Applied Science and Innovative Research
ISSN 2474-4972 (Print) ISSN 2474-4980 (Online)
Vol. 7, No. 4, 2023
www.scholink.org/ojs/index.php/asir

Original Paper

A Quantum Particle Swarm Optimization Algorithm Based on Aggregation Perturbation

H. D. Wang[1], C. N. Zhang[1], H. Zhang[1], Y. C. Wei[1] & X. L. Guan[2]

[1] University of Science and Technology Liaoning, School of Computer Science and Software Engineering, Anshan Liaoning, China

[2] Anshan Information Engineering School, Anshan Liaoning, China

Received: August 16, 2023 Accepted: September 8, 2023 Online Published: September 13, 2023

doi:10.22158/asir.v7n4p21 URL: http://doi.org/10.22158/asir.v7n4p21

Abstract

A quantum particle swarm hybrid optimization algorithm based on aggregation disturbance is proposed for inventory cost control. This algorithm integrates the K-means algorithm on the basis of traditional particle swarm optimization, recalculates the clustering center, initializes stagnant particles, and solves the problem of particle aggregation. Introducing chaos mechanism into the algorithm, changing the position of particles, enhancing their activity, and improving the algorithm's global optimization ability. At the same time, define the aggregation disturbance factor, determine the current state of particles, optimize speed and position to accelerate escape, and solve the problem of particles falling into local optima. Experiments show that M-IKPSO algorithm has strong stability, fast Rate of convergence and high accuracy compared with other algorithms, and the improvement effect is significant.

Keywords

Particle Swarm Optimization; Aggregation disturbance; K-means algorithm

1. Introduction

Inventory cost often occupies an important position in the cost accounting of an enterprise, especially for iron and steel enterprises, which accounts for more than 60%. Changes in market supply and demand determine the necessity of steel mill inventory. How to measure the size of inventory and Market trend is an important way for enterprises to reduce costs. The traditional inventory analysis methods include three types: using year-on-year changes to measure the size of inventory; Using month on month changes to measure inventory consumption speed; Use the data from China Steel Association to estimate the actual consumption in the market. Enterprises can comprehensively consider inventory influencing factors based on the above three analysis methods, determine the true level of inventory, price cycle changes, and seasonal effects, and then calculate reasonable inventory holding and destocking speed to determine the market supply and demand situation and arrange production based on the inventory of steel. Establishing a mathematical model that meets the requirements of inventory optimization and designing an efficient algorithm to solve the optimal procurement quantity is of great practical significance for the cost reduction of enterprises (Kamal, Bakar, & Zainudin, 2022).

Particle swarm optimization (PSO) algorithm was proposed by J. Kennedy and R.C. Eberhart in 1995. The algorithm is a simulation of birds' predatory behavior, iterative in an evolutionary way, and is a global optimization algorithm. The advantages of PSO algorithm lie in its fast search speed, fewer adjustable parameters, and high stability. Once the algorithm is introduced, it is widely praised by scholars and can quickly find the optimal solution, which has been widely applied in multi-objective optimization and constraint problem optimization (Nafiseh & Sodeif, 2020).

Although the PSO algorithm has certain advantages in terms of global optimization ability and convergence, its tendency to fall into local optima is also evident. In algorithm iteration, when particles gather in a certain local area, the usual approach is to enhance the inertia coefficient and give the particles an acceleration to break free from local constraints. However, since the historical global optimal position has been verified in the local range, if the acceleration is not large during the particle's journey, it is easy to quickly return to the original gathering point. In order to solve the above problems, a quantum particle swarm optimization algorithm based on aggregation disturbance is proposed in the paper. The algorithm works when particles fall into local optima (Ramesh, Sundararaman, Sabareeswaran, & Srinivasan, 2022; Ahmed, Osman, & Korovkin, 2021).

2. Particle Swarm Related Issues

2.1 Fundamentals of Particle Swarm Optimization

The PSO algorithm is a simulated evolutionary process that obtains the optimal solution through communication and collaboration between individuals, where each particle has two attributes: velocity and position. The solution of the problem corresponds to the current position of the particle, and its performance depends on the fitness value of the optimization function. In the Q-dimensional space, the particle is initialized as $\{x_1, x_2, \cdots, x_n\}$, the velocity of the i-th particle in the space is defined as $v_i = \{v_{i1}, v_{i1}, \cdots, v_{iQ}\}$, the position is defined as $x_i = \{x_{i1}, x_{i1}, \cdots, x_{iQ}\}, i = 1, 2, \cdots, n$, and the direction of travel follows the trajectory of the optimal particle. Among them, the current optimal position searched by a particle is Pb, and the optimal position obtained by the entire population is Gb. As the algorithm iterates, the particle positions are repeatedly stacked until the termination condition is met, obtaining the optimal solution. For k+1 iterations, each particle updates its own velocity and position according to the following equation:

$$v_{iq}^{k+1} = \alpha v_{iq}^{k} + \beta^{0}(x_{Pb} - x_{iq}^{k}) + \beta^{1}(x_{Gb} - x_{iq}^{k}) \tag{1}$$

$$x_{iq}^{k+1} = x_{iq}^{k} + v_{iq}^{k+1} \tag{2}$$

Among them, α is the inertia weight coefficient used to balance the relationship between the current particle's travel speed; $\beta^0, \beta^1 \in [0,1]$, Random distribution; $q = 1, 2, \cdots, Q$.

In the clustering problem, for vectors in the Q-dimensional space, it ultimately boils down to obtaining a class partition. The partition class is defined as C, and the following formula needs to be met:

$$\begin{cases} C_i \neq \phi \\ C_i C_j = \phi \\ \overset{k}{\underset{i=1}{U}} = X \end{cases} \tag{3}$$

Among them, C_i, $C_j \in C$, $i, j \in [1, k]$, based on this, the inter class dispersion is obtained,

$$J = \sum_{j=1}^{k} \sum_{X_i \in C} D(X_i, Z_i) \tag{4}$$

In the formula, k is the number of clusters, Z_i is the i-th cluster center, and $D(X_i, Z_i)$ represents the distance from the sample to the corresponding cluster center.

For the precocity problem of algorithms, it is mainly reflected in the convergence of the algorithm and the diversity of particles. The trajectory of particles is not random, and it is inevitable that some poorly performing particles will appear during iteration, with a significant decrease in speed or oscillation in a certain dimension of space. This can easily lead to local optima and premature phenomena. Based on this, this article proposes several improvement schemes for particle performance. Firstly, introduce chaos mechanism into the algorithm to enhance particle ergodicity; Furthermore, utilizing quantum theory to enhance particle vitality and enhance particle optimization ability; Meanwhile, an attempt was made to update the particle state using the K-means algorithm in order to achieve better particle performance (Badhera, Verma, & Nahar, 2022; Shrifan, Akbar, & Isa, 2022).

2.2 Chaotic Sequence

Chaos widely exists in nature and is a nonlinear phenomenon. Because of its sensitivity to Initial condition and its characteristics of randomness, ergodicity and regularity, it can search all internal states in a given area according to its own laws without repetition. Based on this, chaos mechanism is used to optimize the particle optimization process, and the search steps are as follows:

(1) Define the initial region. The N-dimensional initial state vector is $R_0 = (R_{01}, R_{02}, \cdots, R_{0N})$, and the values in R_0 are adjacent and have little difference.

(2) The Initialization vector R_0 is calculated by using the logistics equation to generate the chaotic sequence $c_1, c_2, \cdots, c_n$. Here, after several iterations, the system will be completely in a chaotic state. The vector layer can be expressed as:

$$c_{i+1} = c_i (1 - c_{i-1}) \lambda \tag{5}$$

Among them, λ is the iterative control parameter.

(3) Assuming the spatial particle X_i, use formula (3) to obtain a better position for X_i, denote $X_i^{'}$.

$$X_i^{'} = r \cdot rnd \cdot c_j + X_i \tag{6}$$

Among them, r is the active radius of particle X_i, $rnd \in [-1,1]$, $j \in [0,n]$.

The main idea of particle swarm optimization algorithm based on chaotic mechanism is as follows: on the one hand, using chaotic sequences to initialize the position and velocity of particles, due to its ergodic characteristics, it not only maintains the diversity of particles but also enhances their search ability; Furthermore, chaotic states can make the motion of particles persistent.

Particle chaos initialization: Assign initial values to X_i in formula (6) and re-correct the velocity in particle swarm iteration:

$$\begin{aligned} v_{i,j}^{'}(t+1) = \alpha v^{"}(t) + \beta^0(t)(x_{lb}(t) - x_{i,j}(t)) \\ + \beta^1(t)(x_{gb}(t) - x_{i,j}(t)) \end{aligned} \tag{7}$$

Among them, α is the constant of $(0,1]$, β is the random number of Normal distribution $N[0,1]$, $i \in [1,n]$, $j \in [1,m]$, n is the particle number, and m is the spatial dimension. About $v^{"}(t)$

$$v^{"}(t) = \begin{cases} v_{i,j}(t) & \theta = 0 \\ N[0,1] \cdot \delta \cdot \tilde{v} & \theta = 1 \end{cases} \tag{8}$$

$$\theta = \begin{cases} 0 & f(x_{gb}(t-1)) > f(x_{gb}(t)) \\ 1 & f(x_{gb}(t)) = f(x_{gb}(t-1)) = \\ & \cdots = f(x_{gb}(t-5)) \end{cases} \tag{9}$$

Among them, $\tilde{v} = v_{\max} \cdot c_i / 1.1$, $\delta = f(x_{gb}) - f(x_T)$, c_i is a new chaotic sequence, $f(x_{gb})$ is the satisfactory solution, and $f(x_T)$ is the target solution.

The introduction of chaotic mechanisms will have a positive impact on the search process. In this paper propose to replace the weight coefficients of random distributions in PSO with chaotic sequences to complete particle velocity updates.

2.3 Inertia Weight

The improvement of algorithm accuracy introduces chaos mechanism in the article, and for the problem of early maturity and slow evolution in the later stage, it is advisable to improve the inertia weight value in speed update. The ultimate goal of particle evolution is to find the global optimal solution, and the position and velocity change during each iteration. The position is determined by velocity, so velocity is the key to guiding particle flight and optimizing algorithm design. Here, the magnitude of inertia weight directly affects the velocity and direction of particles. Based on the above analysis, the article intends to adjust the inertia weight values that affect speed to make them change according to the cosine law with evolutionary algebra. In this way, in the initial stage of the algorithm, the inertia weight is maintained at a large value to improve the particle's local search ability. The relevant expression is as follows:

$$\alpha = \alpha_0 + (\alpha_1 - \alpha_0) \cdot \cos(\frac{i_0 \pi}{2 i_1}) \tag{10}$$

Among them, α_0、α_1 respectively represent the minimum and maximum weight values of inertia, $\alpha \in [\alpha_0、\alpha_1]$, i_0、i_1 respectively represent the current iteration number and the maximum allowed iteration number.

The use of inertia weights to adjust particle travel speed has significant advantages in global search, but in the later stages of the algorithm, due to the decrease in particle speed, this weight control suppresses particle activity and makes it difficult to avoid falling back into local optima after escaping. The use of decreasing weight can improve the search efficiency of the algorithm to a certain extent, but the motion of particles has a certain degree of randomness and is susceptible to the influence of previous local optima. This strategy has significant hidden dangers. Based on the above reasons, the article intends to further improve the inertia weight value by using an adaptive weight setting strategy to enhance both global search ability and local search ability to improve algorithm performance. The relevant expression is as follows:

$$\alpha' = \begin{cases} \alpha_1 \\ \alpha_0 + \cos\frac{(\alpha_1 - \alpha_0)(i_1 - i_0)}{i_1} \\ \alpha_0 \end{cases} \tag{11}$$

Among them, α' is the current inertia weight value in the iteration; α_0、α_1 respectively is the minimum and maximum weights of inertia, and i_0、i_1 respectively is the current iteration number and the maximum allowed iteration number.

3. Improved Particle Swarm Optimization Hybrid Algorithm

3.1 Quantum Particle Swarm

Quantum particle swarm embodies the superposition state and probability expression characteristics in quantum theory (Ramesh & Haider, 2020). The combined effect of the two can make particles present more states, thereby improving the diversity of the population. Among them, probability expressions can express particles with a certain probability. This article sets the two fundamental states $\overset{\wedge}{0}$ and $\overset{\wedge}{1}$ of particles based on the probability of selection.

In quantum particle swarm, particle quantization is the representation of each bit of a particle using quantum bits, which are represented as a and b. At the same time, this state can also be manifested as a mixed state, which is a linear combination of basic states, as follows:

$$\phi = \lambda_1 \overset{\wedge}{0} + \lambda_2 \overset{\wedge}{1} \tag{12}$$

Among them, λ_1、λ_2 is a complex number, that is, when the quantum state is $\overset{\wedge}{0}$, the probability is

$|\lambda_1|^2$; When the state is $\overset{\wedge}{1}$, the probability is $|\lambda_2|^2$.

The spatial expression of quantum states can evolve as follows:

$$\phi = \cos\theta\overset{\wedge}{0} + \sin\theta\overset{\wedge}{1} \tag{13}$$

Among them, θ is the phase of the quantum state, which satisfies the equation: $\theta = \arctan\frac{\lambda_2}{\lambda_1}$.

In this way, the quantized particles are placed in a quantum space represented by a wave function, where they can freely search for the optimal solution, and their state is not limited by velocity and position vectors. The corresponding wave function representing the particle state is as follows:

$$\int_{-\infty}^{+\infty} L_p dxdydz = 1 \tag{14}$$

The iterative formula for particles obtained by converting quantum states using Monte Carlo algorithm:

$$P_{mb} = \frac{1}{l}\sum_{i=1}^{l} P_i \tag{15}$$

$$P_{mk} = \theta P_{ik} + (1-\theta)P_{gk} \tag{16}$$

$$L_p(t+1) = P_{ik} \pm a\,|\,P_{mk} - L_p(t)\,|\ln^{\frac{1}{\lambda}} \tag{17}$$

In the above formula, P_i is the current position of the particle, P_{gb} is the global extreme value, P_{mb} is the median optimal, l is the number of particles, P_{mk} is the random point between P_{ik} and P_{gk}, a is the expansion and contraction factor, and is a quantitative index used to control the Rate of convergence of the algorithm. The random number between $(0,1)$ is taken during iteration. If $a > 0.5$, the positive value is taken, otherwise the negative value is taken.

3.2 Aggregation Disturbance

During the iteration process, due to the reduced speed of particles in the later stage and being constrained by local optima, the accuracy and convergence ability of the algorithm will significantly decrease when a certain local area is too clustered, making it easy to obtain local extremum (Yang, 2022). Therefore, a concept of particle aggregation perturbation is proposed in the article to regulate the phenomenon of local particles being too clustered. During iteration, particle perturbation is performed in areas with high crowding, increasing the activity of particles and breaking free from the constraints of local optima.

Set a disturbance factor in the text, and the value of this factor should not be in static mode. When the weight value is small, it is beneficial for particles to escape from the aggregation region, and the search ability increases while the convergence ability decreases; When the weight value is large, the particle activity increases, the Rate of convergence is accelerated, and it is easy to quickly approach the local extreme value and fall into the local optimum. Therefore, this factor needs to be dynamic, and the relevant expression is as follows:

$$\eta = \begin{cases} \eta_0 \cdot \log_2(1+\dfrac{i_0}{i_1}) & \eta < \eta_1 \\ \eta_1 & \eta \geq \eta_1 \end{cases} \tag{18}$$

Among them, η is the disturbance factor, η_0 is the current disturbance factor value, and η_1 is the maximum disturbance factor value.

During the algorithm iteration process, the perturbation factor value gradually increases. When the aggregation degree of particles in a certain area reaches the threshold, the perturbation factor takes effect, and particles move within that area. At this time, the particle position needs to be recalculated, and the relevant expression is as follows:

$$X_i^{'} = r \cdot rnd \cdot \eta + X_i \tag{19}$$

Among them, r is the activity radius of particle X_i, $rnd \in [-1,1]$, and η is the disturbance factor.

3.3 Algorithm Analysis

Introducing the K-means algorithm into the PSO algorithm, the basic idea is to use the K-means algorithm to calculate the clustering centers of each generation of particles to form a new particle swarm and obtain global extremum until the algorithm reaches the termination condition. This can effectively increase the activity of particles and avoid falling into local optima. At the same time, the introduction of K-means algorithm improves the overall convergence of the algorithm for fast convergence. The specific steps are as follows:

Step 1: Initialize the particle swarm, set the total number of particles to n, and randomly select k particles as the clustering center, where k<n, set the initial position and velocity of the particles and other attributes;

Step 2: Divide the population, calculate fitness values, calculate local and global optimal values based on their values, and recalculate the position and velocity of particles;

Step 3: Determine whether the current particle is trapped in a local optimum and disturbance it;

Step 4: Integrate the K-means algorithm, optimize the newly generated particles, recalculate the cluster center of the population and the fitness value;

Step 5: Repeat steps 2, 3, and 4 to determine whether the iteration meets the termination condition and obtain the global optimal solution; Otherwise, skip to step 2 and continue iteration.

4. Experimental Analysis

Three experimental tests were conducted on the algorithm studied in the article: clustering performance, convergence ability, and accuracy. Three algorithms are proposed for comparison: PSO (traditional particle swarm optimization algorithm), QPSO (quantum particle swarm optimization algorithm), and the hybrid particle swarm optimization algorithm proposed in this paper (M-KQPSO). For the clustering effect, Analysis of algorithms is proposed for the sample data in the UCI general database.

The datasets are Vehicle, Wine, and Same. Among them, Vehicle and Wine are low dimensional datasets, the former dimension is 18, the number of classes is 4, and the number of samples set is 800; The latter dimension is 13, the number of classes is 3, and the number of samples set is 150; Sam is a high-dimensional dataset, the dimension is 16090, the number of classes is 3, the number of samples set is 300. The maximum iteration number of the algorithm is 400. The maximum iteration number of the algorithm is set to be 400. The normalized Mutual information (NMI) is used to evaluate the clustering effect. It reflects the similarity between the clustering results of the sample set and the real class division, the value range is set to $[0,1]$. The larger the value, the better the clustering effect.

Three benchmark functions were selected for testing in the article, namely *Noncontinuous Rastrigin*, *Griewank*, *Generalized Penalized*, with a focus on examining the algorithm's global search ability. The relevant definitions and parameters of the functions are as follows:

(1) The formula of the *Noncontinuous Rastrigin* function is as follows:

$$Noncontinuous\ Rastrigin = \sum_{i=1}^{D}[x_i^2 - 10\cos 2\pi x_i + 10]$$

$$x_i = \begin{cases} k_i & |k_i| < 0.5 \\ round(2k_i)/2 & |k_i| \geq 0.5 \end{cases}$$

(2) The formula of the *Griewank* function is as follows:

$$Griewank = \frac{1}{4000}\sum_{i=1}^{D}(x_i)^2 - \prod_{i=1}^{D}\cos(x_i/\sqrt{i}) + 1$$

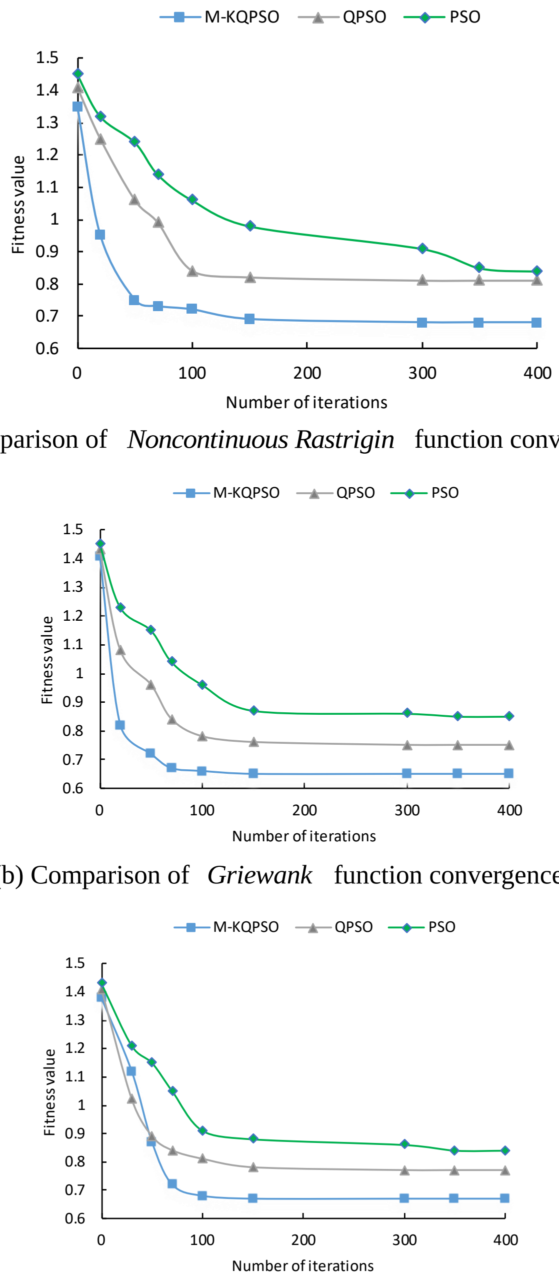
(3) The formula of the *Generalized Penalized* function is as follows:

$$\begin{aligned} Generalized\ Penalized = {} & \frac{\pi}{D}\{10\sin^2(\pi y_i) \\ & + \sum_{i=1}^{D}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}) \\ & + (y_D - 1)^2]\} + \sum_{i=1}^{D}u(x_i) \end{aligned}$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m \\ 0 \\ k(-x_i - a)^m \end{cases}$$

Table 1. Benchmark Function Parameters

Benchmark function	Dimension	Value range	Optimal value
Noncontinuous Rastrigin	30	$[-5.12, 5.12]^D$	0
Griewank	30	$[-600, 600]^D$	0
Generalized Penalized	30	$[-50, 50]^D$	0



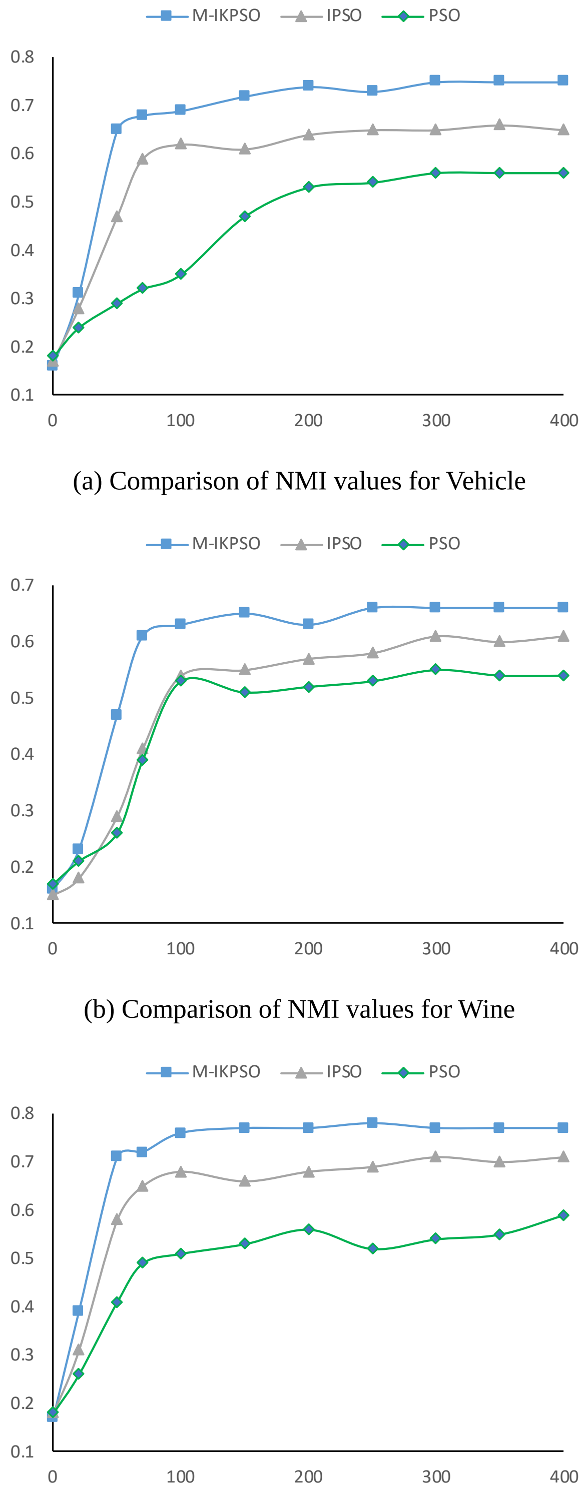
(a) Comparison of *Noncontinuous Rastrigin* function convergence

(b) Comparison of *Griewank* function convergence

(c) Comparison of *Generalized Penalized* function convergence

Figure 1. Algorithm Convergence

Figure 1 selects three benchmark functions for convergence comparison, and the results show that the improved algorithm has significantly better convergence than the other two algorithms. Moreover, the performance of the algorithm after introducing quantum theory is stronger than that of traditional PSO algorithms, especially in the later iteration stage, with good convergence. Because the particle activity of traditional algorithms decreases, making it easy to fall into local optima during iteration. After improvement, the cluster center is recalculated and necessary disturbances are made to improve the state of the particles.



(a) Comparison of NMI values for Vehicle

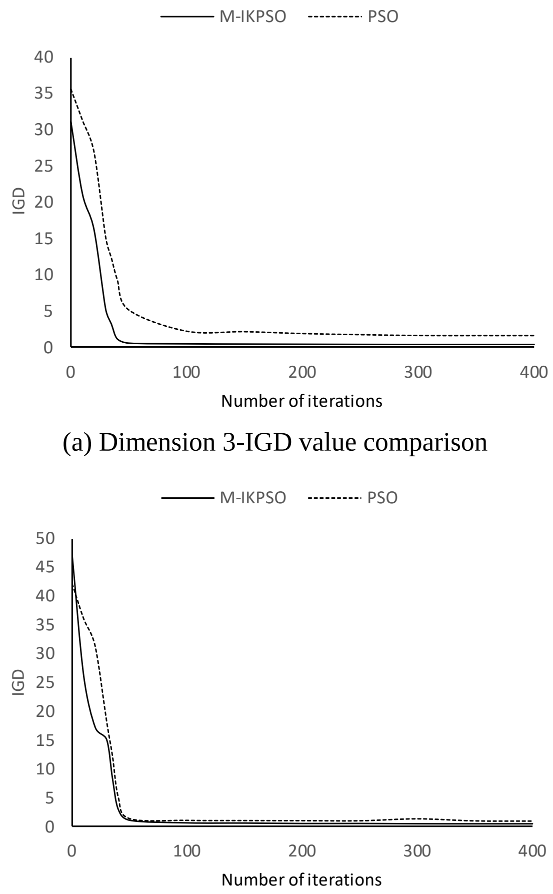
(b) Comparison of NMI values for Wine

(c) Comparison of NMI values for Same

Figure 2. Comparison Results of NMI Values

Figure 2 shows the comparison results of NMI values among three algorithms in the UCI dataset. The results show that the improved algorithm performs significantly better than the other two algorithms. At the beginning of the algorithm, the values of the three algorithms are basically the same. As the algorithm progresses and the iteration number reaches 100, it can be seen that M-IKPSO tends to stabilize significantly, with the highest value and rapid appreciation. At the same time, in the later stage of the algorithm, the other two algorithms have some oscillations and declines, while the stability of M-IKPSO continues to be maintained and the improvement goal is achieved.

Select Inversion General Distance (IGD) to evaluate the accuracy of the algorithm, where the higher the IGD value, the higher the accuracy of the algorithm. When testing, select 3 and 5 as the target dimensions. The number of iterations of the algorithm is set to 400, and the average is calculated by taking 30 iterations. The results are as follows:



(a) Dimension 3-IGD value comparison

(b) Dimension 5-IGD value comparison

Figure 3. Comparison Results of IGD Values

Figure 3 shows that algorithm M-IKPSO has a result closer to 0 compared to PSO, and the difference between the two decreases as the dimension increases. However, the improved algorithm still maintains an advantage in convergence and first reaches the equilibrium point. Therefore, it is concluded that the algorithm has better accuracy and convergence in high dimensions.

5. Conclusion

This article proposes an intelligent optimization algorithm (M-IKPSO) based hybrid mode, which is combined with the K-means algorithm to effectively solve the problem of particles trapped in local optima. The algorithm has certain advantages in accuracy and stability. In traditional PSO algorithms, the concept of clustering disturbance is introduced, which timely disturbances particles when they are stationary. At the same time, K-means is used to recalculate the clustering center, redefine the particle position, and quickly converge. Introducing chaos mechanism to improve the global optimization ability of particles, quantifying particles, adjusting their travel speed, and ensuring the diversity of solutions. Experiments have shown that the improved algorithm has achieved significant improvements in stability, accuracy, and convergence.

Fund Project

Funded by the 2023 College Student Innovation and Entrepreneurship Training Program Fund (202310146007)

References

Ahmed, M. K., Osman, M. H., & Korovkin, N. (2021). Optimal reactive power dispatch in power system comprising renewable energy sources by means of a multi-objective particle swarm algorithm. *Materials Science. Power Engineering*, *27*(115), 5-20.

Badhera Usha, Verma Apoorva, & Nahar Pooja. (2022). Applicability of K-medoids and K-means algorithms for segmenting students based on their scholastic performance. *Journal of Statistics and Management Systems*, *25*(7), 1621-1632. https://doi.org/10.1080/09720510.2022.2130566

Kamal Nor Ashikin Mohamad, Bakar Azuraliza Abu, & Zainudin Suhaila. (2022). Optimization of Discrete Wavelet Transform Feature Representation and Hierarchical Classification of G-Protein Coupled Receptor Using Firefly Algorithm and Particle Swarm Optimization. *Applied Sciences*, *12*(23), 12011-12011. https://doi.org/10.3390/app122312011

Nafiseh Hematpour, & Sodeif Ahadpour. (2020). Execution examination of chaotic S-box dependent on improved PSO algorithm. *Neural Computing and Applications*, *33*(10), 1-23. https://doi.org/10.1007/s00521-020-05304-9

Ramesh Kumar Huda, & Haider Banka. (2020). A group evaluation based binary PSO algorithm for feature selection in high dimensional data. *Evolutionary Intelligence*, *14*(4), 1-15. https://doi.org/10.1007/s12065-020-00482-w

Ramesh, M., Sundararaman, K. A., Sabareeswaran, M., & Srinivasan, R. (2022). Development of Hybrid Artificial Neural Network–Particle Swarm Optimization Model and Comparison of Genetic and Particle Swarm Algorithms for Optimization of Machining Fixture Layout. *International Journal of Precision Engineering and Manufacturing*, *23*(12), 1411-1430. https://doi.org/10.1007/s12541-022-00698-z

Shrifan Nawaf H.M.M., Akbar Muhammad F., & Isa Nor Ashidi Mat. (2022). An adaptive outlier removal aided k-means clustering algorithm. *Journal of King Saud University - Computer and Information Sciences*, *34*(8), 6365-6376. https://doi.org/10.1016/j.jksuci.2021.07.003

Yang, D. Q. (2022). An improved particle swarm optimization algorithm for parameter optimization. *Computer Informatization and Mechanical System*, *5*(4), 35-38.