Les Cahiers du GERAD ISSN: 0711–2440

K-means with controlled center updates: Rethinking the centroid update rule

M. H. Midingoyi, D. Aloise, J. Brimberg, Z. Drezner

G–2026–18

April 2026

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

K-means with controlled center updates: Rethinking the centroid update rule

Mahuton Hugues Midingoyi [a, b]

Daniel Aloise [a, b]

Jack Brimberg [a, c]

Zvi Drezner [d]

[a] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[b] *Department of Computer and Software Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4*

[c] *Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston (On), Canada, K7K 7B4*

[d] *College of Business and Economics, California State University-Fullerton, Fullerton, 92834, CA, United States*

mahuton-hugues.midingoyi@polymtl.ca
daniel.aloise@gerad.ca
jack.brimberg@rmc.ca

April 2026
Les Cahiers du GERAD
G–2026–18

Abstract : The k-means algorithm is one of the most widely used methods for solving the minimum sum-of-squares clustering (MSSC) problem, but it is well known to be sensitive to initialization and prone to convergence to poor local minima. In this work, we revisit the classical centroid update rule and propose a variant of k-means in which cluster centers are updated through a controlled movement toward their corresponding centroids. The proposed approach introduces a parameter that governs the magnitude of the update, allowing the algorithm to deviate from exact centroid updates while preserving the simplicity and computational efficiency of k-means. Computational experiments on benchmark datasets demonstrate that the proposed method consistently improves solution quality compared to standard k-means under equal computational budgets, achieving improvements of up to 15% without incurring additional computational overhead. These results suggest that strictly updating centers to centroids at every iteration is not always optimal from an algorithmic perspective, and that controlled update steps can lead to better clustering performance.

Keywords : K-means clustering; centroid update optimization; minimum sum-of-squares clustering; local search heuristics

Acknowledgements: This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), grants 2020–04846 and 2023–04466.

1 Introduction

Clustering is a fundamental task in data analysis and pattern recognition. It aims to partition a set of data points into subsets, called *clusters*, such that points within the same cluster are similar to each other, while points in different clusters should differ one from another.

Among the clustering methods described in the literature, the k-means algorithm (Forgy, 1965) is by far the most widely known, with over two million citations on Google Scholar as of March 2026. It was also ranked by the IEEE Computer Society as the second most influential algorithm in data mining (Wu et al., 2008).

The k-means algorithm is a local descent method that seeks to minimize the sum of squared Euclidean distances between each data point and the center of its assigned cluster, which is equivalent to minimizing intra-cluster variance. Given an initial partition, k-means alternates between (i) assigning points to their nearest center, and (ii) updating the center positions until stability is reached.

Cluster centers are iteratively updated in (ii) to the centroids of their assigned data points. This follows from the fact that, for a fixed assignment, the intra-cluster sum of squared Euclidean distances is minimized when each cluster center coincides with the centroid of its assigned points, as implied by first-order optimality conditions (see, e.g., (Aloise and Hansen, 2009)). While this update is optimal for the corresponding subproblem, it enforces an exact minimization at each iteration, which may limit the exploration of the solution space and contribute to convergence to poor local minima.

In this paper, we revisit this centroid update rule and propose a variant of the k-means algorithm based on controlled center updates. Instead of moving cluster centers directly to the centroids of their assigned points, we introduce a parameterized update that governs the magnitude of the movement toward the centroid. This modification generalizes the standard k-means algorithm, which is recovered as a special case.

Moreover, we show that the proposed method preserves the convergence properties of k-means for a range of parameter values. Through computational experiments on benchmark datasets, we demonstrate that controlled center updates can improve solution quality compared to standard k-means under equal computational budgets, with improvements of up to approximately 15%. Since k-means serves as a core local descent procedure in many state-of-the-art clustering methods, and is also frequently embedded as a subroutine within algorithms addressing a wide range of problems, improvements to its behavior may translate into enhanced performance across diverse applications.

The remainder of the paper is organized as follows. Section 2 defines mathematically the minimum sum-of-squares clustering problem (MSSC), which is the mathematical optimization problem heuristically approached by the k-means algorithm. Section 3 introduces the proposed algorithm. Moreover, we demonstrate that the method converges to an MSSC local minimum. In Section 4, we report and analyze our computational experiments on benchmark clustering instances used in the literature. Finally, Section 5 concludes the paper.

2 Problem definition

Given a set $P = \{p_1, p_2, \dots p_n\}$ of n data points in $\mathbb{R}^d$, MSSC consists in partitioning P into k clusters such that the sum of squared Euclidean distances from each point to the center of its assigned cluster is minimized. Formally, MSSC can be stated as follows:

$$\min \quad \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \|p_i - \mu_j\|^2 \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{k} x_{ij} = 1 \qquad \forall i \in \{1, \dots, n\}, \tag{2}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in \{1,\dots,n\},\ \forall j \in \{1,\dots,k\}, \tag{3}$$

$$\mu_j \in \mathbb{R}^d \qquad \forall j \in \{1,\dots,k\}, \tag{4}$$

where $\|.\|$ is the Euclidean norm in $\mathbb{R}^d$. The binary decision variables x_{ij}, for $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,k\}$, express whether data point p_i is assigned to cluster j ($x_{ij} = 1$) or not ($x_{ij} = 0$), while μ_j, for $j \in \{1,\dots,k\}$, correspond to the locations of the cluster centers in $\mathbb{R}^d$. Finally, constraints (2) ensure that each point is assigned to exactly one cluster.

MSSC is known to be NP-hard in general dimension for $k \geq 2$ clusters (Aloise et al., 2009). Hence, many heuristics have been proposed in the literature to tackle large-scale instances of the problem (e.g. (Bagirov and Yearwood, 2006; Gribel and Vidal, 2019; Hansen and Mladenović, 2001; Likas et al., 2003; Mansueto and Schoen, 2021; Mussabayev et al., 2023)).

The k-means algorithm steps (i) and (ii), described in Section 1, are based on the following two properties of the problem. First, for fixed centers μ, the optimal decision corresponds to assigning the data points to their closest centers. Second, for fixed assignments x, the resulting optimization problem is convex and optimized by locating the cluster centers at the centroid of their assigned data points. Algorithm 1 presents the pseudo-code of the k-means heuristic according to our notation.

Algorithm 1 **k-means algorithm**

Require: Data points $p_1,\dots,p_n$, number of clusters k, tolerance δ

1: Initialize centers $\mu^{(0)} = \{\mu_1^{(0)},\dots,\mu_k^{(0)}\}$; set $t \leftarrow 0$.

2: **repeat**

3: For each point p_i, assign it to the nearest center:

$$C_j^{(t)} = \{\, p_i : j = \arg\min_{\ell \in \{1,\dots,k\}} \|p_i - \mu_\ell^{(t)}\| \,\}.$$

4: Recompute each center as the centroid of its assigned points:

$$\mu_j^{(t+1)} = \frac{1}{|C_j^{(t)}|} \sum_{p_i \in C_j^{(t)}} p_i, \quad j = 1,\dots,k.$$

5: $t \leftarrow t + 1$.

6: **until** $\max_{j=1,\dots,k} \|\mu_j^{(t)} - \mu_j^{(t-1)}\| \leq \delta$

7: **return** $\{C_1^{(t)},\dots,C_k^{(t)}\}$ and $\{\mu_1^{(t)},\dots,\mu_k^{(t)}\}$.

3 K-means with controlled center updates

The main difference between the standard k-means algorithm presented in Algorithm 1 and our proposed controlled center update k-means, hereafter denoted *controlled k-means*, concerns the center update step.

In the classical k-means algorithm, cluster centers are iteratively updated to the centroids of their assigned data points. This update is optimal for the corresponding subproblem with fixed assignments. However, it enforces an exact minimization at each iteration, which may limit the exploration of the solution space and contribute to convergence to poor local minima.

In our approach, we replace the exact centroid update by a controlled movement toward the centroid. Instead, we follow the direction given by the gradient at the current center, which points towards the centroid, but take a controlled step along the gradient direction.

Our proposed update rule for the cluster centers is given by:

$$\mu_j^{(t+1)} = \mu_j^{(t)} + \alpha(\nu_j^{(t)} - \mu_j^{(t)}), \tag{5}$$

where $\nu^{(t)} = \{\nu_1^{(t)},\dots,\nu_k^{(t)}\}$ correspond to the actual centroids of the clusters $C_1,\dots,C_k$ at iteration t. The parameter $\alpha > 0$ controls the magnitude of the update. When $\alpha = 1$, the method reduces to

the standard k-means algorithm. Values $0 < \alpha < 1$ correspond to conservative updates, while $\alpha > 1$ produces updates that overshoot the centroid, placing the center beyond its current cluster mean.

Algorithm 2 presents the pseudo-code of the controlled k-means algorithm. We observe that the only modification with respect to the standard k-means consists of updating the cluster centers according to (5). The proposed modification is simple to implement and preserves the computational complexity of the standard k-means algorithm presented in Algorithm 1.

Algorithm 2 Controlled k-means algorithm

Require: Data points $p_1, \ldots, p_n$, number of clusters k, tolerance δ, and $\alpha > 0$.

1: Initialize centers $\mu^{(0)} = \{\mu_1^{(0)}, \ldots, \mu_k^{(0)}\}$; set $t \leftarrow 0$.

2: **repeat**

3: For each point p_i, assign it to the nearest center:

$$C_j^{(t)} = \{\, p_i : j = \arg\min_{\ell \in \{1,\ldots,k\}} \|p_i - \mu_\ell^{(t)}\| \,\}.$$

4: Compute, for each cluster, the mean of its assigned points:

$$\nu_j^{(t)} = \frac{1}{|C_j^{(t)}|} \sum_{p_i \in C_j^{(t)}} p_i, \quad j = 1, \ldots, k.$$

5: For each cluster, set the center :

$$\mu_j^{(t+1)} = \mu_j^{(t)} + \alpha(\nu_j^{(t)} - \mu_j^{(t)}), \quad j = 1, \ldots, k.$$

6: $t \leftarrow t + 1$.

7: **until** $\max_{j=1,\ldots,k} \|\mu_j^{(t)} - \mu_j^{(t-1)}\| \leq \delta$

8: **return** $\{C_1^{(t)}, \ldots, C_k^{(t)}\}$ and $\{\mu_1^{(t)}, \ldots, \mu_k^{(t)}\}$.

In the following, we demonstrate that the controlled k-means algorithm is convergent for a certain range of α values.

Proposition 1. *Algorithm 2 converges in a finite number of iterations to a local minimum of the MSSC for* $0 < \alpha < 2$.

Proof. The standard k-means algorithm has two monotone steps: (i) reassignment of data points which can only lead to improving solutions; and (ii) center updates to the cluster centroids, which decreases the MSSC objective to the minimum for fixed assignments. As the MSSC objective is bounded from below (zero), the standard k-means algorithm converges to a minimum.

Step (i) is not modified in the controlled k-means heuristic presented in Algorithm 2, and hence, our demonstration is focused on proving that updating the cluster centers according to (5) decreases the MSSC objective function.

For a fixed assignment, the contribution of cluster C_j to the objective is

$$\Phi(j) = \sum_{p_i \in C_j} \|p_i - \mu_j\|^2.$$

Using the identity

$$\|p - \mu\|^2 = \|p\|^2 - 2p^\top \mu + \|\mu\|^2,$$

we obtain

$$\Phi(j) = \sum_{p_i \in C_j} \|p_i\|^2 - 2 \left(\sum_{p_i \in C_j} p_i \right)^\top \mu_j + |C_j| \|\mu_j\|^2.$$

Since the centroid

$$\nu_j = \frac{1}{|C_j|} \sum_{p_i \in C_j} p_i,$$

then $\sum_{p_i \in C_j} p_i = |C_j|\nu_j$, and then we have

$$\Phi(j) = \sum_{p_i \in C_j} \|p_i\|^2 - 2|C_j|\,\nu_j^\top \mu_j + |C_j|\|\mu_j\|^2. \tag{6}$$

Now, let us focus on the terms of (6) depending on μ_j only, i.e.,

$$|C_j|\|\mu_j\|^2 - 2|C_j|\nu_j^\top \mu_j = |C_j|\big(\|\mu_j\|^2 - 2\nu_j^\top \mu_j\big)$$

By adding and subtracting $|C_j|\|\nu_j\|^2$, we obtain:

$$|C_j|\big(\|\mu_j\|^2 - 2\nu_j^\top \mu_j + \|\nu_j\|^2\big) - |C_j|\|\nu_j\|^2.$$

Now using the identity on $\|\mu_j - \nu_j\|^2$, and bringing it back to (6), we obtain

$$\Phi(j) = \underbrace{\sum_{p_i \in C_j} \|p_i\|^2 - |C_j|\|\nu_j\|^2}_{\text{constant w.r.t. } \mu_j} + |C_j|\|\mu_j - \nu_j\|^2, \tag{7}$$

which means that, for a fixed assignment, the contribution of cluster C_j to the MSSC objective depends only on the distance between μ_j and the cluster centroid ν_j.

After updating the cluster center of C_j according to the modified rule (5), the new cluster center μ_j' of C_j is such that:

$$\mu_j' - \nu_j = (1-\alpha)(\mu_j - \nu_j) \quad \text{and so} \quad \|\mu_j' - \nu_j\|^2 = (1-\alpha)^2\|\mu_j - \nu_j\|^2.$$

Therefore, for that fixed assignment, the MSSC objective changes by the factor $(1-\alpha)^2$. If $0 < \alpha < 2$, then $(1-\alpha)^2 < 1$, which results that the MSSC objective decreases with the utilization of (5). □

The corollary below follows from proposition 1.

Corollary 1. *For a fixed assignment and $0 < \alpha < 2$, the sequence $\{\mu_j^{(t)}\}$ converges to the centroid ν_j in Algorithm 2.*

The k-means algorithm can itself be viewed as a particular case of Cooper's location-allocation framework (Cooper, 1964), which has been extensively studied in facility location theory for a variety of objective function. In this broader context, variations of the update step have been investigated, including strategies that modify the step size or deliberately slow convergence in order to explore alternative descent paths (see, e.g., (Brimberg and Drezner, 2025; Drezner, 1992, 1995; Ostresh Jr, 1978; Vardi and Zhang, 2001)). These approaches have been shown to influence the trajectory of the algorithm and, in some cases, improve solution quality.

However, such ideas have not been explored within the standard k-means framework, where the centroid update is typically performed exactly at each iteration. The proposed controlled update mechanism fills this gap by introducing a parameterized update rule that modifies the descent path of k-means while preserving its simplicity and local minimum convergence.

4 Computational experiments

In this section, we present a series of computational experiments to evaluate the proposed algorithm on real-world datasets. The experiments were conducted on an Intel Core i7-3770 3.40 GHz processor with 16 GB of RAM. We first present a description of the datasets in subsection 4.1 and the details of the implementation and parameter choices in subsection 4.2. Then, in subsection 4.3, we perform an analysis of the numerical results of controlled k-means under different α values in comparison with those obtained by standard k-means. Finally, in subsection 4.4, we assess our proposed controlled k-means algorithm on a more practical setting, where the clustering algorithm is allowed to run for a fixed number of executions, and the best solution is returned.

4.1 Benchmark datasets

The experiments were done on datasets from the UCI machine learning repository (http://archive.ics.uci.edu/ml/). Table 1 reports the number of samples n and the dimensionality d of each used dataset. They contain between 297 and 20,000 samples and from 2 to 34 dimensions. For each dataset, we consider $k \in \{2, \lfloor\sqrt{n}/4\rfloor, \lfloor\sqrt{n}/2\rfloor, \lfloor\sqrt{n}\rfloor\}$. Specifically, $k = \lfloor\sqrt{n}\rfloor$ serves as a boundary case where the number of clusters and the average number of points per cluster are supposedly balanced (both equal to $\sqrt{n}$). This choice allows us to observe the algorithm's behavior as the partition complexity scales with the dataset size.

Table 1: Datasets used in the experiments.

Dataset	n	d
Heart Disease	297	13
Liver Disorders	345	6
Ionosphere	351	34
Congressional Voting	435	16
Breast Cancer	683	9
Pima Indians Diabetes	768	8
TSPLib1060	1,060	2
Image Segmentation	2,310	19
TSPLib3038	3,038	2
Page Blocks	5,473	10
EEG Eye State	14,980	14
D15112	15,112	2
Pendigit	10,992	16
Letters	20,000	16

4.2 Implementation details and parameter settings

We set the tolerance $\delta = 10^{-4}$ for checking the convergence of both k-means and controlled k-means. Nonetheless, the controlled k-means algorithm may perform several additional iterations until convergence, even after the final local optimum partition has already been reached. For this reason, and motivated by Corollary 1, we propose an alternate version of controlled k-means, denoted *hybrid-controlled k-means*, that works as follows: if the partition does not change between two consecutive iterations, the cluster centers are directly updated to the corresponding cluster centroids, and the algorithm proceeds with the next iteration. The qualification 'hybrid' is due to the fact that cluster centers are either updated according to (5), or by the step (ii) used within classical k-means.

Cluster centers were initialized using the k-means++ method of (Arthur and Vassilvitskii, 2007), ensuring that, for each iteration index, all k-means variants start from the same initial solution, even though the total number of iterations may differ across the algorithms.

Finally, the algorithms were implemented in Python and are available at https://github.com/huguesmid/controlled-k-means.

4.3 Impact of α

We evaluated controlled k-means and hybrid-controlled k-means for $\alpha \in \{0.1, 0.5, 0.9, 1.1, 1.5, 1.9\}$. For each instance, both algorithms were allowed to perform multiple independent executions under a total time budget equivalent to that required for 100 runs of the standard k-means heuristic.

Solution quality for each data instance was measured as the average percentage gap relative to the mean objective value obtained from the 100 standard k-means executions. Consequently, a negative gap indicates that the controlled k-means variants outperformed the standard k-means under the same CPU time budget.

Figure 1 shows the distributions of the average gap between controlled k-means and hybrid-controlled k-means with respect to standard k-means across all the datasets listed in Table 1, for $k = 2, \lfloor\sqrt{n}/4\rfloor, \lfloor\sqrt{n}/2\rfloor$ and $\lfloor\sqrt{n}\rfloor$ clusters. The figure is clipped in the y-axis to show values between $\pm 5\%$.

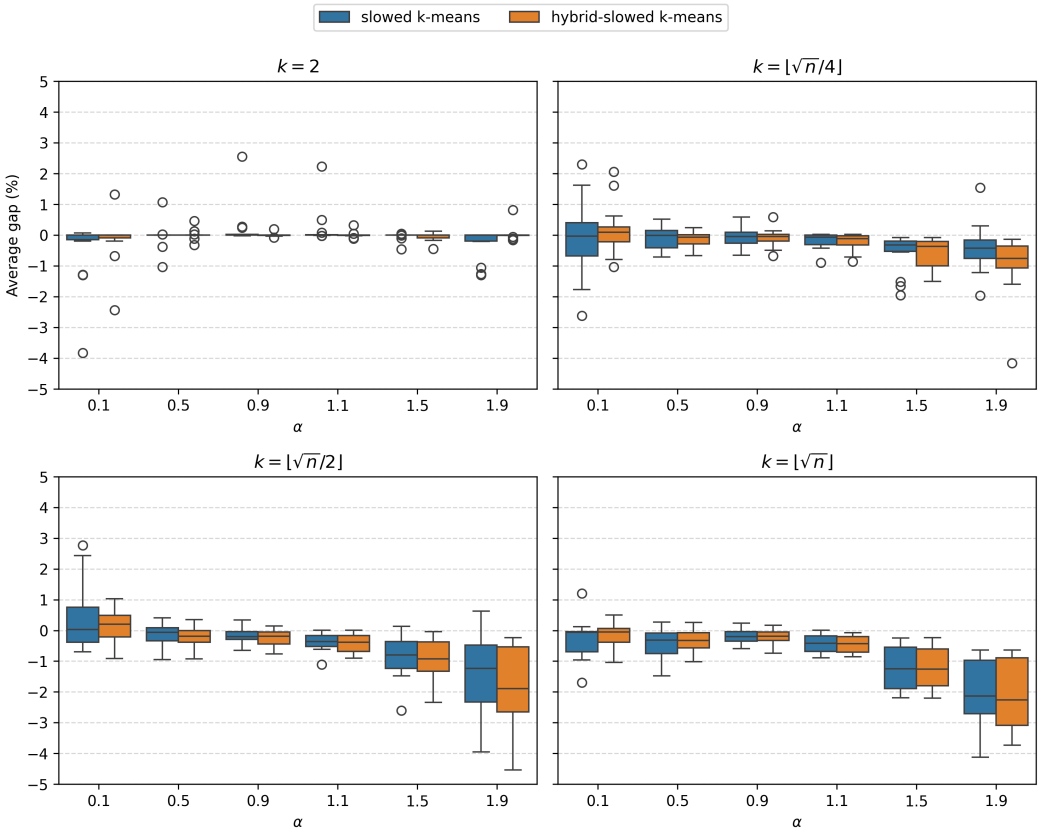

Figure 1: Distribution of the average gaps of controlled and hybrid-controlled k-means solutions relative to those obtained by standard k-means in the 14 tested datasets.

The results indicate that both controlled k-means variants perform increasingly well with respect to k-means as the number of clusters increases. However, we observed in our experiments that the performance of controlled k-means can be quite bad. It reached an average gap of approximately $+25\%$ in instance *EEG Eye State* for $k = 2$ clusters and $\alpha = 0.1$ and $\alpha = 1.9$. This is explained by the fact that a much smaller number of executions are performed in these cases, as compared to CPU time of 100 standard k-means executions used as limiting computational budget in this set of experiments.

Figure 2 reports the average number of executions of the controlled k-means algorithm that are performed for the tested datasets. As can be observed, the number of controlled k-means executions is always smaller than 100, and particularly small for $\alpha = 0.1$ and $\alpha = 1.9$.

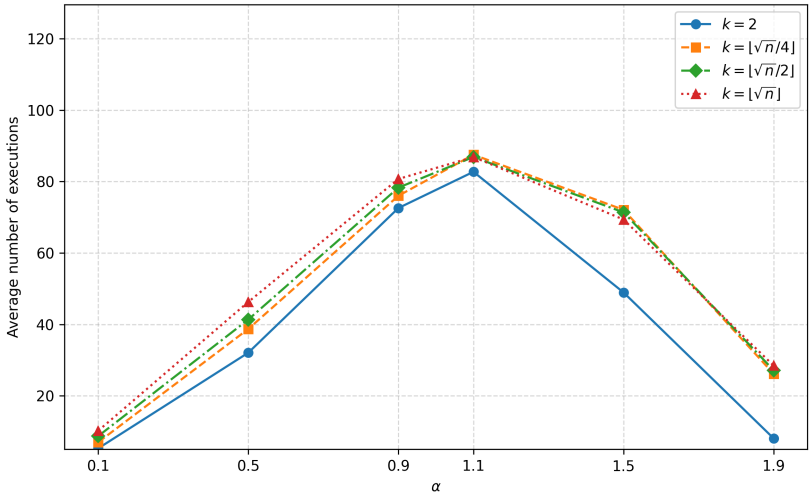

Figure 2: Average number of executions performed by controlled k-means for a CPU time corresponding to 100 k-means executions.

In contrast, the hybrid controlled version is more robust, never exceeding +2.5% in average gap. The most significant gains occur at higher α values as the number of clusters increases. Specifically, for the *Page Blocks* dataset with $k = \lfloor \sqrt{n}/2 \rfloor$ clusters, using $\alpha = 1.9$ yields an average improvement of nearly -4.54% in comparison with standard k-means.

The hybrid version accelerates convergence by reducing the number of algorithm iterations, thereby allowing more executions of the hybrid-slowed k-means within the allowed CPU time. Figure 3 reports the average number of executions of the hybrid-controlled k-means that are performed within the CPU time corresponding to 100 k-means executions. Notably, for $\alpha = 1.1$ and $\alpha = 1.5$, the number of executions exceeded 100 for $k = 2$ as well as for $\alpha = 1.5$ and $k = \lfloor \sqrt{n}/4 \rfloor$ clusters, indicating faster average convergence than standard k-means in these cases.

Finally, we conducted statistical tests to evaluate the impact of α on the controlled and hybrid controlled k-means algorithms while keeping the number of executions fixed at 100. To this end, we implemented two variants of each algorithm: one in which α is randomly selected from the interval $[0.1, 1]$, and another in which α is randomly selected from the interval $[1, 1.9]$. The null hypothesis states that, for both controlled and hybrid-controlled k-means, the average results obtained by the two variants are equal. Student's t-tests were performed only for $k = \lfloor \sqrt{n}/4 \rfloor$, $\lfloor \sqrt{n}/2 \rfloor$, and $\lfloor \sqrt{n} \rfloor$, since for $k = 2$ the number of local minima might be very limited, not leading to an approximately normal distribution of MSSC values.

In general, the null hypothesis is rejected for some data instances, but no consistent conclusion can be drawn regarding which variant performs better for the entire set of tested instances. For example, Figure 4 shows the histograms of the MSSC local minima obtained over 100 runs of the controlled k-means algorithm on the *Page Blocks* dataset for the two variants, with $\alpha \in [0.1, 1]$ and $\alpha \in [1, 1.9]$, for $k = 2$, $\lfloor \sqrt{n}/4 \rfloor$, $\lfloor \sqrt{n}/2 \rfloor$, and $\lfloor \sqrt{n} \rfloor$. For the three latter values of k, the p-values were greater than 0.05, indicating that the Student's t-test did not find sufficient evidence to reject the null hypothesis at the 5% significance level. In fact, the histograms show a substantial overlap between the distributions obtained with the two variants, suggesting that both ranges of α produce similar MSSC values across the runs.

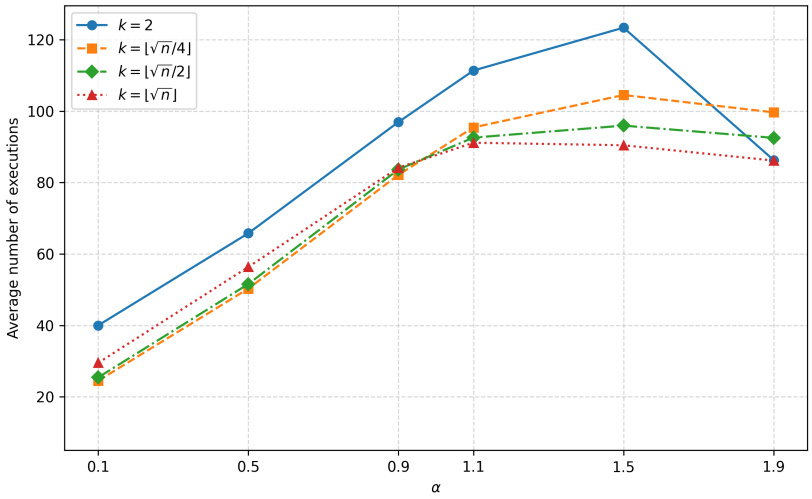

Figure 3: Average number of executions performed by hybrid-controlled k-means for a CPU time corresponding to 100 k-means executions.

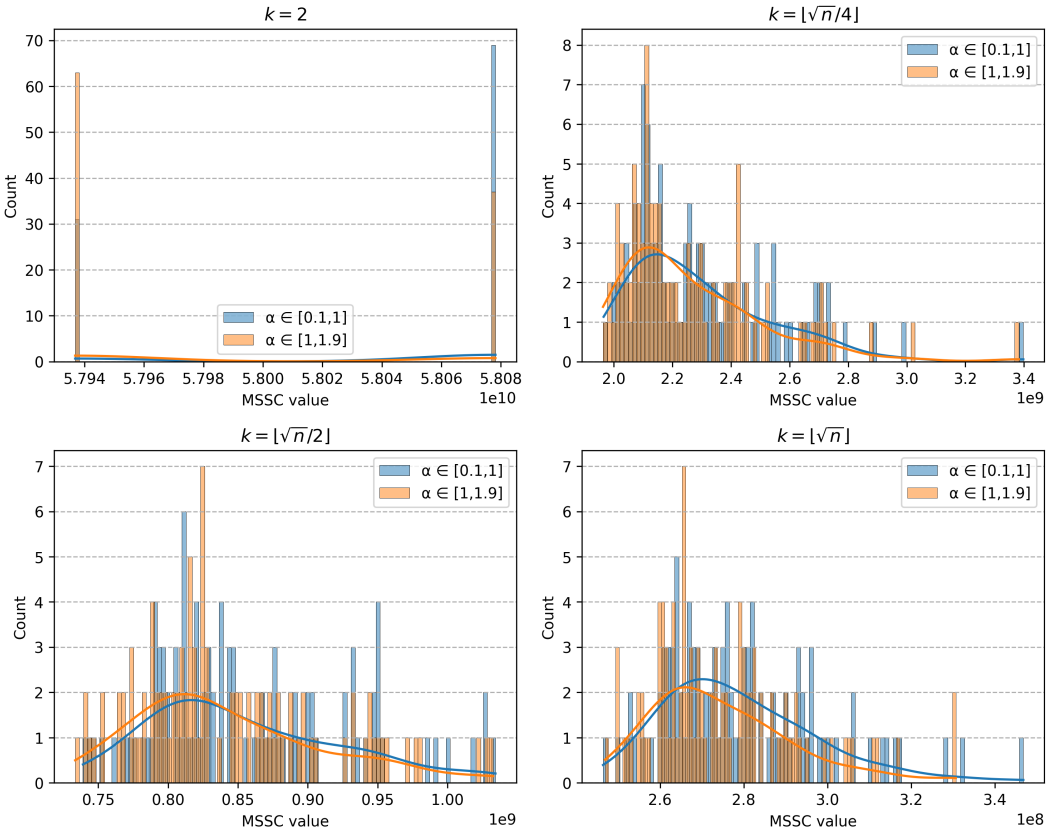

Figure 4: Histograms of MSSC local minima obtained over 100 runs of the controlled k-means algorithm on the Page Blocks dataset for two variants of α: $\alpha \in [0.1, 1]$ and $\alpha \in [1, 1.9]$.

4.4 Practical performance analysis

In this section, we evaluate our approach in a setting that more closely reflects how users and practitioners apply the k-means algorithm in real-world clustering applications.

Popular programming libraries such as `scikit-learn` (Pedregosa et al., 2011) provide implementations of k-means in which the algorithm is executed multiple times, and the best clustering solution is returned as the final output.

Our results presented in the previous section indicate that the hybrid-controlled variant is both the most robust and the fastest when cluster center updates are overshot, i.e, using $\alpha \in [1, 1.9]$. This is the version we compare against standard k-means in this section.

Table 2 reports the results obtained by the hybrid-controlled k-means algorithm and the standard k-means over 100 runs, where each run is initialized by the k-means++ initialization method of (Arthur and Vassilvitskii, 2007). Column *improv.* indicates the relative improvement (in %) of the best solution found by our algorithm with respect to the best solution obtained by the standard k-means in each of the tested data instances. The total computational times spent by both algorithms are also reported in the associated columns.

Table 2: Results of hybrid-controlled k-means with uniform random values of alpha in $[1, 1.9]$ and standard k-means in 100 executions.

Dataset	$k = 2$			$k = \lfloor\sqrt{n}/4\rfloor$			$k = \lfloor\sqrt{n}/2\rfloor$			$k = \lfloor\sqrt{n}\rfloor$		
	h.c. k-means		k-means	h.c. k-means		k-means	h.c. k-means		k-means	h.c. k-means		k-means
	Improv.	T(s)	T(s)	Improv.	T(s)	T(s)	Improv.	T(s)	T(s)	Improv.	T(s)	T(s)
heart	0.00	0.19	0.24	−3.71	0.42	0.46	−3.95	0.86	0.74	−9.45	1.58	1.24
liver	−0.01	0.13	0.16	−2.82	0.39	0.37	−6.42	0.98	0.75	−7.22	1.66	1.20
ionosphere	−1.30	0.24	0.20	−7.93	0.49	0.40	−8.11	0.92	0.88	−6.03	1.90	1.58
congress	−1.31	0.26	0.18	−1.65	0.67	0.59	−3.29	1.15	0.97	−5.37	2.22	1.80
breast	0.00	0.24	0.15	−2.22	0.79	0.70	−4.56	1.73	1.48	−5.78	3.45	2.84
pima	0.00	0.30	0.36	−5.71	0.94	0.85	−6.78	2.32	1.93	−6.83	4.86	3.89
tsplib1060	0.00	0.20	0.19	−2.93	1.70	1.64	−6.59	3.03	2.44	−7.75	5.77	4.58
image2	−3.85	0.64	0.66	−7.30	6.20	6.61	−8.68	15.97	16.43	−6.15	39.25	37.46
tsplib3038	0.00	0.42	0.57	−1.97	6.15	6.52	−2.53	14.46	13.11	−4.71	31.09	27.17
page	−0.17	0.58	0.66	−14.93	19.63	18.10	−14.89	52.84	52.46	−11.86	164.60	157.07
pendigit	−0.55	4.06	2.74	−2.94	94.34	98.81	−2.46	273.30	285.63	−1.85	737.27	760.52
eye	−10.78	0.71	0.75	−1.39	268.93	307.18	−1.94	648.88	707.97	−1.85	1537.70	1574.37
d15112	0.00	2.05	2.59	−1.83	107.53	122.93	−2.22	255.05	250.77	−3.13	612.64	602.02
letter	0.00	8.86	7.11	−1.60	424.01	506.87	−1.43	968.53	1092.63	−1.61	2395.45	2473.97
Average	**-1.28**	**1.35**	**1.18**	**-4.21**	**66.59**	**76.57**	**-5.27**	**160.00**	**173.44**	**-5.69**	**395.68**	**403.55**

We can conclude from the table that:

- The hybrid-controlled k-means consistently produces clustering solutions that are at least as good as those obtained by the standard k-means. In all tested instances, the best MSSC value found by the controlled variant is equal to or better than that obtained by the standard algorithm.
- Significant improvements are observed on several datasets. For instance, on the *Page Blocks* dataset with $k = \lfloor\sqrt{n}/4\rfloor$, the hybrid-controlled k-means improves the best MSSC value by up to 14.93% compared with the standard k-means.
- The magnitude of the improvement tends to increase with the number of clusters. On average, the relative gap improves from -1.28% for $k = 2$ to -5.69% for $k = \lfloor\sqrt{n}\rfloor$, indicating that the proposed controlled approach becomes more effective as the clustering problem becomes more complex.
- The computational times of the hybrid-controlled k-means remain comparable to those of the standard k-means. In fact, for larger values of k, the average runtime of the controlled variant is slightly lower.

These results indicate that overshooting cluster center updates can considerably improve the quality of the solutions obtained by k-means without introducing computational overhead.

5 Conclusion

In this paper, we revisited the classical centroid update rule in the k-means algorithm and proposed a variant based on controlled center updates for the minimum sum-of-squares clustering (MSSC) problem. The main contribution lies in introducing a simple parameterized update mechanism that allows cluster centers to move toward their centroids in a controlled manner, rather than enforcing exact centroid updates at each iteration. From a theoretical perspective, we showed that the proposed method preserves the convergence properties of k-means, reaching a local minimum of the MSSC problem for a range of parameter values.

From a computational standpoint, the proposed approach consistently improves solution quality compared to standard k-means under equal computational budgets, with gains of up to 15% observed on benchmark data instances. These improvements are achieved without increasing computational complexity and with only minimal modifications to the standard algorithm.

Given that the k-means algorithm serves as a fundamental local descent procedure in many state-of-the-art heuristics for MSSC, the proposed controlled update mechanism can be seamlessly integrated into these frameworks. As a result, improvements at the level of k-means may translate into enhanced performance for a broader class of clustering methods.

Overall, our results suggest that the classical centroid update is not necessarily optimal from an algorithmic perspective, and that controlled deviations from exact updates can enhance the exploration of the solution space. This observation motivates the development of improved local search heuristics for clustering and related location-allocation problems, as well as the investigation of controlled update mechanisms in extensions such as online and mini-batch k-means (Sculley, 2010).

References

Aloise, D., Deshpande, A., Hansen, P., Popat, P., 2009. Np-hardness of euclidean sum-of-squares clustering. Machine Learning 75, 245–248.

Aloise, D., Hansen, P., 2009. A branch-and-cut sdp-based algorithm for minimum sum-of-squares clustering. Pesquisa Operacional 29, 503–516.

Arthur, D., Vassilvitskii, S., 2007. k-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. pp. 1027–1035.

Bagirov, A.M., Yearwood, J., 2006. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. European journal of operational research 170, 578–596.

Brimberg, J., Drezner, Z., 2025. Less is more: adjusting convergence of cooper's algorithm in the continuous location-allocation problem. Journal of the Operational Research Society 76, 1599–1611.

Cooper, L., 1964. Heuristic methods for location-allocation problems. SIAM review 6, 37–53.

Drezner, Z., 1992. A note on the weber location problem. Annals of Operations Research 40, 153–161.

Drezner, Z., 1995. A note on accelerating the Weiszfeld procedure. Location Science 3, 275–279.

Forgy, E.W., 1965. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. biometrics 21, 768–769.

Gribel, D., Vidal, T., 2019. Hg-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering. Pattern Recognition 88, 569–583.

Hansen, P., Mladenović, N., 2001. J-means: a new local search heuristic for minimum sum of squares clustering. Pattern recognition 34, 405–413.

Likas, A., Vlassis, N., Verbeek, J.J., 2003. The global k-means clustering algorithm. Pattern recognition 36, 451–461.

Mansueto, P., Schoen, F., 2021. Memetic differential evolution methods for clustering problems. Pattern Recognition 114, 107849.

Mussabayev, R., Mladenovic, N., Jarboui, B., Mussabayev, R., 2023. How to use k-means for big data clustering? Pattern Recognition 137, 109269.

Ostresh Jr, L.M., 1978. On the convergence of a class of iterative methods for solving the weber location problem. Operations Research 26, 597–609.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in python. the Journal of machine Learning research 12, 2825–2830.

Sculley, D., 2010. Web-scale k-means clustering, in: Proceedings of the 19th International World Wide Web Conference, ACM. pp. 1177–1178.

Vardi, Y., Zhang, C.H., 2001. A modified weiszfeld algorithm for the fermat-weber location problem. Mathematical Programming 90, 559–566.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., et al., 2008. Top 10 algorithms in data mining. Knowledge and information systems 14, 1–37.