https://doi.org/10.1093/comjnl/bxab206

Customer Segmentation Using K-Means Clustering and the Hybrid Particle Swarm Optimization Algorithm

Yue Li[1], Jianfang Qi[1], Xiaoquan Chu[1] and Weisong Mu[1,2,*]

[1]*College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China*
[2]*Key Laboratory of Viticulture and Enology, Ministry of Agriculture, Beijing 100083, China*
**Corresponding author: wsmu@cau.edu.cn*

In a competitive market, it is of great significance to divide customer groups to develop customer-centered personalized products. In this paper, we propose a customer segmentation method based on the K-means algorithm and the improved particle swarm optimization (PSO) algorithm. As the PSO algorithm easily falls into local extremum, the improved hybrid particle swarm optimization (IHPSO) algorithm is proposed to improve optimization accuracy. The full factorial design is used to determine the optimal parameter combination; the roulette operator is used to select excellent particles; then, the selected particles are crossed according to their adaptive crossover probabilities; when the population falls into a local optimum, the particles are mutated according to their adaptive mutation probabilities. Aimed at the K-means' sensitivity to selecting the initial cluster centers, IHPSO is used to optimize the cluster centers (IHPSO-KM). We compare IHPSO with the PSO, LDWPSO, GA, GA-PSO and ALPSO algorithms on nine benchmark functions. We also conduct comparative experiments to compare IHPSO-KM with several conventional and state-of-the-art approaches on five UCI datasets. All results show that the two proposed methods outperform existing models. Finally, IHPSO-KM is applied in customer segmentation. The experimental results also prove the rationality and applicability of IHPSO-KM for customer segmentation.

Keywords: K-means clustering algorithm; particle swarm optimization algorithm; hybrid mechanism; cluster analysis; customer segmentation

Received 21 December 2020; Revised 11 November 2021; Editorial Decision 6 December 2021
Handling editor: Yannis Manolopoulos

1. INTRODUCTION

With the increase in market competition, enterprises increasingly highlight the construction of customer-centric management systems to improve enterprises' competitive advantages. Customer segmentation refers to dividing a customer group into different groups according to their consumption behaviors to provide customers with personalized services and enterprises with professional marketing strategies [1]. Traditional customer segmentation methods mainly include empirical segmentation and mathematical statistics. However, the empirical segmentation method has strong subjectivity, and the mathematical statistics method largely depends on segmentation criteria [2]. Cluster analysis is an important technique in data mining and exploratory data analysis and has been widely used in the field of customer segmentation. It can classify datasets and extract valuable information from the characteristics of data objects [3]. Therefore, cluster analysis is used to divide customers according to their consumption values to ensure that scientific and reasonable services are provided for different customer groups.

Clustering assembles objects into groups such that objects within the same group are similar and objects within different groups are dissimilar [4]. Clustering algorithms can generally be divided into partition-based, density-based, model-based

and hierarchical-based [5]. The K-means algorithm is a classical partition-based clustering algorithm. K-means has many advantages, such as it is easy to understand, convergences quickly and has a strong local search ability. It is used widely in statistics, marketing, customer segmentation and so on [6, 7]. However, the K-means algorithm is sensitive to the selection of the initial cluster centers, which causes it to converge to local optima. Once the local optimal solution is produced, K-means has no ability to jump out it [8]. Therefore, selecting reasonable initial cluster centers is of great significance in reducing the volatility of clustering results and obtaining a higher clustering accuracy.

The present study aims to develop a new customer segmentation method to divide customer groups in the grape market in China. To improve the accuracy of customer segmentation, we propose a K-means clustering algorithm based on the improved hybrid particle swarm optimization (IHPSO) algorithm. For the IHPSO algorithm, a full factorial design is used to analyze the timing of fusion, and the selection, crossover and mutation operations are introduced and redesigned into the PSO algorithm to improve the optimization accuracy of PSO. The IHPSO algorithm is then used to optimize the original K-means cluster centers and thereby prevent K-means from relying on the initial values. Our research proves the practicality of the IHPSO-based K-means (IHPSO-KM) algorithm for customer segmentation.

The contributions of this paper are as follows: (i) a new IHPSO algorithm is proposed to improve the optimization performance of PSO and (ii) a new IHPSO-KM algorithm is proposed to prevent the overdependence of the K-means algorithm on initial cluster centers. The optimization performance of the IHPSO algorithm is compared based on nine benchmark test functions. The clustering performance of the IHPSO-KM algorithm is analyzed on five UCI datasets. Finally, the IHPSO-KM algorithm is applied to customer segmentation. The experimental results show that the two proposed methods achieve higher accuracies than several conventional and state-of-the-art methods. The segmentation results obtained by the IHPSO-KM algorithm are reasonable and convenient for enterprises to use to provide personalized service strategies for different customer groups.

The rest of this paper is organized as follows: Section 2 discusses the review of the related work. Section 3 describes our methods in detail. Section 4 presents the experimental results and provides the analysis. Section 5 offers the conclusions and future work from this research.

2. RELATED WORK

In this section, we mainly discuss the K-means clustering algorithm and the PSO algorithm. The focus is on the optimization of the initial centers of K-means and the improvement of the PSO algorithm with a genetic mechanism.

2.1. K-means clustering algorithm

The data are divided into clusters based on the minimized error function [9]. The mean of all the data in cluster E_j represents that the cluster center e_j. $dist(o, e_j)$ represents the Euclidean distance between the numerical data and center point. The square sum of errors (*SSE*) is used as the evaluation function to measure the clustering quality.

$$SSE = \sum_{j=1}^{k} \sum_{o \in E_j} dist\left(o, e_j\right) \tag{1}$$

$$e_j = \frac{1}{n_j} \sum_{o \in E_j} o \tag{2}$$

where o is the sample data. k is the number of clusters. n_j is the number of data points in the jth cluster E_j. *SSE* represents the compactness of clusters. The smaller the *SSE* is, the better the clustering effect.

Currently, the research of K-means on the selection of initial cluster centers mainly includes distance- and density-based optimization methods [10]. Although the distance-based clustering algorithm costs less time, it is too sensitive to outliers in data. While the density-based clustering algorithm can more accurately reflect the data distribution, it takes considerable time. Aimed at random initial centers, many methods, such as K-means++ and K-medoids, have been proposed to solve this problem [11]. The K-means++ algorithm selects data that are as far away as possible as the initial cluster centers [12]. This approach reduces the sensitivity to initial centers and improves the clustering accuracy compared with the traditional methods of randomly selecting initial centers. The K-medoids algorithm determines the cluster centers using the median of data, which reduces the influence of outliers on the clustering results and effectively solves the problem of the traditional K-means algorithm easily falling into local optimum [13]. However, this approach is not suitable for large-scale data clustering.

Although several studies have been conducted on the selection of initial cluster centers in K-means and have obtained better clustering results, they do not solve the problem of the poor global search ability of K-means [14]. Consequently, the K-means algorithm may still fall into the local optimal solution.

2.2. Particle swarm optimization algorithm

Recently, the emergence of swarm optimization algorithms, including ant colony, artificial bee colony, firefly and particle swarm optimization (PSO) algorithms, has solved the problem of K-means being sensitive to the initial cluster centers [15, 16]. Swarm intelligence algorithms have relatively strong global convergence and robustness and have been widely applied in various optimization fields, including data mining and machine learning [17, 18]. As a typical swarm intelligence optimization algorithm, the PSO algorithm originated from exchanging

and sharing information in the process of searching for food among bird individuals. Compared with other evolutionary algorithms, PSO has the advantages of simple implementation, fewer parameters and fast convergence [19–21]. The definition of PSO is as follows.

Suppose there are N particles in the D-dimensional space, the position of the ith particle is a D-dimensional vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, the velocity of the ith particle is also a D-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, the best position experienced by the ith particle is called the individual extremum $p\text{best}_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and the best position experienced by the population is called the global extremum $g\text{best} = (g_1, g_2, \ldots, g_D)$. The velocity and position update formulas of the ith particle with inertia weight are as follows:

$$v_{id}^t = wv_{id}^{t-1} + c_1 r_1 \left(p\text{best}_{id}^{t-1} - x_{id}^{t-1}\right) + c_2 r_2 \left(g\text{best}_d^{t-1} - x_{id}^{t-1}\right) \quad (3)$$

$$x_{id}^t = x_{id}^{t-1} + v_{id}^t \quad (4)$$

,where v_{id}^t is the velocity of the dth dimension of the ith particle at the tth iteration. x_{id}^t is the position of the dth dimension of the ith particle at the tth iteration. $1 \leq i \leq N$, N is the number of particles. w is the inertia weight. c_1 and c_2 are the learning factors. r_1 and r_2 are the random numbers, $r_1, r_2 \in [0,1]$.

After PSO was used to optimize the clustering algorithms proposed by Omran [22], researchers began to combine PSO with K-means for cluster analysis. For example, Zhang *et al.* [23] proposed an improved clustering algorithm based on PSO with dynamic crossover. It not only steadily acquires clustering results but also eliminates the dependence on the initial centers of K-means. Gao *et al.* [24] proposed a PSO-based K-means algorithm, which used Gaussian estimation to assist PSO in updating the population information and adopted Levy flight to escape from the local optimum. The clustering results show that the proposed method has better convergence. These works reduce the probability of K-means falling into local optima by using the global search ability of PSO. The PSO-optimized clustering algorithms can improve the clustering performance to a certain extent, but PSO is also prone to premature convergence and cannot guarantee convergence to the global optimum [25, 26].

At present, the introduction of intelligent operators has become the focus of PSO improvement [27, 28]. These methods combine PSO with other intelligent algorithms so that the advantages of PSO and other intelligent algorithms can be integrated to improve the optimization performance of PSO. PSO has a fast convergence speed but can prematurely converge, resulting in a low optimization accuracy. The genetic algorithm (GA) has strong search accuracy and robustness and has advantages in global optimization [29, 30]. Because PSO and GA are highly complementary, to overcome PSO's tendency to fall into local optima, GA is used to improve the optimization ability of PSO [31, 32].

For example, Gandelli *et al.* [33] proposed a new hybrid strategy between GA and PSO. In each iteration, the population evolves GA and PSO in a certain proportion. The purpose is to exploit the uniqueness of PSO and GA in the most effective way. Fu *et al.* [34] proposed a hybrid optimization algorithm based on chaos GA combined with PSO. By applying the experience of PSO, the information sharing of GA and traversing the pathway of chaos, the adaptive switching of two algorithms is implemented, which can quickly obtain the global optimal solution. Yazdanjue *et al.* [35] proposed an evolutionary algorithm for K-anonymity in social networks based on a clustering approach. To minimize the normalized structural information loss (NSIL) value provided by GA optimization while preserving the high convergence rate obtained from the PSO algorithm, hybrid solutions based on the GA and PSO algorithms are obtained. The simulation results demonstrate the efficiency of the proposed model to balance the NSIL and the algorithm's convergence rate. The hybrid GA-based PSO algorithm adds crossover and mutation operations in the evolution of PSO, which significantly improves the PSO's global optimization ability [36].

Based on the above analysis, we can make full use of the local search capability of K-means and the global optimization ability of the improved PSO algorithm to obtain better clustering results.

3. PROPOSED METHODS

3.1. IHPSO algorithm

PSO is prone to premature convergence in the iterative process, resulting in a low optimization accuracy. How to prevent the population from falling into local convergence and how to jump out of the local optimum in time are the purposes of PSO improvement. PSO with a genetic mechanism uses PSO to quickly converge to the global optimum and uses the natural selection of GA to enhance the optimization accuracy.

In this paper, the initial population is randomly generated. Using real number coding to encode particles can ensure that the initial population is in the solution space [37]. When PSO searches for the global optimum, there is no need to integrate genetic operations. In this way, the advantages of PSO are used to quickly approach the global optimum. To make the particles close to the optimal solution and to not overly destroy the superiority of the original particles, the following improved genetic methods are used to optimize PSO.

3.1.1. *Selection—survival of the fittest*

A particle with high fitness has a high probability of being inherited in the next generation. The roulette operator is used to select half of the particles with high fitness. The purpose is to ensure that good particles are not eliminated and improve the convergence of the population. The specific operations are as follows.

(1) Calculate the fitness of each particle $f(x_i)$ and the fitness sum of all particles $f(s)$, $1 \leq i \leq N$, where N is the number of particles.

(2) Calculate the relative probability of each particle. $p(x_i) = f(x_i)/f(s)$.
(3) Calculate the cumulative probability of each particle. $q(x_i)=\sum_{j=1}^{i} p(x_j)$.
(4) If $q(x_{i-1}) < r \leq q(x_i)$, particle i is selected to enter the next generation, where r is a random number, $r\in[0,1]$.
(5) Repeat step (4) to generate half of the particles.

The average fitness of the population is continuously improved according to the sorting of the fitness of particles, which can enhance the local search ability and convergence of the population. If the population has more particles with high fitness, although the population converges faster, it may fall into local extrema. If the population has more particles with low fitness, although the population can maintain diversity, the convergence speed will be impacted. Therefore, improved crossover and mutation operations are used to prevent the above problems.

3.1.2. *Crossover—prevention of premature convergence*

(1) Adaptive crossover probability

PSO can easily fall into a local optimum in the late iteration, but the selected particles are crossed according to their adaptive crossover probabilities to form a new population, which prevents premature convergence and guarantees the stability of the population. The crossover probability (P_c) determines the evolution speed of the population. If the value of P_c is too large, the excellent genetic information of the population will be destroyed. If the value of P_c is too small, the search speed of the population will be slow [38]. Generally, the value of P_c is between 0.25 and 1 [39, 40]. In the early stage of the iteration, the evolution speed of the population should be increased to improve the search ability. In late iterations, P_c should be reduced to retain the superiority of the population. Therefore, an adaptive crossover probability is designed as shown in Equation (5)

$$p_{ci} = 0.25 + \frac{f(x_i) - f(x_i)_{\min}}{f(x_i)_{\max} - f(x_i)_{\min}} \cdot \frac{0.75}{1 + ((t - T')/281)^7}, \quad T' < t < T_{\max} \tag{5}$$

where p_{ci} is the crossover probability of the ith particle. $f(x_i)_{\max}$ is the individual maximum fitness of the ith particle. $f(x_i)_{\min}$ is the individual minimum fitness of the ith particle. t is the current number of iterations. T' is the fusion number of iterations. $T_{\max}$ is the maximum number of iterations.

A particle with high fitness has a greater crossover probability, which can effectively prevent the population from falling into a local optima. Figure 1a shows the crossover probability of a particle, and the fitted curve is shown in Figure 1b. The overall change declines with the number of iterations, which conforms to the above analysis.

(2) Crossover operator

The crossover operator is conducive to finding excellent particles that would lead to the population quickly converging to the optimal solution. The generation of a new particle is affected by its own and individual best. To improve the convergence of PSO, the selected particles are crossed with their individual best positions. The real single-point crossover formula is shown in Equation (6)

$$x_i' = \alpha x_i + (1 - \alpha)\, p\text{best}_i, \ r < p_{ci}, \tag{6}$$

where $p\text{best}_i$ is the individual best position of the ith particle. α is a random crossover point. r is a random number, $r \in [0,1]$.

3.1.3. *Mutation—reprocessing after premature convergence*

(1) Adaptive mutation probability.

Even through selection and crossover operations, the population may still exists premature convergence. When a particle in the population falls into a local extremum, it will be difficult for it to jump out [41, 42]. The particles mutate according to their adaptive mutation probabilities, which effectively maintain population diversity. The mutation probability (P_m) determines the optimization accuracy of the population. If the value of P_m is too large, good individuals will be destroyed. If the value of P_m is too small, individuals cannot mutate into better solutions. Generally, the value of P_m is between 0.001 and 0.1 [43]. In the early stage of the iteration, the possibility of individual mutation should be reduced. In late iterations, P_m should be increased to encourage the generation of new individuals. Therefore, an adaptive mutation probability is designed as shown in Equation (7)

$$p_{mi} = \frac{f(x_i) - f(x_i)_{\min}}{f(x_i)_{\max} - f(x_i)_{\min}} \cdot \left(0.101 - \frac{0.1}{1 + ((k - K')/240)^5}\right) \quad T' < t < T_{\max} \tag{7}$$

where p_{mi} is the mutation probability of the ith particle. A particle with high fitness has a greater mutation probability, which enables the population to jump out of the local optimum in time. Figure 2a shows the mutation probability of a particle, and the fitted curve is shown in Figure 2b. The overall change increases with the number of iterations, which conforms to the above analysis.

(2) Mutation operator.

The mutation operator is an indispensable auxiliary method in genetic mechanism and helps the population jump out of the local optimum in time. To improve the optimization accuracy of PSO, the real bitwise mutation formula is shown in Equation (8)

$$x_{id} = (X_{\max} - X_{\min})\, r + X_{\min}, r < p_{mi} \tag{8}$$

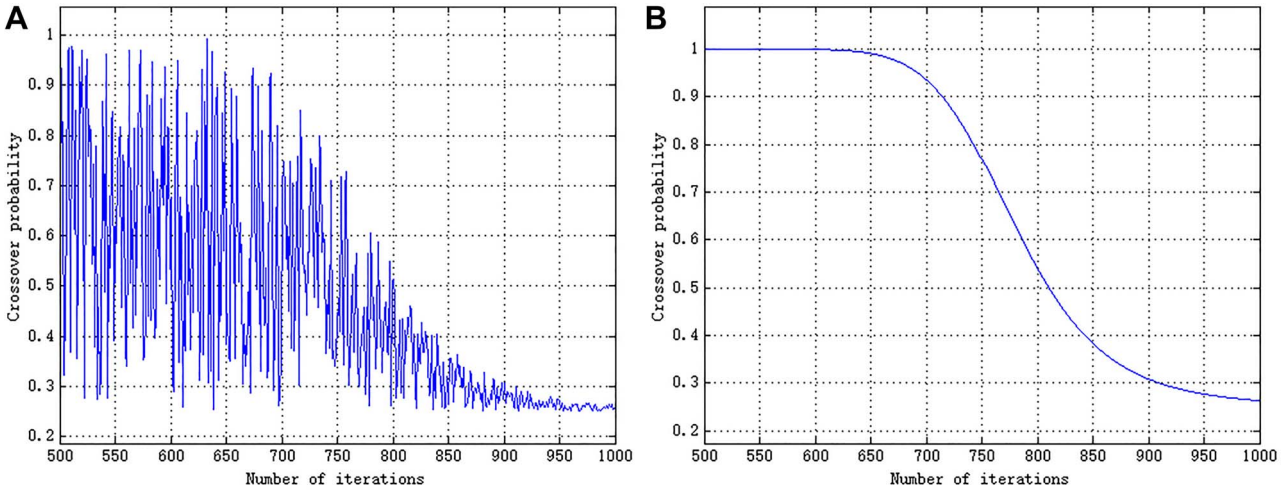

FIGURE 1. Crossover probability adjustment curve. (A) Crossover probability of a particle. (B) Best fit for the crossover probability of a particle.

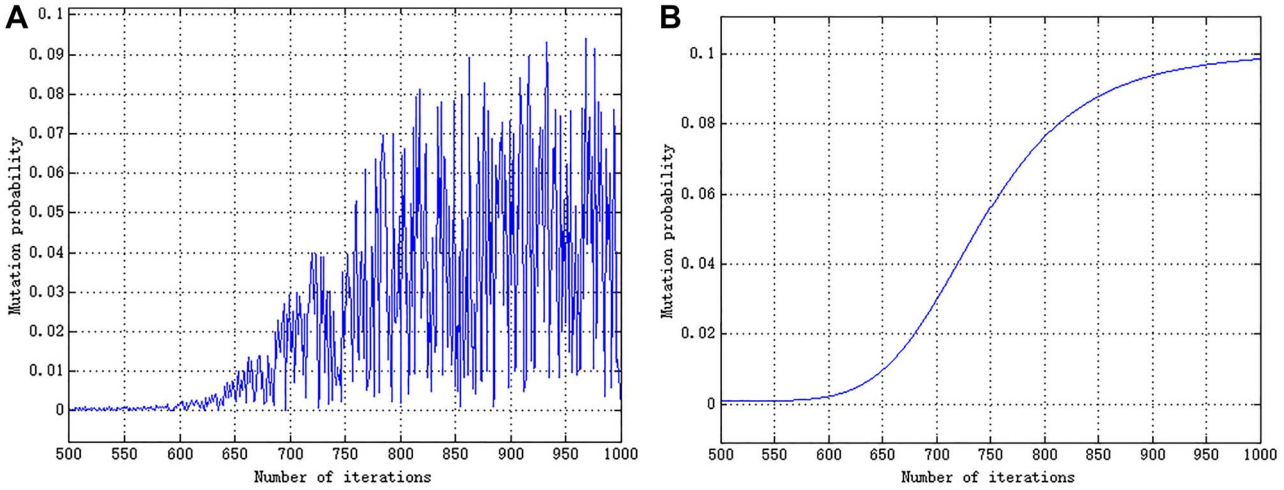

FIGURE 2. Mutation probability adjustment curve. (A) Mutation probability of a particle. (B) Best fit for the mutation probability of a particle.

where $X_{\max}$ and $X_{\min}$ represent the maximum and minimum positions in the solution space, respectively. r is a random number, $r \in [0,1]$.

3.1.4. Parameter update

The inertia weight plays an important role in balancing the global and local search capabilities of the population. When the particles are far from the global optimum, the inertia weight should be increased for the global search, and vice versa [44, 45]. The PSO algorithm with linearly decreasing inertia weight (LDWPSO, Equation 9) has the following problems: PSO has a strong global search ability at the beginning of iteration, but it does not last for a long time, which makes it difficult to find the global optimum. Then, as the inertia weight decreases, the aggregation of particles can cause local minima [46].

To balance the global and local search capabilities of PSO, in early iterations, the inertia weight has a larger value, which it retains for a certain period of time to enhance the global search ability; in the later stage of iteration, the inertia weight keeps a small value for a certain period of time, which can improve the local search performance. In this paper, an adaptive decreasing inertia weight is proposed (Equation 10), and the change curves of linearly decreasing and adaptive decreasing inertia weights are shown in Figure 3

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})\, t}{T_{\max}} \tag{9}$$

$$w = (w_{\max} - w_{\min}) \exp\left(-a\left(\frac{t}{T_{\max}}\right)^{b}\right) + w_{\min} \tag{10}$$

where a and b are the positive constants. $w_{\max}$ and $w_{\min}$ represent the maximum and minimum inertia weights, respectively.

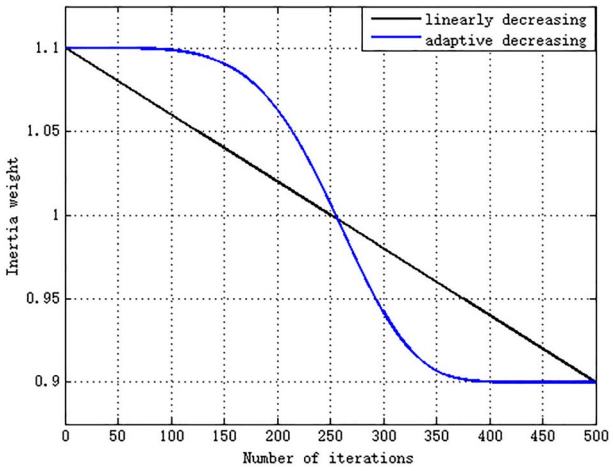

FIGURE 3. The change process of inertial weight.

The proposed IHPSO algorithm evolves PSO in the early stage of the iteration and then integrates the genetic mechanism when the integration time is reached. However, the positions of particles may exceed the boundaries after the selection, crossover and mutation operations. Random perturbation is helpful for the population to find a better solution

$$v_{id} = \begin{cases} V_{\max} - \text{rand}, & v_{id} > V_{\max} \\ V_{\min} + \text{rand}, & v_{id} < V_{\min} \end{cases} \tag{11}$$

$$x_{id} = \begin{cases} X_{\max} - \text{rand}, & x_{id} > X_{\max} \\ X_{\min} + \text{rand}, & x_{id} < X_{\min} \end{cases} \tag{12}$$

where $V_{\max}$ and $V_{\min}$ represent the maximum and minimum velocities in the solution space, respectively.

3.2. IHPSO-KM clustering algorithm

For the IHPSO-KM clustering algorithm, IHPSO with strong global search ability is used to optimize the initial cluster centers of K-means to improve the clustering accuracy. Then, the nearest-neighbor principle is used to determine the final clustering results. The structure of each particle consists of the position, velocity and fitness. The position of each particle $(\bar{x}_{e_1}, \bar{x}_{e_2}, \ldots, \bar{x}_{e_k})$ is composed of k cluster centers. $\bar{x}_{e_j} (j = 1, 2, \ldots, k)$ is the cluster center of the jth cluster. The fitness of each particle can be expressed as Equation (13)

$$\text{fit} = a \cdot \sum_{j=1}^{k} \sum_{o \in E_j} dist\left(o, \bar{x}_{e_j}\right) \tag{13}$$

where a is a positive constant. The data object o and cluster centers satisfy Equation (14)

$$\left\| o, \bar{x}_{e_j} \right\| = \min_{j=1,2,\ldots,k} \left\| o, \bar{x}_{e_j} \right\| \tag{14}$$

.

In this study, IHPSO is used for a global search, and K-means is used for a local search. First, the selection, crossover and mutation operations are used to reconstruct PSO. The particles perform the optimization process in IHPSO, as shown in Figure 4. After the population converges, the global optimum obtained by IHPSO is the cluster centers of K-means. After forwarding IHPSO's output to K-means, particles are reinitialized and clustering is performed again. The flow of IHPSO-KM is shown in Figure 5.

4. DISCUSSION AND RESULTS

4.1. Performance analysis of IHPSO

4.1.1. *Basic test functions*

To evaluate the optimization performance of the IHPSO algorithm, nine benchmark functions with optimization problem 0 are used for simulation experiments, including many local minima (Rastrigin, Griewank, Ackley), bowl-shaped (Sphere, Sum squares, Rotated hyper-ellipsoid), valley-shaped (Rosenbrock, Dixon-price) and plate-shaped (Zakharov) functions [47]. The descriptions of the nine test functions are shown in Table 1. The relevant parameters of PSO and GA are set as follows: the population size is set to 400, the dimension number of the particle is 30, the inertia weight is 1, the learning factors are 2, the positions of particles are randomly initialized between [−10, 10], the crossover probability is 0.9, the mutation probability is 0.001 and the maximum number of iterations is 1000.

The MATLAB environment is used to compile programming for solving and simulating the optimization processes of the comparison algorithms.

4.1.2. *Integration time analysis*

For this section, we are interested in the tuning parameters primarily as experiment design factors. To determine the best integration time of the genetic mechanism (number of iterations), usually, it is more appropriate to use the design of experiment (DOE) to obtain accurate test results with less test times and lower trial costs. DOE full factorial design is a structured data collection and analysis method, which consists of a crossing of all levels of all factors [48]. Thus, a full factorial design is used to systematically explore the effect of parameters on the objective values obtained and select the best parameter values.

The average optimal fitness f_{avg} and the mean standard deviation f_{std} of the test results can reflect the accuracy and stability of the IHPSO algorithm. In this paper, the fusion timing_avg and the fusion timing_std are used as the factor

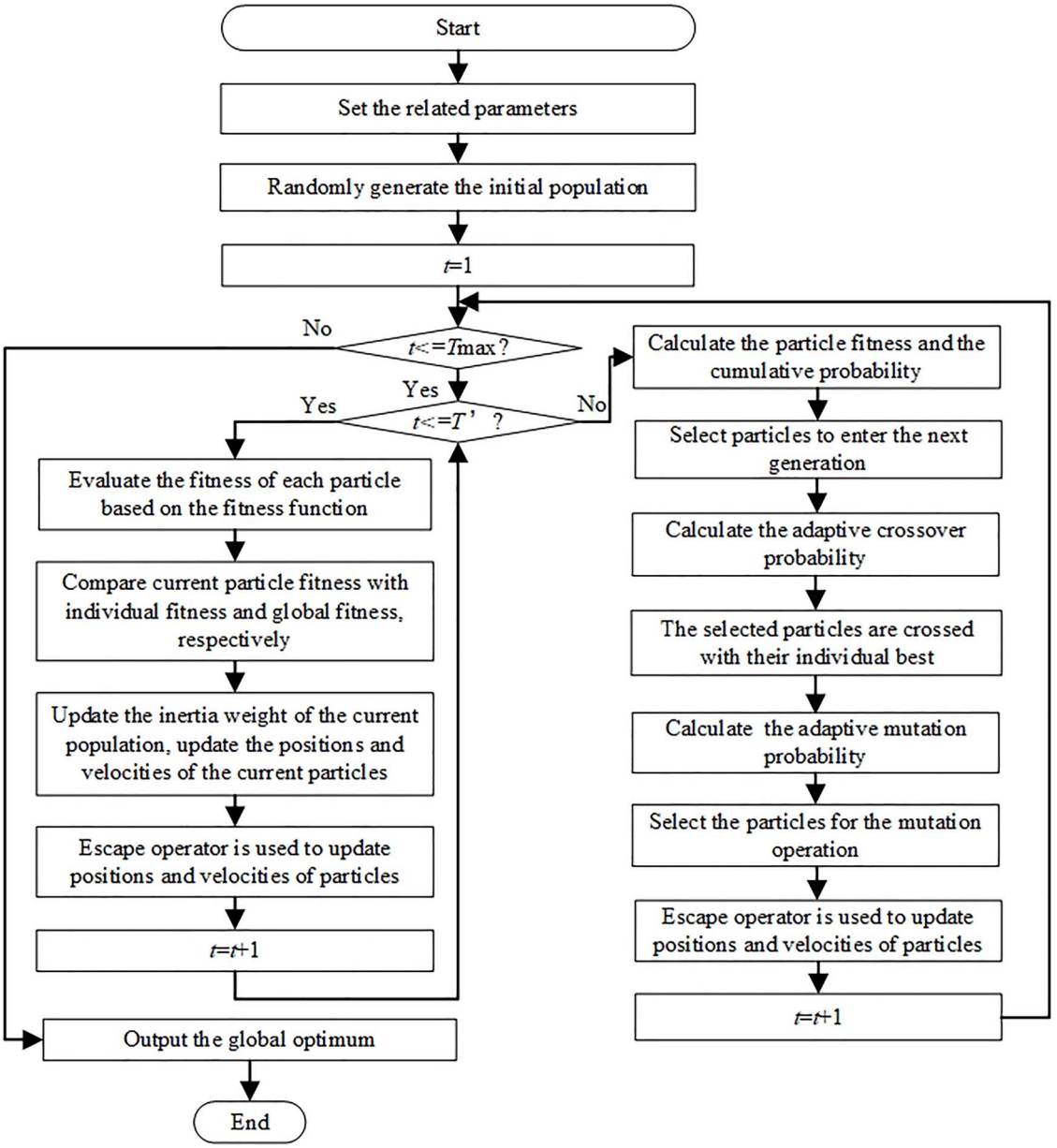

FIGURE 4. Flow chart of IHPSO.

predictors, and the sum of the fusion timing_avg and the fusion timing_std is the response variable. The nine test functions are independently tested 10 times in the same experimental environment, and the full factorial design with two factors and nine levels is shown in Table 2.

The MINITAB 17 software is used to analyze the test data using a full factorial design. The main and interactive effects of factors for the optimal parameter combination are shown in Figure 6. The optimal solution is obtained through the response optimizer in the DOE. The smaller the expected optimization value, the higher the optimization accuracy of the IHPSO algorithm. Therefore, the DOE optimizer is set to the minimum value, and the optimization effect is shown in Figure 7.

Figure 6a shows the effects of changes in the levels of the two factors on the response variable, and it can be seen that both factors are significant. When the number of iterations is 500, the fusion timing_avg and the fusion timing_std reach the minimum and the fusion timing_avg factor has a greater effect than the fusion timing_std factor. Figure 6b illustrates that the change of one factor is affected by the levels of another factor, and it can be seen that there is no interaction between the two

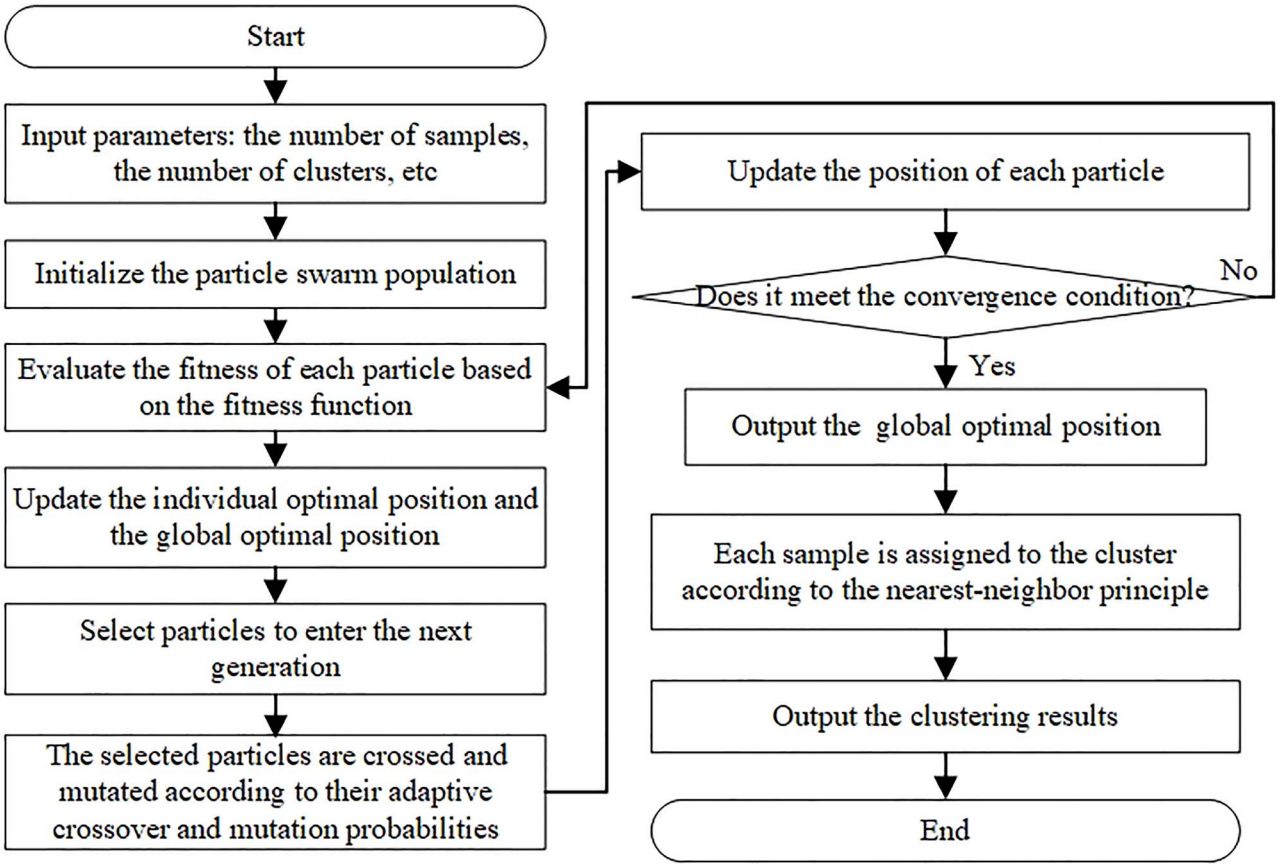

FIGURE 5. Flow chart of IHPSO-KM.

TABLE 1. Basic multidimensional test functions.

Name	Expression
$f_1(x)_{\min}$ = Rastrigin	$\sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$
$f_2(x)_{\min}$ = Griewank	$\frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
$f_3(x)_{\min}$ = Ackley	$-a\exp\left(-b\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(cx_i)\right) + a + \exp(1)$
$f_4(x)_{\min}$ = Sphere	$\sum_{i=1}^{D} x_i^2$
$f_5(x)_{\min}$ = Sum squares	$\sum_{i=1}^{D} ix_i^2$
$f_6(x)_{\min}$ = Rotated hyper-ellipsoid	$\sum_{i=1}^{D}\sum_{j=1}^{i} x_j^2$
$f_7(x)_{\min}$ = Rosenbrock	$\sum_{i=1}^{D-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
$f_8(x)_{\min}$ = Dixon-price	$(x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$
$f_9(x)_{\min}$ = Zakharov	$\sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$

TABLE 2. Design factors and levels for the tuning parameters.

Factors	Type	Levels	Level values								
Fusion timing_avg	Numeric	9	100	200	300	400	500	600	700	800	900
Fusion timing_std	Numeric	9	100	200	300	400	500	600	700	800	900

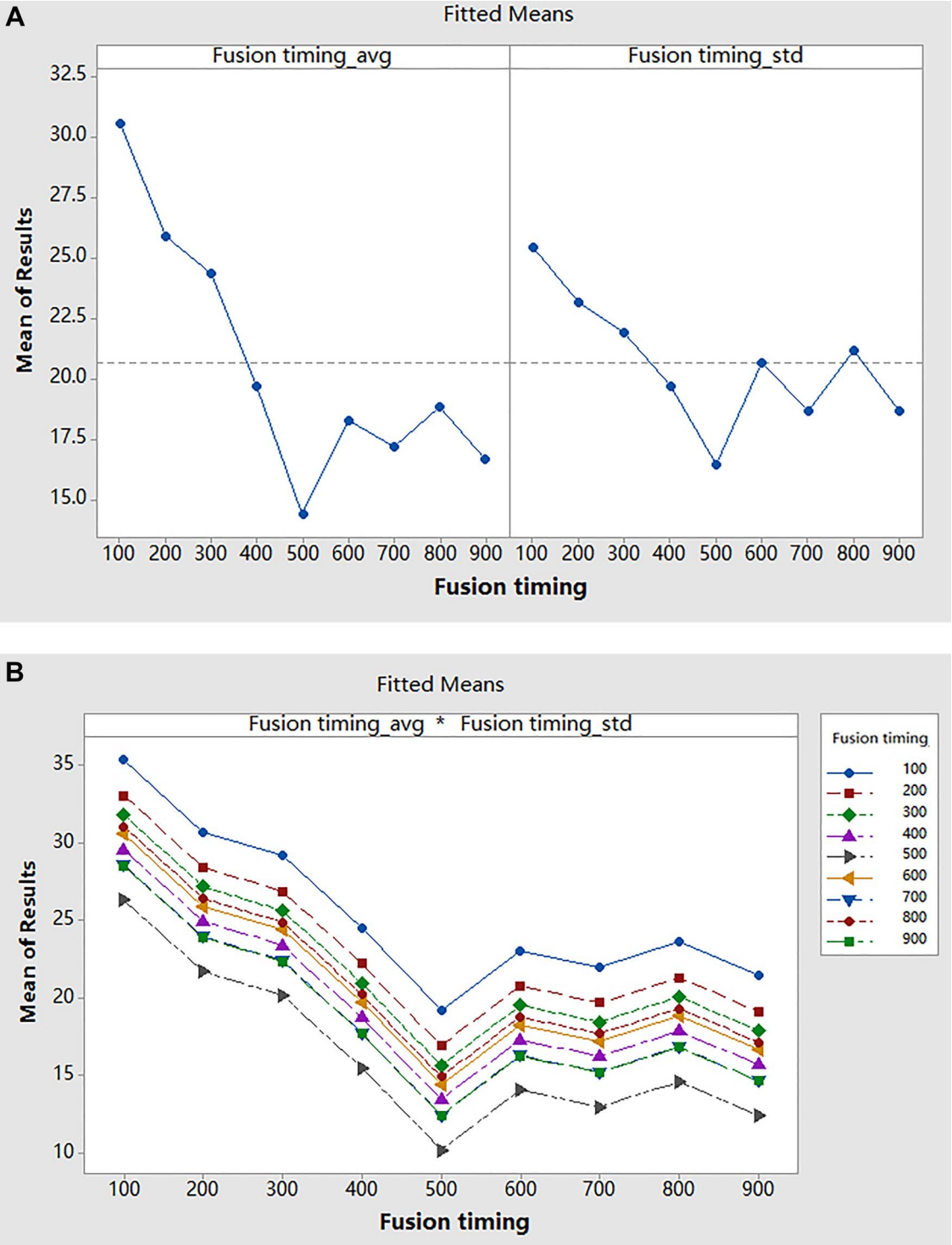

FIGURE 6. Effects of two factors on the optimization results. (A) Main effects for results. (B) Interaction plots for results.

factors. In addition, the DOE full factorial design optimizer is used to determine the optimal parameter combination based on the established optimization model. As can be seen from Figure 7, the optimal level is that both the fusion timing_avg and the fusion timing_std are 500. The minimum value of the IHPSO algorithm is 10.2187, and the satisfaction degree is 0.9240, which is consistent with the conclusion obtained from the main effects. Therefore, it is reasonable to set the fusion

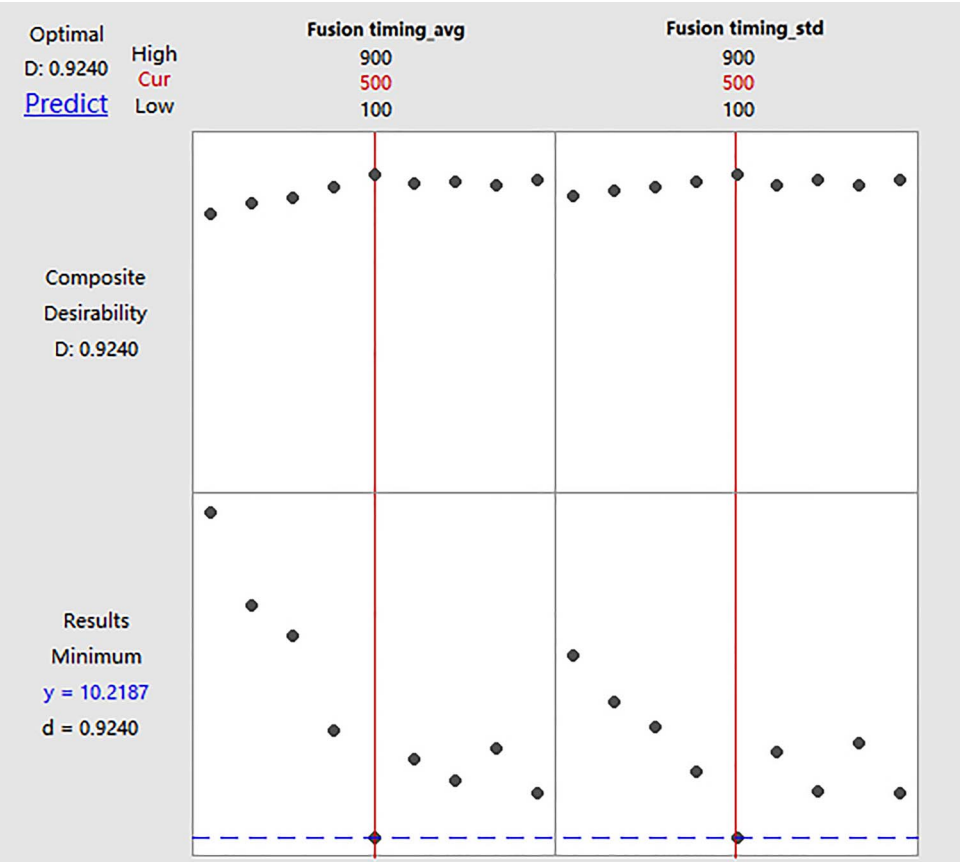

FIGURE 7. Optimization effect of the DOE optimizer.

timing to 500, which guarantees the optimization accuracy and stability of the IHPSO algorithm.

4.1.3. Optimization performance analysis

To analyze and evaluate the optimization performance of IHPSO for different test functions, 30 independent experiments are compared between the PSO, LDWPSO, GA, GA-PSO, ALPSO [49] and IHPSO algorithms. ALPSO is a PSO-based improvement method from the perspective of adaptive learning parameters, and the inertia weight and learning factors are redesigned. The evaluation indicators are as follows: average optimal fitness f_{avg}, maximum optimal fitness $f_{\max}$, minimum optimal fitness $f_{\min}$, median optimal fitness f_{med}, average number of iterations f_{iters}, average running time f_{time} and mean standard deviation f_{std}. The results are shown in Table 3, and the optimal values of all indices are marked in bold. Taking function $f_1(x)$ as an example, Figure 8 shows the adaptive crossover probability and adaptive mutation probability of a particle. Figure 9 shows the optimization curves of different algorithms.

(1) **Optimization accuracy analysis**

As shown in Table 3, the $f_{\text{avg}}, f_{\max}, f_{\min}$ and f_{med} of IHPSO for the nine test functions are the smallest. Compared with PSO, IHPSO can significantly improve the optimization accuracy because the crossover operator makes IHPSO unable to easily fall into the local optimum. Even if the population falls into the local extremum, it will jump out in time because the mutation operator increases the population diversity. For GA, its performance to eliminate the local optimum is also obvious compared with PSO and LDWPSO. Compared with PSO, LDWPSO and GA, the optimization accuracy of GA-PSO is also significantly improved. The reason is that GA-PSO combines the evolution ideas of PSO and GA in each iteration. For the ALPSO algorithm from the literature [49], its optimization accuracy is lower than that of the proposed IHPSO algorithm. However, the ALPSO algorithm improves the global search performance and the ability to jump out of the local optimum, which exhibits higher optimization accuracy than PSO, LDWPSO and GA.

(2) **Convergence analysis**

In terms of algorithm convergence, PSO is better than GA. This is because the random velocity is used to update the

TABLE 3. Performance evaluation results between PSO, LDWPSO, GA, GA-PSO, ALPSO and IHPSO.

Functions	Methods	f_{avg}	f_{max}	f_{min}	f_{med}	f_{iters}	f_{time}	f_{std}
$f_1(x)$	PSO	320.092	401.518	163.735	336.521	664.533	63.824	56.512
	LDWPSO	233.891	367.753	161.320	223.793	659.733	67.308	51.869
	GA	43.669	75.131	18.290	42.363	913.800	**27.244**	13.415
	GA-PSO	39.623	56.001	20.748	38.738	621.633	263.004	8.497
	ALPSO	43.190	86.971	2.342	37.701	600.867	105.650	24.200
	IHPSO	**8.896**	**28.158**	**0.632**	**8.016**	**585.767**	128.742	**5.926**
$f_2(x)$	PSO	0.826	1.011	0.603	0.841	879.033	66.866	0.101
	LDWPSO	0.768	0.973	0.590	0.759	826.933	71.623	0.098
	GA	0.085	0.161	0.035	0.083	947.100	**30.911**	0.026
	GA-PSO	0.022	0.102	0.008	0.019	621.100	164.050	0.016
	ALPSO	0.013	0.158	0.000	0.004	662.035	129.684	0.029
	IHPSO	**0.003**	**0.020**	**0.000**	**0.001**	**569.233**	145.151	**0.005**
$f_3(x)$	PSO	6.894	8.335	5.049	6.898	842.067	76.388	0.814
	LDWPSO	6.690	9.000	5.458	6.633	741.625	61.882	0.782
	GA	2.333	2.963	1.686	2.377	958.300	**30.098**	0.315
	GA-PSO	1.029	1.313	0.545	1.089	593.600	145.455	0.201
	ALPSO	0.506	0.913	0.103	0.511	**320.679**	140.160	0.234
	IHPSO	**0.087**	**0.190**	**0.031**	**0.086**	568.433	134.036	**0.034**
$f_4(x)$	PSO	47.857	84.313	13.134	40.921	940.800	57.707	17.440
	LDWPSO	35.522	74.631	12.725	30.325	903.133	58.017	16.765
	GA	2.188	3.203	0.784	2.144	921.233	**27.304**	0.629
	GA-PSO	0.475	0.966	0.211	0.451	564.567	134.838	0.148
	ALPSO	0.134	0.231	0.040	0.133	753.712	110.961	0.064
	IHPSO	**0.005**	**0.046**	**0.001**	**0.004**	**479.533**	130.052	**0.008**
$f_5(x)$	PSO	661.735	1408.847	167.082	586.911	928.467	63.652	324.699
	LDWPSO	611.851	1043.000	200.470	459.685	849.468	60.474	245.466
	GA	35.370	53.538	15.327	35.218	930.333	**26.193**	9.663
	GA-PSO	5.419	9.911	3.139	5.293	532.900	136.140	1.571
	ALPSO	7.704	38.000	0.544	5.925	647.432	150.894	7.264
	IHPSO	**0.101**	**0.347**	**0.020**	**0.077**	**482.733**	132.470	**0.076**
$f_6(x)$	PSO	756.619	1555.331	364.838	656.962	850.433	69.604	280.263
	LDWPSO	821.641	1279.976	395.566	840.533	817.093	70.528	278.443
	GA	32.742	51.761	11.637	32.578	924.867	**31.378**	7.997
	GA-PSO	6.109	12.695	2.806	5.420	631.400	163.455	2.335
	ALPSO	1.073	5.486	0.071	0.784	**222.789**	157.989	1.161
	IHPSO	**0.128**	**1.155**	**0.013**	**0.064**	582.867	149.730	**0.212**
$f_7(x)$	PSO	53 742.120	10 4914.632	17 702.260	52 044.093	767.033	55.874	24 572.154
	LDWPSO	24 881.410	42 132.089	4705.936	27 376.519	720.933	54.857	10 955.266
	GA	338.674	578.856	125.865	424.878	807.600	**29.295**	100.271
	GA-PSO	40.903	95.078	19.689	38.018	640.033	136.972	15.597
	ALPSO	49.135	98.295	4.661	49.494	404.489	168.970	27.027
	IHPSO	**8.232**	**89.046**	**1.096**	**4.916**	**380.267**	131.987	**14.037**
$f_8(x)$	PSO	14 715.421	54 039.447	970.838	10 944.927	725.067	65.191	11 150.063
	LDWPSO	14 088.686	24 268.587	1339.563	14 163.577	683.765	56.386	6595.058
	GA	75.720	121.729	27.138	72.538	776.733	**39.783**	26.908
	GA-PSO	3.767	5.630	2.061	3.864	587.967	139.057	1.031
	ALPSO	18.210	32.673	0.697	18.320	**330.117**	127.985	9.727
	IHPSO	**1.159**	**2.563**	**0.279**	**1.031**	485.133	122.565	**0.557**

(*Continued*)

TABLE 3. Continued

Functions	Methods	f_{avg}	f_{max}	f_{min}	f_{med}	f_{iters}	f_{time}	f_{std}
$f_9(x)$	PSO	199.054	278.054	115.199	194.807	380.367	60.884	34.107
	LDWPSO	151.722	228.935	111.179	141.990	175.625	65.492	32.719
	GA	8.000	12.152	3.654	8.239	621.167	**33.942**	2.219
	GA-PSO	5.649	14.308	3.410	5.494	501.200	160.747	2.772
	ALPSO	4.048	**6.958**	0.286	3.911	**151.090**	154.171	2.070
	IHPSO	**3.917**	11.292	**0.157**	**3.144**	476.033	97.539	**2.064**

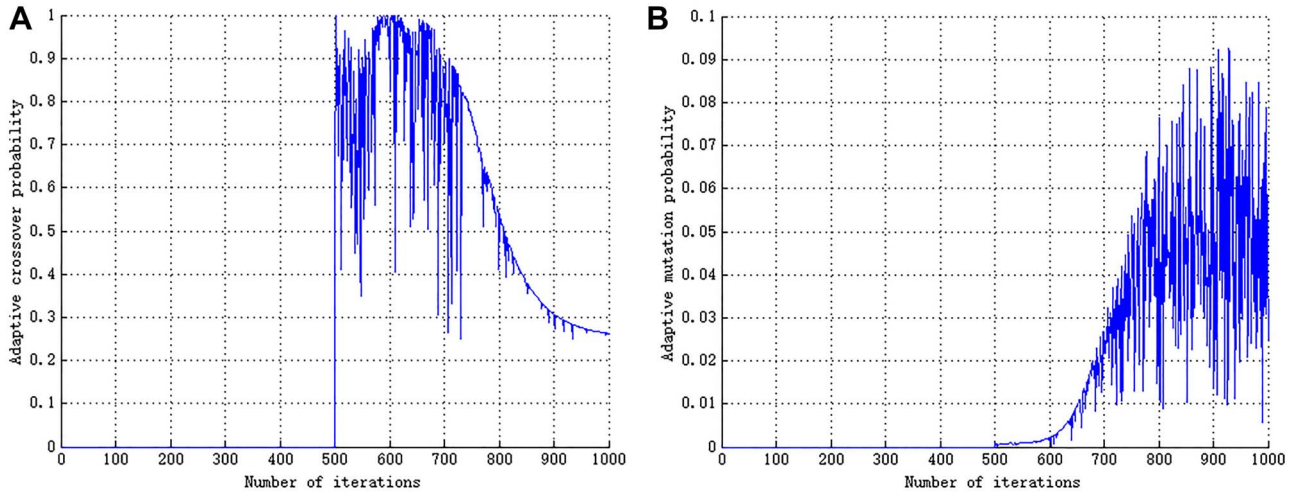

FIGURE 8. Adaptive crossover probability and mutation probability of a particle. (A) Adaptive crossover probability. (B) Adaptive mutation probability.

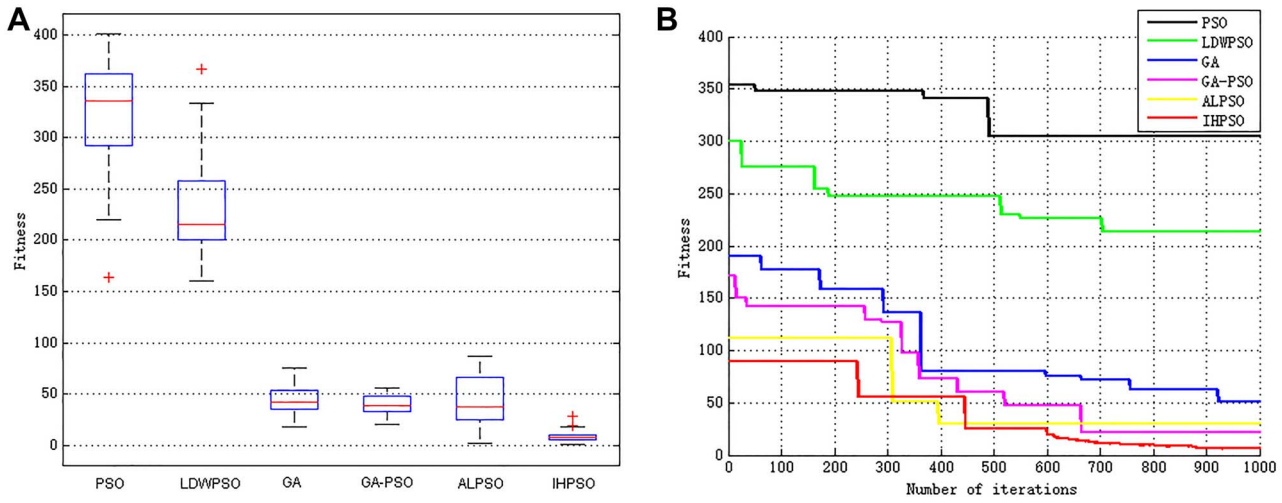

FIGURE 9. Fitness comparison for different improvement methods. (A) Optimal fitness results after 30 runs. (B) Optimization curves for a run.

particle position in the solution space, which has a strong randomness. The convergence speed of GA-PSO is faster than that of PSO and LDWPSO due to its genetic operations. GA-PSO has a better local search ability and can quickly find the global optimum. The ALPSO algorithm has a faster convergence speed than the PSO, LDWPSO, GA and GA-PSO algorithm. IHPSO improves the traditional genetic mechanism and reduces the number of iterations. In terms of running

time, GA is the fastest, and GA-PSO has the longest running time. Although IHPSO improves the optimization accuracy to a large extent compared with PSO, it increases the running time.

(3) **Stability analysis**

The mean standard deviation of IHPSO is the smallest among all compared algorithms. In Figure 9a, the box plot of IHPSO after 30 runs is the smallest, indicating that it has the best optimization stability. As seen from Figure 9b, IHPSO can effectively eliminate the local extremum to reach the global optimum.

Overall, the experimental results indicate that the proposed IHPSO is superior to PSO, LDWPSO, GA, GA-PSO and ALPSO in terms of the optimization accuracy, convergence and stability, and that it can find the lowest fitness values for the optimization problems.

4.2. Performance analysis of IHPSO-KM

4.2.1. *Evaluation metrics*

When the external information of the dataset is available, the Purity, normalized mutual information (NMI) and $F1$ indices are used to evaluate the clustering performance by comparing the clustering results with the real labels [50, 51].

Purity The proportion of correctly classified samples to the total number of samples

$$\text{Purity} = \frac{1}{n}\sum_{i=1}^{K} n_i \tag{15}$$

where n is the number of samples. K is the number of clusters. n_i is the number of data points correctly classified into the ith class.

NMI NMI is used to measure the consistency between the clustering results and the real labels

$$\text{NMI} = 2 \times \frac{\text{MI}\,(U, V)}{H(U) + H(V)} \tag{16}$$

$$\text{MI}\,(U, V) = \sum_{k=1}^{K}\sum_{l=1}^{L} P\,(k \cap l) \log\left(\frac{P\,(k \cap l)}{P(k)P(l)}\right) \tag{17}$$

$$H(U) = -\sum_{k=1}^{K} P(k)\log\,(P(k)) \tag{18}$$

$$H(V) = -\sum_{l=1}^{L} P(l)\log\,(P(l)) \tag{19}$$

where MI is the mutual information. H is the entropy. L is the number of real classes. $P(k)$, $P(l)$ and $P(k \cap l)$ denote the probabilities of the samples belonging to cluster k, class l, and k and l, respectively.

TABLE 4. Summarization of dataset characteristics.

#	Name	# Samples	# Numerical attributes
1	Abalone	4177	8
2	Waveform	5000	21
3	Page blocks	5473	10
4	Wine quality	4898	11
5	Image segmentation	2310	19

F1 $F1$-measure combines Precision and Recall for clustering evaluation. For each real class, the closest cluster is selected as the $F1$ value

$$F1\,(k, l) = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

$$F1 = \sum_{l=1}^{L} p(l) \cdot \max_{1\le k\le K} F1\,(k, l) \tag{21}$$

.

The internal method considers the dispersion and compactness of clusters. In this paper, the silhouette coefficient (SC) is used to evaluate the intrinsic clustering quality. SC is defined as follows:

$$\text{SC}(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}} \tag{22}$$

$$a(o) = \frac{\sum_{o' \in E_i, o \ne o'} dist\,(o, o')}{|\,U_i\,| - 1} \tag{23}$$

$$b(o) = \min_{E_j : 1\le j\le K, j\ne i} \left\{ \frac{\sum_{o' \in E_j} dist\,(o, o')}{|\,U_j\,|} \right\} \tag{24}$$

where $a(o)$ represents the average distance between o and other samples in the same cluster. $b(o)$ represents the minimum average distance between o and all samples in the nearest cluster. To measure the fitness of clusters, the average value of all data SC is used as the final internal evaluation index.

4.2.2. *Algorithm performance evaluation*

To analyze the clustering capacity of the IHPSO-KM algorithm, we compare the K-means, K-means++, K-medoids, PSO-KM, WGPAM (weighted Gower PAM) [52], HPSO-KM (GA-PSO-K-means), ALPSO-KM (ALPSO-K-means) [49] and IHPSO-KM algorithms on five typical UCI machine learning datasets. The basic descriptions of the selected UCI datasets are listed in Table 4. These datasets have similar numbers of samples and attributes as the grape-customer consumption dataset in this study [53]. To eliminate influences

TABLE 5. Comparison results for different clustering algorithms.

#	Methods	Purity	NMI	*F*1	*SC*	Running time
1	K-means	0.378	0.158	0.101	0.366	**19.495**
	K-means++	0.390	0.159	0.154	0.367	44.643
	K-medoids	0.391	0.161	0.150	0.369	42.175
	PSO-KM	0.495	0.160	0.183	0.370	338.865
	WGPAM	0.502	0.288	0.346	0.402	69.657
	HPSO-KM	0.505	0.322	0.486	0.430	443.092
	ALPSO-KM	0.507	0.168	0.183	0.372	358.338
	IHPSO-KM	**0.509**	**0.325**	**0.491**	**0.447**	364.528
2	K-means	0.531	0.362	0.504	0.222	**0.802**
	K-means++	0.636	0.371	0.523	0.227	1.236
	K-medoids	0.627	0.369	0.519	0.226	2.554
	PSO-KM	0.725	0.374	0.594	0.238	258.122
	WGPAM	0.729	0.380	0.582	0.301	11.320
	HPSO-KM	0.738	0.384	0.596	0.319	321.087
	ALPSO-KM	0.742	0.443	0.650	**0.369**	207.887
	IHPSO-KM	**0.746**	**0.451**	**0.653**	0.365	289.466
3	K-means	0.710	0.055	0.806	0.592	**2.193**
	K-means++	0.747	0.056	0.809	0.593	3.672
	K-medoids	0.742	0.056	0.814	0.595	3.387
	PSO-KM	0.758	0.057	0.874	0.666	281.807
	WGPAM	0.759	0.062	0.877	0.685	19.425
	HPSO-KM	0.770	0.073	0.892	0.690	300.654
	ALPSO-KM	0.775	0.062	**0.897**	0.673	282.374
	IHPSO-KM	**0.779**	**0.081**	0.895	**0.694**	297.043
4	K-means	0.456	0.028	0.228	0.318	**5.991**
	K-means++	0.533	0.032	0.335	0.367	6.765
	K-medoids	0.563	0.029	0.356	0.358	7.809
	PSO-KM	0.635	0.050	0.371	0.417	334.870
	WGPAM	0.634	0.058	0.379	0.419	32.909
	HPSO-KM	0.635	0.064	0.384	0.426	386.447
	ALPSO-KM	0.632	0.059	**0.481**	0.462	308.691
	IHPSO-KM	**0.736**	**0.075**	0.472	**0.463**	369.583
5	K-means	0.515	0.518	0.465	0.324	**0.878**
	K-means++	0.544	0.524	0.462	0.339	1.769
	K-medoids	0.544	0.523	0.445	0.345	1.827
	PSO-KM	0.585	0.604	0.443	0.356	172.906
	WGPAM	0.546	0.631	0.477	0.358	9.546
	HPSO-KM	0.647	0.647	0.485	0.420	206.318
	ALPSO-KM	0.675	**0.667**	0.464	0.455	200.087
	IHPSO-KM	**0.695**	0.660	**0.496**	**0.457**	199.785

of the internal randomness of the algorithms on the clustering results, each clustering algorithm is independently run 10 times. The different clustering algorithms are evaluated using the average of each indicator, as shown in Table 5. The bold values indicate the optimal results. Figure 10 shows the clustering performances of different clustering algorithms based on the five datasets.

It can be seen from Table 5 and Figure 10 that K-means has the lowest clustering accuracy due to its excessive dependence on the initial cluster centers. The clustering accuracies of K-means++ and K-medoids are better than that of K-means. K-means and PAM are typical partition-based clustering algorithms. PAM is a variant of the K-means algorithm that has solved the problems of K-means, such as producing empty clusters and sensitivity to outliers. The clustering accuracy of the WGPAM algorithm is significantly improved. Compared with PSO-KM and WGPAM, HPSO-KM makes use of the strong global search ability of HPSO, and the population has a

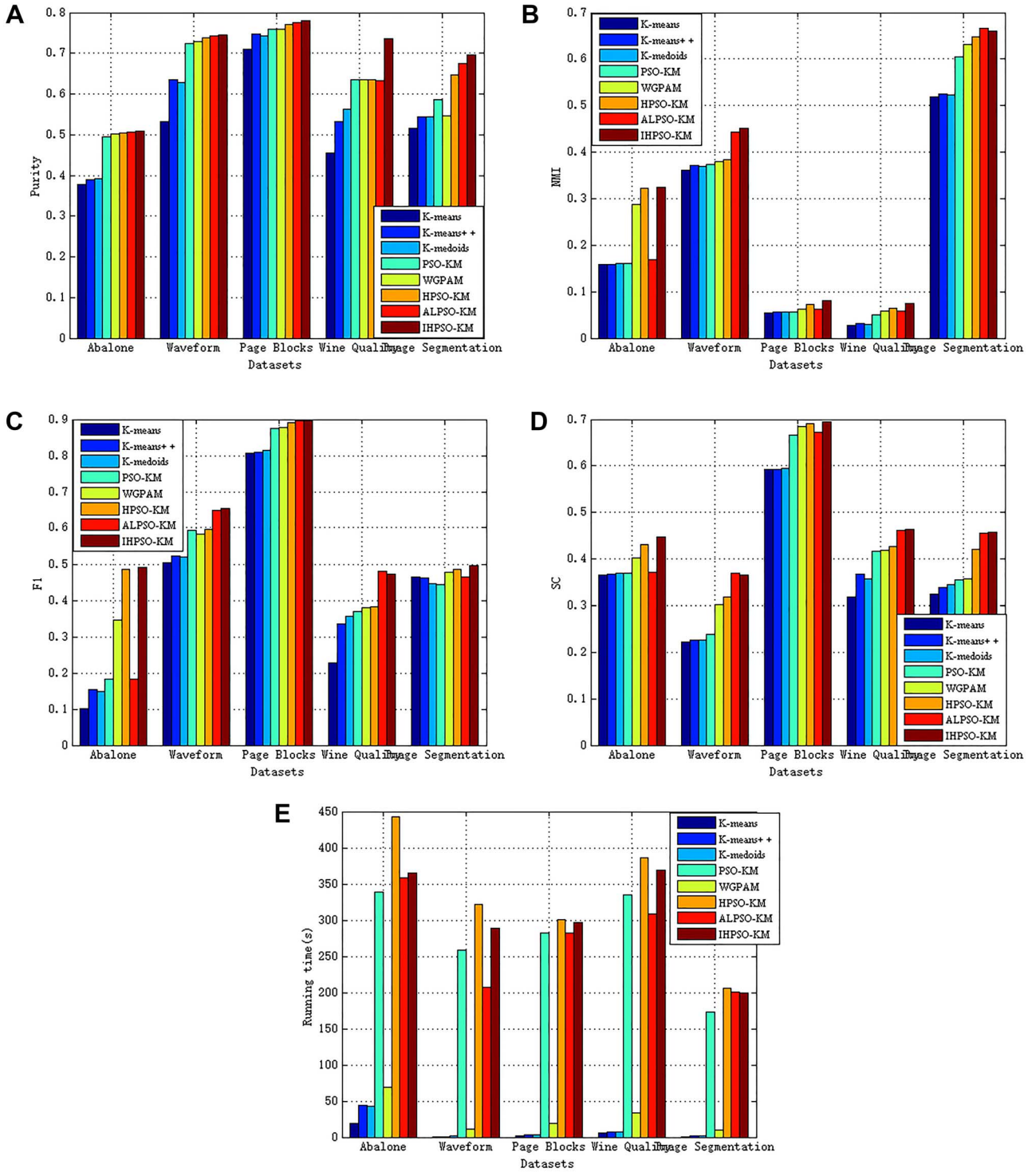

FIGURE 10. Evaluation indicators of different clustering algorithms. (A) *Purity*. (B) *NMI*. (C) *F1*. (D) *SC*. (E) *Running time*.

greater randomness. HPSO-KM also has the local search ability of K-means, so it cannot easily fall into the local minimum. For the ALPSO-KM algorithm proposed in the literature [49], it performs more accurately than the K-means, K-means++, K-medoids, PSO-KM and WGPAM algorithms. Compared with HPSO-KM and ALPSO-KM, IHPSO-KM has higher clustering accuracy and longer running time on the five datasets. This is because the proposed IHPSO-KM monitors the

optimization process of particles and optimizes the premature convergence to make particles jump out of the local extremum in time. Although the IHPSO-KM algorithm has a longer running time than that of K-means, the running time is acceptable for the customer segmentation requirements in this study.

4.3. Customer segmentation method based on IHPSO-KM

To further verify the rationality and feasibility of IHPSO-KM in actual applications, this paper carries out a cluster analysis of customer consumption data in China's table grape market.

4.3.1. Data acquisition

The experimental data come from the consumption values and basic customer information in the questionnaire. The sampling method uses quota sampling, which is commonly used in market research. The respondents are ordinary urban consumers. The questionnaire interviewers were students from China Agricultural University. Finally, 3652 questionnaires were collected, and the effective sample size was 3230. The sample distribution of different data types is shown in Figure 11. The characteristics of categorical variables are described in Table 6.

From Figure 11, the demographic characteristics are described as follows: the sample distributions for gender and age are approximately balanced; the sample education level is generally high; the main occupations are companies and enterprises; the family population is approximately 3∼4 people; the monthly income is concentrated between 3001 and 5000 yuan and the regional economic development is biased toward cities with relatively high development levels.

4.3.2. Analysis of customer segmentation results

Based on normalized and standardized data processing methods, the optimal number of customer groups is determined by comparing the number of iterations and optimal fitness (Figure 12). The results show that when the number of clusters is 5, the number of iterations of IHPSO-KM is the smallest. While the optimal fitness decreases with the number of clusters, it is reasonable to have five customer groups.

To show the clustering effects based on the actual grape customer consumption dataset, the original 14-dimensional features are compressed through the principal component analysis (PCA) method. The data distribution is shown in Figure 13. From Figure 13b, the clustering effect of each customer category is more obvious and has good discrimination. Thus, the proposed IHPSO-KM is a suitable method to divide customer groups.

To describe and summarize the consumption characteristics among different customer groups, the mean values of numerical variables and the frequencies of categorical variables are counted. The clustering results are compared with the

TABLE 6. The characteristics of categorical features.

#	1	2	3	4	5	6	7	8	9
Gender	Male	Female							
Age	17–25	26–35	36–45	46–55	56–75				
Education	Bachelor degree or above	Junior college	High school	Junior high school	Primary school and below				
Occupation	Party and government organ and institution	Company and enterprise	Freelancer	Farmer	Education and research institution	Student	Unemployment and retirement	other	
Family population	1	2	3	4	5	6	7	8	9
Monthly income	<2000	2001–3000	3001–5000	5001–7000	7001–10 000	10 001–15 000	>15 000		
Regional economic development	First-tier	New first-tier	Second-tier	Third-tier	Fourth-tier	Fifth-tier			

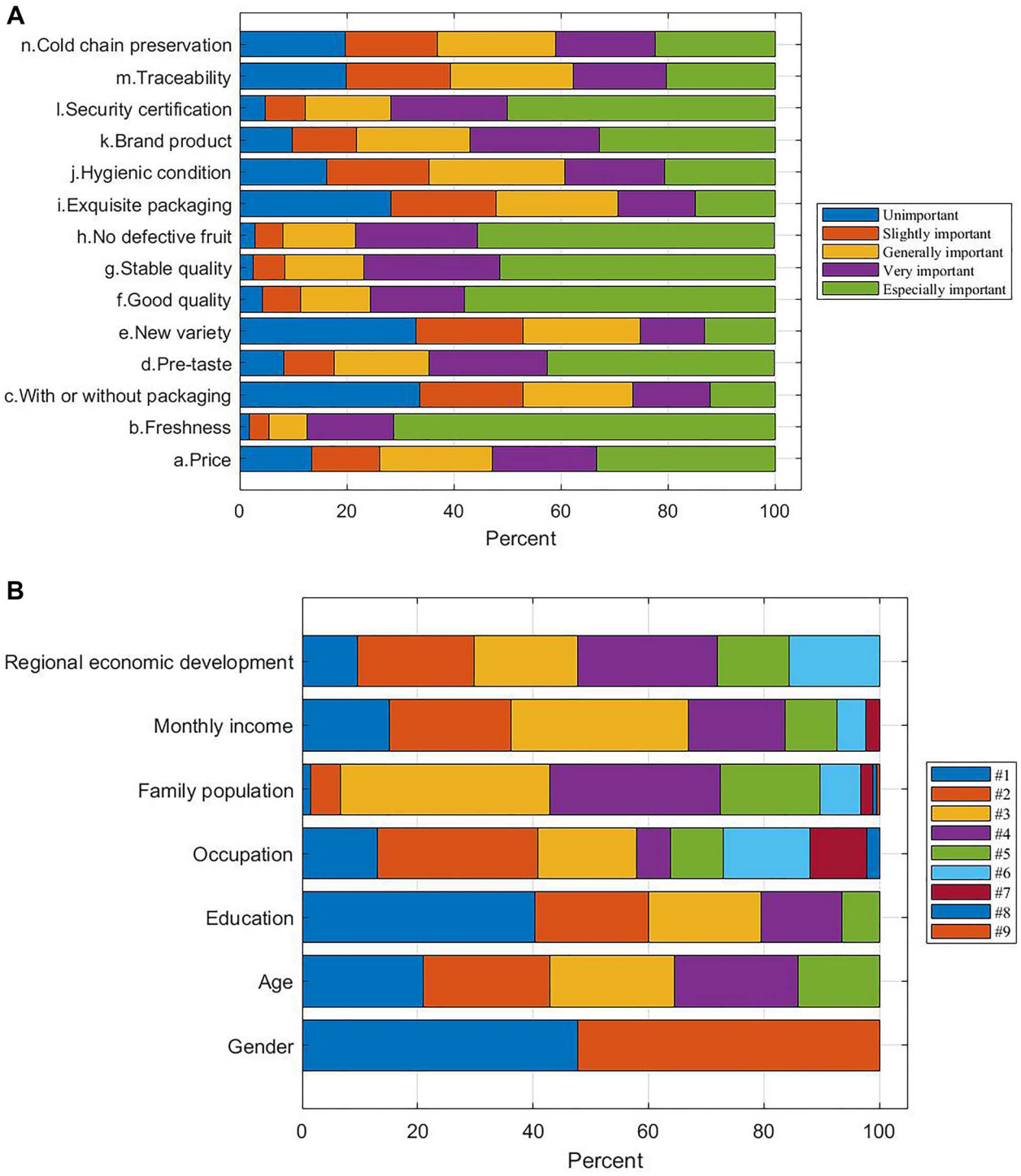

FIGURE 11. Proportion of data characteristics. (A) Sample distribution of numerical features. (B) Sample distribution of categorical features.

consumption level of the China's grape market ('·'), as shown in Figure 14.

Taking 'Customer group 4' which belongs to the professional group with high income, as an example we see that there are more male youths with bachelor's degree or above, a smaller family population and cities concentrated in the first and third tiers. For this group, their requirements for grape quality, safety, appearance and packaging hygiene are higher than the average level of the grape market. This indicates that this group has a high consumption demand for table grapes.

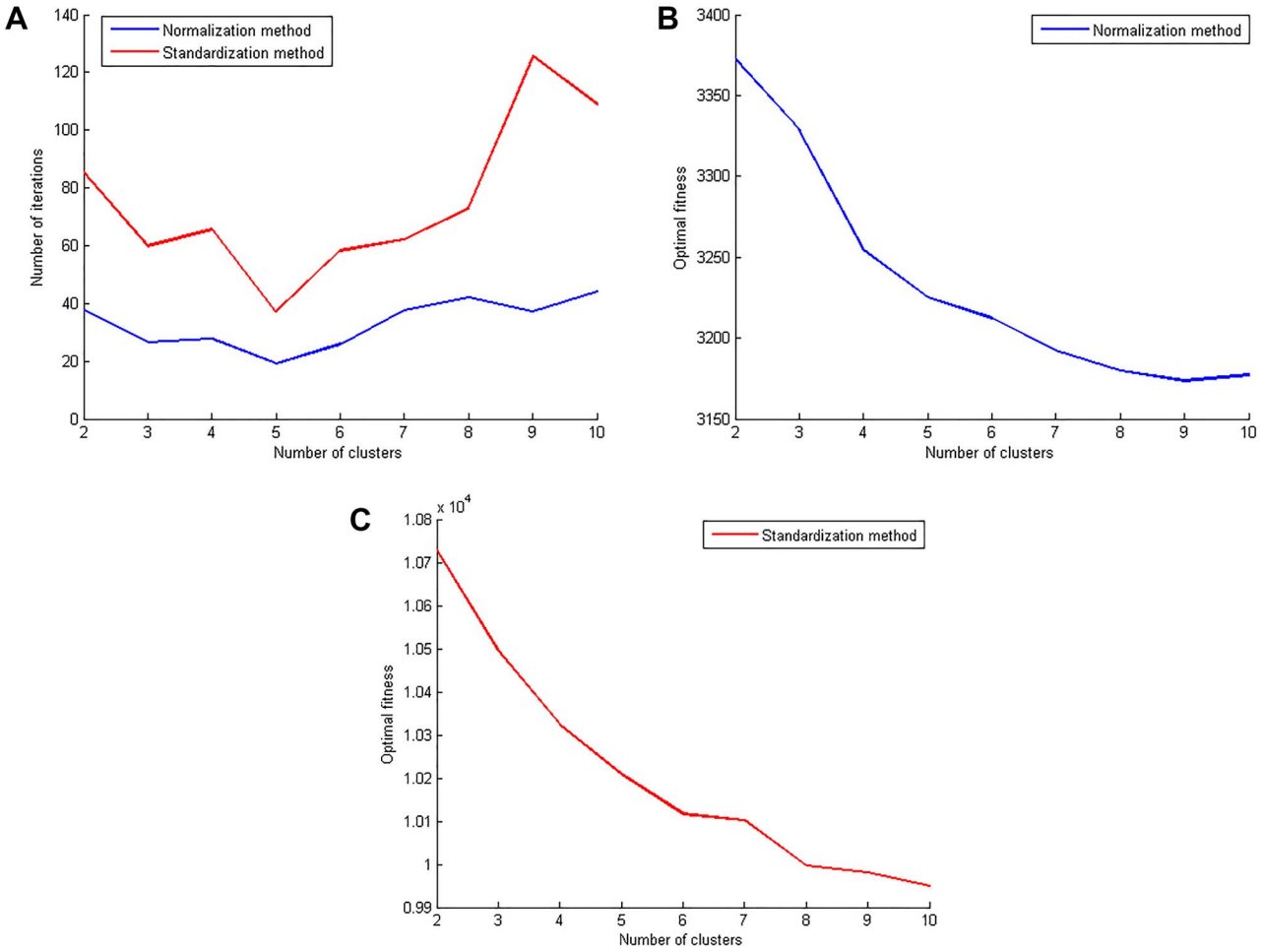

FIGURE 12. Number of iterations and optimal fitness corresponding to the number of clusters (grape dataset). (A) Number of iterations. (B) Normalization. (C) Standardization.

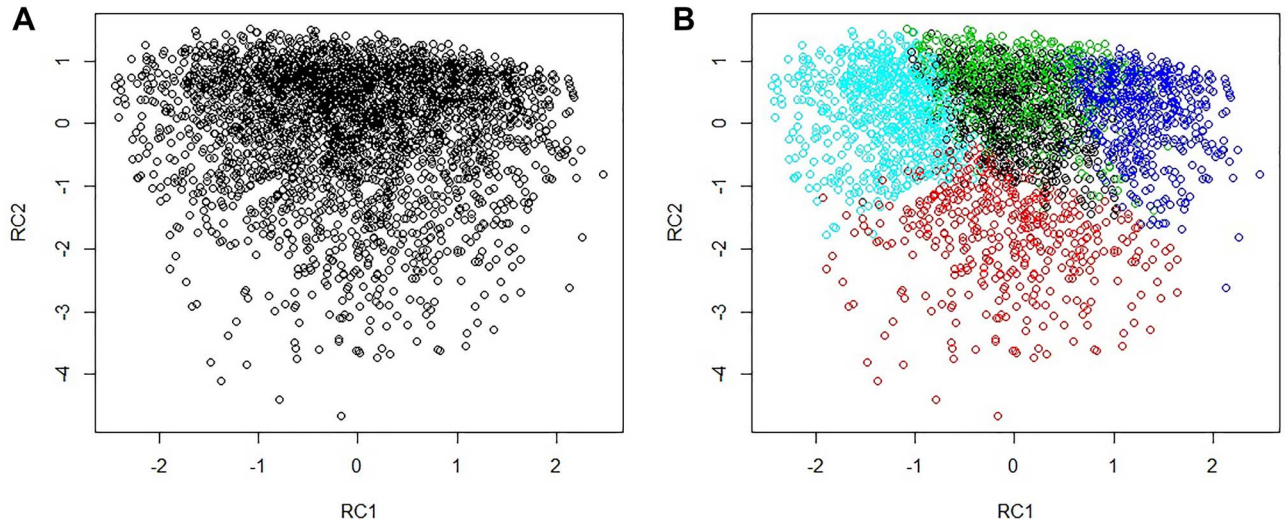

FIGURE 13. Comparison of clustering effects after PCA. (A) Original graph. (B) Clustering graph.

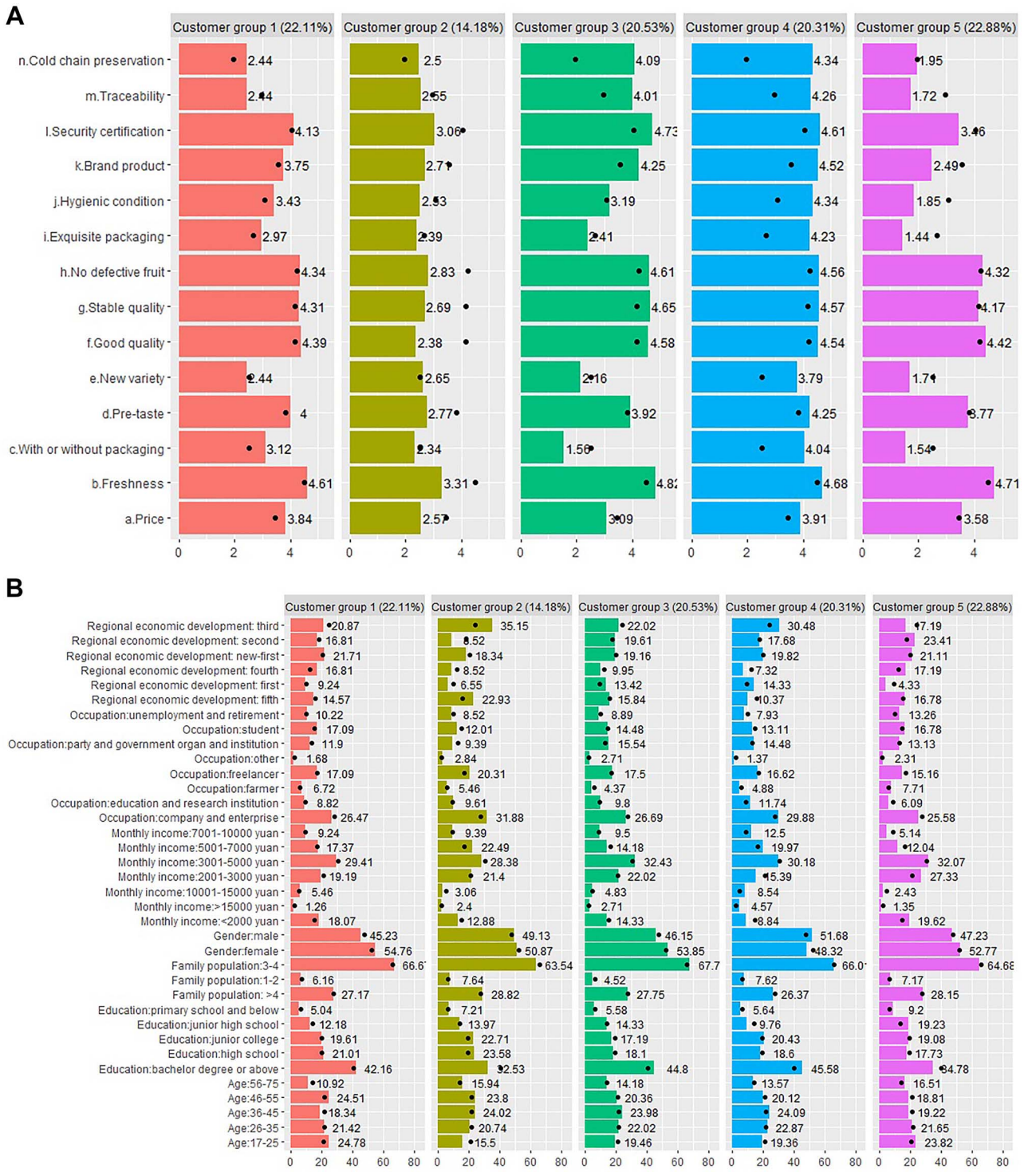

FIGURE 14. Customer segmentation effects based on IHPSO-KM. (A) Numerical variables. (B) Categorical variables.

Through the visual analysis of customer segmentation results, it can be clearly seen that different customer groups have different consumption behaviors. Therefore, the customer segmentation method based on IHPSO-KM can be used to create personalized marketing strategies.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new customer segmentation method using K-means clustering and the hybrid PSO algorithm. This paper has two key innovations: (i) the IHPSO algorithm is proposed to prevent falling into the local

extremum and (ii) the IHPSO-KM algorithm is proposed to improve the clustering accuracy of customer segmentation, and the dependence on the initial centers is effectively restricted.

We use nine test functions to verify the optimization performance of IHPSO. The experimental results illustrate that our proposed IHPSO algorithm has better optimization accuracy, convergence and stability compared with the PSO, LDW-PSO, GA, GA-PSO and ALPSO algorithms. We also compare IHPSO-KM with the K-means, K-means++, K-medoids, PSO-KM, WGPAM, HPSO-KM and ALPSO-KM clustering algorithms on five UCI datasets. The experimental results show that the IHPSO-KM algorithm is of higher accuracy than other state-of-the-art algorithms. Finally, the IHPSO-KM algorithm is applied to divide customer groups into five categories with different consumption behaviors. This verifies the practicability and feasibility of applying the customer segmentation in this method.

We demonstrate the rationality of our customer segmentation method. However, the running time of the proposed IHPSO-KM algorithm has yet to be improved for large-scale datasets. How to reduce the running time while improving the clustering accuracy is the key to future work.

DATA AVAILABILITY

The data underlying this article cannot be shared publicly for the privacy of individuals that participated in the study. The data will be shared on reasonable request to the corresponding author.

SUPPLEMENTARY MATERIAL

Supplementary material is available at www.comjnl.oxfordjournals.org.

ACKNOWLEDGEMENTS

This paper is our original work and has not been published or submitted simultaneously elsewhere. All authors have agreed to the submission and declared that they have no conflict of interest. The authors are grateful to the editor and the anonymous referees for their many helpful comments on earlier version of our paper.

FUNDING

Chinese Agricultural Research System (CARS-29); Key Laboratory of Viticulture and Enology, Ministry of Agriculture, PR China.

CONFLICT OF INTEREST STATEMENT

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

REFERENCES

[1] Xiao, J., Cao, H., Jiang, X., Gu, X. and Xie, L. (2017) GMDH-based semi-supervised feature selection for customer classification. *Knowledge-Based Syst.*, 132, 236–248.
[2] Holý, V., Sokol, O. and Černý, M. (2017) Clustering retail products based on customer behavior. *Appl. Soft Comput.*, 60, 752–762.
[3] Munusamy, S. and Murugesan, P. (2020) Modified dynamic fuzzy c-means clustering algorithm-application in dynamic customer segmentation. *Appl. Intell.*, 50, 1922–1942.
[4] Hayashi, Y., Friedel, J.E., Foreman, A.M. and Wirth, O. (2019) A cluster analysis of text message users based on their demand for text messaging: a behavioral economic approach. *J. Exp. Anal. Behav.*, 112, 273–289.
[5] Jiang, X., Li, C. and Sun, J. (2018) A modified K-means clustering for mining of multimedia databases based on dimensionality reduction and similarity measures. *Cluster Comput.*, 21, 797–804.
[6] Tsai, C.F., Hu, Y.H. and Lu, Y.H. (2015) Customer segmentation issues and strategies for an automobile dealership with two clustering techniques. *Expert Syst.*, 32, 65–76.
[7] Luo, F.L. (2014) An improved K-means algorithm and its application in customer classification of network enterprises. *Appl. Mech. Mater.*, 543-547, 2124–2127.
[8] Xie, H., Zhang, L., Lim, C.P., Yu, Y., Liu, C., Liu, H. and Walters, J. (2019) Improving K-means clustering with enhanced firefly algorithms. *Appl. Soft Comput.*, 84, 105763.
[9] Zhang, G., Zhang, C. and Zhang, H. (2018) Improved K-means algorithm based on density canopy. *Knowledge-Based Syst.*, 145, 289–297.
[10] Bai, L., Cheng, X., Liang, J., Shen, H. and Guo, Y. (2017) Fast density clustering strategies based on the K-means algorithm. *Pattern Recognit.*, 71, 375–386.
[11] Ei-Alfy, E.S.M. (2017) Detection of phishing websites based on probabilistic neural networks and K-medoids clustering. *Comput. J.*, 60, 1745–1759.
[12] Xu, Y., Qu, W., Li, Z., Ji, C., Li, Y. and Wu, Y. (2014) Fast scalable K-means++ algorithm with MapReduce. In *Proc. of 2014 Int. Conf. on Algorithms and Architectures for Parallel Processing (ICA3PP 2014)*, Dalian, China, August 24–27, pp. 15–28. Springer, Cham.
[13] Ushakov, A.V. and Vasilyev, I. (2021) Near-optimal large-scale K-medoids clustering. *Inform. Sci.*, 545, 344–362.
[14] Liu, M., Zhang, B., Li, X., Tang, W. and Zhang, G. (2021) An optimized K-means algorithm based on information entropy. *Comput. J.*, 64, 1130–1143.
[15] Wu, P. and Liu, C. (2013) Financial distress study based on PSO K-means clustering algorithm and rough set theory. *Appl. Mech. Mater.*, 411-414, 2377–2383.
[16] Li, J., Xiao, D., Zhang, T., Liu, C., Li, Y. and Wang, G. (2021) Multi-swarm cuckoo search algorithm with Q-learning model. *Comput. J.*, 64, 108–131.
[17] Bouyer, A. and Hatamlou, A. (2018) An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Appl. Soft Comput.*, 67, 172–182.
[18] Kuo, R.J. and Zulvia, F.E. (2018) Automatic clustering using an improved artificial bee colony optimization for customer segmentation. *Knowl. Inf. Syst.*, 57, 331–357.

[19] Kenny, I. (2020) Hydrographical flow modelling of the river Severn using particle swarm optimization. *Comput. J.*, 63, 1713–1726.
[20] Selvi, M.T. and Jaison, B. (2020) Lemuria: a novel future crop prediction algorithm using data mining. *Comput. J.*, 1–12.
[21] Wu, Z., Wu, Z. and Zhang, J. (2017) An improved FCM algorithm with adaptive weights based on SA-PSO. *Neural Comput. Applic.*, 28, 3113–3118.
[22] Omran, M.G.H., Salman, A. and Engelbrecht, A.P. (2006) Dynamic clustering using particle swarm optimization with application in image segmentation. *Pattern Anal. Appl.*, 8, 332–344.
[23] Zhang, J., Wang, Y. and Feng, J. (2014) A hybrid clustering algorithm based on PSO with dynamic crossover. *Soft Comput.*, 18, 961–979.
[24] Gao, H., Li, Y., Kabalyants, P., Xu, H. and Martínez-Béjar, R. (2020) A novel hybrid PSO-K-means clustering algorithm using Gaussian estimation of distribution method and Lévy flight. *IEEE Access*, 8, 122848–122863.
[25] Qiu, C., Wang, C. and Zuo, X. (2013) A novel multi-objective particle swarm optimization with K-means based global best selection strategy. *Int. J. Comput. Intell. Syst.*, 6, 822–835.
[26] Niu, B., Duan, Q., Liu, J., Tan, L. and Liu, Y. (2017) A population-based clustering technique using particle swarm optimization and K-means. *Nat. Comput.*, 16, 45–59.
[27] Huang, C.L., Huang, W.C., Chang, H.Y., Yeh, Y.C. and Tsai, C.Y. (2013) Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. *Appl. Soft Comput.*, 13, 3864–3872.
[28] Kiran, M.S., Gündüz, M. and Baykan, Ö.K. (2012) A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum. *Appl. Math Comput.*, 219, 1515–1521.
[29] Rahman, M.A. and Islam, M.Z. (2014) A hybrid clustering technique combining a novel genetic algorithm with K-means. *Knowledge-Based Syst.*, 71, 345–365.
[30] Marjani, A., Shirazian, S. and Asadollahzadeh, M. (2018) Topology optimization of neural networks based on a coupled genetic algorithm and particle swarm optimization techniques (c-GA–PSO-NN). *Neural Comput. Applic.*, 29, 1073–1076.
[31] Bertram, A.M., Zhang, Q. and Kong, S.C. (2016) A novel particle swarm and genetic algorithm hybrid method for diesel engine performance optimization. *Int. J. Engine. Res.*, 17, 732–747.
[32] Tan, C., Chang, S. and Liu, L. (2017) Hierarchical genetic-particle swarm optimization for bistable permanent magnet actuators. *Appl. Soft Comput.*, 61, 1–7.
[33] Gandelli, A., Grimaccia, F., Mussetta, M., Pirinoli, P. and Zich, R.E. (2007) Development and validation of different hybridization strategies between GA and PSO. In *Proc. of 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, September 25–28, pp. 2782–2787. IEEE
[34] Fu, Y., Wang, B. and Xu, S.H. (2012) A hybrid evolution algorithm with application based on chaos genetic algorithm and particle swarm optimization. In *Proc. of 2012 National Conf. on Information Technology and Computer Science,* Lanzhou, China, 16 November, pp. 405–409. Atlantis Press.
[35] Yazdanjue, N., Fathian, M. and Amiri, B. (2020) Evolutionary algorithms for K-anonymity in social networks based on clustering approach. *Comput. J.*, 63, 1039–1062.
[36] Choudhary, A., Kumar, M., Gupta, M.K., Unune, D.K. and Mia, M. (2020) Mathematical modeling and intelligent optimization of submerged arc welding process parameters using hybrid PSO-GA evolutionary algorithms. *Neural Comput. Applic.*, 32, 5761–5774.
[37] Lu, Z.J., Xiang, Q., Li, B.Z. and Wei, W. (2013) Support vector machine with real code genetic algorithm for yarn quality prediction. *Adv. Sci. Lett.*, 19, 2468–2472.
[38] Zheng, L., Wang, Y., Luo, R., Tong, X. and Liang, H. (2016) Study on delivery route optimization based on improved genetic algorithm. *Adv. Appl. Math.*, 5, 516–522.
[39] Liu, M.H. and Peng, X.F. (2010) Improved adaptive genetic algorithms for job shop scheduling problems. *Adv. Mat. Res.*, 97–101, 2473–2476.
[40] Majumder, S., Saha, B., Anand, P., Kar, S. and Pal, T. (2018) Uncertainty based genetic algorithm with varying population for random fuzzy maximum flow problem. *Expert Syst.*, 35, e12264.
[41] Pereira, A.G.C., Campos, V.S.M., de Pinho, A.L.S., Vivacqua, C.A. and de Oliveira, R.T.G. (2020) On the convergence rate of the elitist genetic algorithm based on mutation probability. *Commun. Stat.*, 49, 769–780.
[42] Zuo, M., Dai, G. and Peng, L. (2019) Multi-agent genetic algorithm with controllable mutation probability utilizing back propagation neural network for global optimization of trajectory design. *Eng. Optimiz.*, 51, 120–139.
[43] Chinnasri, W. (2013) Adaptive probability of crossover and mutation in genetic algorithm on university course timetabling problem. In *Proc. of 2013 IEEE Int. Conf. on Computer Science and Automation Engineering (CSAE 2013)*, Guangzhou, China, November 2013, pp. 724–728. IEEE.
[44] Chen, K., Zhou, F. and Liu, A. (2018) Chaotic dynamic weight particle swarm optimization for numerical function optimization. *Knowledge-Based Syst.*, 139, 23–40.
[45] Nagra, A.A., Han, F. and Ling, Q.H. (2019) An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search. *Eng. Optimiz.*, 51, 1115–1132.
[46] Yan, C., Lu, G., Liu, Y. and Deng, X. (2017) A modified PSO algorithm with exponential decay weight. In *Proc. of 2017 13th Int. Conf. on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017)*, Guilin, China, July 29–31, pp. 239–242. IEEE.
[47] Hakli, H. and Uğuz, H. (2014) A novel particle swarm optimization algorithm with levy flight. *Appl. Soft Comput.*, 23, 333–345.
[48] Ridge, E. and Kudenko, D. (2010) Tuning an algorithm using design of experiments. In Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M. (eds) *Experimental Methods for the Analysis of Optimization Algorithms*, pp. 265–286. Springer, Berlin Heidelberg.
[49] Li, Y., Chu, X., Tian, D., Feng, J. and Mu, W. (2021) Customer segmentation using K-means clustering and the adaptive particle swarm optimization algorithm. *Appl. Soft Comput.*, 113, 107924.
[50] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M. and Perona, I. (2013) An extensive comparative study of cluster validity indices. *Pattern Recognit.*, 46, 243–256.

[51] Kannan, S.R., Ramthilagam, S., Devi, R. and Huang, Y.M. (2013) Novel quadratic fuzzy c-means algorithms for effective data clustering problems. *Comput. J.*, 56, 393–406.

[52] Li, Y., Chu, X., Mou, X., Tian, D., Feng, J. and Mu, W. (2020) An optimized hybrid clustering algorithm for mixed data: application to customer segmentation of table grapes in China. In *Proc. of 2020 10th Int. Conf. on Computer Engineering and Networks (CENet 2020)*, Xi'an, China, October 16–18, pp. 20–32. Springer, Singapore.

[53] Cortez, P., Cerdeira, A., Almeida, F., Matos, T. and Reis, J. (2009) Modeling wine preferences by data mining from physicochemical properties. *Decis. Support Syst.*, 47, 547–553.