Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **1**, February **2024**

Density Peaking Clustering Algorithm Based on Improved Tuna Swarm Optimization

Hang Zhang*
College of Computer Science and Technology
Henan Polytechnic University
Jiaozuo 454000, China
melodyzkh@163.com

Yong-Li Liu
College of Computer Science and Technology
Henan Polytechnic University
Jiaozuo 454000, China
yongli.buaa@gmail.com

Hao Chao
College of Computer Science and Technology
Henan Polytechnic University
Jiaozuo 454000, China
chaohao1981@163.com

*Corresponding author: Hang Zhang

Received March 21, 2023, revised May 7, 2023, accepted September 8, 2023.

Abstract. *The density peak clustering algorithm shows good clustering performance by rapidly determining each cluster division with a high-density region as the kernel. However, the cut-off distance (dc) as the only parameter of the algorithm usually needs to be specified manually, so it is limited in practical applications. In order to automatically select the optimal value of the cut-off distance parameter, a Density Peak Clustering (DPC) algorithm based on Improved Tuna Swarm Optimization (ITSO) was designed. To begin, the ITSO algorithm introduces multi-strategy chaotic mapping initialization and a self-adaptive factor, which boosts tuna swarm search performance, and conducts simulation experiments on 8 benchmark functions, with the results demonstrating that the ITSO algorithm outperforms other comparison algorithms; then the cut-off distance in the DPC is optimized using the ITSO algorithm, and the objective function is developed based on the accuracy index. In order to evaluate the effectiveness of the ITSO-DPC algorithm, five synthetic datasets and three UCI standard datasets were chosen for experiments. The results of the experiments demonstrate that the proposed algorithm can automatically select a better cut-off distance value and guarantee a higher clustering accuracy.*

Keywords: density peak clustering; cut-off distance; tuna swarm optimization; multi-strategy

1. Introduction.

Artificial intelligence technologies have become more and more significant in a variety of industries in recent years. AI technologies are revolutionizing our lives and work in a variety of ways, including speech recognition [1], image processing [2], natural language processing [3], and intelligent recommendation systems. Cluster analysis is one of the key AI technologies, and the application range of these technologies is growing along with the amount of data and advancements in computing power.

Clustering techniques group data samples according to the potential relationships between them: data samples within the same group are more correlated, and data samples between different groups are less correlated [4]. Currently, clustering analysis is widely used in machine learning [5], computer vision, image analysis [6], and biology. Based on different data grouping rules, the existing clustering algorithms are mainly divided into partition-based clustering methods [7], hierarchy-based clustering methods [8], density-based clustering methods [9], and grid-based clustering methods [10], etc. Cluster analysis still has several issues, despite the fact that it has been frequently employed to solve practical issues [11, 12]. For instance, the number of clusters is typically unknown in real-world issues, while k-means [13] and hierarchical clustering algorithms [14] in cluster analysis typically require defining the number of clusters. Furthermore, because clustering analysis algorithms are so sensitive to noise and outliers, the quality of the clustering results can be impacted.

New cluster analysis algorithms are constantly being proposed to solve these problems. In 2014, Rodriguez and Laio [15] proposed Density Peaks Clustering (DPC). The algorithm can quickly discover the class cluster centers of arbitrarily shaped datasets; it does not need to determine the number of class clusters in advance; it can quickly assign data samples other than the class cluster centers; it is effective in detecting outliers; and it is suitable for large-scale datasets. Despite DPC has the above-mentioned advantages, it will suffer some limitations when manually setting the cut-off distance dc in real applications. The exact dc selection will affect the clustering capability of DPC.

In the problem of finding the optimal solution, metaheuristic algorithms provide good solutions, such as genetic algorithm [16], particle swarm optimization algorithm [17], phasmatodea population evolution algorithm [18], etc. In order to select the dc parameter of DPC more precisely, it can be transformed into an optimization problem. Based on the good performance of Tuna Swarm Optimization (TSO) [19] in convergence performance and optimization accuracy, this paper studies the selection of the dc parameter. The creation of the TSO algorithm was motivated by the foraging strategies of two distinct tuna species, and the algorithm demonstrated good performance in terms of sensitivity, scalability, and robustness. For reasons of efficiency and ease of use, this paper introduces multi-strategy hybrid initialization and a self-adaptive factor into the TSO algorithm to help further improve convergence performance, whereby the Density Peaking Clustering Algorithm Based on Improved Tuna Swarm Optimization (ITSO-DPC) algorithm is proposed. Stronger global search capabilities and quicker convergence speed are advantages of the modified TSO algorithm, and the algorithm is able to find globally optimal cut-off distances on different datasets and solve the problem of setting the parameters of the DPC algorithm.

2. Related work.

DPC has the advantages of simple implementation, fast clustering, no need to determine the number of cluster centers, and few parameters. Benefiting from the above advantages, the DPC algorithm quickly became a research hotspot after it was proposed, and a large number of improved algorithms emerged. Mehmood Rashid [20] provides a non-parametric approach for determining the probability distribution of a specified dataset using thermal diffusion. The algorithm introduces diffusion partial differential variance on the one hand, using the kernel density prediction as a particular solution to its equation, and narrows the sensitivity of the cut-off distance through the time parameter of the thermal diffusion equation on the other hand. Unfortunately, the algorithm is under a high computational cost due to the inherent limitations of kernel density estimation. In order to boost the effectiveness of the DPC algorithm, Chen et al. [21] designed a fast-density peak clustering algorithm that is appropriate for handling huge amounts of data and distinguishing between local and non-local density peaks using fast KNN retrieval. In order to overcome the difficulties of variable density clustering and joint assignment error, Liu et al. [22] developed a density peak clustering based on shared nearest neighbors, redefined local density and distance, and incorporated shared neighbour and local density data between points. In order to fix the problem of uneven density computation in the DPC algorithm, Xie et al. [23] devised a fuzzy weighted KNN density peak clustering algorithm (FKNN-DPC), which tries to find the clustering center and recreates the local density of points through KNN. To enhance the efficiency of the DPC algorithm on outliers, Jiang et al. [24] designed a density peak clustering based on gravity theory, which uses universal gravity theory to boost the effectiveness of the DPC algorithm for anomaly detection.

The aforementioned algorithms have generated good clustering accuracy and efficiency results, although the DPC algorithm's *dc* parameter still needs to be manually determined. *Dc* is very sensitive to the value and small changes may have a large impact on the clustering results. To automate the process of obtaining the optimal *dc* parameter, In order to achieve automatic determination of *dc*, Gao et al. [25] designed an adaptable density peaking clustering algorithm utilizing the Gini index as the objective optimization function and solved by an extended pattern search algorithm. In order to optimize the *dc* parameter, Zhao et al. [26] devised a DPC algorithm based on firefly optimization by using Rand Index as the objective function. Zhou et al. [27] used the Silhouette Validity Index as the clustering objective function to optimize the Silhouette Validity Index using the fruit fly optimization algorithm and obtained better *dc* value in the experiment. It can be observed that the meta-heuristic optimization algorithm is more effective in optimizing the *dc* parameter. To eventually get trustworthy image segmentation findings, Zhu et al. [28] merged the Fruit Fly Optimization Algorithm with the DPC algorithm and employed the image entropy, which responds to the whole image information, as the smell objective function.

3. Density Peaking Clustering Algorithm Based on Improved Tuna Swarm Optimization.

This paper focuses on three main improvements: firstly, the population is initialized by a mixture of Piecewise chaos mapping and cubic chaos mapping fused with a reverse refraction mechanism; secondly, the parameter β is updated using an adaptive inertial weight coefficient formula; and finally, the *dc* value in the DPC algorithm is optimized using the stronger optimization-seeking performance of ITSO.

3.1. Density Peaks Clustering.

The main goal of the DPC algorithm is to find high-density districts split by low-density districts. The method makes the assumption that the density of the class cluster centroid must be higher than the density of its immediate neighbors and that there must be a significant distance between the class cluster centroid and higher-density sites. Therefore, DPC has two main variables that have to be calculated: the local density and the distance to the higher-density locations.

For a given dataset $\{x_1, x_2, \ldots, x_N\}$, the Euclidean distance between the datasets x_i and x_j can be expressed as $d_{ij} = dist(x_i, x_j)$. The cut-off kernel can determine the local density of any point x_i as:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1}$$

where:

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Local densities can also be calculated from the Gaussian kernel as follows:

$$\rho_i = \sum_j \exp(-\frac{{d_{ij}}^2}{{d_c}^2}) \tag{3}$$

δ_i is determined by the shortest distance between x_i and every other point of higher density, as showed in the Equation (4) is shown.

$$\delta_i = \begin{cases} \max\limits_j (d_{ij}) & \rho = \max(\rho_1, \rho_2, ..., \rho_n) \\ \min\limits_{j:\rho_j<\rho_i} (d_{ij}) & \text{otherwise} \end{cases} \tag{4}$$

Based on the predicted local densities ρ and distances δ, a decision map is produced. The relatively larger values of ρ and δ are manually selected from the decision diagram as the cluster centers. To automatically determine the cluster centers, the metric γ is proposed to consider both ρ and δ. As the Equation (5) is indicated.

$$\gamma_i = \rho_i \delta_i \tag{5}$$

Finally, after the DPC algorithm assigns labels to the clustering centers according to a one-step label assignment strategy, the rest of the points are then distributed to the centers with greater density than them and closest to them to get their labels.

3.2. **Tuna Swarm Optimization.** The Tuna Swarm Optimization algorithm is characterised by fast convergence and high solution accuracy of single-objective problems, and its design idea is based on two collaborative foraging behaviors of tuna swarms: spiral foraging and parabolic foraging.

3.2.1. *spiral foraging.* Swarms of tuna hunt their prey by closely enclosing each other in a spiral pattern. In addition to this, swarms of tuna transmit important information to each other, and as the heads and tails of the tuna are connected, the swarms can share information. The mathematical formula of spiral foraging created according to the above concepts is as follows:

$$X_i^{t+1} = \begin{cases} \alpha_1(X_{rand}^t + \beta\,|X_{rand}^t - X_i^t|\ ) + \alpha_2 X_i^t, i = 1 & \text{if rand} < \frac{t}{t_{\max}} \\ \alpha_1(X_{rand}^t + \beta\,|X_{rand}^t - X_i^t|\ ) + \alpha_2 X_{i-1}^t, i = 2, 3, ..., N & \\ \alpha_1(X_{best}^t + \beta\,|X_{best}^t - X_i^t|\ ) + \alpha_2 X_i^t, i = 1 & \text{if rand} \geq \frac{t}{t_{\max}} \\ \alpha_1(X_{best}^t + \beta\,|X_{best}^t - X_i^t|\ ) + \alpha_2 X_{i-1}^t, i = 2, 3, ..., N & \end{cases} \tag{6}$$

$$\alpha_1 = a + (1 - a)\frac{t}{t_{\max}} \tag{7}$$

$$\alpha_2 = (1 - a) - (1 - a)\frac{t}{t_{\max}} \tag{8}$$

$$\beta = e^{bl}\cos\ (2\pi b) \tag{9}$$

$$l = e^{3\cos(((t_{\max}+1/t)-1)\pi)} \tag{10}$$

In Equation(6), t is the current count of cycles, t_{max} is the highest iteration threshold and N is the count of individual tuna; X_i^{t+1} is the position of the ith individual after the $t+1$st iteration; X_{rand}^t and X_{best}^t denote the current random individual and the optimal individual, respectively. α_1 and α_2 are weighting factors that constrain the individual tuna to swim to the best position; a is a constant used to constrain the tuna to follow a life pattern in the early stages. β is related to the random individual or the best individual. An evenly spaced random value within [0,1] makes up variable b.

3.2.2. *parabolic foraging.* When parabolic hunting of prey is carried out, the swarms of tuna form a parabolic pattern with the prey as the target of the attack to feed or by searching around itself for food, at which point the specific mathematical model of the position of the population can be described as:

$$X_i^{t+1} = \begin{cases} X_{best}^t + \text{rand}(X_{best}^t - X_i^t) + TFp^2(X_{best}^t - X_i^t) & \text{if rand} < 0.5 \\ TFp^2X_i^t & \text{if rand} \geq 0.5 \end{cases} \tag{11}$$

$$p = \left(1 - \frac{t}{t_{\max}}\right)^{(t/t_{\max})} \tag{12}$$

In Equation(11), TF is a randomly generated integer of 1 or -1, which determines the direction of population exploitation; p is an important parameter that varies adaptively with the number of iterations and affects the range of population exploitation.

Tuna forage cooperatively through the two foraging strategies described above. The TSO algorithm will eventually return the best tuna and its relative fitness by continuously updating the individual positions. Each tuna will either choose a foraging strategy to hunt its prey or be relocated to the spatial location where it is located by probability z in each iteration of the TSO algorithm.

The following is the specific design process of the TSO algorithm:

Step 1: Set t_{max}, t, and the population size NP, and randomly initialize the tuna swarms.

Step 2: Assign a and z, calculate the fitness value of each individual by the objective function, and update X_{best}^t.

Step 3: Update the weight coefficients α_1 and α_2 and the key parameter p. Update the location of the individual according to the obtained value of the *rand*.

When $rand < z$, Reinitialization of the population position.

When $rand \geq z$ and $rand < 0.5$, Equation (6) is executed for a position update.

When $rand \geq z$ and $rand \geq 0.5$, Equation (11) is executed for a position update.

Step 4: Calculate the updated tuna fitness values and update the retention of the best tuna positions and their relative fitness values.

Step 5: Determine whether $t < t_{max}$ is valid, if so, make $t = t + 1$ and go to Step 2; otherwise, the algorithm ends and outputs the best solution and the best value.

3.3. **ITSO strategy.**

3.3.1. *Piecewise chaotic map initialization strategy.* Chaos is a relatively common motion phenomenon in non-linear systems, which has the characteristics of pseudo-randomness, boundary, and ergodicity. By using these characteristics to set the initial position of tuna swarm individuals, not only can the tuna swarm be diversified, but also the distribution of tuna swarm individuals can be made more uniform. The common chaotic maps mainly include Logistic map and Tent map, of which Piecewise chaotic map is one of the typical representatives of the chaotic map, and its mathematical model is as Equation (13) The mathematical model is shown as follows:

$$X_{k+1} = \begin{cases} X_k/P & 0 \leq X_k < P \\ (X_k - p)/(0.5 - P) & P \leq X_k < 0.5 \\ (1 - P - X_K)/(0.5 - P) & 0.5 \leq X_k < 1 - P \\ (1 - X_k)/P & 1 - P \leq X_k < 1 \end{cases} \tag{13}$$

where P and X range from [0,1]. The Piecewise chaotic map is distributed between [0,1], using its chaos as an alternative to random initialization to make the tuna individuals more evenly distributed in the search space.

3.3.2. *Cubic map fusion inverse refraction initialization.* In the original TSO algorithm, the positions of the tuna swarm are generated randomly by means of random initialization. When the positions are unevenly distributed, it has a tendency to lead to low solution accuracy. In this paper, cubic mapping is used to make the tuna swarm positions have better uniformity. The cubic map chaos operator formula is demonstrated below:

$$y^{i+1} = 2{y_i}^3 - 3y_i/2 \tag{14}$$

$$-1 < y_i < 1, y_i \neq 0, i = 0, 1, ..., N \tag{15}$$

$$x_i = \frac{(X_{ub} - X_{lb})(y_i + 1)}{2} + X_{lb} \tag{16}$$

where x_i in Equation (16) is the mapped tuna location; X_{ub} and X_{lb} correspond to the highest and lowest thresholds in the calculation space, respectively. The result of the cubic chaos mapping calculation is mapped onto the individual tuna swarm positions by Equation (16). The tuna positions in Equation (16) are fused with the tuna positions generated by the reverse refraction mechanism shown in Equation (17), then the fitness values are ranked and the number of tuna in the previous population size is removed in order as the initial position of the tuna swarm.

$$X_{ij} = (a_j + b_j)/2 + (a_j + b_j)/2k - X_{ij}/k \tag{17}$$

where the parameter k is the crucial parameter in Opposition-Based Learning (OBL). The Opposition-Based Learning optimization mechanism was proposed by Tizhooshs [29]. This mechanism extends the search by computing the inverse of the current solution, thus finding a better candidate solution for a given problem. However, OBL has some drawbacks, for example, although the algorithm introduced in the early iterations can enhance the optimization performance of the algorithm, in the later iterations OBL tends to fall into local optimality, which leads to poor convergence performance. Refractive backward learning is an improved mechanism that essentially combines the law of refraction of light with backward learning to find a better candidate solution.

3.3.3. *Adaptive factor updating strategy.* During the spiral foraging phase, the weight coefficients of the TSO algorithm are generated randomly within a specified range. This situation fixes the search step of the algorithm and tends to slow down the convergence of the optimization algorithm. β is associated with optimal or random individuals and affects the ability of individual tuna to exploit the search space during the spiral foraging phase.

The adaptive inertia factor strategy designed in this section is as follows:

$$b_{new} = 1 - \exp(-((2(T_{max} - t_r)/t_r))^2) \tag{18}$$

where T_{max} is the maximum cycle threshold, and t_r is the present count of cycles. The parameter b_{new} will vary exponentially according to the count of cycles. In the initial stage of the spiral foraging stage, b_{new} provides a large search step to speed up the optimization speed; in the late spiral foraging stage, b_{new} is relatively small, which increases the algorithm's local search accuracy.

3.4. **Optimized selection of *dc* with ITSO..** The accuracy (ACC) evaluation metric is chosen for the merit-seeking process to measure the clustering quality of the clustering algorithm in this research, which uses the robust merit-seeking performance of the ITSO algorithm to select the best dc. The ACC [30] evaluation metric is defined as follows:

$$\mathrm{ACC} = \sum_{i=1}^{\mathrm{n}} \delta(c_i, \mathrm{map}(q_i))/n \tag{19}$$

where the dataset's total amount of data points is n; $\delta(x,y)$ is a δ function, if $x=y$, then $\delta(x,y)$ outputs 1, otherwise 0; c_i is the original label of the data in the prior knowledge; q_i is the experimentally obtained clustering label; $\mathrm{map}(q_i)$ is the mapping function that matches the clustering result label with the original label.

In the optimization process, the ACC metric is used by the ITSO algorithm as the objective function to optimize the *dc* in the DPC algorithm. Using the ITSO algorithm, the *dc* value corresponding to the ACC metric of the maximization DPC algorithm is output and used as the optimal solution for parameter *dc* of the present dataset.

The ITSO-DPC algorithm has the merit of finding the best parameter values quickly and outputting the best clustering results. The algorithm optimizes the selection of parameter *dc* in the DPC algorithm and promotes the global detection efficiency of the TSO algorithm. Its algorithm flow is as follows:

The flow chart of ITSO-DPC algorithm is shown in Figure 1.

According to Figure 1, the operation steps of the ITSO-DPC algorithm are as follows:

Step 1: Set the maximum iteration threshold T, tuna swarm size $SizeParticles_no$, and *dc* value range from *ub* to *lb*.

Step 2: Initialize the population by using the tuna population location X as the cut-off distance *dc* value and performing a hybrid chaotic mapping in the search space.

Step 3: Substitute dc into Equations (3) and (4). the results of its calculation are input into Equation (5) for the automatic selection of clustering centers.

Step 4: Use Equation (19) to calculate ACC evaluation metrics and the *dc* value corresponding to the maximum ACC evaluation metric is recorded as dc_{max}.

Step 5: The adaptive factor is updated and substituted into Equation (9), and the position of the tuna is updated according to Equations (6) and (11) to record the current best individual and its fitness value.

Step 6: Determine whether T is reached, if the condition is met, the current main loop will be skipped and the best *dc* value and the maximum ACC value will be output, otherwise it will return to Step 3 to continue the optimization search.

Step 7: Clustering is completed using the optimal cut-off distance of the output.

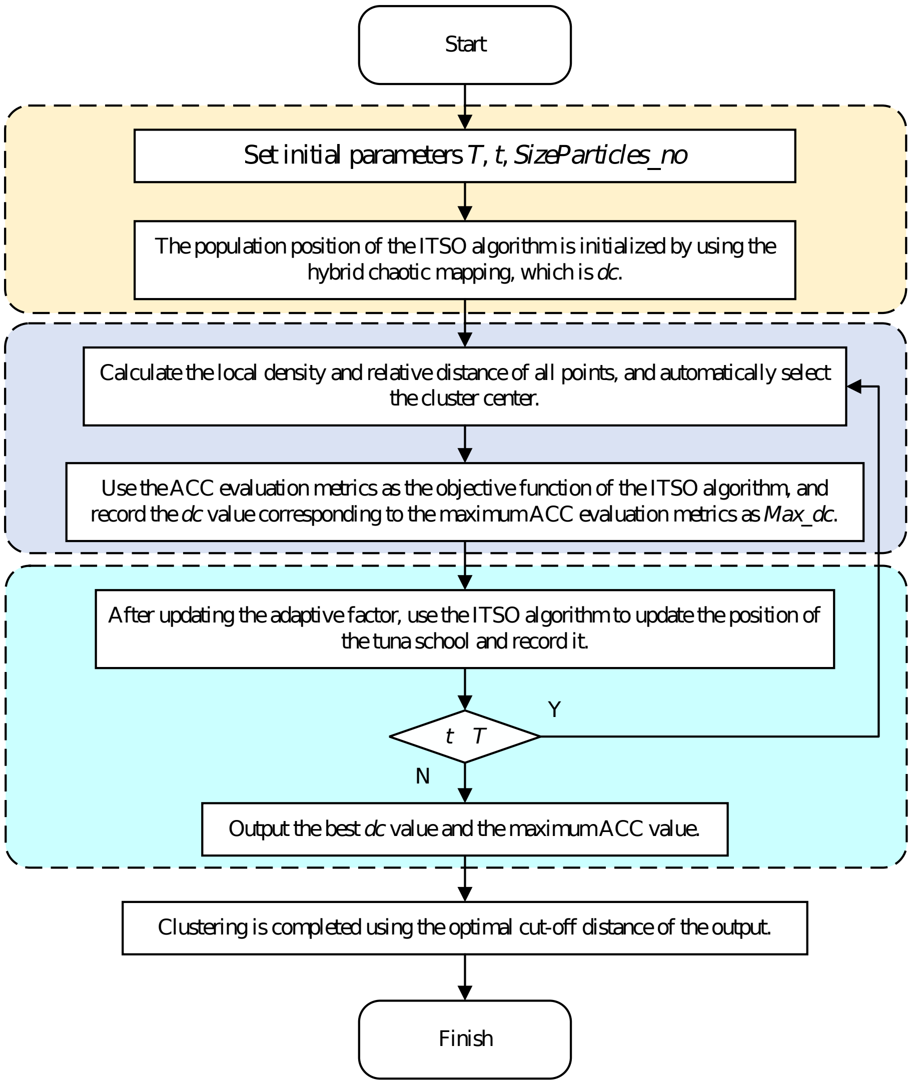

Start
Set initial parameters T, t, SizeParticles_no
The population position of the ITSO algorithm is initialized by using the hybrid chaotic mapping, which is dc.
Calculate the local density and relative distance of all points, and automatically select the cluster center.
Use the ACC evaluation metrics as the objective function of the ITSO algorithm, and record the dc value corresponding to the maximum ACC evaluation metrics as Max_dc.
After updating the adaptive factor, use the ITSO algorithm to update the position of the tuna school and record it.
t T
Y
N
Output the best dc value and the maximum ACC value.
Clustering is completed using the optimal cut-off distance of the output.
Finish

FIGURE 1. ITSO-DPC algorithm flow chart

4. **Simulation experiments and analysis.** The simulation experiments were conducted in an Intel(R) Core(TM) i5-6300HQ, 8.00GB RAM, the x64-based processor with Windows 10 operating system, and Matlab 2022a environment.

4.1. **Optimization algorithm section.** In order to evaluate the ITSO algorithm's efficiency, the grey wolf optimization algorithm (GWO) [31], the sine cosine algorithm (SCA) [32], and the wind driven algorithm (WDO) [33], which are group intelligence optimization algorithms, were selected for comparison experiments. For the simulation experiments, eight benchmark test functions were chosen, and Table 1 displays the functions' details. Single-peaked functions (f_1-f_3) are typically used to evaluate how well algorithms perform local searches; multi-peaked functions (f_4-f_8) have function solutions that are more complicated and are more likely to find a local optimum. To guarantee the test algorithm's performance is accurate and equitable, the experimental environment and parameters used in the comparison algorithm and the ITSO algorithm remain the same. The experiments were set to T=500, N=30, and d=30. For the eight benchmark functions mentioned above, each function was run independently for 30 independent experiments and its mean and standard deviation were found, where the mean was used to reflect the convergence accuracy and the standard deviation was utilized to reflect the stability.

Table 1. Benchmarking function information and results

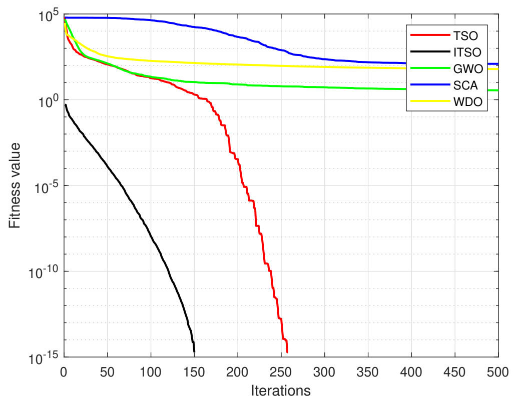
		ITSO	TSO	GWO	SCA	WDO
$f_1(Sphere)$	Mean	6.20E-246	3.04E-235	1.00E-31	4.18E+00	1.00E-15
	Std	0.00E+00	0.00E+00	3.25E-02	8.75E+00	4.60E-02
$f_2(Schwefel2.22)$	Mean	3.12E-124	4.80E-117	1.04E-18	4.25E-02	1.17E-08
	Std	1.40E-01	2.65E-02	7.63E-02	5.97E-02	2.55E-02
$f_3(Schwefel2.21)$	Mean	1.11E-121	3.50E-114	1.63E-08	2.72E+01	1.73E-09
	Std	6.01E-02	1.90E-02	1.34E-01	1.20E+01	2.32E-02
$f_4(Quartic)$	Mean	1.14E-04	3.00E-04	1.90E-03	1.03E+05	2.58E-04
	Std	9.07E-02	3.00E-04	4.00E-03	2.69E-01	1.92E-01
$f_5(Rastrigin)$	Mean	0.00E+00	0.00E+00	3.56E+00	1.23E+02	6.14E+01
	Std	0.00E+00	0.00E+00	4.44E+00	6.62E+01	2.69E+01
$f_6(Ackley)$	Mean	4.44E-16	4.44E-16	2.09E+01	2.03E+01	6.98E-01
	Std	0.00E+00	0.00E+00	7.35E-02	9.16E-02	3.83E+00
$f_7(Griewank)$	Mean	0.00E+00	0.00E+00	3.10E-02	4.08E-01	7.30E-03
	Std	0.00E+00	0.00E+00	6.00E-02	2.45E-01	1.93E-02
$f_8(Penalized2)$	Mean	5.00E-06	1.50E-03	4.52E-01	3.22E+15	4.54E-01
	Std	1.84E-02	3.80E-03	2.07E-01	7.32E-01	9.27E-01

The results in Table 1 demonstrate that all algorithms did not reach the theoretical optimum for (f_1-f_3) under the same parameter environment, but the ITSO algorithm outperformed the other comparative algorithms. For (f_4-f_8), the optimum values reached by the ITSO algorithm were smaller than those of the other algorithms, its standard deviation was smaller and its performance was stable. In conclusion, the stability and performance of the ITSO algorithm are superior. The convergence curves of all the algorithms in the 30-dimensional scenario with four multi-peaked benchmark test functions are displayed in Figure 2 to visualize the speed and efficacy of the ITSO algorithm. This figure also illustrates the effectiveness of the proposed optimization strategy.

4.2. **Experimental results and analysis of the ITSO-DPC section.**

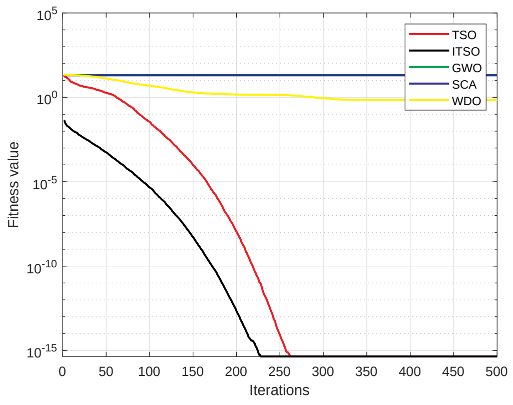
4.2.1. *Experimental dataset.* In order to confirm that the ITSO-DPC worked clustering with greater accuracy than the DPC. Five adult datasets combined and three UCI standard datasets were selected for comparison. The experimental data are shown in Table 2.

4.2.2. *Evaluation Indicator.* To avoid reusing ACC metrics to assess clustering algorithm efficiency, two evaluation metrics independent of label absolute values were utilized to assess clustering capabilities: Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI). The upper bound of these three indicators is 1, and the closer the value of the indicator is to the upper bound, the better the clustering result. ARI is defined as

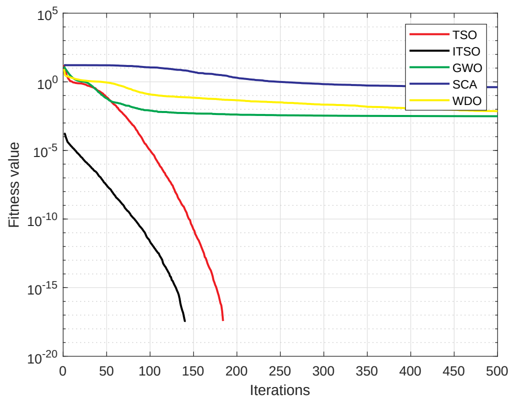


Fitness value
Iterations
TSO
ITSO
GWO
SCA
WDO

(a) (b)

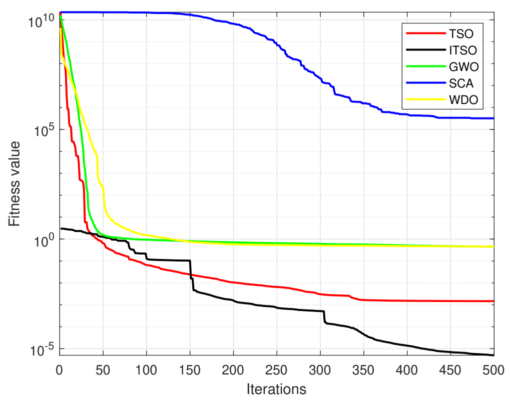


Fitness value
Iterations
TSO
ITSO
GWO
SCA
WDO



Fitness value
Iterations
TSO
ITSO
GWO
SCA
WDO

(c) (d)



Fitness value
Iterations
TSO
ITSO
GWO
SCA
WDO

Figure 2. Convergence curves for test function optimization. (a) f_5. (b) f_6. (c) f_7. (d) f_8.

Table 2. Experimental data

Dataset	Samples	Features	Clusters
Aggregation	788	2	7
Jain	373	2	2
R15	600	2	15
Spiral	312	2	3
D31	3100	2	31
Seeds	210	7	3
Ecoli	336	8	8
Iris	150	4	3

shown below:

$$ARI = \frac{\sum_{ij}\binom{M_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_l}{2}\right] / \binom{M}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{M}{2}} \tag{20}$$

The value of ARI is calculated by introducing a column linkage table to represent the degree of overlap between the actual data sample category divisions and the cluster divisions. The rows of the column linkage table represent the actual category divisions, the columns of the column linkage table represent the cluster markers of the cluster divisions, M_{ij} represents the number of overlapping instances, and a_i and b_j are the marginal sums of the column linkage table. The columnar table is shown in Table 3.

Table 3. Column linkage table

	Cluster 1	Cluster 2	$\cdots$	Cluster s
Category 1	M_{11}	M_{21}	$\cdots$	M_{1s}
Category 2	M_{12}	M_{22}	$\cdots$	M_{2s}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Category r	M_{1r}	M_{2r}	$\cdots$	M_{rs}

AMI is based on Shannon's information theory and uses mutual information to compare different clusters. Its definition is shown below:

$$AMI(T,R) = \frac{MI(T,R) - E\{MI(T,R)\}}{\max\{H(T),H(R)\} - E\{MI(T,R)\}} \tag{21}$$

where T is the actual label of the data sample and R is the clustering result label. $MI(T,R)$ denotes the mutual information between T and R; $H(T)$ and $H(R)$ denote the Shannon entropy of the category labels of T and R, respectively. $E\{MI(T,R)\}$ denotes the mathematical expectation of $MI(T,R)$.

4.3. **Experimental results and analysis.** Table 4 displays the results of four different clustering algorithms on eight datasets using the ACC, ARI, and AMI. From the values of the evaluation metrics obtained in Table 4, it is known that it is feasible and effective to optimize the parameter *dc* parameter of the DPC algorithm by tuna swarms. On the synthetic dataset, among the proposed algorithms, Aggregation and D31 were higher than the other comparative algorithms in all three clustering evaluation metrics, and ITSO-DPC had the best clustering performance. On the Jain dataset, the proposed ITSO-DPC algorithm is second only to DBSCAN in terms of clustering metrics and outperforms the other comparative algorithms. In terms of ACC metrics, ITSO-DPC achieved 92.2%, succeeding only to DBSCAN's 96.5%, but it is higher than DPC's 86.1% and FKNN-DPC's 65.1%. On the R15 and Spiral datasets, the four algorithms work admirably and similarly. In the real dataset, taking the ACC index of the Seeds dataset as an example, the proposed ITSO-DPC has reached 89.1%, which is second only to 89.5% of FKNN-DPC, but it is about 10% better than the original DPC and 25.3% better than DBSCAN. Taking the ACC metric of Iris as an example, the proposed algorithm achieved an accuracy of 95.3%, which is better than DPC by about 12%, FKNN-DPC by about 4.6%, and significantly better than DBSCAN by about 22.6%. On the Ecoli dataset, taking the ARI metric as an example, the proposed ITSO-DPC algorithm achieved a clustering accuracy of 68.6%, which is significantly better than the DPC algorithm by about 34.9% and higher than FKNN-DPC by about 13.1% and better than DBSCAN by about 16.1%. This demonstrates that the proposed ITSO-DPC outperforms all other comparative algorithms on such real datasets.

Figures 3-5 show comparison charts of the ACC, ARI, and AMI clustering evaluation metrics for each of the four algorithms for each dataset to make them easier to observe. Despite ACC, AMI, and ARI having different definitions and the clustering algorithms generating different results, their average rankings are fairly similar. On the Jain dataset, ITSO is ranked higher than DPC and FKNN-DPC but not as high as DBSCAN. The DPC has the lowest average rating for the Ecoli dataset, whereas ITSO-DPC has the best. The clustering metric values of the ITSO-DPC algorithm have a significant improvement on both the artificial and UCI datasets. Compared with the original DPC, the ITSO-DPC algorithm reduces the drawback of manually selecting parameter *dc* and its clustering results are much better than the DPC.

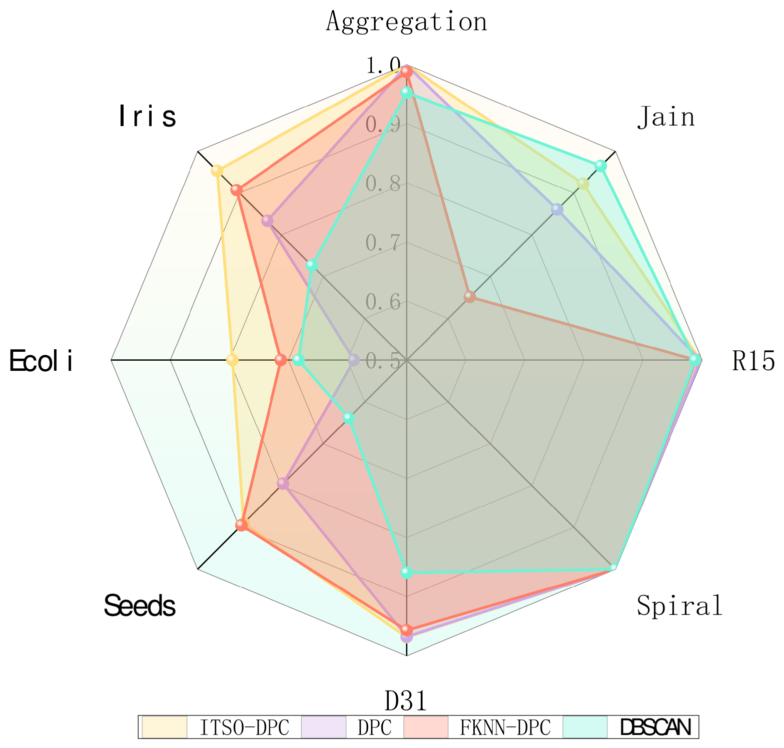


FIGURE 3. Comparison of ACC metrics for the four algorithms

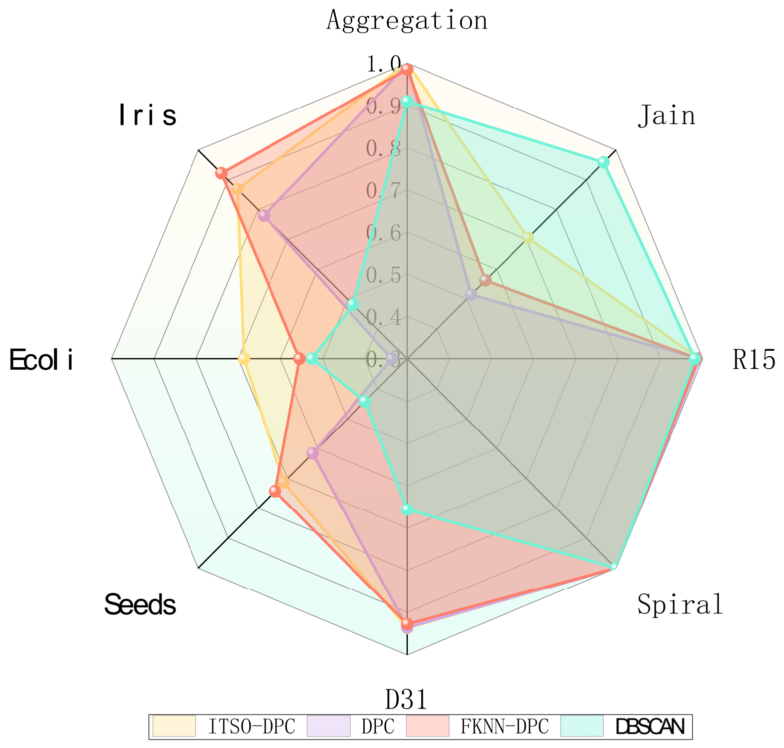


FIGURE 4. Comparison of ARI metrics for the four algorithms

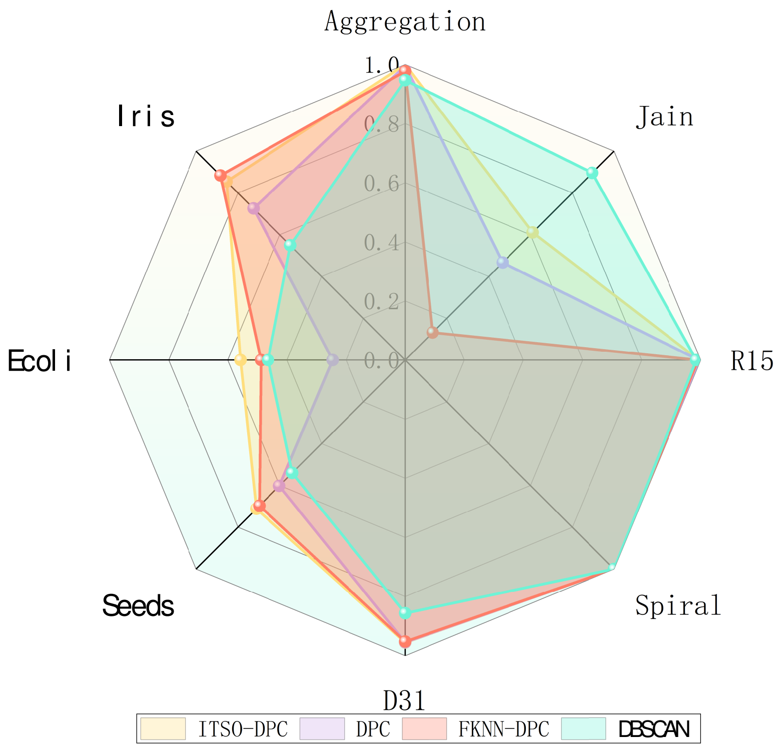


FIGURE 5. Comparison of AMI metric for the four algorithms

TABLE 4. Monitoring data query contract algorithm

	ITSO-DPC			DPC			FKNN-DPC			DBSCAN		
	ACC	ARI	AMI	ACC	ARI	AMI	ACC	ARI	AMI	ACC	ARI	AMI
Aggregation	1.0000	1.0000	1.0000	0.9987	0.9978	0.9957	0.9876	0.9855	0.9775	0.9530	0.9097	0.9468
Jain	0.9223	0.7055	0.6103	0.8606	0.5146	0.4667	0.6514	0.562	0.1318	0.9651	0.9583	0.8955
R15	0.9967	0.9928	0.9938	0.9967	0.9928	0.9938	0.9898	0.9892	0.9907	0.9883	0.9819	0.9846
Spiral	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
D31	0.9687	0.9370	0.9558	0.9674	0.9354	0.9547	0.9568	0.9275	0.9522	0.8593	0.6563	0.8564
Seeds	0.8905	0.7132	0.7119	0.7952	0.6163	0.6018	0.8952	0.7422	0.6971	0.6381	0.4416	0.5402
Ecoli	0.7946	0.6863	0.5567	0.5892	0.3369	0.2453	0.7126	0.5547	0.4876	0.6829	0.5255	0.4636
Iris	0.9533	0.8682	0.8541	0.8333	0.7794	0.7257	0.9067	0.9222	0.8831	0.7267	0.4817	0.5491

5. **Conclusion.** Considering the limitations of the DPC in the cut-off distance taking, this paper proposes a density peak clustering algorithm (ITSO-DPC) incorporating tuna swarm optimization. Firstly, the algorithm introduces a mixture of Piecewise chaotic map and cubic map fused with a reverse refraction mechanism to initialize the population, updates the parameters of the spiral foraging phase based on an adaptive inertia factor, and evaluates its convergence with eight benchmark test functions. Stronger global search capabilities and quicker convergence speed are advantages of the modified TSO algorithm. Secondly, in order to automatically determine the cluster centers, the metric γ was used to consider computational variables including ρ and δ. Finally, this paper incorporates the stronger search performance of ITSO to optimize the dc value in the DPC algorithm. Experimental results on synthetic datasets and UCI standard datasets are shown that the algorithm has a better adaptive capability of the cut-off distance parameter on datasets

of different dimensions and scales. However, the ITSO-DPC algorithm proposed in this paper still has some detection challenges in dividing the data points around the boundary and the data points with lower-density peaks. The problem in this direction needs further research.

Acknowledgment. The authors would like to acknowledge the support of the National Science Fund subsidized project under Grant (No. 62273290, No.61872126).

REFERENCES

[1] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16–24, 2010.

[2] M. D. Hossain, and D. Chen, "Segmentation for Object-Based Image Analysis (OBIA): A review of algorithms and challenges from remote sensing perspective," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 115–134, 2019.

[3] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[4] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence*, vol. 110, pp. 104743, 2022.

[5] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[6] A. L. H. P. Shaik, M. K. Manoharan, A. K. Pani, R. R. Avala, and C.-M. Chen, "Gaussian Mutation–Spider Monkey Optimization (GM-SMO) Model for Remote Sensing Scene Classification," *Remote Sensing*, vol. 14, no. 24, pp. 6279, 2022.

[7] S.-S. Yu, S.-W. Chu, C.-M. Wang, Y.-K. Chan, and T.-C. Chang, "Two improved k-means algorithms," *Applied Soft Computing*, vol. 68, pp. 747–755, 2018.

[8] A. K. Varshney, P. K. Muhuri, and Q. D. Lohani, "PIFHC: The probabilistic intuitionistic fuzzy hierarchical clustering algorithm," *Applied Soft Computing*, vol. 120, pp. 108584, 2022.

[9] M. Li, X. Bi, L. Wang, and X. Han, "A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm," *Computer Communications*, vol. 167, pp. 75–84, 2021.

[10] C. Hireche, H. Drias, and H. Moulai, "Grid based clustering for satisfiability solving," *Applied Soft Computing*, vol. 88, pp. 106069, 2020.

[11] G. Punj, and D. W. Stewart, "Cluster analysis in marketing research: Review and suggestions for application," *Journal of Marketing Research*, vol. 20, no. 2, pp. 134–148, 1983.

[12] S. H. Kwon, "Threshold selection based on cluster analysis," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1045–1050, 2004.

[13] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[14] F. Murtagh, and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[15] A. Rodriguez, and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[16] C.-M. Chen, S. Lv, J. Ning, Y.-K. Chan, and J. M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, no. 1, pp. 124, 2023.

[17] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018.

[18] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, pp. 1977, 2023.

[19] L. Xie, T. Han, H. Zhou, Z.-R. Zhang, B. Han, and A. Tang, "Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–22, 2021.

[20] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, 2016.

[21] Y. Chen, X. Hu, W. Fan, L. Shen, Z. Zhang, X. Liu, J. Du, H. Li, Y. Chen, and H. Li, "Fast density peak clustering for large scale data based on kNN," *Knowledge-Based Systems*, vol. 187, no. C, pp. 104824, 2020.

[22] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Information Sciences*, vol. 450, pp. 200–226, 2018.

[23] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

[24] J. Jiang, D. Hao, Y. Chen, M. Parmar, and K. Li, "GDPC: Gravitation-based density peaks clustering algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 502, pp. 345–355, 2018.

[25] T. Gao, D. Chen, Y. Tang, B. Du, R. Ranjan, A. Y. Zomaya, and S. Dustdar, "Adaptive density peaks clustering: Towards exploratory EEG analysis," *Knowledge-Based Systems*, vol. 240, pp. 108123, 2022.

[26] J. Zhao, J. Tang, A. Shi, T. Fan, and L. Xu, "Improved density peaks clustering based on firefly algorithm," *International Journal of Bio-Inspired Computation*, vol. 15, no. 1, pp. 24–42, 2020.

[27] R. Zhou, Q. Liu, Z. Xu, L. Wang, and X. Han, "Improved Fruit Fly Optimization Algorithm-based density peak clustering and its applications," *Tehnički vjesnik*, vol. 24, no. 2, pp. 473–480, 2017.

[28] H. Zhu, H. He, J. Xu, Q. Fang, and W. Wang, "Medical image segmentation using fruit fly optimization and density peaks clustering," *Computational and Mathematical Methods in Medicine*, vol. 2018, 3052852, 2018.

[29] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06).* IEEE, 2005, pp. 695–701.

[30] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.

[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[32] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[33] A. A. Abd El-Mageed, A. G. Gad, K. M. Sallam, K. Munasinghe, and A. A. Abohany, "Improved binary adaptive wind driven optimization algorithm-based dimensionality reduction for supervised classification," *Computers & Industrial Engineering*, vol. 167, pp. 107904, 2022.