***111公式章 1 节 1Type: Article***

An Improved FCM Clustering Method Based on Multi-Strategy Tuna Swarm Optimization

Zeming Yang1, Guinan Guo2, \*

1 School of Computer Science, China University of Geosciences, Wuhan, 430000, China

2Aerospace Information Research Institute, Chinese Academy of Sciences, Guangdong, 510000, China

\*Corresponding Author: Guinan Guo. Email: guoguinan7@gmail.com

Received: 06 December 2022; Accepted: 15 May 2023

**Abstract:** To overcome the shortcoming of the Fuzzy C-means algorithm (FCM) that it is easy to fall into local optimum due to the dependence of sub-spatial clustering on initialization, a Multi-Strategy Tuna Swarm Optimization-Fuzzy C-means (MSTSO-FCM) algorithm is proposed. Firstly, a chaotic local search strategy and an offset distribution estimation strategy proposed algorithm are proposed to improve the performance, enhance the population diversity of the Tuna Swarm Optimization (TSO) algorithm, and avoid falling into local optimum. Secondly, the search and development characteristics of the MSTSO algorithm are introduced into the fuzzy matrix of Fuzzy C-means (FCM), which overcomes the defects of poor global searchability and sensitive initialization. Not only has the searchability of the Multi-Strategy Tuna Swarm Optimization algorithm been employed, but the fuzzy mathematical ideas of FCM have been retained, to improve the clustering accuracy, stability, and accuracy of the FCM algorithm. Finally, two sets of artificial datasets and multiple sets of the University of California Irvine (UCI) datasets are used to do the testing, and four indicators are introduced for evaluation. The results show that the MSTSO-FCM algorithm has better convergence speed than Tuna Swarm Optimization Fuzzy C-means (TSO-FCM) algorithm, and its accuracies in the heart, liver, and iris datasets are 89.46%, 63.58%, 98.67%, respectively, which have an outstanding improvement.

**Keywords:** fuzzy C-means; tuna swarm optimization; local search; data cluster

**1 Introduction**

As technology improves by leaps and bounds, the emergence of data expansion has had a certain impact on all walks of life. Countries around the world have gradually paid attention to the analysis of data and internal knowledge. A commonly used data analysis method, data mining is used to transform the original data into useful data or information through a certain identification method [1]. The existing supervised learning has a very strong dependence on data labels. If the data labels are accurate, the supervised algorithm can learn well, but when the label is wrong, the supervised algorithm is difficult to analyze, and the internal knowledge mined will produce certain deviations. The ways of labeling massive data and mining internal connections are the unique function of clustering algorithms [2].

The current clustering algorithms are divided into five categories: density-based methods, grid-based methods, hierarchical-based methods, model-based methods, and division-based methods [3]. The fuzzy C-means clustering method is a division-based clustering method, which is widely used in the segmentation and classification of brain tumors [4], the evaluation of power grid reliability performance [5], and the recognition of the whitewashing behavior of college accounting statements [6]. However, because the FCM algorithm belongs to the hard subspace clustering algorithm, its computational complexity is high. Besides, it is not sensitive to noise, and the processing effect of high data is poor. It also does not overcome the dependence of the sub-spatial clustering method on initialization, so it is very easy to fall into the local optimum. To overcome the above shortcomings, an improved particle swarm optimization algorithm was proposed by using the foraging behavior of birds, which is applied to the clustering algorithm and has achieved certain improvements [7]. The hybrid Fuzzy c‑means clustering and gray wolf optimization (GWO) for image segmentation were proposed to overcome the shortcomings of Fuzzy c‑means clustering [8]. A new collaborative multi-population differential evolution method with elite strategies was proposed to identify approximate optimal initial clustering prototypes, determine the optimal number of clusters in the data, and optimize the initial structure of FCM [9]. A hybrid optimization algorithm for simulated annealing and ant colony was put forward to improve the clustering method and enhance the accuracy of data clustering [10]. The particle swarm algorithm was improved by using the levy flight strategy and applied to the clustering algorithm, which improves its initialization process, greatly reduces the computational complexity, and improves the clustering accuracy [11]. The particle swarm optimization was combined with FCM, and the FCM fitness was optimized for each iteration [12]. Three improved particle swarm optimization algorithms were proposed and used differently according to data characteristics, which improves the clustering performance and weakens data sensitivity [13]. An improved artificial bee colony optimization algorithm was proposed to optimize the clustering problem and significantly improve the efficiency of data processing [14].

From the above research, the heuristic optimization algorithm plays a positive role in the clustering problem and overcomes a considerable part of the defects of the clustering algorithm itself according to the different improvement methods. Tuna Swarm Optimization (TSO) is a novel algorithm with excellent searchability and excellent performance in various problems. An improved TSO is used to segment images of forest canopies [15]. An improved nonlinear tuna swarm optimization algorithm based on a Circle Chaos map and levy flight operator (CLTSO) is proposed to optimize a BP neural network [16]. A hybrid model based on the long short-term memory (LSTM) and the TSO algorithm is established to predict wind speed [17]. TSO is blended with improved adaptive differential evolution with an optional external archive to form a new heuristic algorithm, which has been well demonstrated in the problem of photovoltaic parameter identification [18]. TSO is used to have a higher efficiency of the global maximum power point tracking, and have a good performance in the Hardware-in-loop and Rapid Control Prototype platform [19]. A Chaotic Tuna Swarm Optimizer (CTSO) is proposed to find the optimal parameters of the Three-Diode Model, and it is hybridized with the Newton-Raphson method to improve the ability of convergence [20]. An improved particle swarm optimization algorithm can adaptively adjust the relevant parameters of the fuzzy decision tree, effectively improving the recognition accuracy in lithology recognition [21]. A multi-objective algorithm using a grid-based method is proposed to do the privacy-preserving data mining and achieves excellent performance [22]. A multi-objective neural evolutionary algorithm based on decomposition and dominance (MONEADD) was proposed to evolve neural networks [23]. [24] proposed a multi-objective optimization algorithm, which is used to optimize the cluster center of FCM to shorten the calculation time. An improved Fruit Fly Optimization (IFFO) algorithm is proposed to minimize makespan and costs of scheduling multiple workflows in the cloud computing environment [25]. An advanced encryption standard (AES) algorithm was used to deal with workflow scheduling [26]. [27] proposed a Hybrid Enhanced Glowworm Swarm Optimization Traffic-Aware Vehicular Networks, and technical delay is significantly reduced.

FCM dependence on initialization was not overcome in the above research. It is very easy to fall into the local optimum. In summary, to overcome the defects, an FCM based on multi-strategy tuna swarm optimization (MSTSO-FCM) is proposed. The contribution of this paper is that a multi-strategy tuna swarm optimization (MSTSO) is proposed. A chaotic local search strategy is introduced to improve the development ability of the algorithm, and the dominant population is fully utilized by an offset distribution estimation strategy to enhance the performance of the algorithm. Another contribution is that the search and development characteristics of the MSTSO algorithm are introduced into the fuzzy matrix of FCM, which overcomes the defects of poor global search ability and sensitive initialization. It not only uses the searchability of MSTSO but also retains the fuzzy mathematical ideas of FCM, to improve the clustering, stability, and accuracy of the FCM algorithm.

The rest of this paper is organized as follows. The second section introduces the existing related algorithms which are the tuna swarm optimization algorithm and the Fuzzy C-means clustering algorithm. Section 3 proposes the Multi-Strategy Tuna Swarm Optimization algorithm, and MSTSO-FCM is proposed based on it. The fourth section makes the simulation experiment and comparative analysis. Finally, the conclusion is described in section 5.

**2 Existing Related Algorithms**

***2.1 Tuna Swarm Optimization***

As a top marine predator, tuna is a social animal. They choose the corresponding predation strategy according to the object they forage. The first strategy is spiral foraging, that is, when tuna feed, they swim in a spiral to take their prey into shallow water, attack, and catch it. The second strategy is parabolic foraging, in which each tuna follows the previous individual to form a parabolic shape to surround its prey. Tuna successfully forages through the above two methods. TSO is modeled based on these natural foraging behaviors, and the algorithm follows the basic rules as follows.

(1) Population initialization: TSO starts the optimization process by randomly generating the initial swarm evenly to update the position.

 23\\* MERGEFORMAT ()

(2-1) Spiral foraging. Schools of tuna chase their prey by forming a tight spiral, and in addition to chasing their prey, they also exchange information with each other. Each tuna follows the previous one, so information can be shared between neighboring tuna. Based on the above principles, the mathematical formulas of the spiral foraging strategy are as follows:

 45\\* MERGEFORMAT ()

 67\\* MERGEFORMAT ()

 89\\* MERGEFORMAT ()

 1011\\* MERGEFORMAT ()

 1213\\* MERGEFORMAT ()

where  is the position of the th individual in the t+1 generation.  is the current optimal individual.  and  are the coefficient that controls the degree of follow-up of the individual towards the optimal individual and the previous individual on the chain.  and  represent the current number of iterations and the maximum number of iterations respectively.  is a random number evenly distributed from 0 to 1. When the optimal individual cannot find food, blindly following the optimal individual to forage is not conducive to group foraging. Therefore, a random coordinate in the search space is generated to serve as a reference point for the spiral search. It enables each individual to explore in a wider space and enables TSO to explore globally. The specific mathematical model is described as follows:

 1415\\* MERGEFORMAT ()

In particular, meta-heuristics typically perform extensive global exploration in the early stage, followed by a gradual transition to precise local development. Therefore, as the number of iterations increases, TSO changes the reference point for spiral foraging from random individuals to optimal individuals. In summary, the final mathematical model of the spiral foraging strategy is as follows:

 1617\\* MERGEFORMAT ()

(2-2) Parabolic foraging. In addition to feeding through spiral formation, tuna can also form parabolic cooperative feeding. The first method is that tuna feeds in a parabolic shape using food as a reference point. Another is that tuna can forage for food by looking around. And assume that the selection probability of both is 50%, and both methods are executed at the same time. Tuna hunt synergistically through these two foraging strategies and then find their prey.

 1819\\* MERGEFORMAT ()

 2021\\* MERGEFORMAT ()

(3) Termination condition. The algorithm continuously updates and calculates all individuals of tuna until the final condition is met, and then returns the optimal individual and the corresponding fitness value.

***2.2.1 Fuzzy C-means Clustering Algorithm***

The FCM algorithm [28] is a clustering algorithm based on division, which measures the clustering effect by dividing the similarity between objects. The similarity between the same cluster is high, but that of different clusters is low. So, the fuzzy mean is used for flexible division, which is mainly divided into the following steps:

Step 1: Initialize the membership matrix. The FCM algorithm blurs the concept of 0, 1 binary values to between 0~1, or the degree of membership which measures the relationship between each data point and the cluster center.

 2223\\* MERGEFORMAT ()

Where  represents the cluster category.  represents the number of objects in the dataset.  represents the membership degree of different categories of each set of data, and its sum is 1. The larger the relative membership degree, the greater the probability of the category.

Step 2: Iterative termination judgment. The maximum number of iterations, , is set to determine whether the current number of iterations has reached the upper limit of iterations. Output the clustering result if it exceeds , or update the cluster center;

Step 3: Calculate the cluster center. After the FCM cluster center  is established, the membership degree sum of all points to the cluster center is calculated first, and then the cluster center is updated by multiplying the proportion of the membership degree of each category by the original data. The calculation formula is below:

 2425\\* MERGEFORMAT ()

where  represents the jth cluster center.  is the current data point.

Step 4: Update the membership matrix. Based on the cluster center and data points, the membership degree is updated through the membership calculation formula:

 2627\\* MERGEFORMAT ()

As can be seen from the above equation,  represents the distance from the data point  to the cluster center .  represents the distance from the data point  to all cluster centers. The closer the data points to , the greater the membership value.

Step5: Judgment of iteration termination conditions. When the current number of iterations is less than the maximum number of iterations, the iteration termination conditions are as follows:

 2829\\* MERGEFORMAT ()

Where  indicates the degree of membership under the current number of iterations .  indicates the degree of membership before the update. When the difference between the two is less than the set threshold , it means that a better solution has been solved, hence the ending of iterations and the return of the fuzzy clustering result. If the termination condition is not met, return to step 3.

**3 MSTSO-FCM**

In this section, two improvement strategies, which are the Chaotic local search strategy and the Offset distribution estimation strategy, are introduced in detail, and the design scheme of MSTSO is described. On this basis, the MSTSO-FCM algorithm is proposed.

***3.1 Chaotic Local Search Strategy***

The chaotic local search strategy finds a better solution by searching the vicinity of each solution. Therefore, this strategy can effectively improve the exploitation ability of algorithms. Besides, chaos mapping has the characteristics of randomness and ergodicity, so the use of chaos mapping could further improve the effectiveness of the local search strategy. In MSTSO, the chaotic local search strategy is applied only to the dominant population of the population. The formula of the chaotic local search strategy is as follows:

 3031\\* MERGEFORMAT ()

where  is a new solution generated using the chaotic local search strategy.  and  are two different individuals randomly selected from the dominant population.  is the chaos value generated by the chaos mapping. In MSTSO, a tent chaos mapping is used to generate . Tent chaos mapping is a classical one, and compared with logistic mapping, it has better traversal uniformity, which can improve the optimization speed of the algorithm, and at the same time produce a more evenly distributed initial value between [0,1]. The tent mapping expression is as follows:

 3233\\* MERGEFORMAT ()

***3.2 Offset Distribution Estimation Strategy***

The distribution estimation strategy represents relationships between individuals through probabilistic models. This strategy uses the current dominant population to calculate the probability distribution model, generates new offspring populations based on the sampling of the probability distribution model, and finally obtains the optimal solution through continuous iteration. In this paper, the top half of the individuals with better performance are sampled and the strategy is used for the poor individuals. The mathematical model of the strategy is described below:

 3435\\* MERGEFORMAT ()

 3637\\* MERGEFORMAT ()

 3839\\* MERGEFORMAT ()

 4041\\* MERGEFORMAT ()

 4243\\* MERGEFORMAT ()

Where  represents the weighted mean of the dominant population.  is the population size.  represents the weight coefficient in the descending order of fitness value in the dominant population.  is the weighted covariance matrix of the dominant population.

***3.3 MSTSO Design***

MSTSO adds the proposed Chaotic local search strategy and Offset distribution estimation strategy to the original TSO. The specific algorithm design is shown in **Algorithm 1**.

|  |  |
| --- | --- |
| **Algorithm 1**: The procedure of MSTSO | |
| **Input：** | *f*, [***x*min**, ***x*max**], *FEs*max. |
| **Output：** | ***x***best. |
|  | // Initialization. |
| **1.** | Initialize population ***X*** by using Eq.Error: Reference source not found. |
| **2.** | Assign *a* and *z* |
| **3.** | Evaluate ***X*** to determine their fitness value by using *f*(***X***). |
| **4.** | Fes= Fes+NP |
|  | ***// Main loop.*** |
| **5.** | **While** FEs< FEsmax **do** |
| **6.** | Update *a1*, *a2*, *p*; |
| **7.** | **If** *rand<z* |
| **8.** | Update ***X*** through Eq.(1); |
| **9.** | **else** |
| **10.** | **If** *rand*<0.5 |
| **11.** | Update ***X*** through Eq.(17); |
| **12.** | **else** |
| **13.** | **If** *t/tmax*<*rand* |
| **14.** | Update ***X*** through Eq.(7); |
| **15.** | **else** |
| **16.** | Update ***X*** through Eq.(2); |
| **17.** | **end** |
| **18.** | **end** |
| **19.** | **end** |
| **20.** | Update ***X*** through Eq.(15); |
| **21.** | Evaluate ***X*** to determine their fitness value by using *f*(***X***); |
| **22.** | Updata ***xbest***; |
| **23.** | **end** |

***3.4 FCM Cluster Based On MSTSO Optimization***

Due to the fuzzy theory, FCM has more flexible clustering ideas than traditional hard subspace clustering (mean clustering) [29]. The iterative process of FCM clustering can be understood as a central continuous moving process, which leads to a large improvement in global search ability. FCM does not overcome the dependence of the sub-spatial clustering method on initialization, so it is very easy to fall into the local optimum. It is difficult to effectively cluster when dealing with high-dimensional spatial data. Considering the above problems using the MSTSO algorithm to improve FCM, the search and development characteristics of the MSTSO algorithm are introduced into the fuzzy matrix of FCM to overcome the shortcomings of poor global searchability and sensitive initialization. It not only uses the searchability of MSSO but also retains the fuzzy mathematical ideas of FCM, and improves the clustering, stability, and accuracy of the FCM algorithm.

To obtain a better clustering center for FCM, the MSTSO algorithm is used to optimize the membership matrix and replace the original random initialization and update method. It is necessary to know the number of categories of clustered data.

The process of MSO optimizing the FCM cluster is as follows, the flowchart is shown in Fig. 1, pseudo-code is shown in **Algorithm 2.**

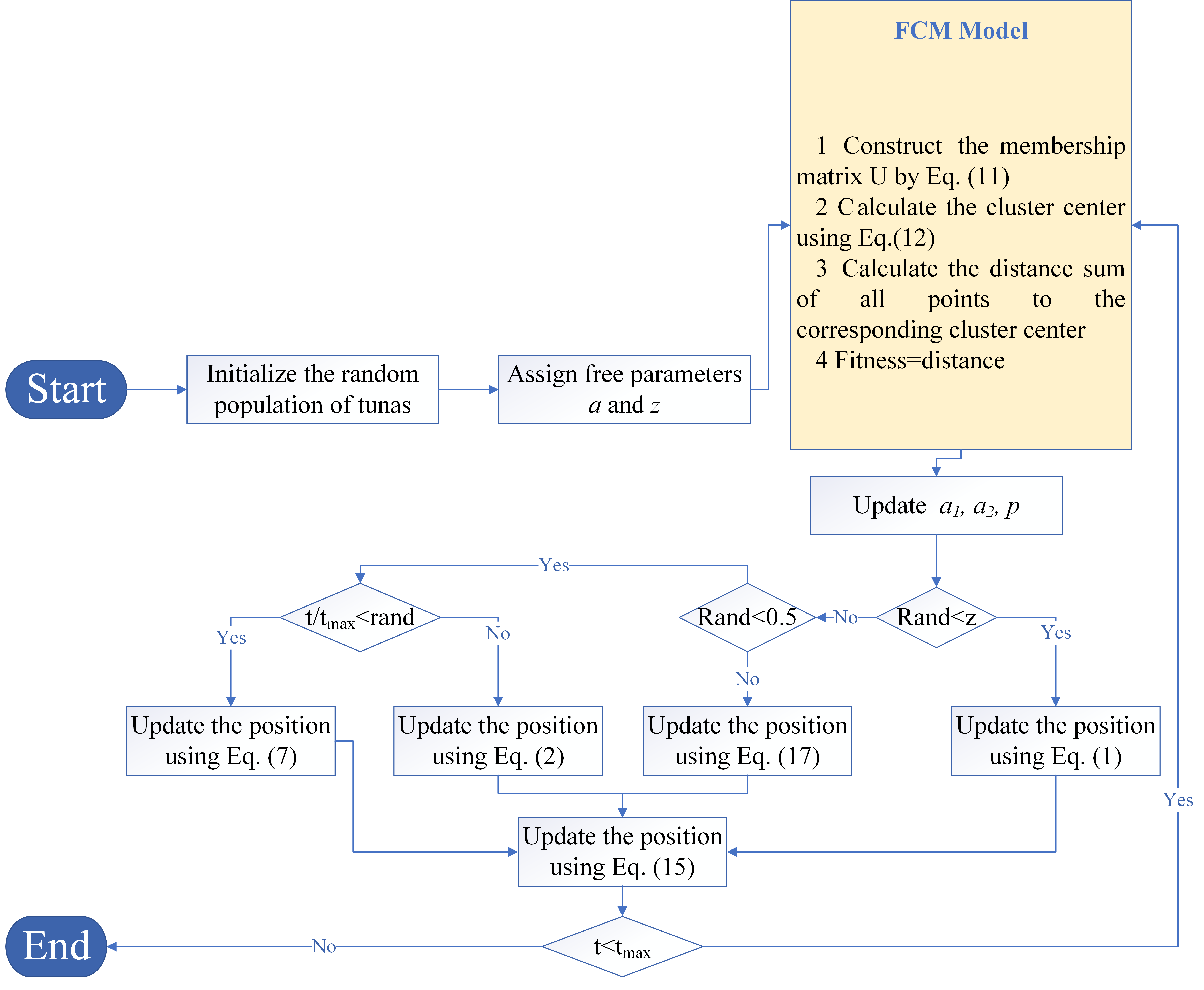


Figure 1. MSTSO optimizes the FCM cluster

**Step 1:** Use Eq. (1) to randomly initialize the population and assign the parameters *a* and *z*, where  means the population dimension, with each dimension including all the elements in the membership matrix , as shown in Fig.2;

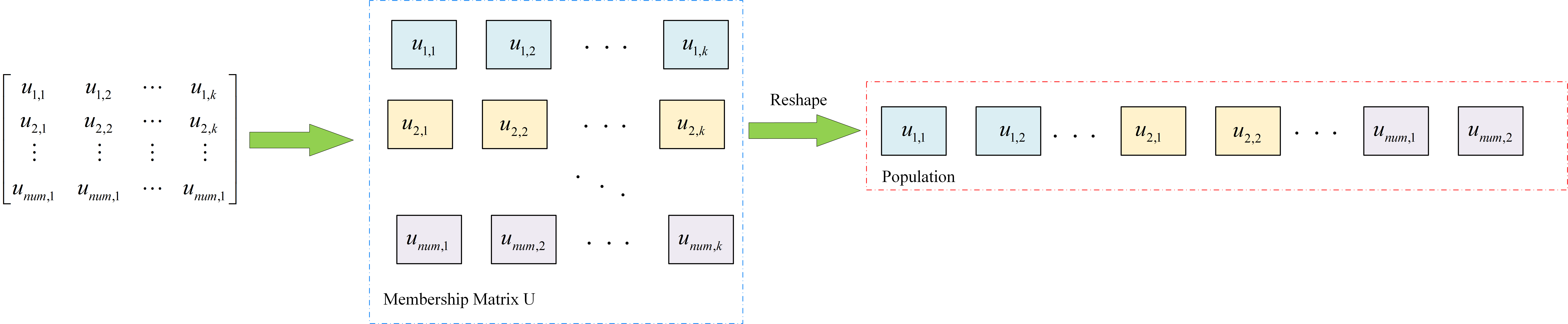


Figure 2. Membership Matrix U reshape to the population in MSTSO

**Step 2:** Introduce the FCM module to calculate the fitness value, Eq. (18) is used to reconstruct the membership matrix into the data, in which the dimension of each population and the number of parameters of the membership matrix in the MSTSO algorithm are the same. and Eq. (19) is used to calculate the cluster center point, and finally calculate the sum of the distances from all arrays to the cluster center, as shown in the following

 4445\\* MERGEFORMAT ()

**Step 3:** Update the population position and calculate the fitness value. Update . When random number , use Eq. (1) to initially update the population position. When , make a finer division. Use Eq. (13) to initially update the position, and vice versa. When , use Eq. (7) to conduct preliminary update. When , use Eq. (2) to perform a preliminary update and finally use Eq. (11) to update the position information again that enters the FCM model before calculating the fitness value;

**Step 4:** Determine whether the iteration is terminated. When the iteration is terminated, output the membership matrix U corresponding to the optimal fitness value at this time. As the optimal FCM model is constructed, if the iteration termination condition is not met, return to **Step 2**.

|  |  |
| --- | --- |
| **Algorithm 2**: The procedure of MSTSO-FCM | |
| **Input：** | *fFCM*, [***X*min**, X**max**], *FEs*max. |
| **Output：** | membership matrix U |
| **1.** | Reshape the membership matrix U to population ***X***; |
| **2.** | Initialize population ***X*** by using Eq.Error: Reference source not found; |
| **3.** | Assign *a* and *z* |
| **5.** | Evaluate ***X*** to determine their fitness value by using *fFCM*(***X***) (Eq.22). |
| **6.** | Fes= Fes+NP |
| **7.** | Execute MSTSO's Main Loop; |
| **8.** | Reshape the ***x***best. to U,and output. |

**4 Simulation Experiment And Comparative Analysis**

***4.1 MSTSO Algorithm Performance Verification***

To illustrate the performance of the MSTSO algorithm proposed in this paper, the CEC2017 test set is selected for verification, including 1 unimodal function, 7 multimodal functions, 10 mixed functions, and 10 combined functions. All simulations in this paper are run on MATLAB R2016b software, and the experimental environment is an Inter(R) Core(TM) i7-8700 CPU, 16GB memory computer. In this paper, the butterfly optimization algorithm (BOA), reptile search algorithm (RSA), arithmetic optimization algorithm (AOA), tunicate swarm algorithm (tunicate swarm algorithm), and sparrow search algorithm (SSA) are selected as comparison algorithms to illustrate the performance of the improved algorithm MSTSO. To ensure the fairness of the experiment, the population of all algorithms is 500, the maximum number of iterations is 600, and the parameter settings of the rest of the algorithms are based on the parameter values of the original paper. Each algorithm runs independently 51 times, and the statistical average results are shown in Annex-1.

From Annex-1, we can see that MSTSO performs well on most test functions. Specifically, for the unimodal test function F1, although MSTSO fails to stably obtain the optimal solution, the MSTSO optimization accuracy is significantly better than other algorithms, reaching 9 orders of magnitude, illustrating that MSTSO has excellent local search ability. The multimodal functions F2-F8 are often used to test the global search ability of algorithms, which usually have multiple local minimums, so excellent algorithms can jump out of local minimums to obtain global optimal values. MSTSO performs best on all 7 of these functions, ranking behind RSA only on F8. For TSO, MSTSO performs better on all multimodal functions, indicating that MSTSO's improvement strategy significantly improves TSO's global search ability. F9-F18 and F19-F28 are mixed and combination functions respectively. These two types of functions have complex structures, able to better test the ability of algorithms to solve complex optimization problems. As can be seen from Annex-1, MSTSO only ranked second on F19. In the remaining mixed functions and combination functions, MSTSO is the best performer, achieving better results significantly better than TSO, which shows that the improvement strategy proposed in this paper has well enhanced the ability of algorithms to solve complex structural problems. It has the potential to better solve complex optimization problems in the real world. In summary, MSTSO can strike a balance between the development and exploration capabilities of the algorithm and has a good local optimal avoidance ability, which can effectively avoid the premature convergence of the algorithm.

***4.2 Cluster Dataset And Comparison Algorithm***

To verify the clustering ability of MSTSO-FCM, two sets of artificial large datasets and four sets of UCI datasets are selected [30], of which two sets of artificial datasets contain three categories, with each category containing 5000 sets of data, respectively, two-dimensional characteristic data and three-dimensional characteristic data, which conformed to Gaussian distribution, and the basic information of the data is shown in Table 1.

**Table 1：** Information about the dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset name | Number of data objects | Characteristic dimensions | Categorized categories |
| Artificial data set1 | 15000 | 2 | 3 |
| Artificial data set1 | 15000 | 3 | 3 |
| UCI-iris | 150 | 4 | 2 |
| UCI-liver | 345 | 6 | 2 |
| UCI-heart | 303 | 13 | 2 |
| UCI-pima | 768 | 8 | 2 |
| UCI-waveform | 5000 | 21 | 3 |

To evaluate the clustering performance of each algorithm, three clustering indicators, Accuracy (Ac), Silhouette (Sil), Davies-Bouldin (DB), and Area Under Curve (AUC) are introduced, and the specific information of each index is shown in Table 2.

**Table 2：** Cluster indicators

|  |  |  |
| --- | --- | --- |
| Indicators | Equations | Specific information |
| Ac |  | is the correct sample classification number, is the total number of samples. The larger the value of A, the better the clustering. |
| Sil |  | The larger the value , the better the clustering effect. |
| DB |  | The ratio of the sum of scatters within a cluster between cluster separations. |
| AUC |  | represents the number of positive and negative samples， represents the adding of the positive sample numbers， represents the *i-th* Sample. |

Four comparison algorithms are selected, namely the TSO-FCM algorithm, PSO-FCM algorithm [13], FCM algorithm, Gaussian mixed clustering (GMM) [31], and K-means clustering algorithm [32]. Table 3 shows the clustering algorithm parameters.

**Table 3：** Settings of Comparison algorithm settings

|  |  |  |
| --- | --- | --- |
| Number | Algorithm | Parameter settings |
| 1 | MSTSO-FCM | The index of membership matrix , Maximum iterations , Termination condition .. |
| 2 | TSO-FCM | ,, . |
| 3 | PSO-FCM | ,,,. |
| 4 | FCM | ,,. |
| 5 | GMM | Non-negative regularized number . |
| 6 | K-MEANS | The number of clusters varies with the dataset. |

***4.3 Evaluation Of Clustering Results***

To verify that MSTSO can achieve a better improvement effect on FCM clustering. In the above five datasets, the loss reduction curve of the traditional TSO algorithm is compared, and the comparison results are shown in Fig. 3, in which the convergence ability of MSTSO in the early stage is not as good as TSO, but its development performance is better. After TSO stagnates, MSO can still search for a better development posture. Generally, after iterating to 8 times, it will stagnate. From the decline curve of the UCI public data set in the figure, it can be intuitively seen that the late convergence ability of MSTSO is stronger than that of traditional TSO, and it can better explore the global situation and obtain smaller loss values.

To further verify the clustering effect, AC, Sil, DB, and AUC are used for evaluation, in which AC indicates the clustering accuracy, Sil and DB represent the internal indicators of the cluster, and AUC is to evaluate the classification ability of positive and negative cases of the cluster. It can be seen from Table 4 that MSTSO-FCM has achieved the first good score in the AC indicators of the six datasets. Besides, there is significant progress compared with TSO-FCM, PSO-FCM, and FCM. It has good clustering ability. MSTSO-FCM also has progressed in the Sil index. For artificial dataset 1, the Sil index of FCM is only 0.60, the Sil index of TSO-FCM is 0.74 compared with FCM, and the Sil value of PSO-FCM is 0.73. However, the Sil index of MSTSO-FCM reaches 0.84, showing that the discrimination between clusters is significantly improved on dataset 1. On dataset 2, the DB index of FCM is 1.39, while the DB index of TSO-FCM on dataset 2 is 0.87, the DB index of PSO-FCM is 0.85, and the DB index of MSTSO-FCM on dataset 2 is 0.4735. Compared with TSO-FCM and PSO-FCM, the intra-cluster compactness and inter-cluster separation have been improved, and the clustering effect is better. Regarding the AUC index, MSTSO-FCM achieved first in the four groups of UCI data sets, and had significant advantages compared with TSO-FCM and PSO-FCM, indicating that MSTSO-FCM has excellent classification ability.

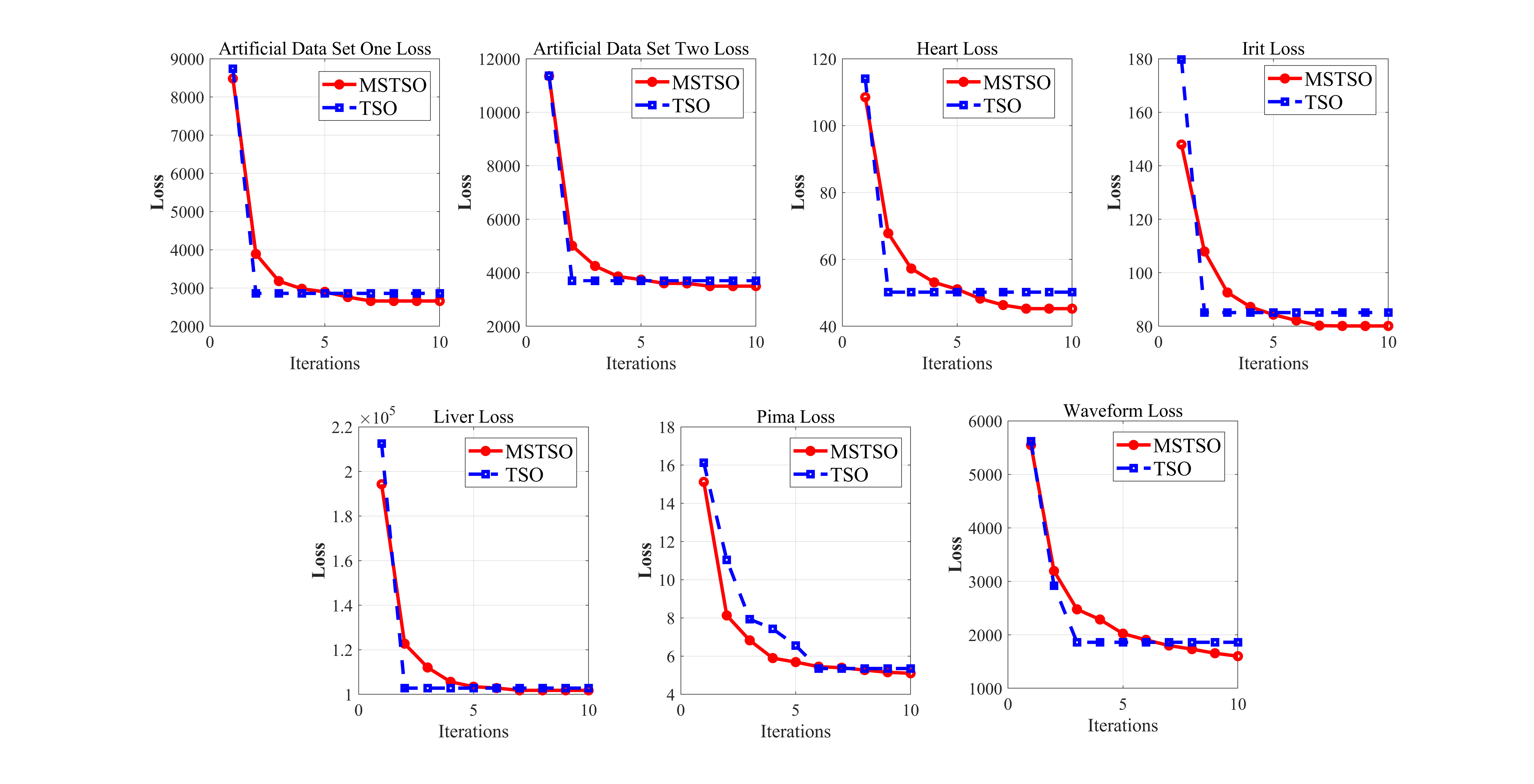


Figure 3: MSTSO optimization and TSO optimization loss reduction curve comparison

**Table 4:** Comparison of clustering performance results

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Dating | AC | Sil | DB | AUC | Algorithm | Dating | AC | Sil | DB | AUC |
| MSTSO-FCM | set1 | 97.13% | 0.84 | 0.15 | 0.96 | FCM | set1 | 89.97% | 0.60 | 0.45 | 0.88 |
| set2 | 90.04% | 0.61 | 0.47 | 0.90 | data | 75.19% | 0.26 | 1.39 | 0.74 |
| iris | 98.67% | 0.50 | 0.64 | 0.97 | iris | 90.67% | 0.54 | 0.59 | 0.89 |
| liver | 63.48% | 0.23 | 2.48 | 0.62 | liver | 50.14% | 0.51 | 1.17 | 0.51 |
| heart | 89.44% | 0.13 | 2.49 | 0.90 | heart | 74.59% | 0.19 | 1.93 | 0.72 |
| pima | 69.12% | 0.19 | 2.34 | 0.65 | pima | 58.18% | 0.32 | 2.72 | 0.55 |
| waveform | 77.62% | 0.58 | 0.75 | 0.75 | waveform | 65.84% | 0.51 | 0.89 | 0.64 |
| PSO-FCM | set1 | 92.7% | 0.73 | 0.26 | 0.91 | GMM | set1 | 89.71% | 0.12 | 0.77 | 0.86 |
| set2 | 77.45% | 0.39 | 0.85 | 0.75 | set2 | 73.26% | 0.21 | 1.46 | 0.72 |
| iris | 92.67% | 0.46 | 0.59 | 0.91 | iris | 96.67% | 0.50 | 0.65 | 0.95 |
| liver | 57.97% | 0.34 | 1.90 | 0.55 | liver | 50.14% | 0.50 | 1.07 | 0.51 |
| heart | 79.54% | 0.21 | 2.39 | 0.77 | heart | 54.78% | 0.18 | 2.12 | 0.53 |
| pima | 65.17% | 0.25 | 2.19 | 0.62 | pima | 61.93% | 0.11 | 2.20 | 0.62 |
| waveform | 70.37% | 0.52 | 0.82 | 0.69 | waveform | 70.62% | 0.53 | 1.21 | 0.68 |
| TSO-FCM | set1 | 94.8% | 0.74 | 0.25 | 0.93 | K-Means | set1 | 80.54% | 0.36 | 0.97 | 0.80 |
| data | 79.36% | 0.38 | 0.87 | 0.77 | set2 | 70.12% | 0.18 | 1.59 | 0.68 |
| iris | 94.67% | 0.51 | 0.64 | 0.93 | iris | 89.33% | 0.55 | 0.58 | 0.88 |
| liver | 56.81% | 0.33 | 1.92 | 0.59 | liver | 46.00% | 0.63 | 0.77 | 0.48 |
| heart | 80.86% | 0.15 | 2.25 | 0.78 | heart | 72.61% | 0.21 | 1.82 | 0.71 |
| pima | 66.26% | 0.23 | 2.51 | 0.63 | pima | 66.92% | 0.17 | 1.78 | 0.67 |
| waveform | 72.42% | 0.56 | 0.79 | 0.61 | waveform | 60.12% | 0.35 | 1.98 | 0.89 |

To visualize the clustering effect, the two-dimensional features of the dataset Heart are selected for display. Each color in Fig. 4 corresponds to a cluster, and the same color representation of different algorithms may not be the same, but it does not affect the results. It can be seen from the figure that MSTSO-FCM has achieved better clustering results than FCM, TSO-FCM, and PSO-FCM.

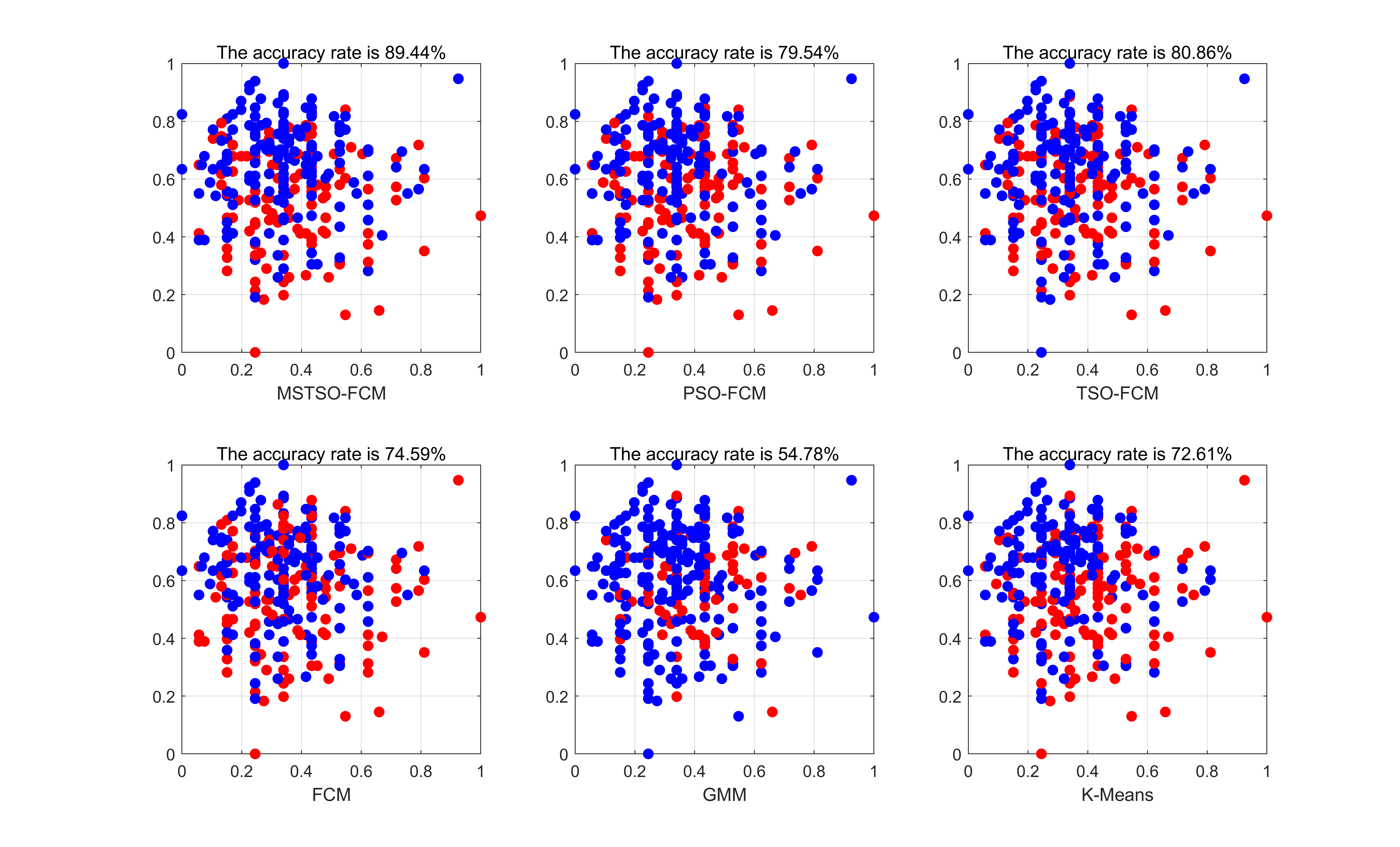


Figure 4： Heart dataset two-dimensional feature clustering effect visualization

It can be seen from Table 4 that MTSO-FCM can achieve better classification results in a variety of different sample situations. Compared with FCM, MTSO-FCM overcomes the data sensitivity better and shows excellent classification performance. It can be seen from the AUC index that this algorithm has a better classification ability of positive and negative cases. In Fig. 3, the MSTSO algorithm has a better global search ability. However, its computational complexity has not been significantly improved, and it belongs to an improvement of the traditional hard clustering algorithm.

**5 Conclusion**

In this paper, a clustering algorithm of MSTSO-FCM is proposed, which first integrates the chaotic local search strategy and the shift distribution estimation strategy into the TSO algorithm, which improves the performance of basic TSO by enhancing the development ability and maintaining population diversity. Secondly, the search and development characteristics of the MSTSO algorithm are introduced into the fuzzy matrix of FCM, which overcomes the defects of poor global search ability and sensitive initialization. It not only uses the searchability of MSTSO but also retains the fuzzy mathematical ideas of FCM, to improve the clustering accuracy, stability, and accuracy of the FCM algorithm. In two groups of artificial datasets and four groups of UCI datasets, MSTSO-FCM achieves the first good results in the AC index and AUC index, indicating that it has excellent clustering and classification ability, and has significant improvement compared with TSO-FCM and PSO-FCM in Sil and DB two internal indicators, it shows that the intra-cluster compactness and inter-cluster separation have been improved. The simulation results show that the improved clustering algorithm has better convergence ability and clustering effect.

In the future research direction, the density clustering method will be studied, and FCM will be combined with density clustering, and it is necessary to focus more on solving real-world engineering application problems.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:**

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

# References:

1. G. Atluri, A. Karpatne and V. Kumar, “Spatio-temporal data mining: a survey of problems and methods,” *Acm Computing Surveys,* vol. 51, no. 4, pp. 1-41, 2018.
2. H. M. Jia, Z. C. Jiang and Y. Li, “Simultaneous feature selection optimization based on improved bald eagle search algorithm,” *Control and Decision*, vol. 37, no. 2, pp. 445-454, 2022.
3. [D. Banerjee](https://scholar.cnki.net/home/search?sw=6&sw-input=Debasish Banerjee), “Recent progress on cluster and meron algorithms for strongly correlated systems,” [*Indian Journal of Physics*](https://scholar.cnki.net/journal/index/SSJD097314582052)*,* vol. 95, no. 8, pp. 1669-1680,2021.
4. [M. S. Kumar,](https://scholar.cnki.net/home/search?sw=6&sw-input=Mohapatra Srikanta Kumar) [S. Premananda](https://scholar.cnki.net/home/search?sw=6&sw-input=Sahu Premananda), A. Jasem, A. Roobaea and R. Saeed et al., “Segmentation and classification of encephalon tumor by applying improved fast and robust FCM Algorithm with PSO-based ELM Technique,” [*Computational Intelligence and Neuroscience*](https://scholar.cnki.net/journal/index/SPQD16875265018D), vol. 2022, no. 1, pp. 1-9, 2022.
5. M. Mehran, K. Ali and [H. Hamed](https://scholar.cnki.net/home/search?sw=6&sw-input=Hashemi-Dezaki Hamed), “Clustering-based reliability assessment of smart grids by fuzzy c-means algorithm considering direct cyber–physical interdependencies and system uncertainties,” [*Sustainable Energy, Grids and Networks*](https://scholar.cnki.net/journal/index/SJES235246773378), vol. 31, pp. 100757, 2022.
6. Q. H. Yang. “An FCM clustering algorithm based on the identification of accounting statement whitewashing behavior in universities,” *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 345-355, 2022.
7. R. Poli, J. Kennedy and T. Blackwell. “Particle swarm optimization,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33-57, 2007.
8. [M. Maryam,](https://scholar.cnki.net/home/search?sw=6&sw-input=Mohammdian khoshnoud Maryam) [S. Ali Reza](https://scholar.cnki.net/home/search?sw=6&sw-input=Soltanian Ali Reza) and [D. Arash,](https://scholar.cnki.net/home/search?sw=6&sw-input=Dehghan Arash) “Optimization of fuzzy c-means (FCM) clustering in cytology image segmentation using the gray wolf algorithm,” [*BMC Molecular and Cell Biology*](https://scholar.cnki.net/journal/index/SJPD26618850165X), vol. 23, no. 1, pp. 9, 2022.
9. B. Amit and [A. Issam](https://scholar.cnki.net/home/search?sw=6&sw-input=AbuMahfouz Issam), “A novel adaptive FCM with cooperative multi-population differential evolution optimization,” [*Algorithms*](https://scholar.cnki.net/journal/index/SJPD19994893068D), vol. 15, no. 10, pp. 380, 2022.
10. T. Niknam, J. Olamaei and B. Amiri, “A hybrid evolutionary algorithm based on ACO and SA for cluster analysis,” *Journal of Applied Sciences*, vol. 8, no. 15, pp. 2695-2702, 2008.
11. H. Gao, Y. Li, P. Kabalyants, H. Xu and R. Martinez-Bejar, “A novel hybrid PSO-K-Means clustering algorithm using gaussian estimation of distribution method and lévy flight,” *IEEE Access*, vol. 8, pp. 122848-122863, 2020.
12. H. Izakian and A. Abraham, “Fuzzy C-means and fuzzy swarm for fuzzy clustering problem,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 1835-1838, 2011.
13. Z. Y. Qian, Y. H. Cao, X. S. Sun, L. Ni and Z. Y. Chen, et al., “Clustering optimization for triple-frequency combined observations of BDS-3 based on improved PSO-FCM algorithm,” [*Remote Sensing*](https://scholar.cnki.net/journal/index/SJDJ207242921425), vol. 14, no. 15, pp. 3713-3713, 2022.
14. O. Celal, H. Emrah and K. Dervis, “Dynamic clustering with improved binary artificial bee colony algorithm,” *Applied Soft Computing*, vol. 28, pp. 69-80, 2015.
15. J. Y. Wang, L. K. Zhu, B. W. Wu and A. Ryspayev, “Forestry canopy image segmentation based on improved tuna swarm optimization,” *Forests*, vol. 13, no. 11, pp. 1746, 2022.
16. W. T. Wang and J. Tian, “An improved nonlinear tuna swarm optimization algorithm based on circle chaos map and levy flight operator,” *Electronics*, vol. 11, no. 22, pp. 3678, 2022.
17. W. Tuerxun, C. Xu, H. Y. Gou, L. Guo and N. M. Zeng, et al., “An ultra-short-term wind speed prediction model using LSTM based on modified tuna swarm optimization and successive variational mode decomposition,” *Energy Science & Engineering*, vol.10, no.8, pp.3001-3022, 2022,
18. M. L. Tang, Y. T. Li, D. L. Ding, R. Zhou and C. Q. Huang, “An improved JADE hybridizing with tuna swarm optimization for numerical optimization problems,” *Mathematical Problems in Engineering*, vol. 2022, pp. 7726548, 2022.
19. C. X. Fu and L. X. Zhang, “A novel method based on tuna swarm algorithm under complex partial shading conditions in PV system,” *Solar Energy*, vol. 248, pp. 28-40, 2023.
20. C. Kumar and D. M. Mary, “A novel chaotic-driven tuna swarm optimizer with newton-raphson method for parameter identification of three-diode equivalent circuit model of solar photovoltaic cells/ modules,” *Optik*, vol. 264, no.1, pp. 169379, 2022.
21. Q. Ren, H. B. Zhang, D. L. Zhang and X. Zhao, “Lithology identification using principal component analysis and particle swarm optimization fuzzy decision tree,” *Journal of Petroleum Science and Engineering*, vol. 220, pp. 111233, 2023.
22. T. Y. Wu, J. C. W. Lin, Y. Y. Zhang and C. H. Chen, “A grid-based swarm intelligence algorithm for privacy-preserving data mining,” *Applied Science*, vol. 9, no.4, pp. 774, 2019.
23. Y. A. Shao, J. C. W. Lin, G. Srivastava, D. D. Guo and H. C. Zhang et al., “Multi-objective neural evolutionary algorithm for combinatorial optimization problems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 2133-2143, 2021.
24. L. Wang, G. Cui and X. Cai, “Fuzzy clustering optimal k selection method based on multi-objective optimization,” *Soft Computing*, vol. 27, no. 3, pp. 1289-1301, 2023.
25. A. Aggarwal, P. Dimri, A. Agarwal, V. Madhushi and H. A. Alhumyani, et al., “IFFO: an improved fruit fly optimization algorithm for multiple workflow scheduling minimizing cost and makespan in cloud computing environments,” *Mathematical Problems in Engineering*, vol. 2021, pp. 5205530, 2021.
26. A. Aggarwal, S. Kumar, A. Bhatt and M. A. Shah, “Solving user priority in cloud computing using enhanced optimization algorithm in workflow scheduling,” *Computational Intelligence and Neuroscience*, vol.2022, pp. 7855532, 2022.
27. P. Upadhyay, V. Marriboina, A. Kumar, K. Sunil and M. A. Shah, “An enhanced hybrid glowworm swarm optimization algorithm for traffic-aware vehicular networks,” *IEEE Access*, vol. 10, pp. 110136-110148, 2022.
28. H. M. Jia, Z. Q. Zhang, Z. C. Jiang and Y. Q. Feng, “An optimization fuzzy C-means clustering algorithm based on the hybrid identity search and slime mold algorithms,” *CAAI Transactions on Intelligent Systems*, vol. 17, no. 5, pp. 999-1011, 2022.
29. Q. Usman, “A dissimilarity measure based fuzzy c-means (FCM) clustering algorithm,” *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 1, pp. 229-238, 2014.
30. M. Tanveer, C. Gautam and P.N. Suganthan, “Comprehensive evaluation of twin SVM based classifiers on UCI datasets,” *Applied Soft Computing*, vol. 83, pp. 105617, 2019.
31. Y. H. Ma and Y. Hao, “Antenna classification using gaussian mixture models (GMM) and machine learning,” *IEEE Open Journal of Antennas and Propagation*, vol. 1, pp. 320-328, 2020.
32. B. Mao and B. C. Li, “Building facade semantic segmentation based on K-means classification and graph analysis,” *Arabian Journal of Geosciences*, vol. 12, no. 7, pp. 253, 2019.

ANNEX-1

Test results of seven algorithms at CEC2017

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Indicator | BOA | RSA | AOA | TSA | SSA | TSO | MSTSO |
| F1 | Mean | 3.82E+04 | 6.76E+03 | 6.91E+04 | 3.83E+04 | 8.40E+04 | 7.52E+04 | 2.67E-06 |
| Std | 6.97E+03 | 2.46E+03 | 1.15E+04 | 1.19E+04 | 6.59E+03 | 8.60E+03 | 9.20E-07 |
| Rank | 3 | 2 | 5 | 4 | 7 | 6 | 1 |
| F2 | Mean | 9.33E+03 | 8.48E+01 | 7.61E+03 | 1.62E+03 | 1.44E+03 | 8.10E+03 | 5.90E+01 |
| Std | 1.29E+03 | 3.03E+01 | 2.45E+03 | 1.40E+03 | 1.09E+03 | 2.45E+03 | 1.51E+00 |
| Rank | 7 | 2 | 5 | 4 | 3 | 6 | 1 |
| F3 | Mean | 3.49E+02 | 1.23E+02 | 2.95E+02 | 2.76E+02 | 3.50E+02 | 4.04E+02 | 3.42E+01 |
| Std | 2.16E+01 | 2.19E+01 | 3.20E+01 | 4.09E+01 | 4.40E+01 | 2.46E+01 | 1.25E+01 |
| Rank | 5 | 2 | 4 | 3 | 6 | 7 | 1 |
| F4 | Mean | 6.63E+01 | 2.49E+01 | 6.21E+01 | 6.15E+01 | 8.06E+01 | 8.29E+01 | 6.44E-03 |
| Std | 5.76E+00 | 7.04E+00 | 6.71E+00 | 1.43E+01 | 8.84E+00 | 8.30E+00 | 8.25E-03 |
| Rank | 5 | 2 | 4 | 3 | 6 | 7 | 1 |
| F5 | Mean | 5.57E+02 | 2.12E+02 | 6.00E+02 | 4.83E+02 | 7.12E+02 | 6.59E+02 | 6.07E+01 |
| Std | 3.17E+01 | 5.10E+01 | 5.66E+01 | 7.78E+01 | 6.85E+01 | 5.57E+01 | 1.48E+01 |
| Rank | 4 | 2 | 5 | 3 | 7 | 6 | 1 |
| F6 | Mean | 2.93E+02 | 9.91E+01 | 2.25E+02 | 2.34E+02 | 2.72E+02 | 3.28E+02 | 3.64E+01 |
| Std | 1.54E+01 | 2.68E+01 | 2.67E+01 | 3.99E+01 | 4.31E+01 | 1.94E+01 | 1.30E+01 |
| Rank | 6 | 2 | 3 | 4 | 5 | 7 | 1 |
| F7 | Mean | 6.82E+03 | 1.53E+03 | 4.50E+03 | 8.57E+03 | 9.35E+03 | 8.55E+03 | 1.56E-01 |
| Std | 8.69E+02 | 6.00E+02 | 7.24E+02 | 3.01E+03 | 1.85E+03 | 9.42E+02 | 2.74E-01 |
| Rank | 4 | 2 | 3 | 6 | 7 | 5 | 1 |
| F8 | Mean | 7.33E+03 | 3.76E+03 | 5.51E+03 | 5.55E+03 | 7.05E+03 | 7.22E+03 | 4.84E+03 |
| Std | 2.85E+02 | 6.30E+02 | 5.83E+02 | 6.07E+02 | 7.45E+02 | 3.98E+02 | 7.52E+02 |
| Rank | 7 | 1 | 3 | 4 | 5 | 6 | 2 |
| F9 | Mean | 2.19E+03 | 1.00E+02 | 1.72E+03 | 2.23E+03 | 3.91E+03 | 7.28E+03 | 2.42E+01 |
| Std | 6.72E+02 | 4.16E+01 | 9.74E+02 | 1.69E+03 | 1.64E+03 | 1.81E+03 | 2.31E+01 |
| Rank | 4 | 2 | 3 | 5 | 6 | 7 | 1 |
| F10 | Mean | 2.08E+09 | 8.30E+04 | 6.27E+09 | 8.88E+08 | 4.69E+08 | 1.32E+10 | 3.09E+02 |
| Std | 7.43E+08 | 7.32E+04 | 2.56E+09 | 1.07E+09 | 3.76E+08 | 2.71E+09 | 2.15E+02 |
| Rank | 5 | 2 | 6 | 4 | 3 | 7 | 1 |
| F11 | Mean | 3.15E+08 | 1.01E+04 | 3.80E+04 | 1.75E+08 | 8.55E+07 | 8.92E+09 | 4.96E+01 |
| Std | 2.10E+08 | 1.20E+04 | 1.71E+04 | 4.14E+08 | 4.66E+08 | 3.93E+09 | 1.67E+01 |
| Rank | 6 | 2 | 3 | 5 | 4 | 7 | 1 |
| F12 | Mean | 1.19E+05 | 2.45E+03 | 5.72E+04 | 3.73E+05 | 1.50E+06 | 5.68E+06 | 3.23E+01 |
| Std | 7.62E+04 | 2.63E+03 | 4.92E+04 | 6.73E+05 | 1.21E+06 | 4.39E+06 | 8.21E+00 |
| Rank | 4 | 2 | 3 | 5 | 6 | 7 | 1 |
| F13 | Mean | 1.82E+06 | 7.17E+03 | 2.35E+04 | 2.48E+07 | 1.83E+07 | 5.67E+08 | 2.41E+01 |
| Std | 1.46E+06 | 7.41E+03 | 1.22E+04 | 7.80E+07 | 2.37E+07 | 3.47E+08 | 4.53E+00 |
| Rank | 4 | 2 | 3 | 6 | 5 | 7 | 1 |
| F14 | Mean | 3.18E+03 | 9.68E+02 | 1.98E+03 | 1.43E+03 | 2.74E+03 | 3.90E+03 | 4.12E+02 |
| Std | 4.12E+02 | 3.29E+02 | 5.09E+02 | 2.92E+02 | 5.38E+02 | 6.13E+02 | 2.62E+02 |
| Rank | 6 | 2 | 4 | 3 | 5 | 7 | 1 |
| F15 | Mean | 1.22E+03 | 4.21E+02 | 9.12E+02 | 6.06E+02 | 1.20E+03 | 2.69E+03 | 1.04E+02 |
| Std | 2.49E+02 | 1.77E+02 | 2.67E+02 | 2.30E+02 | 3.85E+02 | 1.14E+03 | 4.28E+01 |
| Rank | 6 | 2 | 4 | 3 | 5 | 7 | 1 |
| F16 | Mean | 9.60E+05 | 1.21E+05 | 1.29E+06 | 2.08E+06 | 1.51E+07 | 3.51E+07 | 3.02E+01 |
| Std | 6.22E+05 | 1.09E+05 | 1.60E+06 | 4.09E+06 | 1.51E+07 | 2.38E+07 | 2.15E+00 |
| Rank | 3 | 2 | 4 | 5 | 6 | 7 | 1 |
| F17 | Mean | 4.61E+06 | 8.43E+03 | 1.08E+06 | 1.11E+07 | 4.23E+07 | 6.66E+08 | 2.17E+01 |
| Std | 4.06E+06 | 9.45E+03 | 1.39E+05 | 3.45E+07 | 1.23E+08 | 3.75E+08 | 3.32E+00 |
| Rank | 4 | 2 | 3 | 5 | 6 | 7 | 1 |
| F18 | Mean | 7.29E+02 | 3.97E+02 | 6.94E+02 | 7.24E+02 | 8.59E+02 | 9.81E+02 | 1.76E+02 |
| Std | 9.88E+01 | 1.32E+02 | 1.54E+02 | 2.09E+02 | 2.42E+02 | 1.38E+02 | 6.63E+01 |
| Rank | 5 | 2 | 3 | 4 | 6 | 7 | 1 |
| F19 | Mean | 1.97E+02 | 3.03E+02 | 4.87E+02 | 4.68E+02 | 5.06E+02 | 6.05E+02 | 2.35E+02 |
| Std | 3.01E+01 | 2.44E+01 | 5.23E+01 | 4.96E+01 | 5.36E+01 | 4.53E+01 | 1.51E+01 |
| Rank | 1 | 3 | 5 | 4 | 6 | 7 | 2 |
| F20 | Mean | 4.71E+02 | 1.01E+02 | 5.13E+03 | 4.47E+03 | 4.18E+03 | 6.15E+03 | 1.00E+02 |
| Std | 7.76E+01 | 2.02E+00 | 1.21E+03 | 2.09E+03 | 1.88E+03 | 1.31E+03 | 5.68E-01 |
| Rank | 3 | 2 | 6 | 5 | 4 | 7 | 1 |
| F21 | Mean | 6.97E+02 | 4.97E+02 | 9.68E+02 | 7.86E+02 | 8.60E+02 | 1.13E+03 | 3.86E+02 |
| Std | 5.59E+01 | 4.04E+01 | 9.10E+01 | 8.15E+01 | 1.00E+02 | 1.10E+02 | 1.59E+01 |
| Rank | 3 | 2 | 6 | 4 | 5 | 7 | 1 |
| F22 | Mean | 1.10E+03 | 5.59E+02 | 1.14E+03 | 8.47E+02 | 8.99E+02 | 1.20E+03 | 4.57E+02 |
| Std | 1.68E+02 | 5.59E+01 | 1.09E+02 | 8.08E+01 | 1.37E+02 | 1.92E+02 | 1.41E+01 |
| Rank | 5 | 2 | 6 | 3 | 4 | 7 | 1 |
| F23 | Mean | 1.75E+03 | 3.93E+02 | 1.67E+03 | 7.61E+02 | 7.84E+02 | 2.54E+03 | 3.87E+02 |
| Std | 2.01E+02 | 1.44E+01 | 4.55E+02 | 3.02E+02 | 1.30E+02 | 5.87E+02 | 6.52E-02 |
| Rank | 6 | 2 | 5 | 3 | 4 | 7 | 1 |
| F24 | Mean | 5.21E+03 | 2.34E+03 | 6.40E+03 | 5.01E+03 | 6.30E+03 | 8.12E+03 | 1.32E+03 |
| Std | 1.49E+03 | 9.05E+02 | 7.22E+02 | 8.76E+02 | 1.11E+03 | 6.89E+02 | 1.46E+02 |
| Rank | 4 | 2 | 6 | 3 | 5 | 7 | 1 |
| F25 | Mean | 8.14E+02 | 5.59E+02 | 1.34E+03 | 7.30E+02 | 9.56E+02 | 1.49E+03 | 5.00E+02 |
| Std | 9.81E+01 | 4.78E+01 | 2.14E+02 | 9.92E+01 | 1.65E+02 | 3.99E+02 | 8.10E+00 |
| Rank | 4 | 2 | 6 | 3 | 5 | 7 | 1 |
| F26 | Mean | 3.28E+03 | 3.71E+02 | 2.95E+03 | 1.27E+03 | 1.06E+03 | 3.57E+03 | 3.53E+02 |
| Std | 3.99E+02 | 5.38E+01 | 6.15E+02 | 4.52E+02 | 3.22E+02 | 7.74E+02 | 6.03E+01 |
| Rank | 6 | 2 | 5 | 4 | 3 | 7 | 1 |
| F27 | Mean | 3.04E+03 | 1.01E+03 | 2.43E+03 | 1.58E+03 | 2.64E+03 | 4.07E+03 | 5.42E+02 |
| Std | 4.72E+02 | 2.69E+02 | 5.22E+02 | 4.08E+02 | 6.35E+02 | 1.11E+03 | 7.38E+01 |
| Rank | 6 | 2 | 4 | 3 | 5 | 7 | 1 |
| F28 | Mean | 3.98E+07 | 6.41E+03 | 1.47E+07 | 1.33E+07 | 4.85E+07 | 1.58E+09 | 2.00E+03 |
| Std | 2.31E+07 | 3.26E+03 | 1.01E+07 | 1.07E+07 | 3.75E+07 | 5.48E+08 | 8.39E+01 |
| Rank | 5 | 2 | 4 | 3 | 6 | 7 | 1 |