Central European Journal of Operations Research (2024) 32:11–27
https://doi.org/10.1007/s10100-023-00881-1

Mixed integer linear programming formulation for K-means clustering problem

Kolos Cs. Ágoston[1] · Marianna E.-Nagy[2]

Accepted: 31 August 2023 / Published online: 27 October 2023

Abstract

The minimum sum-of-squares clusering is the most widely used clustering method. The minimum sum-of-squares clustering is usually solved by the heuristic KMEANS algorithm, which converges to a local optimum. A lot of effort has been made to solve such kind of problems, but a mixed integer linear programming formulation (MILP) is still missing. In this paper, we formulate MILP models. The advantage of MILP formulation is that users can extend the original problem with arbitrary linear constraints. We also present numerical results, we solve these models up to sample size of 150.

Keywords Mathematical programming · Linear programming formulation · Clustering · K-means

1 Introduction

Clustering is one of the most widely used methods in data science. Within this area, K-means clustering is the most widely used approach that aims to minimize the within-cluster sum of squared distances. It is known to be an NP-hard problem (Aloise et al. 2009) even if the cluster sizes are equal (Kondor 2022). The well-known KMEANS clustering algorithm is a very fast method, but it is a heuristic algorithm without any guarantee of global optimum. In data science, it is said that the KMEANS algorithm is sensitive to the initial cluster centers, in optimization terminology the KMEANS

Earlier results of this paper have been presented in conference paper (Ágoston and E.-Nagy 2021).

✉ Kolos Cs. Ágoston
kolos.agoston@uni-corvinus.hu

Marianna E.-Nagy
marianna.eisenberg-nagy@uni-corvinus.hu

[1] Institute of Operations and Decision Sciences, Corvinus University of Budapest, Budapest, Hungary

[2] Corvinus Centre for Operations Research, Corvinus University of Budapest, Budapest, Hungary

algorithm converges to a local optimum. As Hansen and Jaumard (1997) reported in their paper, "experiments show that the best clustering found with KMEANS may be more than 50% worse than the best known one". This phenomenon is well known, and even so, this method has been implemented in most commonly used statistical and data science softwares until today, contrary to the fact that exact algorithms are known (see, for instance, du Merle et al. 1999).

Solving the clustering problem using Linear Programming (LP) appeared early in the literature (see Vinod 1969; Rao 1971). Later, different types of clustering problems were solved using LP, (see, for instance, Cornuejols et al. 1980; Kulkarni and Fathi 2007; Dorndorf and Pesch 1994; Gilpin et al. 2012), but the most frequently used minimum sum-of-squares clustering was less investigated. du Merle et al. (1999) proposed an exact algorithm to solve the minimum sum-of-squares clustering problem, but this approach did not appear in statistical packages, probably due to the fact that it is a rather complicated algorithm.

We can also form the minimum sum-of-squares clustering problem based on Semidefinite Programming (SDP) (see Peng and Wei 2007; Picciall et al. 2021). The drawback of this approach is that it is not a pure SDP problem, since it has an additional nonlinear constraint, and moreover, only moderate size SDP problems can be solved. A more detailed overview of the mathematical background of clustering problems can be found in Hansen and Jaumard (1997) and Peng and Wei (2007).

In this paper, we present Mixed Integer Linear Programming (MILP) formulations for the minimum sum-of-squares clustering problem. Rujeerapaiboon et al. (2019) described a MILP formulation for minimum sum-of-squares problems, but their formulation works only with a priori fixed cluster sizes. However (as we will see), the main source of the nonlinearity in the model is that the cardinality of the clusters is unknown. The suggested formulation can be extended to problems with many types of constraint (for instance, lower bound on the cardinality of clusters or must-link constraints Bradley et al. 2000; Davidson and Ravi 2007). The suggested MILP models are based on the nonlinear formulation appeared in Awasthi et al. (2015), which is recalled in Sect. 2. In the rest of Sect. 2, we investigate our two MILP models and propose additional cuts which can result in tighter LP relaxations. Finally, the computational results are presented in Sect. 3.

The following notations are used throughout the paper: Let $\mathcal{H}$ be a set, then $|\mathcal{H}|$ is the cardinality of the set $\mathcal{H}$. If K is a positive integer, then $[K] := \{1, \dots, K\}$. The Euclidean distance between the points a and b is denoted by $d(a, b)$.

2 MILP formulation for minimum sum-of-squares clustering problem

We have N points in the n-dimensional space: $\mathcal{A} = \{a_1, \dots, a_N\} \subset \mathbb{R}^n$. Our aim is to group these points into K clusters in a way that minimizes the sum of the squared distances. Clusters of points are denoted by $\mathcal{A}_k$, $k \in [K]$. These sets form a partition of $\mathcal{A}$ and none of them is empty, that is,

$$\cup_{k=1}^{K} \mathcal{A}_k = \mathcal{A}, \qquad \mathcal{A}_k \cap \mathcal{A}_\ell = \emptyset, \quad \mathcal{A}_k \neq \emptyset \quad \forall\, k \neq \ell \in [K].$$

Let $\mathcal{P}_{\mathcal{A}}$ denote the set of partitions of $\mathcal{A}$ into exactly K nonempty subsets. The center of the cluster $\mathcal{A}_k$ is denoted by c_k, which is defined as the multidimensional mean, i.e., $c_k = \frac{1}{|\mathcal{A}_k|}\sum_{a_i \in \mathcal{A}_k} a_i \in \mathbb{R}^n$. The sum of squared distances within the cluster $\mathcal{A}_k$ is given by the formula: $\sum_{a_i \in \mathcal{A}_k} d(a_i, c_k)^2$. We can reformulate this sum of squares formula as $\frac{1}{|\mathcal{A}_k|}\sum_{a_i, a_j \in \mathcal{A}_k} d(a_i, a_j)^2$ (see du Merle et al. 1999; Awasthi et al. 2015). Consequently, *minimum sum-of-squares clustering problem* is the following:

$$\min_{(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k) \in \mathcal{P}_{\mathcal{A}}} \sum_{k=1}^{K} \sum_{a_i, a_j \in \mathcal{A}_k} \frac{d(a_i, a_j)^2}{|\mathcal{A}_k|}. \tag{1}$$

2.1 An almost linear modell

In Awasthi et al. (2015), we can find a promising reformulation:

$$\sum_{i,j} d(a_i, a_j)^2 z_{ij} \to \min \tag{2}$$

s.t.

$$\sum_{j=1}^{N} z_{ij} = 1 \qquad \forall\ i \in [N] \tag{3}$$

$$z_{ij} \le z_{ii} \qquad \forall\ i,j \in [N] \tag{4}$$

$$\sum_{i=1}^{N} z_{ii} = K \tag{5}$$

$$\begin{aligned} & z_{ij} \ge 0 && \forall\ \ i,j \in [N] \\ & z_{ij} \in \{0, 1/|\mathcal{A}_{t(j)}|\} && \forall\ i,j \in [N] \end{aligned} \tag{6}$$

where $t(j)$ is the index of the cluster, which contains a_j, namely $a_j \in \mathcal{A}_{t(j)}$. This is a nonlinear problem, however, except for the last constraint, this is a linear model with nonnegative decision variables z_{ij} which indicates whether elements i and j belong to the same cluster or not. There are two problems with the last constraint: we do not know a priori the value of $t(j)$ and the cardinality of the cluster $\mathcal{A}_{t(j)}$. However, it can be reformulated as $z_{ij}(z_{ij} - z_{ii}) = 0$, but it is still not a linear constraint. We note here that the 0–1 SDP model of Peng and Wei (2007) is very similar to this one. Their variable is a symmetric matrix Z, but its elements correspond exactly to the variables z_{ij} here. Their objective function and constraints (3)–(6) are the same only in matrix form. Finally, instead of the last, problematic constraint in the model of Awasthi et al. (2015), they have $Z^2 = Z$, that is, the matrix Z has to be a projection matrix. This is a nonlinear constraint, therefore we cannot use directly an SDP algorithm to solve it.

2.2 Minimum sum of squares linear relaxation

The optimal solution of the problem minimizing (2) subject to (3)–(6) does not give a 'legal' clustering. To ensure this, we need further constraints.

It is worth prescribing the *symmetry of the variables* z_{ij}, that is,

$$z_{ij} = z_{ji} \quad \forall\, i,j \in [N]. \tag{7}$$

We suggest another type of possible linear constraint that makes the linear relaxation significantly tighter, this is the *'triangle inequality'*:

$$z_{ij} + z_{i\ell} - z_{j\ell} \le z_{ii} \quad \forall\, i,j,\ell \in [N]. \tag{8}$$

Indeed, if both variables z_{ij} and $z_{i\ell}$ take positive values (which means that elements i and j are in the same cluster and also elements i and ℓ are in the same cluster), then variable $z_{j\ell}$ has to take a positive value, and in this case, the values of all three variables must be equal to variable z_{ii}. If both variables z_{ij} and $z_{i\ell}$ are 0, then the value of the variable $z_{j\ell}$ is not constrained.

We refer to the model that minimizes (2) subject to (3)–(8) as *MSSR: Minimum Sum of Squares Relaxation*. It still does not surely result in a 'legal' clustering structure, but as the numerical tests show, we already get an optimal clustering with this model in most cases. To obtain an exact model, we use binary variables. It can be done in different ways, we will discuss two of them.

2.3 Binary minimum sum of squares formulation

First, we introduce the binary variable ζ_{ij}, which takes the value 1, if elements i and j are in the same cluster, otherwise, it takes the value 0:

$$\zeta_{ij} \in \{0,1\} \quad \forall\, i,j \in [N]. \tag{9}$$

The values of variables z_{ij} and ζ_{ij} are not independent, hence we need constraints to ensure the relationship between them:

$$z_{ij} \le \zeta_{ij} \quad \forall\, i,j \in [N]. \tag{10}$$

$$z_{ii} - z_{ij} \le 1 - \zeta_{ij} \quad \forall\, i,j \in [N]. \tag{11}$$

Theorem 1 *The problem of minimizing* (2) *subject to* (3)–(11) *gives an exact MILP model for the K-means problem.*

Proof First of all, consider a partition of points with exactly K nonempty subsets, and let $z_{ij} = 1/|\mathcal{A}_{t(j)}|$ for all $i,j \in [N]$, and $\zeta_{ij} = 1$ if i and j are in the same clusters and zero otherwise for all $i, j \in [N]$. Then, this is a feasible solution to the problem (as we have discussed above), and its objective function value is exactly the sum-of-squared distances according to the given clustering.

On the other hand, based on constraints (3) and (4), $z_{ii} > 0$ for all i. Furthermore, by (7), (10) and (11),

$$0 < z_{ii} = z_{ii} - z_{ij} + z_{ji} \leq 1 - \zeta_{ij} + \zeta_{ji} \quad \forall\, i, j \in [N]$$

therefore

$$\zeta_{ij} = \zeta_{ji} \quad \forall\, i, j \in [N]. \tag{12}$$

To prove the triangle inequality on variables ζ_{ij}, add the constraint (11) for indices i, j and i, l and then use (8), the positivity of z_{ii}, and finally (10),

$$\zeta_{ij} + \zeta_{il} \leq 2 - z_{ii} + z_{ij} + z_{il} - z_{ii} < 2 + z_{jl} \leq 2 + \zeta_{jl}.$$

Combining it with the binary nature of the variable ζ, we get the desired inequality

$$\zeta_{ij} + \zeta_{i\ell} \leq 1 + \zeta_{j\ell} \quad \forall\, i, j, \ell \in [N]. \tag{13}$$

Summarizing the above, we have shown that for each feasible solution of the proposed MILP, ζ gives a proper clustering since (9), (12) and (13). Moreover, the number of clusters is exactly K as a consequence of the constraint (5). So, we only need to prove that the objective value is appropriate. Multiplying constraints (10) and (11) and using that ζ's are binary variables, we get that

$$0 \leq (z_{ii} - z_{ij}) z_{ij} \leq (1 - \zeta_{ij}) \zeta_{ij} = 0 \quad \forall\, i, j \in [N].$$

In other words, z_{ij} is either zero or equal to z_{ii}. Comparing it with Eq. (3), $z_{ii} = 1/|\mathcal{A}_{t(j)}|$, namely, $z_{ij} \in \{0, 1/|\mathcal{A}_{t(j)}|\}$ for all $i, j \in [N]$. This completes the proof. □

Adding additional constraints (cuts) can help the MILP solver find an optimal solution faster. We considered two possibilities which are the following constraints

$$(N - K + 1) z_{ij} \geq \zeta_{ij} \quad \forall\, i, j \in [N]. \tag{14}$$

$$(N - K + 1)(z_{ii} - z_{ij}) \geq 1 - \zeta_{ij} \quad \forall\, i, j \in [N]. \tag{15}$$

We reach *BMSS (Binary Minimum Sum of Squares) formulation*: minimize (2) subject to (3)–(11) and (14)–(15). By Theorem 1, this is an exact formulation for the K-means problem, but with redundant additional constraints to improve the quality of the solution of its LP relaxation. It is easy to see that the constraints (14)–(15) ensure: if an optimal solution of MSSR is a ‘legal’ clustering, then all binary variables ζ_{ij} take integer values, i.e., the branch and bound tree will only contain the root node.

2.4 Assignment-type minimum sum of squares formulation

In the BMSS formulation, the number of binary variables can be quite large, and its number increases quadratically as the number of elements increases. Therefore, we tried another approach where the number of binary variables is significantly less. Let γ_{ik} denote the binary variable that indicates whether the element i is assigned to the cluster k:

$$\gamma_{ik} \in \{0, 1\} \quad \forall\, i \in [N], k \in [K]. \tag{16}$$

Since every element belongs to exactly one cluster,

$$\sum_{k=1}^{K} \gamma_{ik} = 1 \quad \forall\, i \in [N] \tag{17}$$

furthermore, every cluster contains at least one element:

$$\sum_{i=1}^{N} \gamma_{ik} \geq 1 \quad \forall\, k \in [K]. \tag{18}$$

In this way, namely by constraints (16)–(18), we define a 'legal' clustering, where the number of clusters is exactly K.

We need to connect the variables γ_{ik} to the variables z_{ij}. If elements i and j are in different clusters, then z_{ij} has to be zero, therefore

$$z_{ij} \leq 1 + \gamma_{ik} - \gamma_{jk} \quad \forall\, i \neq j \in [N]. \tag{19}$$

Theorem 2 *The problem of minimizing* (2) *subject to* (3)–(6) *and* (16)–(19) *gives an exact MILP model for the K-means problem.*

Proof We have already noted that γ which fulfills constraints (16)–(18) gives a 'legal' clustering with exactly K clusters.

Furthermore, the constraint (19) with $k = t(j)$ ensures that z_{ij} is zero if i and j are in different clusters. So, by (3) and (4), we get $z_{ii} \geq 1/|\mathcal{A}_{t(j)}|$. But there is no empty cluster due to (18), namely $z_{ii} = 1/|\mathcal{A}_{t(j)}|$ by (5). This again means that $z_{ij} \in \{0, 1/|\mathcal{A}_{t(j)}|\}$ for all $i, j \in [N]$ based on the Eq. (3). Hence, any feasible solution of the problem gives a K-clustering and its objective function value is exactly the sum of squared distances within clusters.

On the other hand, we consider all possible partitions into exactly K nonempty subsets $\{\mathcal{A}_1, \dots, \mathcal{A}_K\}$. Indeed, if $z_{ij} = 1/|\mathcal{A}_{t(j)}|$ for all $i, j \in [N]$, and $\gamma_{ik} = 1$ if $i \in \mathcal{A}_k$ and zero otherwise for all $i \in [N]$ and $k \in [K]$, then it satisfies constraints (3)–(6) and (16)–(19), and its objective function value is the sum of squared distances within clusters. Therefore, we proved the statement. □

Let us again show some further constraints that can help a MILP solver. One possibility is to enforce i and j in different clusters if $z_{ij} = 0$:

$$\gamma_{ik} + \gamma_{jk} \le 1 + (N - K + 1)z_{ij} \quad \forall\, i,j \in [N], k \in [K]. \tag{20}$$

Furthermore, in a clustering problem, the essential result is a grouping, meaning which elements are in the same cluster and which are in different ones. The 'label' of the cluster is irrelevant. If we have K clusters, the labels can be assigned in $K!$ way. We can break this symmetry by prescribing that the first element belongs to the first cluster:

$$\gamma_{1,1} = 1. \tag{21}$$

We could go further. If the second element belongs to the same cluster as the first element, it will also be assigned to cluster 1. Otherwise, let it be in the second cluster, so we have $\gamma_{2,k} = 0$, for $k \ge 3$. Similarly, for the third element, $\gamma_{3,k} = 0$ for $k \ge 4$. Surprisingly, these constraints also slow down the process, and it is not worth using all of them.

We call the problem of minimizing (2) subject to (3)–(11) and (16)–(21) as *AMSS (Assignment-type Minimum Sum of Squares)* formulation. It is again an exact model (by Theorem 2) with some redundant constraints. AMSS has significantly fewer binary variables than BMSS ($N \times K$ vs. $(N-1) \times (N-1)$). Another advantage of AMSS formulation is that more constraints can be formulated with the help of variables γ_{ik} than with the help of ζ_{ij}. On the other hand, it is not true that if the optimal solution of the MSSR formulation gives a legal clustering, then all binary variables in the relaxation of AMSS take integer values. Therefore, it is not enough for AMSS to check the integrality of the solution to the continuous relaxation.

We close this section by generalizing the idea of the triangular inequality (8). Instead of three points, let us take four nodes, then

$$z_{ij} + z_{ik} + z_{i\ell} \le z_{ii} + z_{jk} + z_{j\ell} + z_{k\ell} \quad \forall\, i,j,k,\ell \in [N] \tag{22}$$

constraints should hold. These are valid inequalities for a 'legal clustering'. It can be shown that these constraints cut down feasible basic solutions of BMSS which do not give a 'legal clustering', but, in the meantime, they can generate new feasible basic solutions which are not legal clustering. Furthermore, we can formulate similar constraints on more than four points as well, but already for four points, its number is quite huge.

3 Numerical results

We tested the above described two MILP formulations (BMSS and AMSS) and their common LP core (MSSR) on randomly generated data points and on real-world data points as well. We used a desktop computer with 3.60 GHz Intel Pentium processor and 8 GB RAM. The operating system is Windows 10 Enterprise. We used Gurobi 9.1.1 solver with default parameter settings to solve MILP problems.

Table 1 Essential information about the MSSR formulation (LP problem)

(N,K)	#var. (bin.)	#const	#nonzero	#iter	Time (s)	o.f. value
(25,2)	625 (0)	7826	30,050	379	0.16	4.2121
(50,2)	2500 (0)	62,526	245,100	44,891	5.41	8.7706
(75,2)	5625 (0)	210,976	832,650	47,323	29.55	14.4857
(100,2)	10,000 (0)	500,051	1,980,200	113,292	146.49	18.8850
(25,3)	625 (0)	7826	30,050	274	0.15	2.3742
(50,3)	2500 (0)	62,526	245,100	32,493	2.30	4.8643
(75,3)	5625 (0)	210,976	832,650	112,711	17.09	8.6184
(100,3)	10,000 (0)	500,051	1,980,200	271,365	110.24	11.8003
(25,5)	625 (0)	7826	30,050	132	0.10	1.0266
(50,5)	2500 (0)	62,526	245,100	21,525	1.40	2.7192
(75,5)	5625 (0)	210,976	832,650	74,264	9.46	4.3356
(100,5)	10,000 (0)	500,051	1,980,200	134,661	68.26	**6.0120**

We get an integer optimal solution except for the last instance (the objective value is in bold)

3.1 Randomly generated data points

In order to test the MSSR, BMSS, and AMSS formulations, we generated uniformly distributed random points in the unit square. From a clustering perspective, it is difficult to make a grouping of uniformly distributed data points since such a set of points is quite homogeneous. Considering real-world instances, the larger problems can be solved in less time since the clustering structure can be more obvious. In this sense, these running times can be considered as an upper bound.

The important statistics on the size of the problem (number of variables (binary variables), number of constraints and number of nonzero coefficients), the number of iterations, the running time (in seconds), and the optimal objective function value can be found in Tables 1, 2 and 3.

As we can see in Tables 1, 2 and 3, except for the instance (100,5), the optimum solution of MSSR will result in a legal clustering structure, actually, we do not need the integer variables. For all the instances presented, the running times are less than 2.5 min for the MSSR formulation. Not surprisingly, the running times for the BMSS and AMSS formulations are higher but still tolerable (except for the instance (100,5)). There is no strict dominance between BMSS and AMSS formulations, BMSS seems to have slightly better performance (mainly for case (100,5)).

3.2 Real-world instances

We chose three well-known data sets to test our models. The first one is the so-called Ruspini data set, which contains 75 data points in the plane (see Fig. 1). The Ruspini data set appeared first in Ruspini (1970), but was also analyzed in Kaufman and Rousseeuw (1990). The second data set is the Iris data set (see Fisher 1936), which is a well-known benchmark data set for classification problems. This data set contains information on 150 flowers. We used this data set for clustering purposes,

Table 2 Essential information about the BMSS formulation (MILP problem)

(N,K)	#var. (bin.)	#const	#nonzero	#iter	Time (s)	o.f. value
(25,2)	925 (300)	9026	33,050	767	0.41	4.2121
(50,2)	3725 (1225)	67,426	257,350	13,057	4.86	8.7706
(75,2)	8400 (2775)	222,076	860,400	85,215	69.00	14.4857
(100,2)	14,950 (4950)	519,851	2029,700	199,582	547.48	18.8850
(25,3)	925 (300)	9026	33,050	566	0.32	2.3742
(50,3)	3725 (1225)	67,426	257,350	7231	3.53	4.8643
(75,3)	8400 (2775)	222,076	860,400	51,086	45.91	8.6184
(100,3)	14,950 (4950)	519,851	2,029,700	122,879	345.28	11.8003
(25,5)	925 (300)	9026	33,050	290	0.28	1.0266
(50,5)	3725 (1225)	67,426	257,350	8950	3.58	2.7192
(75,5)	8400 (2775)	222,076	860,400	42,963	38.07	4.3356
(100,5)	14,950 (4950)	519,851	2,029,700	199,143	1064.93	6.0156

Table 3 Essential information about the AMSS formulation (MILP problem)

(N,K)	#var. (bin.)	#const	#nonzero	#iter	Time (s)	o.f. value
(25,2)	675 (50)	9702	35,601	662	0.45	4.2121
(50,2)	2600 (100)	70,029	267,451	559,853	306.13	8.7706
(75,2)	5775 (150)	227,854	883,201	64,316	45.55	14.4857
(100,2)	10,200 (200)	530,054	2,070,101	155,495	270.05	18.8850
(25,3)	700 (75)	10,627	38,376	678	0.51	2.3742
(50,3)	2650 (150)	73,755	278,776	211,463	88.09	4.8643
(75,3)	5850 (225)	236,255	908,476	81,300	56.23	8.6184
(100,3)	10,300 (300)	545,005	2,115,051	205,222	312.18	11.8003
(25,5)	750 (125)	12,477	43,926	432	0.46	1.0266
(50,5)	2750 (250)	81,207	301,226	182,144	72.10	2.7192
(75,5)	6000 (375)	253,057	959,026	73,654	56.49	4.3356
(100,5)	10,500 (500)	574,907	2,204,951	766,669	16,128.02	6.0156

so we neglected the iris type (class), and we used only the parameters sepal length, sepal width, petal length, and petal width during clustering. The third data set is the Breast Tissue data set.[1] In this dataset, 106 different instances were analyzed with the help of 9 features. Since the measurements are not uniform, we used standardized variables in this case.

We calculated the optimal clustering for cases where the number of clusters is 2–6 (2–7 in the case of Iris). The MSSR relaxation gave an optimal solution for clustering problems (except Iris with 7 clusters and Breast Tissue with 6 clusters).

[1] Available in the UCI Machine Learning Repository (see Dua and Graff 2019, http://archive.ics.uci.edu/ml/datasets/breast+tissue).

Table 4 Running times (in seconds) and optimal objective function values for real-world instances

Data set	Cluster	MSSR	BMSS	AMSS	o.f. value
Ruspini	2	8.94	28.02	26.67	178,675.66
Ruspini	3	9.79	24.36	30.53	102,126.95
Ruspini	4	5.17	17.16	15.81	25,762.10
Ruspini	5	4.92	17.05	39.14	20,253.44
Ruspini	6	5.39	17.62	46.3	17,150.81
Iris	2	431.82	2239.58	1258.98	304.70
Iris	3	407.73	1160.21	1168.48	157.70
Iris	4	371.25	1286.70	1458.97	114.46
Iris	5	323.21	1002.96	2135.83	92.89
Iris	6	262.72	964.06	1813.30	78.08
Iris	7	362.47	12,886.20	86,121.75	68,60 (68.56)
Breast tissue	2	150.88	514.04	240.84	1070.41
Breast tissue	3	129.84	414.71	290.88	791.85
Breast tissue	4	131.29	404.48	310.15	590.58
Breast tissue	5	66.09	298.95	260.42	441.95
Breast tissue	6	89.56	3252.46	133,100.11	373.77(373.62)
Breast tissue	20	71.01	514.58	6520.16	95.15(95.09)

The optimal objective value for the MSSR model is in parentheses if it differs from the other models' optimal value

The running times and the optimal value of the objective function can be found in Table 4.

As we see from Table 4, we can apply our formulations for real-world instances and it is worth emphasizing that the running times for real-world instances are less than for the same size randomly generated data sets. For example, let us consider the case of three clusters. The running time is 17.09 for the MSSR formulation for the randomly generated problem, while the running time is 9.79 for the Ruspini data set. Although in the majority of cases, the MSSR formulation gave legal clusters, in two cases a MILP formulation was needed.

It is worth to say a few words about the optimal clustering structure for the Ruspini data set. The Ruspini data set is a two-dimensional problem, optimal clusters can be represented in a scatter plot, see Fig. 1.

This data set was an example of the silhouette method (see Rousseeuw 1987). In this paper, the clustering structure for the Ruspini data is also published. We get almost the same clusters, except for cluster numbers 5 and 6. For the case of five clusters, Rousseeuw writes (Rousseeuw 1987, pg. 62): "When $k = 5$ is imposed, the algorithm splits C into two parts. The second part contains the three 'lowest' points of C..., that is, the three points of C with smallest y-coordinates. This trio has a rather prominent silhouette, and indeed some people consider it as a genuine cluster". As we see in Fig. 1 in light green color, the above mentioned 3 points do not form alone a cluster, there is a forth point as well in this cluster.

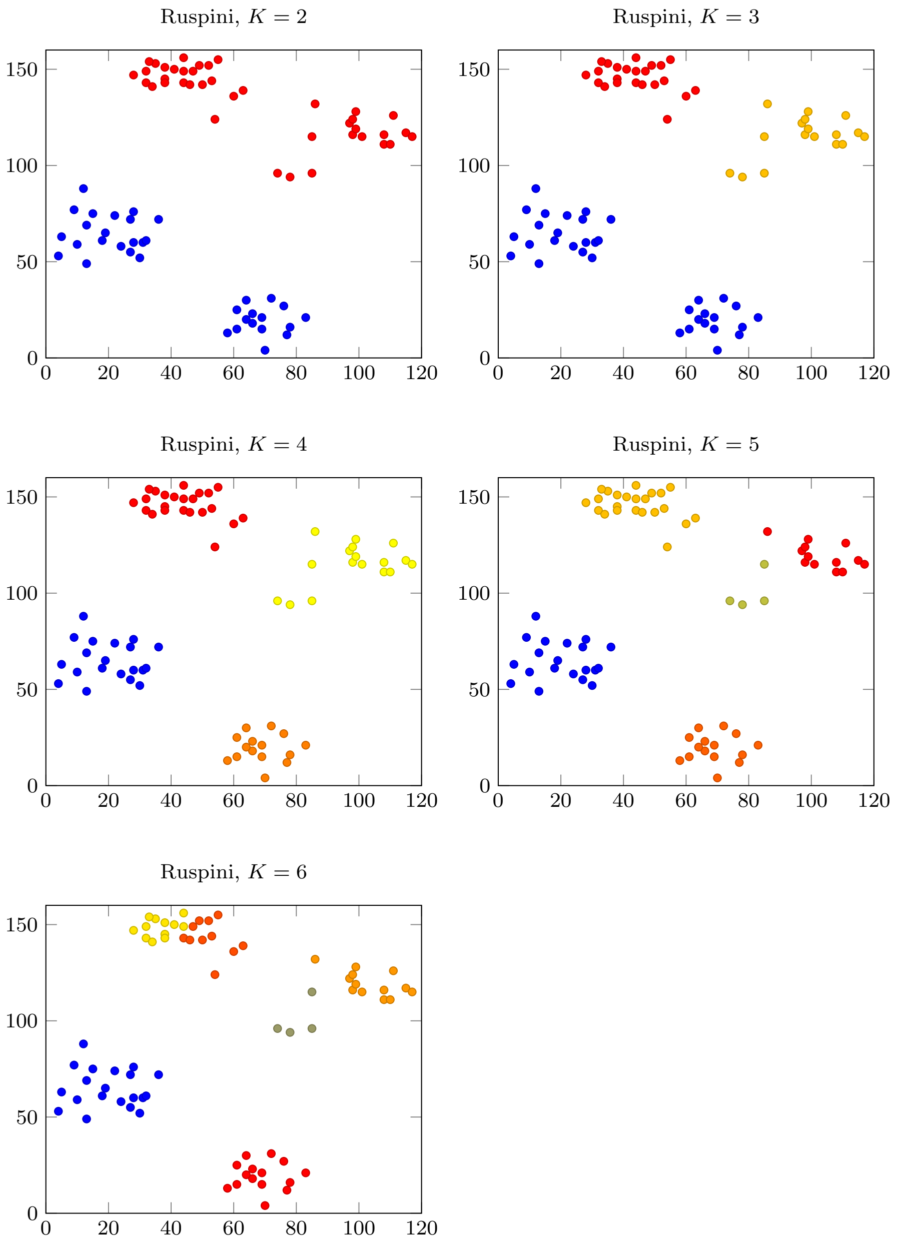

Fig. 1 Optimal minimum sum of clusters for the Ruspini data set

To draw attention to the importance of the exact clustering method, we performed a test on the data sets mentioned above with the given number of clusters. We run the KMEANS algorithm (we used `kmeans()` function in `R` programming language) for randomly chosen initial cluster centers 10,000 times. The

Table 5 Computational results of KMEANS algorithm

Data set	Cluster	o.f. value	ObjVal	ARI	Exact (%)
GenPoints25	2	4.21	4.21	1	100.00
GenPoints50	2	8.77	8.77	1	100.00
GenPoints75	2	14.48	14.98	0.7176	35.38
GenPoints100	2	18.88	19.56	0.7571	70.69
GenPoints25	3	2.37	2.51	0.8599	80.07
GenPoints50	3	4.86	5.18	0.8873	83.11
GenPoints75	3	8.62	8.97	0.8265	74.48
GenPoints100	3	11.80	12.12	0.7352	43.35
GenPoints25	5	1.03	1.12	0.8052	27.35
GenPoints50	5	2.72	2.84	0.7801	21.34
GenPoints75	5	4.33	4.47	0.7249	26.15
GenPoints100	5	6.01	6.17	0.7257	27.52
Ruspini	2	178,675.66	178,675.70	1	100.00
Ruspini	3	102,126.95	102,215.90	0.7422	51.63
Ruspini	4	25,762.10	56,704.84	0.8471	57.75
Ruspini	5	20,253.44	37,038.84	0.8217	19.93
Ruspini	6	17,150.81	26,660.35	0.7676	10.72
Iris	2	304.70	304.70	1	100
Iris	3	157.70	182.98	0.8897	80.22
Iris	4	114.46	125.72	0.8016	26.49
Iris	5	92.89	104.34	0.7327	10.04
Iris	6	78.08	89.45	0.7169	7.20
Iris	7	68.60	78.73	0.7284	9.28
Breast tissue	2	1070.41	1079.04	0.9808	98.02
Breast tissue	3	791.85	850.45	0.6604	3.49
Breast tissue	4	590.58	676.87	0.7171	3.75
Breast tissue	5	441.95	573.26	0.6396	2.20
Breast tissue	6	373.77	502.15	0.5286	1.84
Breast tissue	20	95.15	150.40	0.6045	0.03

computational results are summarized in Table 5. In the first column (Data set) are the names of the data sets (GenPoints n are the randomly generated data sets with n points described in Sect. 3.1). The second column (Clusters) contains the number of clusters. The third column (o.f. value) is the optimal objective value of the exact model, while the fourth column (ObjVal) gives the average optimal objective value found by the KMEANS algorithm. ARI is the Adjusted Rand Index, which is a widely used measure of similarity between two clusters (Rand 1971). Here, we take the average value of ARI for clusters given by the KMEANS algorithm and the exact K-means clustering. The last column (Exact) contains the percentage in which KMEANS finds an exact clustering.

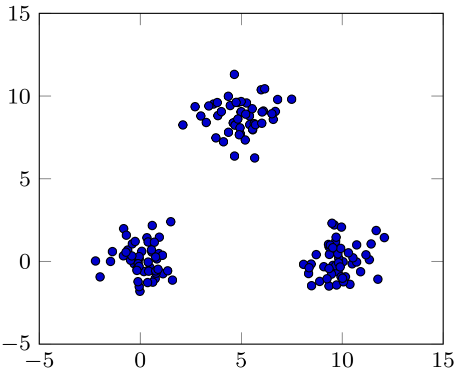

Fig. 2 Generated sample

Table 6 Running times (in s) for generated sample of Fig 2

Cluster size	2	3	4	5	6
Running time (AMSS)	3662.03	578.35	1366.24	1126.76	1947.17
Running time (BMSS)	1587.44	680.31	1174.91	1030.96	994.8

It can be clearly seen that the success rate rapidly decreases with the number of clusters, and the clusterings provided by KMEANS are significantly different from the exact clusterings (ARI values and the differences between the objective values). Therefore, already for these medium-sized problems, there is a relevance of an exact algorithm.

We mention here that although it is well known that the KMEANS algorithm gives maybe only a locally optimal clustering, the possible low chance to get a global optimum is less known. The default method for the R function `kmeans()` is the Hartigan-Wong algorithm (see Hartigan and Wong 1979); Slonim et al. (2013) gave an upper bound on the number of local minima of the Hartigan-Wong algorithm.

Another important issue is the solution time for different approaches. Since the problem is NP-hard, it is unrealistic to expect that the running time of a MILP solver on the exact reformulation BMSS or AMSS should be competitive with heuristic approaches (for instance, with the KMEANS algorithm). The running times for the MILP reformulations are higher than that for the heuristic approaches, this is the price of the exact solution. There are other exact algorithms, but somehow all of them depend on external solvers, and in this sense, the running times are somehow the competition of solvers. We do not know any other MILP formulation with which the comparison would be reliable. The advantage of the MILP model is the usage of widely available LP solvers. On the other hand, a MILP formulation is more flexible to extend the original model with special considerations.

Although the running times in our case are higher than for the KMEANS algorithm, they still can be tolerated for small and medium size samples. On the other hand, the running time itself may provide additional information. Consider the generated sample in Fig. 2. The cluster centers are chosen to be the three vertices of an equilateral triangle, and we generated 50 points around each vertex (sampled from a multivariate normal distribution). We solved the BMSS and AMSS formalization for cluster number 2–6. The running times can be seen in Table 6. The running times are the smallest if the cluster number is 3, which is somehow the only acceptable choice in this example. The running times are the highest when the cluster number is 2, which is really counterintuitive for this problem. Therefore, the high running time may be a sign of an unclear cluster structure, but a more detailed experiment is needed for a direct statement.

3.3 Special constraints

As we mentioned previously, the advantage of the MILP formulation is its flexibility, that is, we can add further constraints to the model. In the literature, different types of such kind of considerations appear. In the following example, we concentrate on the cluster sizes. For the minimum sum-of-squares clustering problem, it is a quite frequent phenomenon that the cluster sizes are unbalanced (see Bradley et al. 2000). However, users may want to avoid the possibility of a very small cluster size. For instance, in the case of the Ruspini data set with 5 and 6 clusters, we want to add a constraint on the minimum number of cluster elements. We seek the minimum sum-of-squares clustering such that each cluster has at least 10 elements. We can incorporate this requirement into the model in multiple ways. The easiest and most efficient way is to impose constraints on z_{ii} variables. The z_{ii} variables are the reciprocal of the number of elements in the corresponding cluster. The constraint that there is no cluster with less than 10 elements means that

$$z_{ii} \leq 0.1 \qquad \forall\, i \in [N]. \tag{23}$$

The advantage of the constraint (23) is that we can insert it into any of the aforementioned formulations. The running times for the formulations are 5.43 s (MSSR), 69.04 s (BMSS), and 2182.29 s (AMMS) in the case of 5 clusters and 5.78 s (MSSR), 224.64 s (BMSS) and 5682.87 s (AMSS) in the case of 6 clusters. The optimal value of the objective function is 22,659.48 in the case of 5 clusters and 19,834.48 in the case of 6 clusters; the optimal solution of the MSSR formulation is not a ‘legal clustering in neither case. The optimal clustering can be seen in Fig. 3.

Another possible example to add further constraints could be the following: we know some sort of categorization on the data set and we assume that at most 2 different categories (or 3 or 4) can appear in each cluster. For instance, we would like to assign abstracts to sections in a conference. We have some general categorization of topics. Homogeneous sections are not achievable, but at most two different categories can appear in a section.

Finally, it is also an important advantage of the proposed models that we do not need to calculate the cluster centers, a distance matrix is a sufficient input to solve the clustering problem. However, in Euclidean spaces, it is easy to calculate the cluster centers (take the mean in every dimension), but in certain applications (see, for instance, Majstorović et al. 2018) it is the drawback of using the KMEANS algorithm.

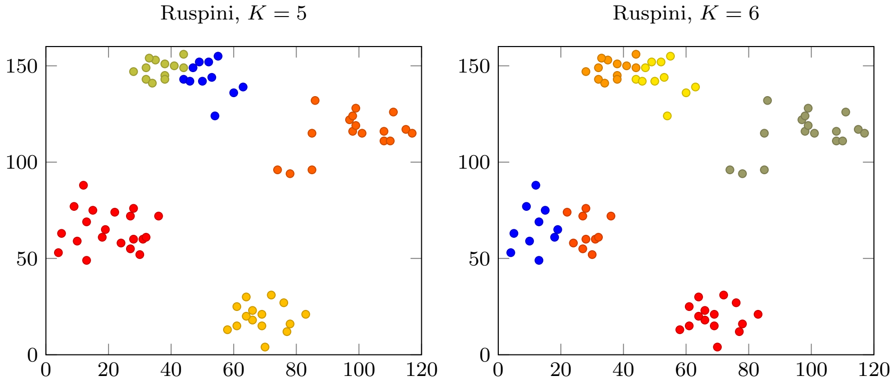

Fig. 3 Optimal minimum sum-of-squares clusters for constrained clustering problems

4 Conclusion

In this paper, we investigated MILP formulations for the minimum sum-of-squares clustering problem. However, these formulations have longer running times than the well-known KMEANS algorithm, however, for sample size at most 100 it is still tolerable. If in some application it is crucial to work with global optimum, these formulations give a possibility for it. The advantage of MILP formulation compared to the other methods is that we can take into consideration further aspects by posing them as a linear constraint. We presented this possibility with further requirements on cluster sizes.

Funding Open access funding provided by Corvinus University of Budapest.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Ágoston KCs, E.-Nagy M (2021) Mixed integer linear programming formulation for K-means cluster problem. In: Drobne S, Stirn LZ, Kljajić BM, Povh J, Žerovnik J (eds) Proceedings of the 16th international symposium on operational research in Slovenia, pp 49–54

Aloise D, Deshpande A, Hansen P, Popat P (2009) NP-hardness of Euclidean sum-of-squares clustering. Mach Learn 75(2):245–248

Awasthi P, Bandeira AS, Charikar M, Krishnaswamy R, Villar S, Ward R (2015) Relax, no need to round: integrality of clustering formulations. In: ITCS '15: proceedings of the 2015 conference on innovations in theoretical computer science, pp 191–200

Bradley PS, Bennett KP, Demiriz A (2000) Constrained K-means clustering. https://www.microsoft.com/en-us/research/publication/constrained-k-means-clustering/

Cornuejols G, Nemhauser GL, Wolsey LA (1980) A canonical representation of simple plant location-problems and its applications. SIAM J Algebr Discrete Methods 1:261–272

Davidson I, Ravi SS (2007) The complexity of non-hierarchical clustering with instance and cluster level constraints. Data Min Knowl Disc 14:25–61

Dorndorf U, Pesch E (1994) Fast clustering algorithms. ORSA J Comput 6:141–153

du Merle O, Hansen P, Jaumard B, Mladenovic N (1999) An interior point algorithm for minimum sum-of-squares clustering. SIAM J Sci Comput 21:1485–1505

Dua D, Graff C (2019) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml. University of California, School of Information and Computer Science, Irvine

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7:179–188

Gilpin S, Nijssen S, Davidson IN (2012) Formalizing hierarchical clustering as integer linear programming. In: Proceedings of the twenty-seventh AAAI conference on artificial intelligence, July 14–18, 2013, Bellevue, Washington, USA, pp 372–378

Hansen P, Jaumard B (1997) Cluster analysis and mathematical programming. Math Program 79(B):191–215

Hartigan JA, Wong MA (1979) A K-means clustering algorithm. J R Stat Soc Ser C 28(1):100–108

Kaufman L, Rousseeuw PJ (1990) Finding groups in data: an introduction to cluster analysis. Wiley, Hoboken

Kondor G (2022) NP-hardness of m-dimensional weighted matching problems. Theoret Comput Sci 930:33–36

Kulkarni G, Fathi Y (2007) Integer programming models for the q-mode problem. Eur J Oper Res 182:612–625

Majstorović S, Sabo K, Jung J, Klarić M (2018) Spectral methods for growth curve clustering. CEJOR 26(3):715–737

Malinen MI, Fränti P (2014) Balanced k-means for clustering. In: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), vol 2014. Springer, pp 32–41

Peng J, Wei Y (2007) Approximating K-means-type clustering via semidefinite programming. SIAM J Optim 18:186–205

Piccialli V, Sudoso AM, Wiegele A (2021) SOS-SDP: an exact solver for minimum sum-of-squares clustering. INFORMS J Comput 34:2144–2162

Pyatkin A, Aloise D, Mladenović N (2017) NP-hardness of balanced minimum sum-of-squares clustering. Pattern Recogn Lett 97:44–45

Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66:846–850

Rao MR (1971) Cluster analysis and mathematical programming. J Am Stat Assoc 66:622–626

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Comput Appl Math 20:53–65

Rujeerapaiboon N, Schindler K, Kuhn D, Wiesemann W (2019) Size matters: cardinality-constrained clustering and outlier detection via conic optimization. SIAM J Optim 29:1211–1239

Ruspini EH (1970) Numerical methods for fuzzy clustering. Inf Sci 2:319–350

Slonim N, Aharoni E, Crammer K (2013) Hartigan's K-means versus Lloyd's K-means: is it time for a change? In: Proceedings of the twenty-third international joint conference on artificial intelligence, Bejing, China, pp 1677–1684

Vinod HD (1969) Integer programming and the theory of grouping. J Am Stat Assoc 64:506–519

Zhu S, Wang D, Li T (2010) Data clustering with size constraints. Knowl-Based Syst 23:883–889

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.