International Journal of Applied Optimization Studies
IJAOS
İSÜ İSTİNYE ÜNİVERSİTESİ İSTANBUL

An Enhanced Metaheuristic Approach for Optimizing Graph-Based K-Means Clustering through Collective Intelligence

Cheng-Jui Tseng[1,*], Tzu-Chia Chen[2]

[1] Graduate School of Technology in Finance, CTBC Business School, Taiwan
[2] Department of Artificial Intelligence, Tamkang University, New Taipei City, 251301, Taiwan
*E-mail (corresponding author): EricTseng@ctbc.edu.tw

ABSTRACT

Data clustering is a fundamental aspect of data mining, serving as a crucial tool for data description and organization. The objective of clustering is to group data points into distinct or overlapping clusters, where data within each cluster exhibit high similarity while being dissimilar to data in other clusters. The emerging clustering technique, known as clustering based on graph K nearest reciprocal neighborhood, has demonstrated notable advancements by addressing issues associated with conventional clustering algorithms, such as the reliance on predefined cluster counts and initial center selection. However, a notable challenge remains in determining the optimal neighborhood parameter 'K' that governs the selection of nearest neighbors. In this paper, we introduce a solution leveraging the power of collective intelligence and metaheuristic algorithms to effectively address the sensitivity issue surrounding the 'K' parameter. Specifically, we employ the Jaya algorithm, distinguished by its minimal control parameters, which not only facilitates population convergence towards optimal solutions but also efficiently distances itself from suboptimal ones. This unique characteristic contributes to a heightened convergence rate when compared to similar metaheuristic methods. We conduct rigorous experiments utilizing datasets from the UCI repository, demonstrating the superiority of our proposed approach over existing methods in terms of clustering accuracy. Our research offers a robust and innovative solution for refining graph-based K-means clustering, further enhancing its applicability and effectiveness in real-world data analysis scenarios.

Paper type: *Research Article*

Received 2022-04-09
Revised 2022-06-21
Accepted 2022-06-25

Keywords:
Graph-based clustering;
Meta-heuristic algorithms;
Collective intelligence;
Jaya algorithm.

1. Introduction

Clustering, an unsupervised method, involves the grouping of data points based on their degree of similarity, as expounded by Kanungo et al. (2002) and Xu and Wunsch (2005). Its primary objectives are twofold: to ensure that data within each cluster exhibit maximum similarity while data across different clusters display maximum dissimilarity. Consequently, clustering hinges on identifying an

appropriate similarity criterion for the data. This methodology represents a harmonious integration of mathematics and computer science, with applications spanning diverse fields such as medicine, biology, seismology, management, and economics.

Clustering algorithms can primarily be categorized into hierarchical and partitional classes (Suo et al., 2019). Partitional clustering focuses on successive cluster formation through iterative processes. Each iteration seeks to enhance clustering quality based on predefined criteria and halts upon meeting predetermined termination conditions. Conversely, the hierarchical approach addresses the issue of overlapping clusters. In this method, clusters are either merged or separated at each stage, guided by predetermined criteria. Partitional algorithms generally exhibit lower time complexity, resulting in faster processing and suitability for dynamic data. Nevertheless, these methods are typically more sensitive to the initial cluster configuration and necessitate additional information and parameters for effective cluster initiation.

The use of clustering methods, despite their numerous advantages in data mining and data description across various clustering groups, presents notable challenges. For instance, most clustering methods necessitate input parameters, such as the number of clusters, neighborhood size, and initial cluster centers, which users must specify. This parameter specification directly impacts clustering efficiency, as established by Jafarzadegan et al. (2019). Moreover, the quest for the optimal clustering solution for a given dataset is inherently complex.

Consequently, substantial research efforts have been devoted to tackling these challenges. One viable approach to automatically determine optimal parameters and achieve the best clustering results involves integrating clustering methods with metaheuristic techniques. For instance, Majdi and Beiki (2019) employed genetic algorithms (GA) and particle swarm optimization (PSO) to enhance the efficiency of Fuzzy c-means clustering. Another study optimized the performance of the widely used k-means clustering method by combining it with PSO, as demonstrated by Prabha and Visalakshi (2014). Aljarah et al. (2020) applied gray wolf optimization (GWO) to optimize clustering outcomes. In yet another investigation, Mann and Singh (2019) introduced a clustering method optimized by the bee evolutionary genetic algorithm to cluster nodes within a wireless system.

The consideration of proximity to the nearest data points stands out as one of the most effective techniques within data clustering. In this approach, rather than relying solely on conventional similarity criteria, it leverages common neighborhoods of data points to establish similarity (Ertöz et al., 2003). A recent advancement in this domain is clustering based on the graph k-reciprocal nearest neighborhood, as introduced by Qin et al. in 2018. This method not only resolves several challenges posed by previous techniques but also excels in identifying outliers. Despite its numerous advantages, the crucial task of selecting the appropriate parameter values for determining neighbors remains an open challenge. Furthermore, there is no assurance of discovering the optimal clusters.

In this study, we employ the Jaya algorithm to address this issue. The Jaya algorithm has emerged as a highly successful metaheuristic in recent years. Distinguished from many evolutionary methods, it boasts minimal control parameters. Moreover, it steers the population towards optimal solutions and distances itself from inferior ones. This innovation effectively mitigates the often-encountered drawback of sluggishness in metaheuristic algorithms, a significant impediment to their deployment in high-speed data processing. Additionally, it enhances the likelihood of identifying more suitable solutions. The ensuing section elucidates the theoretical underpinnings of both the Jaya algorithm and the concept of k-reciprocal nearest neighborhoods.

2. Literature Review

2.1. Jaya Algorithm

The Jaya algorithm has been proposed for optimization and solving search problems. Similar to other metaheuristics, the technique starts from a random initial population in an iterative manner to search the problem space and find the optimal solution. Imagine that F(x) is the objective function of the problem, and the Jaya algorithm tends to find its minimum (or maximum). In each i-th iteration, there are m decision-making variables (j=1,2,…,m) and n number of proposed solutions. Imagine that in the i-th iteration, the best value obtained from all candidate solutions for the F(x) objective function is equal to the value of $X_{j,best,I}$, whereas the worst value of the F(x) objective function is equal to $X_{j,worst,I}$. If $X_{j,k,I}$ expresses the value of the j-th variable for the j-th candidate in the i-th iteration, the decision-making variable of $X_{j,k,I}$ must be converted into Eq. (1) and updated in order to find the optimal solution:

$$X_{j,k,i}^{new} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - \left|X_{X_{j,k,i}}\right|\right) - r_{X_{2,k,i}}\left(X_{j,worst,i} - \left|X_{j,k,i}\right|\right), \quad (1)$$

where $X_{j,best,I}$ is recognized as the best solution while $X_{j,best,I}$ is the worst solution. In addition, $X_{j,k,i}^{new}$ is the updated value of $X_{j,k,I}$ and $r_{1,j,i}$ and $r_{2,j,I}$ are random numbers in the range of (0 and 1). Moreover, $\left(X_{j,best,i} - \left|X_{X_{j,k,i}}\right|\right)$ indicates that the solution is getting closer to the best solution, whereas $\left(X_{j,worst,i} - \left|X_{j,k,i}\right|\right)$ shows that the current solution is distancing itself from the worst solution. According to the formulation of the problem, it seems that the method has no regulatory parameters, contrary to other evolutionary techniques, which solves the problem of possible changes in different executions and the unpredictability (randomness) of the method.

2.2. Clustering Based on K-reciprocal Nearest Neighborhood

To provide a clearer understanding of clustering based on k-reciprocal nearest neighbors, we present an illustrative example. In this sample dataset, several key points warrant attention. When we set the value of the k-means neighborhood parameter to two, we select two nearest neighbors for each data point. Subsequently, we distinguish these neighbors by assigning them either a blue or red square, with the red square indicating an outlier.

The connections resulting from assessing the two k-nearest neighbors are visually depicted for each data point. These edges, with bidirectional links between pairs of data points, signify a specific type of relationship. To analyze subclusters within the dataset, we identify data points that contain edges with reciprocal links and are interconnected by multiple edges, akin to what is shown in Figures 1 and 2. Importantly, during the extraction of these subclusters, the directionality of the edges is eliminated.

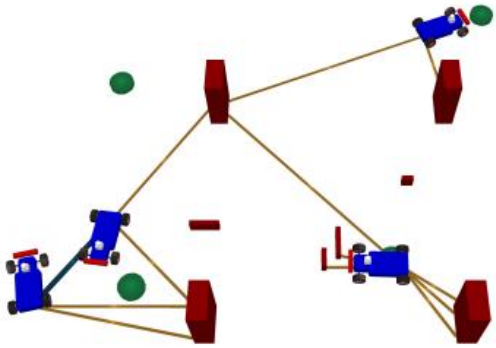


Fig. 1. Finding the nearest neighborhood of each sample data (k neighborhood size=2).

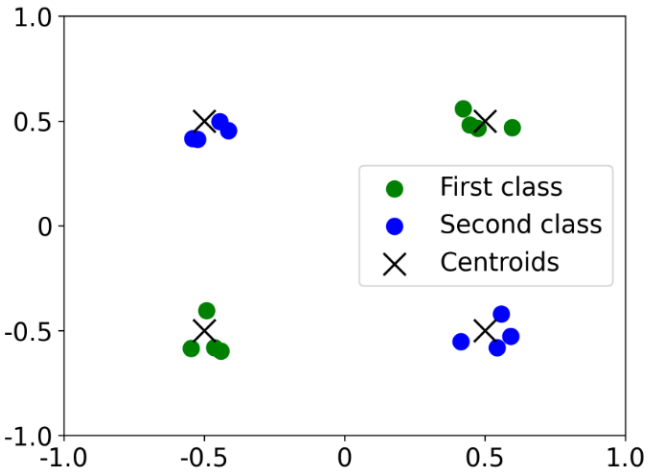


Fig. 2. Clusters found by using reciprocal links (k neighborhood size=2).

3. Proposed Method

This section proposes the optimization of clustering based on a k-reciprocal nearest neighborhood by using the Jaya algorithm.

3.1. Representation of the Problem

The position of each solution in the Jaya algorithm is considered in the form of a string with an n length, where n is indicative of the number of sample data in the database. In this regard, each bit of the intended bit string (the element in the formed bit string) is randomly initialized with the sample data number in the database.

3.2. Fitness Function

The accuracy rate obtained from clustering based on graph k-reciprocal nearest neighborhood was applied as the objective function.

3.3. Optimization of Clustering Based on Graph K-reciprocal Nearest Neighborhood

In the context of optimizing clustering based on the k-reciprocal nearest neighbor using the Jaya algorithm, the initial phase involves identifying solutions within the problem space (Rao, 2016; Goli et al., 2020, 2021). Subsequently, we must ascertain the position of each solution, which corresponds to the sample data. To clarify, this implies that the position of each solution is analogous to a database, with data samples being randomly distributed in binary format. Therefore, the position of each solution equates to that of a database.

The primary positioning of each solution is determined following the establishment of particle positions, and the initial solutions are evaluated by referencing the fitness function. Up to this juncture, position initialization is conducted, and the fitness of each solution is computed. Subsequently, in each iteration, each solution undergoes updates. In this context, as particles traverse the algorithm's iteration space, solutions are iteratively updated until a specified stopping condition is met.

In each iteration, the update of each solution entails getting closer to global optimization while distancing itself from the least optimal outcome. Post-update, the fitness function is once again invoked to assess the fitness of each solution, which informs the decision-making process for selecting a more efficient solution. It's noteworthy that the fitness function in question represents the accuracy rate of the k-reciprocal nearest neighbor. For a clear representation of the proposed method's stages, please refer to the pseudocode presented in Figure 3.

Compute Objective function f(x), x =(x1,...,xd)
Initialize the Jaya population xi (i = 1,2,...,n)
While (t <Max number of iterations)
Generate and updating locations/ solutions [eq. (1)]
For each particle
If (New locations, solution (xi(t+1)) > Current locations /solution xi(t))
Select a New solution
else
Select a Current solution
End if
End for
Rank the bats and find the current best x*
End while
Results Best Solution

Fig. 3. Pseudocode of the proposed method.

4. Experiments and Results

In this research study, we evaluate the effectiveness of the proposed methodology using a set of seven datasets. Four of these datasets are sourced from the standard reservoir of the UCI dataset (reference 18), while the remaining three are generated datasets as described in references 4 and 16. Detailed information about these datasets is provided in Table 1.

For the implementation of the Jaya algorithm, we consider regulatory parameters, specifically a population size of 40 and a maximum iteration count of 100. We compare our proposed method with clustering techniques based on k-reciprocal nearest neighbors and k-reciprocal nearest neighbors optimized by Particle Swarm Optimization (PSO). PSO is renowned as one of the most widely used metaheuristic algorithms. These methods are evaluated based on two criteria: the Rand Index (RI) and the normalized reciprocal data.

Table 1. Characteristics of datasets used for comparison.

Dataset	Number of Characteristics	Number of Samples	Number of Classes
Iris	4	150	3
Wine	13	178	3
Glass	9	214	5
Data1	2	1000	4
Data2	2	1000	7
Data3	2	1000	4

The RI serves as a metric for assessing the similarity between two labeling methods. Our results, which are summarized in Table 2, showcase the performance of our proposed method across all datasets, providing an accurate evaluation of its efficiency.

Table 2. Results related to accuracy assessed by the RI.

Data	KNN-based [16]	KNN-based+PSO	The proposed method
Iris	0.98	0.94	1
Wine	0.42	0.67	0.77
Glass	0.80	0.75	0.84
Data1	0.87	0.91	0.97
Data2	0.95	0.91	1
Data3	0.95	0.88	0.97

4.1. Normalized Reciprocal Data

The criterion of normalized reciprocal data is calculated using Eq. (2):

$$NMI = \frac{2\sum_i \sum_j n_{ij} \log \frac{n_{ij} n}{b_i d_j}}{-\sum_i b_i \log \frac{b_i}{n} - \sum_j d_j \log \frac{d_j}{n}} \quad (2)$$

Table 3 and Figure 4 present the results pertaining to the accuracy of the normalized reciprocal data. A higher value of the normalized reciprocal criterion indicates superior performance, with clustering results closely aligning with the actual class labels present in the database. Our findings unequivocally demonstrate the superior performance of the proposed method when compared to alternative techniques.

Table 3. Results of accuracy assessed by the normalized reciprocal data.

Data	KNN-based [16]	KNN-based+PSO	The proposed method
Iris	0.99	0.92	1
Wine	0.45	0.69	0.76
Glass	0.79	0.72	0.82
Data1	0.89	0.90	0.98
Data2	0.96	0.90	1
Data3	0.92	0.89	0.98

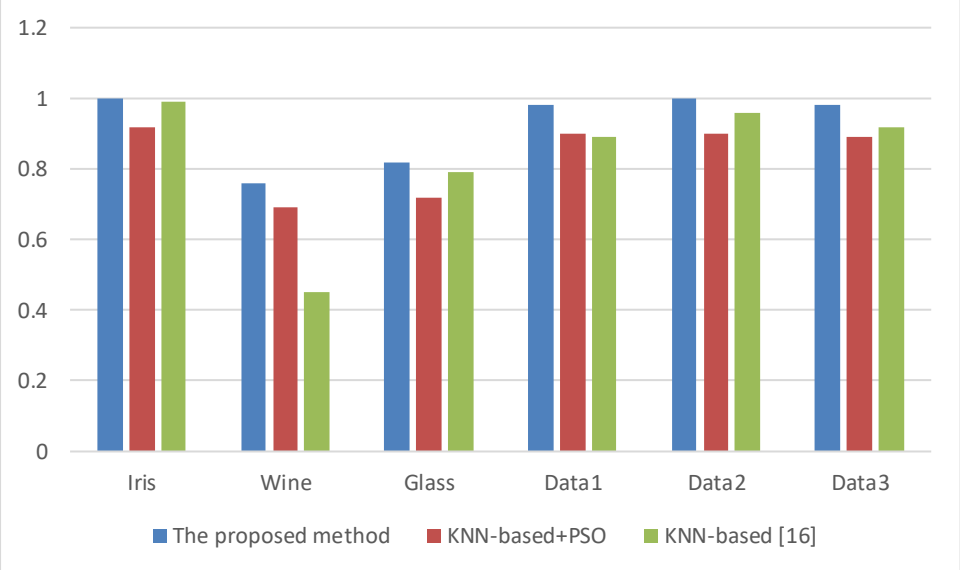


Fig. 4. A comparison of results with different data.

5. Discussion and Conclusion

In this study, In conclusion, the utilization of the Jaya algorithm for clustering purposes in this research has demonstrated its effectiveness and superiority over traditional methods, particularly when confronted with the complexities of graph k-reciprocal nearest neighborhood-based clustering. The strengths of the Jaya algorithm, such as its ability to steer away from suboptimal solutions while

simultaneously converging towards the best possible solution, have proven instrumental in enhancing the quality and precision of clustering outcomes.

Our findings underscore the potential of the Jaya algorithm to elevate the quality of clustering results significantly. By incorporating mechanisms that encourage diversity and simultaneously drive convergence towards both local and global optima, this approach harnesses the strengths of both exploration and exploitation, leading to the discovery of more robust and accurate clusters. Moreover, it accomplishes this with notable efficiency, delivering optimal solutions in a shorter timeframe compared to alternative metaheuristic techniques.

In light of these promising results, it is clear that clustering based on the Jaya algorithm offers a valuable and versatile tool for addressing complex clustering problems. Further research and exploration of this approach may uncover additional applications and refinements, ultimately contributing to advancements in data analysis, pattern recognition, and decision-making in various fields. As we continue to grapple with ever-expanding datasets and intricate clustering challenges, the Jaya algorithm emerges as a promising avenue for fostering innovation and achieving optimal clustering solutions.

References

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., & Mirjalili, S. (2020). Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. Knowledge and Information Systems, 62, 507-539.

Ertöz, L., Steinbach, M., & Kumar, V. (2003, May). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In Proceedings of the 2003 SIAM international conference on data mining (pp. 47-58). Society for Industrial and Applied Mathematics.

Goli, A., & Malmir, B. (2020). A covering tour approach for disaster relief locating and routing with fuzzy demand. International Journal of Intelligent Transportation Systems Research, 18(1), 140-152.

Goli, A., Khademi-Zare, H., Tavakkoli-Moghaddam, R., Sadeghieh, A., Sasanian, M., & Malekalipour Kordestanizadeh, R. (2021). An integrated approach based on artificial intelligence and novel meta-heuristic algorithms to predict demand for dairy products: a case study. Network: Computation in Neural Systems, 32(1), 1-35.

Jafarzadegan, M., Safi-Esfahani, F., & Beheshti, Z. (2019). Combining hierarchical clustering approaches using the PCA method. Expert Systems with Applications, 137, 1-10.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 881-892.

Majdi, A., & Beiki, M. (2019). Applying evolutionary optimization algorithms for improving fuzzy C-mean clustering performance to predict the deformation modulus of rock mass. International Journal of Rock Mechanics and Mining Sciences, 113, 172-182.

Mann, P. S., & Singh, S. (2019). Improved artificial bee colony metaheuristic for energy-efficient clustering in wireless sensor networks. Artificial Intelligence Review, 51, 329-354.

Prabha, K. A., & Visalakshi, N. K. (2014, March). Improved particle swarm optimization based k-means clustering. In 2014 International Conference on Intelligent Computing Applications (pp. 59-63). IEEE.

Qin, Y., Yu, Z. L., Wang, C. D., Gu, Z., & Li, Y. (2018). A novel clustering method based on hybrid k-nearest-neighbor graph. Pattern Recognition, 74, 1-14.

Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, 7(1), 19-34.

Suo, M., Zhu, B., Zhou, D., An, R., & Li, S. (2019). Neighborhood grid clustering and its application in fault diagnosis of satellite power system. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 233(4), 1270-1283.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3), 645-678.