguwaizani (✉)
Math department, King Faisal University, AlAhsa , Kingdom of Saudi Arabia
e-mail: aalguwaizani@kfu.edu.sa

P. Hansen
GERAD and École des Hautes Etudes Commerciales, 3000 ch. de la Cote-Sainte-Catherine, Montreal H3T 2A7, Canada
e-mail: pierreh@crt.umontreal.ca

N. Mladenović
LAMIH, University of Valenciennes, Le Mont Houy, 59313 Valenciennes CEDEX 9, France
e-mail: nenad@mi.sanu.ac.rs

Keywords Clustering · K-harmonic means heuristic · Variable neighborhood search · Degeneracy

1 Introduction

Let $E = \{e_1, \ldots, e_N\}$ be a set of N entities or objects to be clustered ($e_i \in \mathbb{R}^q$), and let C be a subset of E. Then $P_K = \{C_1, C_2, \ldots, C_K\}$ is a partition of E into K clusters if it satisfies: (i) $C_k \neq \emptyset$; $k = 1, 2, \ldots, K$, (ii) $C_i \cap C_j = \emptyset$; $i, j = 1, 2, \ldots, K$; $i \neq j$, and (iii) $\bigcup_{k=1}^{K} C_k = E$. One of the most popular models for partitioning points in Euclidean space is the minimum-sum-of squares clustering (MSSC) model [2,3,9,14,15]. It considers simultaneously the homogeneity and the separation criteria. centroid. The partition that minimizes the sum of squared distances from the entities to the centroid of their cluster is searched. Then the MSSC can be expressed as follows:

$$f_{MSSC}(P) = \min_{P \in \mathcal{P}_K} \sum_{i=1}^{N} \min_{j=1,\ldots,K} \|e_i - c_j\|^2, \tag{1}$$

where $\mathcal{P}$ denotes the set of all K-partitions of I, and c_j the centroid of cluster j i.e.,

$$c_j = \frac{1}{|C_j|} \sum_{i \in C_j} e_i.$$

The most popular heuristics for solving minimum sum-of-squares clustering (MSSC) alternate (ALT) between keeping one type of variables fixed while determining the other type: for fixed centroid points, the best assignments of entities to clusters are found and then, for a given N allocations of entities to clusters, the best centroid points are found. Such heuristics are known as ALT heuristics. Used for solving MSSC, the ALT heuristic is called K-means (see Algorithm 1).

Function KM ($E, K, Maxit$)
Choose initial centroids c_k ($k = 1, \ldots, K$)
$l \leftarrow 0$
repeat
  $l \leftarrow l + 1$
  for $i := 1, \ldots, n$ **do**
    $m(e_i) \leftarrow \text{argmin}_{j \in \{1,\ldots,K\}} (\|e_i - c_j\|_2)^2$
  $z = f_{MSSC}$ as in (1)
  for $j := 1, \ldots, K$ **do**
    Calculate centroid c_j
until *m does not change or* l *=Maxit*

Algorithm 1: K-Means heuristic (KM) for the MSSC problem

Most of the ALT heuristics have an undesirable property known as degeneracy [6,19]: one or more clusters become empty during their execution. In other words, the better solution in the next iteration of ALT may be found but with smaller number of clusters. Clearly, such solutions may easily be improved by adding a new centroid to the position of an entity that does not already coincide with some centroid. Papers investigating the reason for a poor performance of ALT heuristics for some data sets are devoted to initialization of K means, which is still a subject of debate (see e.g. [4,8,12,22,24]).

K-harmonic means clustering problem (KHMCP) does not depend much on the choice of its initial solution. This fact is empirically confirmed in the next section. A question that naturally arises is then, whether K-harmonic means (KHM) heuristic, the most popular heuristic for solving KHM problem suffers from degeneracy as well, and if so, to what degree.

In this paper, we show that the KHM heuristic suffers from two types of degenerate solutions: (i) Type-1, there are some cluster centroids without any entities allocated to them; (ii) Type-2,there are two cluster centroids that coincide or are at a distance smaller than an arbitrary small number ε. We suggest an efficient and fast method for removing empty clusters immediately when they occur in KHM heuristic. Since the KHMCP belongs to non-convex optimization problems, several heuristics have been proposed in the literature for solving it [1,17,26,27]. Among them, VNS based heuristic proposed recently in [1] can be considered as the state-of-the-art method. In this paper, our simple removing degeneracy procedure is introduced into the VNS based heuristic [1].

Surprisingly, it significantly improves the quality of the final solution obtained by VNS. We believe that it could also be successfully embedded into other recent heuristic, such as [17,26,27]. In order to understand degeneracy better, and to show importance of its immediate removal, we performed an extensive computational analysis on test instances from the literature. It shows that the degeneracy may also harm the quality of the final KHMCP solution, but less than K-means heuristic for solving MSSCP.

The paper is organized as follows. In the next section, we give for completeness, pseudocode for the ALT procedures for solving the KHMCP. In the same section, we show empirically that KHM is indeed less sensitive than KM to the choice of the initial solution. Also, we prove by constructing examples that KHM could stop at the degenerate solutions of both Type-1 and Type-2. At the end of this section, we propose a method for removing degeneracy. In Sect. 3, we show the impact of removing degeneracy on efficiency of variable neighborhood search (VNS) and Multi-start KHM local search (MLS) heuristics. In Sect. 4, we perform extensive computational analysis. Section 5 concludes the paper.

2 Degeneracy of K-harmonic means clustering

K-harmonic means clustering problem. In the KHMCP, the sum of harmonic averages of the distances between each entity and all centroids is minimized:

$$f_{KHM}(P) = \min_{P\in\mathcal{P}_K} \sum_{i=1}^{N} \frac{K}{\sum_{j=1}^{K} \frac{1}{\|e_i - c_j\|^p}}, \quad \forall \quad i = 1, \ldots, N, \tag{2}$$

and parameter p is a power of the Euclidean norm, used as a distance function. P and $\mathcal{P}_K$ have the same meaning as in (1).

K-harmonic means (KHM) *heuristic.* The most popular heuristic for solving KHMCP is one of the ALT type, here referred to as the KHM [13,29]. For the sake of completeness, let us recall the steps of this heuristic. The set of variables is divided into a set of cluster centroids and a set of membership variables for each entity and each cluster. KHM uses a weight function which allows that the same entities can belong to different clusters. A weight function w_i, recalled below, determines the weight of each entity with respect to harmonic means. In contrast to the K-means algorithm (KM) (see Algorithm 1), where equal weight (i.e. w_i=1) are given to all data, in the KHM algorithm, the weights at each step vary. Another function used in the

KHM algorithm is called the membership function m_{ij}, which assigns each entity or point e_i to a cluster c_j. This function should satisfy the following:

$$(i)\ \sum_{j=1}^{K} m_{ij} = 1 \quad \forall \quad i = 1, \ldots, N;$$
$$(ii)\ \ 0 \le m_{ij} \le 1 \quad \forall \quad i = 1, \ldots, N, \quad \forall \quad j = 1, \ldots, K.$$

The membership function and the weight function are defined as follows:

$$m_{ij} = m_{KHM}(c_j/e_i) = \frac{\|e_i - c_j\|^{-p-2}}{\sum_{l=1}^{K} \|e_i - c_l\|^{-p-2}}, \quad \forall \quad i = 1, \ldots, N, \quad \forall \quad j = 1, \ldots, K \tag{3}$$

$$w_i = w_{KHM}(e_i) = \frac{\sum_{j=1}^{K} \|e_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{K} \|e_i - c_j\|^{-p}\right)^2}, \quad \forall \quad i = 1, \ldots, N \quad , \tag{4}$$

where the centroids are given by the formula [13,28,29]:

$$c_j^{(new)} = \frac{\sum_{i=1}^{N} m_{KHM}(c_j/e_i) \cdot w_{KHM}(e_i) \cdot (e_i)}{\sum_{i=1}^{N} m_{KHM}(c_j/e_i) \cdot w_{KHM}(e_i)}, \quad \forall \quad j = 1, \ldots, K \quad . \tag{5}$$

The local search algorithm KHM starts by generating K distinct centroids randomly, chosen among the given entities $e_i \quad (i = 1, \ldots, N)$. Then, new centroids are obtained from Eqs. (3), (4), and (5). This process is repeated until the difference between the centroids in two consecutive iterations is less than ε (a small number)or a maximum number of iterations is reached (see Algorithm 2).

Function KHM $(E, K, C, Maxit, \varepsilon)$
- $C^{(new)} = \{c_1, c_2, \ldots, c_K\}$ // K centroids are chosen from E
- $\ell \leftarrow 0$ // ℓ-iteration counter
- **repeat**
  - $\ell \leftarrow \ell + 1$; $C \leftarrow C^{(new)}$
  - $z \leftarrow f_{KHM}(C)$ as in (2)
  - **for** $i = 1, \ldots, n$ **do**
    - **for** $j = 1, \ldots, K$ **do**
      - Calculate $m(c_j/e_i)$ as in (3)
    - Calculate $w(e_i)$ as in (4)
  - **for** $i = 1, \ldots, n$ **do**
    - **for** $j = 1, \ldots, K$ **do**
      - Find new centroids $c_j^{(new)}$, as in (5)
- **until** $(\|c_j^{(new)} - c_j\| \le \varepsilon,\ \forall j = 1, \ldots, K)$ *or* $\ell = Maxit$

Algorithm 2: The local search algorithm KHM for KHMCP

Sensitivity on initial solution. As mentioned before, the KHMCP is introduced to reduce the sensitivity of choosing the initial centroids of the MSSC [18,28,29]. To check this, we performed computational analysis on several well-known test instances (more detailed

Table 1 MSSC objective functions for KM and KHM partitions obtained in 100 restarts

DATASET	DIM	M	ALG	WORST-SOL	BEST-SOL	DIFF
Ruspini (75)	2	3	KM	50,298.04	10,126.72	40,171.32
			KHM	12,415.12	10,126.72	2,288.39
Iris (150)	4	3	KM	145.53	78.85	66.68
			KHM	78.85	78.85	0.00
Wine (178)	13	3	KM	2,633,555.33	2,370,689.69	262,865.64
			KHM	2,371,841.59	2,371,841.59	0.00
Glass (214)	9	2	KM	1,240.11	819.63	420.48
			KHM	820.03	819.63	0.40
B-Cancer (699)	10	50	KM	7,700.88	6,112.12	1,588.76
			KHM	7,298.29	5,954.09	1,344.20
		100	KM	5,853.25	4,348.77	1,504.48
			KHM	5,028.09	4,348.92	679.17
TSP (1060)	2	50	KM	349,545,617.68	275,703,293.57	73,842,324.11
			KHM	293,226,666.88	257,897,808.70	35,328,858.18
		100	KM	157,827,133.11	111,301,083.09	46,526,050.02
			KHM	122,563,406.21	102,361,445.84	20,201,960.37
I-Segmentation (2310)	19	50	KM	4,182,208.78	2,819,337.21	1,362,871.57
			KHM	3,182,598.13	2,294,420.45	888,177.68
		100	KM	2,908,213.19	1,839,231.27	1,068,981.92
			KHM	1,947,788.50	1,340,153.25	607,635.25
TSP (3038)	2	50	KM	113,402,496.67	99,913,944.85	13,488,551.83
			KHM	105,470,392.38	100,278,655.58	5,191,736.80
		100	KM	58,159,312.92	50,568,302.39	7,591,010.53
			KHM	50,614,550.76	48,540,001.30	2,074,549.46

description of these test instances is given below, in Sect. 4). In Table 1, we show the differences between the worst and the best objective function values obtained with 100 restarts of KM and KHM heuristics in turn. Since the objective functions of these two problems are different, we took crisp partitions obtained from the solution of KHM by setting to 1 the variable with the largest membership value in each column and 0 otherwise, and then computing the corresponding MSSC objective function values. In that way, we were able to compare the influence of the initial solutions on the final solutions of KM and KHM.

The first and the second columns in Table 1 display the number of entities and the corresponding dimension of data set, respectively. The desired number of clusters is shown in column 3. In columns 5 and 6, we show the best and the worst values out of 100 restarts. The last column gives the difference between the worst (largest) and the best (smallest) values obtained.

Table 1 confirms that the final solution of KHM is not as sensitive as the KM to the choice of the initial solution, since the differences between the worst and the best solutions obtained by KHM are much smaller than the differences obtained by the KM heuristics. Note also that in some cases, better objective function values are obtained with KHM despite the fact that MSSC problems are considered (see, e.g., the TSP-1060 dataset).

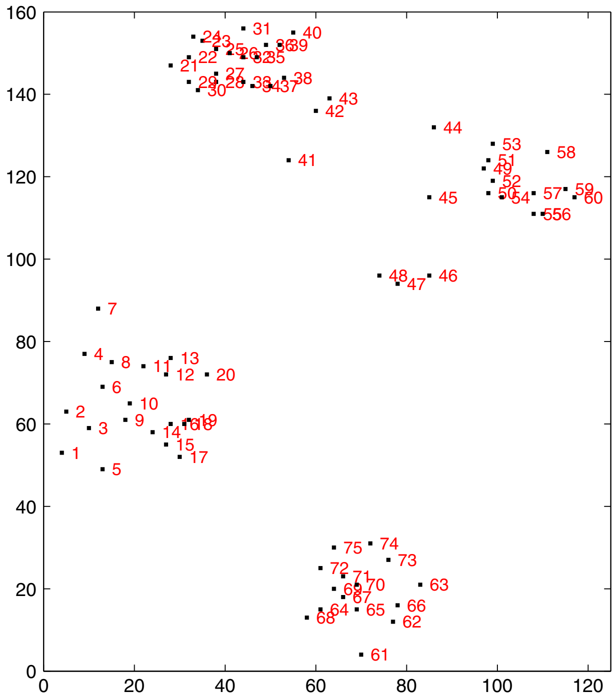


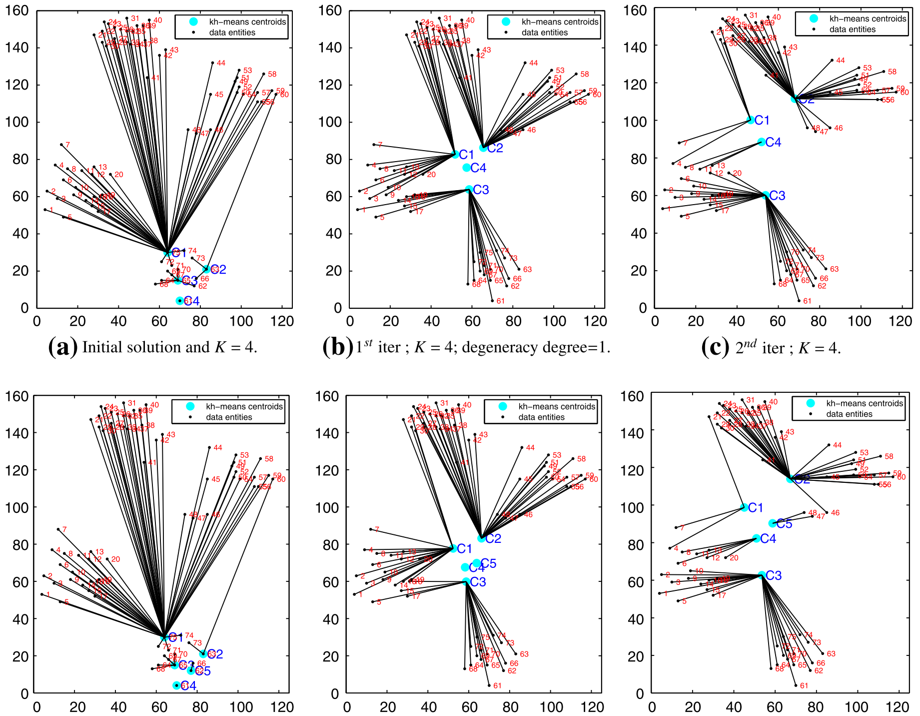
Fig. 1 Ruspini data set

2.1 Degeneracy of `KHM`

In this subsection we show, by counter-example, that the solution obtained by `KHM` could also be degenerate. First, we distinguish 2 types of degeneracy. We can say that the solution of the clustering problem is *degenerate Type-1* if there is one, or more cluster centres without allocated entities. We can say that the solution of the clustering problem is *degenerate Type-2* if there exist at least two identical cluster centres. We also define the degree d of degeneracy [6] as the number of empty clusters.

Type-1 degeneracy of `KHM`. We illustrate *degenerate Type-1* on the following well-known Ruspini data set [23] (entities are 75 points in the plane, as given in Fig. 1). We show that a degenerate solution of Type-1 may occur even in the first iteration of the `KHM` algorithm if $K = 4$. Indeed, if the initial cluster centres are located at entities 75, 63, 65 and 61 (see Fig. 2a), then after the allocation step, the objective function of such a proper solution is $f = 669,408.938$. Entities are divided into 4 clusters, as follows: 63 entities $\{1, 2, \ldots, 59, 71, 72, 74, 75\}$ are the closest to entity 75; 4 entities $\{60, 63, 66, 73\}$ are the closest to entity 63; 7 entities $\{62, 64, 65, 67, 68, 69, 70\}$ are the closest to entity 65. The last entity, i.e. 61 is a cluster by itself. The next step shows the degeneracy in cluster 4 (see Fig. 2b). It is interesting to note that Type-1 degeneracy could appear in the `KHM` algorithm and then, it can disappear without applying additional rules. Indeed, it happens in our example in the next iteration, as shown in Fig. 2c. However, the objective function is almost 3 times smaller: 252,499.813.

Degeneracy of degree 2 also exists if $K = 5$. It is shown in Fig. 2d, where we use the same initial solution as in Fig. 2a but entity 62 is added as the cener of the fifth cluster. Thus,



(a) Initial solution and $K = 4$. **(b)** 1^{st} iter ; $K = 4$; degeneracy degree=1. **(c)** 2^{nd} iter ; $K = 4$.

(d) Initial solution and $K = 5$. **(e)** 1^{st} iter ; $K = 5$; degeneracy degree=2. **(f)** 2^{nd} iter ; $K = 5$.

Fig. 2 KHM clustering degeneracy for the *Ruspini* dataset

the degenerate solution is already obtained in the first iteration (see Fig. 2e). However, it is removed from the solution as before, in the next step of KHM. Therefore, type 1 degeneracy may be automatically corrected during the execution of KHM. But for many other data sets, the degenerate solutions are significantly affected at the end of the local search, as explained in Sect. 2.2.

Although the KHM heuristic eventually reduces Type-1 degeneracy automatically, we show in Sect. 4 that it is better to remove degeneracy immediately when it appears. Moreover, in this way, the number of iterations of the new KHM is reduced as well (see Table 2).

Type-2 degeneracy of KHM. The following example illustrates the degeneracy of Type-2 for KHM local search. The position of entities $e(i, \ell)$ and the initial solution $c(j, \ell)$ are as follows:

$$E = \begin{pmatrix} 0.5 & 0 \\ 1 & 1 \\ 1 & 5 \\ 1 & -5 \\ 1.5 & 0 \end{pmatrix}; \qquad C = \begin{pmatrix} 1 & 5 \\ 1 & 1 \\ 1.5 & 0 \\ 0.5 & 0 \end{pmatrix}.$$

The initial solution of this step is shown in Fig. 3a. The next step is to calculate the objective function, as in (2), to get new centroids as in (5), and to calculate the membership and

Table 2 Comparison between methods KHM and KHM+ on Ruspini dataset

K	mth	Obj	Dev %	Maxit	Type	Maxdeg	Time
4	KHM	42,980.7852	0.000	10	1	1	0.062
4	KHM+	42,980.7812		9			0.000
5	KHM	41,442.8750	0.000	23	1	2	0.016
5	KHM+	41,442.8711		13			0.016
6	KHM	38,989.2109	0.000	45	1	2	0.016
6	KHM+	38,989.2109		28			0.016
7	KHM	40,957.8125	2.577	59	1	2	0.016
7	KHM+	39,928.8477		60			0.031
8	KHM	35,056.9453	1.431	42	1	3	0.016
8	KHM+	34,562.2109		40			0.016
9	KHM	32,716.4531	0.000	34	1	4	0.031
9	KHM+	32,716.4512		41			0.031
10	KHM	32,406.1074	10.126	42	2	6	0.047
10	KHM+	29,426.3652		34			0.031
11	KHM	30,778.1641	4.234	41	2	7	0.016
11	KHM+	29,527.8652		65			0.047
12	KHM	30,869.2480	0.394	41	2	8	0.016
12	KHM+	30,748.0254		80			0.047
13	KHM	31,482.8633	7.965	41	2	7	0.047
13	KHM+	29,160.1875		107			0.062
14	KHM	36,413.9570	3.864	59	2	8	0.078
14	KHM+	35,059.1758		30			0.031
15	KHM	37,569.1562	3.501	54	2	9	0.078
15	KHM+	36,298.4766		41			0.031

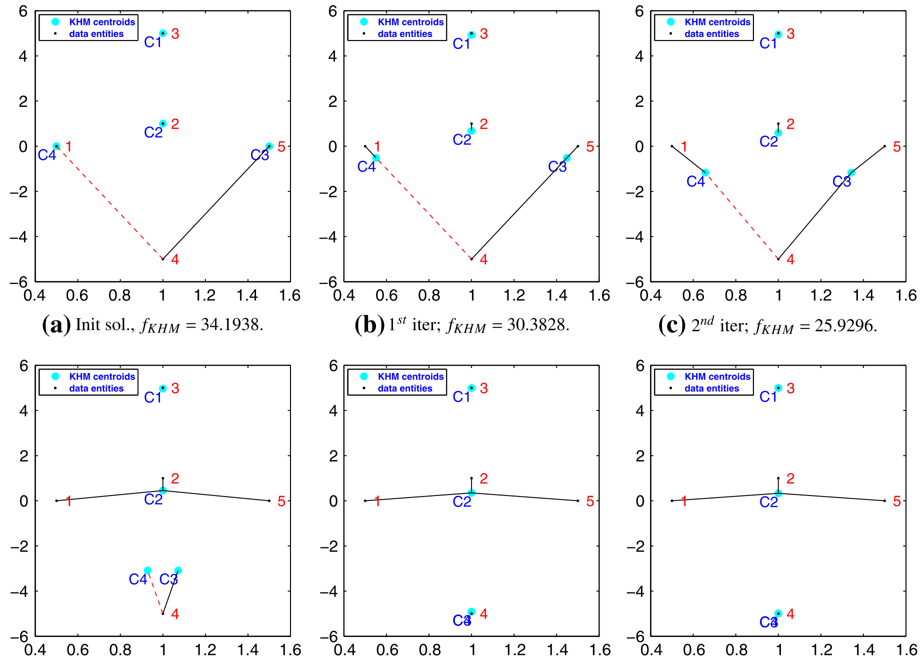
weight matrices as in (3) and (4). We choose $p = 2$ and $\varepsilon = 0.01$. The objective function is:

$$f_{KHM}(P)=\sum_{i=1}^{N} HA_i(K,P) = \sum_{i=1}^{N} \frac{4}{\|e_i-c_1\|^{-2}+\|e_i-c_2\|^{-2}+\|e_i-c_3\|^{-2}+\|e_i-c_4\|^{-2}}$$
$$= \frac{4}{10,001.84}+\frac{4}{10,001.66}+\frac{4}{10,000.14}+\frac{4}{0.12}+\frac{4}{10,001.84}$$
$$= 34.1938$$

By simple calculations, we get the membership matrix:

$$m_{KHM}(c_j/e_i) = \begin{pmatrix} 0.0000 & 0.0000 & 1.0000 & 0.0249 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.1925 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.3913 & 1.0000 \\ 1.0000 & 0.0000 & 0.0000 & 0.3913 & 0.0000 \end{pmatrix}.$$

The fact is that $m_{34} = m_{44} = 0.3913$ will cause future degeneracy. From this matrix, in fact, we can obtain a crisp clustering matrix, as follows:



(a) Init sol., f_{KHM} = 34.1938. **(b)** 1^{st} iter; f_{KHM} = 30.3828. **(c)** 2^{nd} iter; f_{KHM} = 25.9296. **(d)** 3^{rd} iter; f_{KHM} = 11.2037. **(e)** 4^{th} iter; f_{KHM} = 4.4748. **(f)** 5^{th} iter; f_{KHM} = 4.4587.

Fig. 3 KHM clustering degeneracy for dataset-2

$$M_1 = max_j(m_{KHM}(c_j/e_i)) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \text{or} \quad M_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

By repeating the same steps in the next iteration we get:

$$w_{KHM}(e_i) = \begin{pmatrix} 0.9996 & 0.9997 & 1.0000 & 0.2929 & 0.9996 \end{pmatrix}^T.$$

Now the new centroids, calculated from (5) are

$$C = \begin{pmatrix} 1.0000 & 4.9275 \\ 1.0000 & 0.6797 \\ 1.4486 & -0.5143 \\ 0.5514 & -0.5143 \end{pmatrix}.$$

The results are shown in Fig. 3b. We see in Fig. 3b, c that all five entities are clustered in 4 groups, as desired. But in Fig. 3d, two centroid points are almost joined in one cluster. In the rest of the Fig. 3e, f, we can see clearly how they coincide. This step implies that the degeneracy in this example is considered to be of type 2. The final solutionis:

$$C(final) = \begin{pmatrix} 1.0000 & 4.9953 \\ 1.0000 & 0.3291 \\ 1.0000 & -4.9891 \\ 1.0000 & -4.9891 \end{pmatrix}.$$

Note that entity 4 belongs to clusters 3 and 4 equally in the initial solution, as well as in all 5 iterations. At the end, cluster centroids 3 and 4 become identical, producing a solution with degeneracy of Type-2.

2.2 Removing degeneracy (KHM+)

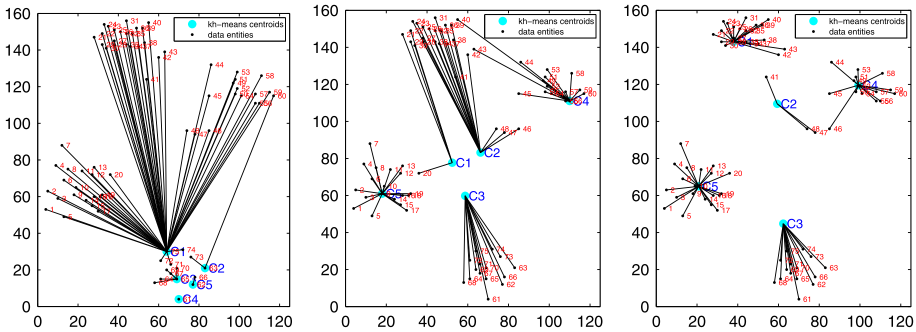
There are many efficient ways to remove degeneracy from the solution. Such procedures are found, for example in Cooper's ALT type algorithm for solving the Multi-Source Weber problem in [6,19]. For some data sets (for example in the *B-Cancer data set 699*), the degeneracy remains in all iterations of KHM, i.e., it does not automatically vanish as in the example in Fig. 2. This result lead us to design a heuristic for removing degeneracy immediately when it appears. Our pseudo-code is given in Algorithm 3. If a degeneracy of degree d occurs, our algorithm randomly selects d new distinct centroid points among the positions of the existing entities, which do not already coincide with existing centroids. Such a new solution is obviously not degenerate since all K centroids have at least one entity allocated to them, but may become degenerate again after the assignment step.

We also tested some different strategies for choosing the entities to be taken as a new centroid points. However, it appears that the most efficient rule is the random selection described above, although the solution qualities do not differ significantly. We found that the computing time for any deterministic search is long, and does not usually improve the quality of the final solution.

```
Function KHM+ (E, K, C, Maxit, ε)
C^(new) = {c_1, c_2, ..., c_K}          // K centroids are chosen from E at random
i ← 0                                     // i-iteration counter
repeat
    i ← i + 1; C ← C^(new)
    z ← f_KHM(C) as in (2)
    Calculate m as in (4) and w as in (5) for all entities
    Find new centroids c_j^(new), j = 1, ..., K as in (6)
    Find indices b_ℓ of degenerate clusters (ℓ = 1, ..., g), where g is the degree of degeneracy
    if (g > 0) then
        for ℓ := 1, ..., g do
            h = ⌊1 + N · rnd⌋      // choose an unoccupied entity h at random
            t ← b_ℓ ; c_t ← e_h     // replace degenerate centroid by chosen entity
until (‖c_j^(new) − c_j‖ ≤ ε, ∀j = 1, ..., K or i = Maxit)
```

Algorithm 3: KHM+ local search with removing degeneracy

It is clear that KHM+ converges to a local optimum in a finite number of steps. In fact, most of ALT type heuristics converge since they iterate until there is an improvement in the objective function value. Obviously, for the KHM clustering problem, the lower bound 0 exists and thus, the decreasing sequence of objective function values cannot tend to $-\infty$. The fact that the obtained solution is locally the best in continuous space is confirmed by the the



(a) Init solution; $f_{KHM} = 685329.3$. **(b)** 1^{st} it; $f_{KHM} = 440700.4$. **(c)** 2^{nd} it; $f_{KHM} = 97519.8$.

Fig. 4 KHM clustering for the *Ruspini* dataset after removing degeneracy

stopping condition $\|c_j^{(new)} - c_j\| \leq \varepsilon$. Since KHM+ is an ALT type heuristic, it converges as well.

By applying Algorithm 3, we can simply remove the degeneracy in the previous example for $k = 5$ in Sect. 3. Figure 4 shows the solutions obtained by our KHM+. Although the solutions obtained by KHM and KHM+ are both proper, after the second iteration, it appears that the objective function value of the former is more than twice larger that the latter one (compare $f_{KHM} = 212,390$ with $f_{KHM+} = 97,519$).

To make a precise comparison between KHM and KHM+, we use the same initial solutions for both heuristics. Table 2 contains results of the comparison of the two local searches on *Ruspini* data and different values of the number K of clusters. In column 4 of Table 2, we give the % difference between the two heuristics calculated as:

$$\frac{f_{KHM} - f_{KHM+}}{f_{KHM+}} \cdot 100 \tag{6}$$

In column 5, we report the number of iterations used, and in column 6 the type of degeneracy occurred.

3 VNS for KHM

Variable neighborhood search is a metaheuristic for solving combinatorial and global optimization problems whose basic idea is a systematic change of neighborhoods both within a descent phase, finding a local optimum, and in a perturbation phase,getting out of the corresponding valley. The efficiency of VNS is based on three simple facts: (i) A local minimum with respect to (w.r.t.) one neighborhood structure is not necessarily the same for another neighborhood structure; (ii) A global minimum is a local minimum w.r.t. all possible neighborhood structures; (iii) For many problems, the local minima w.r.t. one or several neighborhoods are relatively close to each other. The VNS metaheuristic is well-established in the literature. For an overview of the method and its numerous applications, the reader is referred to [7,20], and for the most recent surveys, to [16,21]. For solving KHMCP, the VNS based heuristic (VNS-KHM) has already been proposed in [1], but without degeneracy removal subroutine. For the sake of completeness, we repeat its steps in Algorithm 4.

In our VNS-KHM+, the initial solution is obtained by randomly selecting K centroid points among the existing entities. The method terminates when the given running time t_{max}

```
Function VNS+(X, K, k_max, t_max, C)
repeat
    k ← 1                                         // the neighborhood index
    repeat
        C' ← Shake(X, k, C)                       // Shaking
        C'' ← KHM+ (X, K, C', Maxit, ε)           // Local search
        NeighborhoodsChange(C, C'', k)            // Change centroid
    until k = k_max
    t ← CpuTime()
until t > t_max
```

Algorithm 4: Steps of the basic VNS+

is reached. The inner loop iterates until there is no better solution in the last neighborhood (k_{max}) of the incumbent solution C. The inner loop consists of 3 steps: Shaking, Local search and Neighborhood change. The only difference between the new VNS based heuristic suggested here (`VNS-KHM+`) and the `VNS-KHM` as in [1] is that `KHM+` local search, given in Algorithm 3 is used in the new method, instead of the `KHM` used in the old VNS. Details regarding the functions of `Shake` and `NeighborhoodChange` may be found in [1]. For the sake of completeness, we give their pseudo-codes here.

```
Function Shaking (X, K, C)
j ← 0                              // initializing iteration counter
repeat
    j ← j + 1
    r1 ← ⌊(K − j + 1) · rnd⌋       // a cluster is chosen at random
    r2 ← ⌊(N − j + 1) · rnd⌋       // an entity is chosen at random
    c_r1 ← x_r2                    // a cluster centroid is positioned at entity
until j = K
```

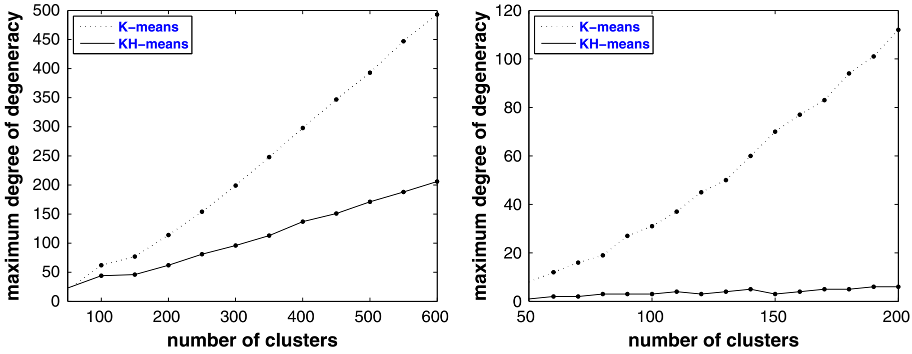
Algorithm 5: Shaking step

Main purpose of the Shaking step is to diversify the incumbent solution C. Neighborhood k ($k = 1, \ldots, k_{max}$) consists of random centroid-to-entity swaps. Such a random solution is the initial one for the `KHM+` local search, as it is usual in VNS.

```
Function NeighborhoodChange (C, C', k)
1 if f(C') < f(C) then
2     C ← C'; k ← 1                // make a move
  else
3     k ← k + 1                    // next centroid
```

Algorithm 6: Neighborhood change or move or not function

Function `NeighborhoodChange()` compares the new value $f_{KHM}(C')$ with the incumbent value $f_{KHM}(C)$ obtained in the neighborhood k (line 1). If an improvement is obtained, k is returned to its initial value, and the new incumbent is updated (line 2). Otherwise, the next neighborhood is considered (line 3).



(a) Dataset:*Breast Cancer-699* **(b)** Dataset:*Image Segmentation-2310*

Fig. 5 Comparison between the degeneracy degrees of K-means and KHM local searches after 100 restarts

4 Computational results

Computer. All experiments were performed on a personal computer Intel(R) Core(TM)2 with 0.98 GB of RAM and a speed of 2.40 GHz. All our methods were coded in Lahey/Fujitsu FORTRAN 95. For plotting, we used MATLAB 7.6.

Test instances. We choose the following test instances: (i) *Ruspini* which has 75 entities in 2-dimensions [23]; (ii) *Iris* which has 150 entities in 4-dimensions; (iii) *Wine* which has 178 entities in 13-dimensions; (iv) *Glass* which has 214 entities in 9-dimensional space; (v) *Breast-cancer* which has 699 entities in 10-dimensions, and (vi) *Image Segmentation* with 2,310 entities in 19-dimensions. For more details about them, see [5].

Parameters. We choose $\varepsilon = 0.01$ in all our experiments. In Algorithm 2, the parameter $Maxit = 180$. For all data sets and for all number of clusters K, we put the power p of the KHM objective function equal to 2 ($p = 2$).

Maximum degree of degeneracy. As mentioned above, the KHM heuristic for harmonic means clustering has smaller degree of degeneracy than the KM heuristic for Minimum sum-of-squares clustering. In Fig. 5, we show the maximum degree of degeneracy obtained during the execution of these two heuristics. Comparative results on two well-known data sets are presented: (i) *Breast-cancer* and (ii) *Image Segmentation*. The difference between these two heuristics is clear: the maximum degree of degeneracy is much larger for KM than for KHM. It is interesting to note that the maximum degree of degeneracy is empirically closer to linear function of the number K of clusters.

Comparison between KHM and KHM+. In the following tables, we present a comparison between the objective function values obtained with KHM and KHM+. The first column indicates the desired number K of clusters. The second column indicates the method used. Column 3 gives the corresponding objective function values. Column 4 shows the percentage improvement obtained by KHM+ in comparison with KHM. The number of local search iterations is displayed in Columns 5. The type and the maximum degree of degeneracy are displayed in columns 6 and 7, respectively. The last column shows the computing time (in s) for each method.

Based on the comparative results between KHM and KHM+ given in Table 3, one can make the following observations:

(i) Using KHM+ instead of KHM, the solution qualities are improved up to 58 % in smaller number of iterations and less computing times, on average.

Table 3 Comparison between KHM and KHM+ based on one run

K	Method	Obj	Dev %	*Maxit*	Type	Maxdeg	Time
(a) Dataset: *Wine-178*							
50	KHM	1,010,980.81	0.160	153	1	1	1.25
50	KHM+	1,009.370.12		138			0.84
60	KHM	772,767.75	50.279	119	1	1	0.93
60	KHM+	514,221.22		93			0.73
70	KHM	1,214,848.75	57.338	93	1	1	0.86
70	KHM+	772,127.44		73			0.78
(b) Dataset: *Glass-214*							
180	KHM	136.40	26.538	4	2	1	0.09
180	KHM+	107.79		4			0.21
190	KHM	89.61	55.899	4	2	1	0.12
190	KHM+	57.48		5			0.37
200	KHM	37.74	21.598	5	2	1	0.09
200	KHM+	31.04		5			0.60
(c) Dataset: *Breast Cancer-699*							
100	KHM	29,219.30	0.535	125	2	42	4.06
100	KHM+	29,063.82		68			2.57
150	KHM	31,468.25	9.242	48	2	39	2.35
150	KHM+	28,806.04		7			0.57
200	KHM	30,548.25	28.757	3	2	50	0.20
200	KHM+	23,725.55		3			0.62
250	KHM	26,197.01	37.502	3	2	62	0.25
250	KHM+	19,052.13		2			0.85
300	KHM	23,265.06	57.817	2	2	78	0.20
300	KHM+	14,741.80		2			1.09
(d) Dataset: *Image Segmentation-2310*							
100	KHM	32,480,866	0.197	104	2	1	14.75
100	KHM+	32,416,846		131			16.67
200	KHM	31,760,192	0.017	157	2	1	38.18
200	KHM+	31,754,658		150			40.87
300	KHM	30,295,272	0.023	105	2	3	37.68
300	KHM+	30,288,258		91			38.03
400	KHM	29,596,908	0.968	105	2	6	49.70
400	KHM+	29,313,292		102			48.68
500	KHM	28,287,296	0.419	50	2	10	29.28
500	KHM+	28,169,312		43			26.18

(ii) The degeneracy type depends on the instance considered. There is no instance with both types of degeneracy: *Wine-178* exhibits only type 1 and the other instances only type 2 degeneracy.

(iii) The number of clusters without entity (the maximum degree of degeneracy) can be more than 40 % of the total number of clusters K (see the *Breast Cancer-699* instance).

Table 4 Comparison between `KHM-VNS` and `KHM-VNS+`

Dataset	K	Obj		*Dev%*	obj		Dev %	*tmls*	tvns	tmax
Wine (178)	50	MLS	619,330.5630	8.754	VNS	466,665.6250	0.777	0.02	0.04	0.17
		MLS+	569,476.3750		VNS+	463,067.2190		0.02	0.15	0.16
	60	MLS	571,392.2500	11.118	VNS	435,012.0310	9.155	0.00	0.13	0.13
		MLS+	514,221.2190		VNS+	398,527.0000		0.00	0.03	0.47
	70	MLS	463,024.7500	3.282	VNS	353,198.4380	3.812	0.02	0.17	0.29
		MLS+	448,310.0940		VNS+	340,229.7810		0.02	0.21	0.31
Glass (214)	180	MLS	106.0072	30.034	VNS	28.8215	4.237	0.05	0.14	0.17
		MLS+	81.5225		VNS+	27.6500		0.02	0.05	0.12
	190	MLS	48.2871	3.967	VNS	19.9370	1.294	0.09	0.14	0.14
		MLS+	46.4445		VNS+	19.6822		0.46	0.45	0.56
	200	MLS	29.4370	3.111	VNS	8.5587	3.984	0.08	0.71	0.75
		MLS+	28.5487		VNS+	8.2308		0.12	1.16	1.16
Breast Cancer (699)	100	MLS	28,901.5645	0.673	VNS	27,519.3906	0.255	0.08	0.23	0.23
		MLS+	28,708.2246		VNS+	27,449.3926		0.19	0.21	0.21
	150	MLS	27,848.8496	2.505	VNS	24,057.1543	0.148	0.09	0.40	0.40
		MLS+	27,168.3398		VNS+	24,021.5273		0.22	0.31	0.34
	200	MLS	27,974.0879	22.398	VNS	20,640.6973	10.334	0.43	0.40	0.48
		MLS+	22,854.9551		VNS+	18,707.5488		0.50	0.50	0.50
	250	MLS	26,197.0117	44.040	VNS	18,345.2031	21.149	0.06	0.55	0.65
		MLS+	18,187.2559		VNS+	15,142.7334		0.62	0.62	0.62
	300	MLS	23,265.0645	71.161	VNS	16,581.6523	47.983	0.08	0.67	0.70
		MLS+	13,592.4814		VNS+	11,205.0742		0.70	0.54	0.70
Image Segmentation (2310)	100	MLS	32,480,866	7.213	VNS	24,805,398	0.142	0.09	0.26	0.98
		MLS+	30,295,774		VNS+	24,770,124		0.76	0.83	0.95
	150	MLS	25,970,568	0.041	VNS	21,838,422	1.360	1.92	1.55	2.14
		MLS+	25,960,054		VNS+	21,545,372		0.25	0.41	2.06
	200	MLS	24,675,180	3.892	VNS	18,951,112	0.608	2.18	2.78	3.64
		MLS+	23,750,836		VNS+	18,836,514		1.25	1.01	4.25
	250	MLS	24,025,238	1.047	VNS	17,222,892	0.061	2.17	2.75	4.42
		MLS+	23,776,396		VNS+	17,212,400		2.12	2.54	4.43
	300	MLS	21,686,348	3.819	VNS	16,094,108	0.572	2.17	2.67	5.34
		MLS+	20,888,608		VNS+	16,002,548		1.03	1.98	5.18

Comparison between VNS-KHM and VNS-KHM+. The next table illustrates the influence of `KHM+` when applied within two metaheuristics: MLS and VNS. Heuristics which use KHM+ as a local search within MLS and VNS, we denote as MLS+ and VNS+, respectively. Table 4 presents comparative results obtained by these 4 methods. For each data set, we first run KHM and KHM+ 100 times to get the maximum time allowed for VNS and VNS+ (t_{max}). These values are given in the last column of Table 4. Then, in the two columns before the last one, we report on running times when the best solution is found for each method.

It appears that:

(i) The results of best quality are obtained by the `VNS+` heuristic in less CPU time than those reported by `VNS`, the current state-of-the-art heuristic for KHM clustering.

(ii) VNS is always better than MLS+, except for two *Breast cancer* instances (for $K = 250$ and $K = 300$).
(iii) MLS+ improves the solution quality of MLS significantly. Thus, the four heuristics can be ranked as follows: VNS+, VNS, MLS+ and MLS.
(iv) Better results are obtained by VNS+ than VNS even when the final solution obtained by this last heuristic is not degenerate. This means that removing degeneracy immediately when it appears during the KHM iteration is better than waiting for possible auto correction in future iterations.
(v) Regarding CPU time, MLS+ and VNS+ are slightly faster on average than MLS and VNS respectively.

5 Conclusions

In this paper, we consider the KHMCP and an ALT type of heuristic (ALT) to solve it. We show that the KHM clustering heuristic for solving KHMCP suffers from degeneracy, i.e., that some clusters could become or remain empty (without entities) during the execution of the heuristic. We distinguish two types of degenerate solutions, and provide an efficient procedure which removes degeneracy immediately when it appears. Moreover, this new routine is used as a local search within a recent variable neighborhood search (VNS+) heuristic, which appears to represent the current state-of-the-art. Extensive computational analysis on some usual data sets from the literature confirms that degeneracy could seriously damage the solution qualities of both KHM and VNS-KHM.

Future research may include: (i) application of our approach to investigating effectiveness of ALT procedures for solving k-Heronian, Generalized k-Heronian mean [11] and $k-$logarithmic mean clustering [10]; (ii) the development of a general statement regarding degeneracy in ALT iterative procedures; (iii) the design of different methods for correcting degenerate solutions for ALT methods; (iv) an investigation of the relation between initial solution methods of K-means and KHM with degeneracy, i.e., whether the proper initialization method could avoid degeneracy completely.

Acknowledgments The research of E. Carrizosa is partially supported by Grants MTM2009-14039 (Ministerio de Educación y Ciencia, Spain) and FQM329 (Junta de Andalucía, Spain). Part of this research was done while N. Mladenović was visiting the Instituto de Matemáticas de la Universidad de Sevilla (Grant SAB2009-0144, Ministerio de Educación y Ciencia, Spain).

References

1. Alguwaizani, A., Hansen, P., Mladenovic, N., Ngai, E.: Variable neighborhood search for harmonic means clustering. Appl. Math. Model. **35**, 2688–2694 (2011)
2. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: Np-hardness of Euclidean sum-of-squares clustering. Mach. Learn. **75**, 245–248 (2009)
3. Aloise, D., Hansen, P.: Clustering. In: Sheir D.r. (ed.) Handbook of Discrete and Combinatorial Mathemaics. CRC Press (2009)
4. Bai, L., Liang, J., Dang, C., Cao, F.: A cluster centers initialization method for clustering categorical data. Expert Syst. Appl. **39**, 8022–8029 (2012)
5. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/datasets.html (1998)
6. Brimberg, J., Mladenović, N.: Degeneracy in the multi-source Weber problem. Math. Program. **85**(1), 213–220 (1999)

7. Brimberg, J., Hansen, P., Mladenovic, N.: Attraction probabilities in Variable neighborhood search. 4OR **8**, 181–194 (2010)
8. Cao, F., Liang, J., Jiang, G.: An initialization method for the K- Means algorithm using neighborhood model. Comput. Math. Appl. **58**, 474–483 (2009)
9. Carrizosa, E., Mladenovic, N., Todosijevic, R.: Sum-of-squares clustering on networks. Yugosl. J. Oper. Res. **21**, 157–161 (2011)
10. Chua, Y.-M., Xiab, W.-F.: Two optimal double inequalities between power mean and logarithmic mean. Comput. Math. Appl. **60**, 83–89 (2010)
11. Čižmešija, A.: A new sharp double inequality for generalized Heronian, harmonic and power means. Comput. Math. Appl. **64**, 664671 (2012)
12. Erisoglu, M., Calis, N., Sakallioglu, S.: A new algorithm for initial cluster centers in k-means algorithm. Pattern Recognit. Lett. **32**, 1701–1705 (2011)
13. Hamerly, G., Elkan, C.: Alternatives to the k-means algorithm that find better clusterings. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 600–607. ACM (2002)
14. Hansen, P., Jaumard, B., Mladenovic, N.: Minimum sum of squares clustering in a low dimensional space. J. Classif. **15**, 37–56 (1998)
15. Hansen, P., Mladenovic, N.: J-means: a new local search heuristic for minimum sum of squares clustering. Pattern Recognit. **34**, 405–413 (2001)
16. Hansen, P., Mladenovic, N., Pérez, J.A.M.: Variable neighbourhood search: methods and applications. 4-OR **6**, 319–360 (2008)
17. Hua, J., Yi, S., Li, J., et al.: Ant clustering algorithm with K-harmonic means clustering. Expert Syst. Appl. **37**, 8679–8684 (2010). doi:10.1016/j.eswa.2010.06.061
18. Li, Q., Mitianoudis, N., Stathaki, T.: Spatial kernel K-harmonic means clustering for multi-spectral image segmentation. Image Process. IET **1**(2), 156–167 (2007)
19. Mladenović, N., Brimberg, J.: A degeneracy property in continuous location-allocation problems. Les Cahiers du GERAD, G-96-37 (1996)
20. Mladenović, N., Hansen, P.: Variable neighbourhood search. Comput. Oper. Res. **24**, 1097–1100 (1997)
21. Mladenovic, N., Todosijevic, R., Urosevic, D.: An efficient general variable neighborhood search for large TSP problem with time windows. Yugosl. J. Oper. Res. **23**, 19–31 (2013)
22. Pakhira, M.K.: A modified k-means Algorithm to avoid empty clusters. Int. J. Recent Trends Eng. **1**, 220–226 (2009)
23. Ruspini, E.H.: Numerical methods for fuzzy clustering. Inf. Sci. **2**, 319–350 (1970)
24. Steinley, D., Brusco, M.J.: Initializing k-means batch clustering: a critical evaluation of several techniques. J. Classif. **24**, 99–121 (2007)
25. Xu, R., Wunsch, D.: Clustering. IEEE Press, New York (2009)
26. Yang, Fengqin, Sun, Tieli, Zhang, Changhai: An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization original research article. Expert Syst. Appl. **36**, 9847–9852 (2009)
27. Yin, M., Hu, Y., Yang, F., et al.: A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering. Expert Syst. Appl. **38**, 9319–9324 (2011). doi:10.1016/j.eswa.2011.01.018
28. Zhang, B.: Generalized k-harmonic means—boosting in unsupervised learning. Technical Report, HPL-2000-137, Hewlett-Packard Laboratories (2000)
29. Zhang, B., Hsu, M., Dayal, U.: K-harmonic means—a data clustering algorithm. Technical Report, HPL-1999-124, Hewlett-Packard Laboratories (1999)