Xiaojun Yu Jiangsu University of Technology, Changzhou, Jiangsu, China

DYNAMIC MULTI-SWARM PSO BASED ON K-MEANS CLUSTERING

Сяоцзюнь Юй Технологічний університет Цзянсу, м. Чанчжоу, провінція Цзянсу, КНР

ДИНАМІЧНА БАГАТОРОЄВА ОПТИМІЗАЦІЯ МЕТОДОМ РОЮ ЧАСТОК НА ОСНОВІ КЛАСТЕРИЗАЦІЇ МЕТОДОМ K-СЕРЕДНІХ

Purpose. Objective optimization is a very important area in scientific research and practical applications, many problems are related to the objective optimization. The research investigates combinational measures of Particle Swarm Optimization (PSO) and K-means clustering. The dynamic multi-swarm particle swarm optimization based on K-means clustering (KDMPSO) has been obtained, which is a hybrid clustering algorithm integrating PSO and K-means clustering, and it can nicely find global extreme in different problems.

Methodology. The comprehensive and in-depth analysis on PSO and K-means clustering was carried out, and improvement strategies have been found by adopting combinational measures of PSO and K-means clustering. For both continuous and discrete optimization problems, it has strong global search capacity; it effectively reduces the premature convergence of the traditional PSO.

Findings. Combination of the advantages of PSO and K-means clustering solves convergence to local optimum and inefficiency of traditional PSO algorithm in complex optimization problems, ensures that PSO is stable and can maintain the population diversity, avoids prematurity, and enhances the algorithm accuracy.

Originality. The multi-swarm PSO and K-means clustering were studied. In the iteration process, PSO is easy to get trapped in local optimal solution, causing the phenomenon of premature convergence, on the other hand, K-means is extensively used in clustering since it is easy to realize and it is also a highly-efficient algorithm with linear time complexity. For the first time the complementary combinational method of PSO and K-means clustering was considered.

Practical value. Since the optimization measures is widely used in bu-siness management, market analysis, engineering design and scientific exploration and other fields, the research results can be applied in in various fields. KDMPSO can effectively make up for the deficiency of the traditional PSO, and have achieved good results.

Keywords: *objective optimization, particle swarm optimization, k-means clustering, hybrid clustering, multi-swarm, global extreme*

Introduction. An optimization technique is the applied technology based on mathematics, which is used to resolve various combinational optimization problems. It refers to the process to search a group of parameters to make the objective function maximum or minimum when meeting certain constraints [1]. So far, many branches have emerged, such as linear programming, integer programming, non-linear programming, geometric programming, dynamic programming and stochastic programming and the application of optimization technique in each field has generated tremendous economic and social benefits. However, with the increase of the scale of the object to be processed, optimization problem has also become increasingly complex. The newly emerging swarm intelligence method in recent years is very effective in solving these problems [2].

The concept of swarm intelligence originates from the observation and research of the swarm behaviors of such gregarious animals as bees, ants, and birds. It is quite common for us to see a shoal of fish, a flock of birds or a group of other animals [3]. Their clustering behaviors are good for their foraging. Generally, the populations of these animals do not have a unified commander; therefore, there must be certain potential ability or rule to ensure the synchronization of these behaviors. Eberhart and Kennedy are the first to come up with PSO, simulating the coordination and collaboration between the individuals and the group in the group activities such as foraging and immigration of the gregarious animals [4]. Currently, progress has been made in PSO in solving optimization problems, but the theoretical foundation of this algorithm is still quite weak. Most of the researchers have been focusing on how to accelerate the convergence velocity of PSO and avoid premature convergence. Some researchers make analysis on the convergence of the algorithm while the majority investigates the structure and performance improvements of the algorithm, including parameter analysis, topology structure, maintenance of particle diversity, algorithm fusion and performance comparison. Since the particle of PSO cluster towards its previous optimal position and neighborhood or swarm previous optimal position, forming the rapid convergence effect of the particle swarm and making it easy to be trapped in local extremum, premature convergence or stagnation phenomenon [5]. In the meanwhile, the performance of PSO also relies on its parameters. In order to overcome the above-mentioned defects, the researchers from different countries have brought about various improvement measures. Based on the basic principle of PSO, this paper will improve this algorithm so as to enhance the optimization performance of PSO in the complex and high-dimensional situations, make it possible for it to effectively avoid the prematurity problem in the search process and increase its stability by integrating K-means clustering.

This paper firstly introduces the basic principle of PSO. Then it analyzes K-means clustering. On this basis, it raises the corresponding improvement strategy for the main defects of PSO and proposes the dynamic multi-swarm particle swarm optimization based on K-means clustering. Finally, in order to test the performance of the algorithm of this paper, it makes experimental comparison between two commonly-used improved PSO and the algorithm of this paper and make brief analysis of the experimental results.

Principles and technological process of particle swarm algorithm. ***Principles of particle swarm algorithm.*** The basic idea of PSO comes from the research on the foraging behavior of birds. Imagine such a scene: a flock of birds searches for food randomly in a region where there is only a piece of food and other birds do not know where to find it but they know how far they are away from the food. Under this circumstance, what is the optimal strategy to find the food? The simplest and the most effective method is to search the surrounding area of the bird that is nearest to the food [6]. In PSO, the potential solution to each optimization problem is a bird in the search space, which is called particle. Afterward, the particles will follow the current optimal particle and search in the solution space. Every particle has a fitness value determined by the optimization function and a velocity, which determines their flying direction and distance. The final optimal solution can be found after several rounds of iterations. Every particle updates itself through two factors in every iteration, one is the optimal solution searched by the particle itself, which is called "self-consciousness" and which is closely related to the local search performance of the algorithm and the other is the so-called "swarm intelligence", which is the optimal solution found by the entire swarm. In the velocity update, it leads the entire swarm to cluster the global optimum. The particle velocity can determine the search path and it searches along the gradient direction at fast search velocity. In most conditions, all the particles can convergence to the optimal solution [7].

Assume that in a D-dimensional objective search space, N particles form a swarm and the ith particle refers to a D-dimensional. X_i is the position vector of the ith particle.

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), i = 1, 2, \cdots, N \circ .$$

The "flying" velocity of the ith particle V_i is also a D-dimensional vector, which is recorded as

$$V_i = (v_{i1}, v_{i2}, \cdots, v_{iD}), i = 1, 2, \cdots 3 \circ .$$

The optimal position the ith particle has searched so far is called individual extremum p_{best} and recorded as

$$p_{best} = (p_{i1}, p_{i2}, \cdots, p_{iD}), i = 1, 2, \cdots, N \circ .$$

The optimal position the entire particle swarm has searched so far is called global extremum g_{best} and recorded as

$$g_{best} = (p_{g1}, p_{g2}, \cdots, p_{gD}) .$$

When finding these two optimal values, the particle updates its own velocity and position according to the following Formulas (1, 2)

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}); \quad (1)$$

$$x_{id} = x_{id} + v_{id} . \quad (2)$$

Here, v_i is the flying speed of ith particle, x_i is the position of ith particle, w is the inertia weight, c_1 and c_2 are learning factors and r_1 and r_2 are random numbers within $[0,1]$. The formula includes three parts. The first part is the inertia part and it reflects the motion habit of the particle and means that the particle has the trend to maintain its previous velocity. The second part is the cognition modal. It is a vector pointing from the current point to the optimal point of the particle and it means that the motion of the particle results from its experience. The third part is the social modal and it reflects the previous swarm experience of collaboration and knowledge share of the particles. These three parts co-decide the spatial search capacity of the particle. Each particle of PSO uses uni-directional information flow way to exchange information. The entire search update process is the process following the current optimal solution. Every particle has the memorability to make the neighborhood operator impossible to damage the searched solution [8].

Technological process of particle swarm algorithm. Every particle is evaluated according to the well-defined fitness function, which is related to the problem to be solved. The procedures of PSO are classified as follows, as shown in Fig. 1.

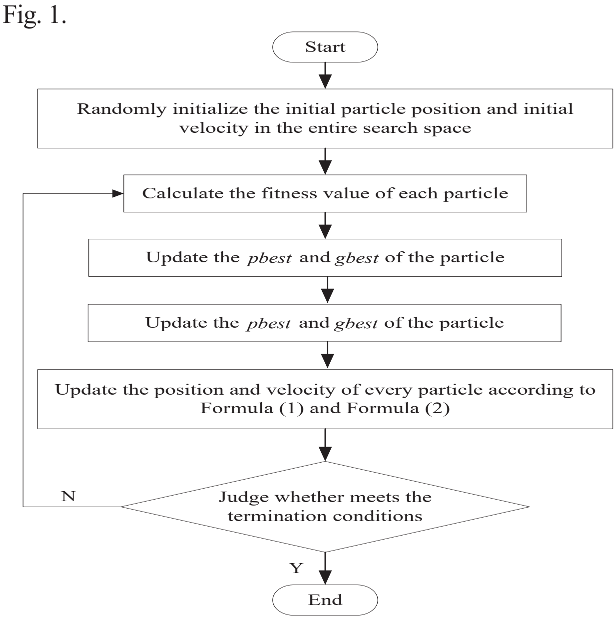

Fig. 1. Procedures of PSO

Step 1: Initialize the particle swarm, including the population scale as well as the initial velocity and position of the particle.

Step 2: Calculate the fitness of every particle, save the best position *pbest* and fitness of every particle and select the particle with the best fitness as the *gbest* of the population.

Step 3: Compare the fitness value of every particle with the individual extremum *pbest*. If it is better, replace the *pbest*.

Step 4: Compare the fitness value of every particle with the global extremum $gbest$. If it is better, replace the $gbest$.

Step 5: Update the speed and position of particles according to the Formulas (1, 2).

Step 6: Judge whether the search result satisfies the constraint conditions set by the algorithm (normally, reach the satisfactory fitness value or the preset maximum iterations). If it does not, turn back to Step 2, if it does, stop the iteration and output the optimal solution.

K-means clustering algorithm. Clustering is to classify the data objects in the data space. The data objects of the same class are quite similar while those of different classes are different. K-means clustering problem assumes that there is a sample set $X=\{x_1, x_2, x_3,\dots, x_n\}$ which have n data to be clustered. The problem of K-means clustering is to find a partition of X, i.e. $P_k=\{C_1, C_2, C_3,\dots, C_k\}$ to make the objective function $f(p_k)=\sum_{i=1}^{k}\sum_{x\in K_i}\delta(x,p_i)$ minimum. Its steps are as follows [9].

Step1: Initialization. Select k representative points $p_1, p_2, p_3 ... p_k$, and select k cluster centers.

Step2: Build k spatial clustering table $K_1, K_2 ... K_k$, calculate the distance between each data point and cluster centers.

Step3: Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.

Step4: Classify the sample set X one by one according to the minimum distance method.

$$f(p_k)=\arg\min_i \delta(x,p_i), add(x,K_j) , \quad (3)$$

where x is the sample, p_i is the center of every class, δ is the distance between each sample and the center of the class, $add(x,K_j)$ is classification coefficient, $i,j=[1,2,3,...k]$.

Step5: Calculate $f(p_k)$ and the clustering mean value with various clustering tables and take them as the new representative points of various clusters.

Step6: Recalculate the distance between each data point and new obtained cluster centers. If $f(p_k)$ remains the same or the representative point does not change, stop it, otherwise, turn to Step 2.

Step7: Output operation result

$$f(p_k)=\sum_{i=1}^{k}\sum_{x\in K_i}\delta(x,p_i) ,$$

where δ is the distance between each sample and the center of the class. x is the sample, p_i is the center of every class.

In K-means clustering, the main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different results. An initial partition needs to be determined according to the initial clustering center and then optimize the initial partition. Therefore, the better choice is to place them as much as possible far away from each other. From the framework of K-means clustering, it can be seen that this algorithm needs to classify and adjust the samples continuously and take each point belonging to a given data set and associate it to the nearest center. K-means clustering is relatively flexible and highly efficient in processing big data. When the resulted cluster is intensive and the difference between the clusters is significant, it has a better effect. Different initial values of K-means clustering may lead to different results [10]. K-means clustering algorithm operating results are shown in Fig. 2.

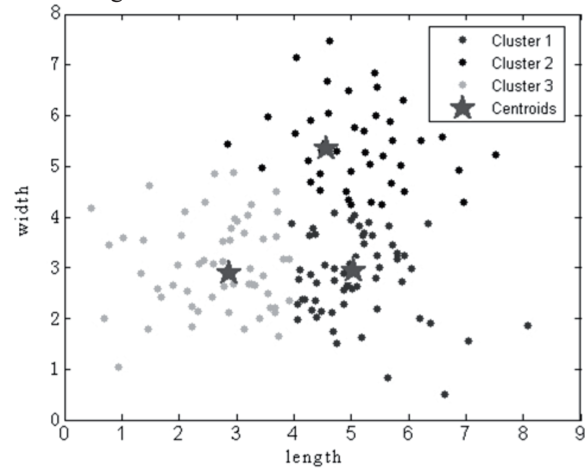

Fig. 2. Comparison of the algorithm performance test

Procedure of the K-means clustering dynamic multi-swarm PSO. The average particle distance shows the degree of dispersion of the distribution of every particle in the swarm. Assume that Q is the population size, then the average particle distance $D(t)$ of the tth generation is

$$D(t)=\frac{1}{Q.L}\sum_{i=1}^{S}\sqrt{\sum_{d=1}^{n}(x_{id}^t-\overline{r_d})^2} , \quad (4)$$

where L is the individual with the furthest distance in the solute-on space, n is the number of dimensions of the vector in the solution space, x_{id}^t is the d-dimensional component value of the ith particle in the tth generation, $\overline{r_d}$ is the d-dimensional mean component value.

Initialize the particle velocity with the formula $V_{ik}(d)=rand()*\omega*\phi(d)$. Here, $rand()$ is a random number within $[0,1]$ and $\phi(d)$ is the difference of the d-dimensional maximum and minimum values, ω is the weight factor. Adjust the initial velocity of every dimension with ω in order to enhance the global optimization ability of the algorithm.

In the clustering algorithm based on particle swarm, particle is represented as the center of k clusters. Every particle is represented as $X_i=\{c_1^i, c_2^i,\dots,c_k^i\}$. Here, c_j^i is the center of the jth cluster of the ith particle. Every particle represents a candidate clustering scheme and evaluates the clustering scheme in every iteration of the particle swarm so as to make the particle swarm move towards the optimal clustering and adopt the following fitness function.

$$f(x_i)=\frac{L}{1+\sum_{j=1}^{n}J_j} . \quad (5)$$

Here, $\sum_{j=1}^{n} J_j$ is the total sum within cluster dispersion, $J_j = \sum_{S_i \in X_j} d(S_i, X_j)$ is the total distance from the jth -cluster samples to the center of such cluster and L is the constant, L is the individual with the furthest distance in the solution space.

The implementation steps of KDMPSO are as follows.

Step1: Initialize the population. Calculate the particle dimensions according to (4), select algorithm to calculate the initial position and velocity of the particle with K-means clustering center, generate the initial particle swarm. Assume that $t = 1$ and take it as the initial clustering partition. Calculate every clustering center according to the partition result and take it as the initial position encoding. Calculate the fitness function of the particle according to (5) and initialize the particle velocity.

Step2: Calculate the global optimal solutions which have been searched by all individuals of the population and compare its fitness with the fitness of the best position it has passed. If it is better, update the individual extremum and the optimal position of that individual.

Step3: Set the convergence sign and the maximum iterations of each individual. The algorithm begins to iterate, classify according to the dimensions and obtain the dimension set.

Step4: For every dimension, update the inertia weight W as well as the position and velocity of the particle according to (1, 2). Compare the fitness value of every particle and that of the best position all particles in this dimension have passed. If it is better, update the optimal value and the optimal position of this dimension.

Step5: Calculate the corresponding fitness function value to every particle, compare their fitness, update the global optimal value G and the optimal position X of the particle as well as the values of individual extremum p_{best} and global extremum g_{best}, otherwise keep the particle position unchanged.

Step6: Repartition all the particles of the new generation according to the nearest neighbor rule. Calculate every clustering center according to the partition result, update the position, velocity and dimensions of the particle and recalculate the new clustering center as the new position of the particle through K-means clustering rule.

Step7: Judge whether t meets $t \le M$ (M is the minimum number of iterations). If not, turn to Step 2, if it does, terminate the circulation and output the global optimal particle.

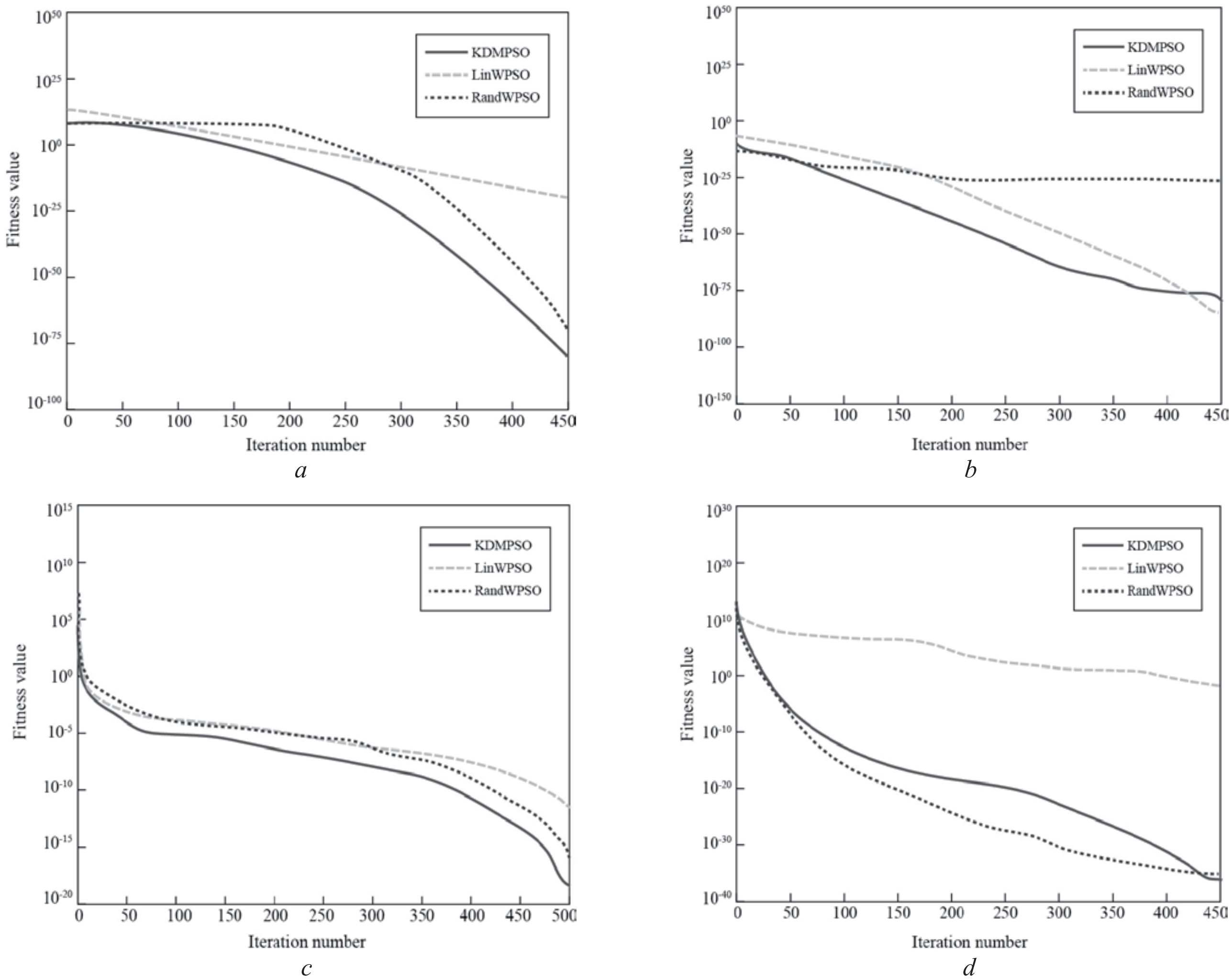

Fig. 3. Comparison of the algorithms performance test: a – test function 1; b – test function 2; c – test function 3; d – test function 4

Performance test of the K-means clustering dynamic multi-swarm PSO. In order to verify the effectiveness of the algorithm of this research, the KDMPSO with the common improved PSO, namely Linearly Decreasing Weight PSO (LinWPSO) and Random Weight PSO (RandWPSO) have been compared and the following 4 optimization problems have been resolved (Test function and the number of dimensions is 20); each of the optimal solutions for these 4 optimization problems is 0. The experimental parameters were set as follows: the population scale of PSO is $N = 30$, $c_1 = 1.3$, $c_2 = 2.0$, $w_{\max} = 0.8$ and $w_{\min} = 0.2$. In order to compare the search process of the extremum optimization of these three algorithms, the following graph is the evolution curve of the average fitness value after 30 independent operations on the test function by the three algorithms. Comparison of three algorithms performance test as shown in Fig. 3.

For $f_1(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$, the feasible solution space is $[-100,100]^n$.

For $f_2(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$, the feasible solution space is $[-100,100]^n$.

For $f_3(x) = \sum_{i=1}^{n} x_i^2$, the feasible solution space is $[-100,100]^n$.

For $f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$, the feasible solution space is $[-50,50]^n$.

It can be seen from the optimization numerical results of the above-mentioned algorithms that whether in the solution quality and the function evaluations, the optimization solution accuracy and robustness performance of KDMPSO are better than the other two methods and KDMPSO can find all the optimal solutions for the 4 test functions. In one word, KDMPSO has demonstrated excellent performance in the optimization problems. On one hand, it can find the solution with higher quality; on the other hand, it requires less evaluations and lower time complexity. Besides, the variance of the solutions through KDMPSO is smaller, suggesting that the performance of the hybrid algorithm is stable and robust. KDMPSO can quickly converge to the global optimal solution within certain evolutionary generations. It has both strong global search capacity and fast convergence velocity and it effectively reduces the premature convergence of the traditional PSO and the phenomenon to be trapped in local optimal solution.

Discussion and conclusion. In processing the optimization problems and in order to overcome sensitivity of K-means clustering in the initial value and the easiness of PSO to get trapped in prematurity, the dynamic multi-swarm particle swarm optimization based on K-means clustering has been proposed, which integrates K-means clustering into particle swarm optimization and makes proper clustering of the particles through internal spatial features. This algorithm not only improves the local search ability of the particle swarm optimization, but also increases the population diversity, avoids the emergence of premature convergence and increases the algorithm accuracy and efficiency. The experimental result has proved the effectiveness of the algorithm.

Acknowledgments. This work was supported by School Youth Research Foundation of Jiangsu University of Technology (No.KYY13029).

References / Список літератури

1. Min-Yuan Cheng, Kuo-Yu Huang and Hung-Ming Chen (2012), "K-means Particle Swarm Optimization with Embedded Chaotic Search for Solving Multidimensional Problems", *Applied Mathematics and Computation*, vol. 219, no.6, pp. 3091–3099.

2. Satish Gajawada and Durga Toshniwal (2012), "Projected Clustering Using Particle Swarm Optimization", *Procedia Technology*, vol.35, no. 4, pp. 360–364.

3. Sung-Kwun Oh, Wook-Dong Kim, Witold Pedrycz and Su-Chong Joo (2012), "Design of K-means Clustering-based Polynomial Radial Basis Function Neural Networks (PRBF NNS) realized with the aid of Particle Swarm Optimization and Differential Evolution", *Neurocomputing*, vol. 78, no.1, pp. 121–132.

4. Shafiq Alam, Gillian Dobbie, Yun Sing Koh, Patricia Riddle and Saeed Ur Rehman (2014), "Research on Particle Swarm Optimization based clustering: a systematic review of literature and techniques", *Swarm and Evolutionary Computation*, vol.17, no.8, pp. 1–13.

5. Cheng-Lung Huang, Wen-Chen Huang, Hung-Yi Chang, Yi-Chun Yeh and Cheng-Yi Tsai (2013), "Hybridization strategies for Continuous Ant Colony Optimization and Particle Swarm Optimization applied to data clustering", *Applied Soft Computing*, vol.13, no.9, pp. 3864–3872.

6. Lige S. Hage and Ramsey F. Hamade (2013), "Segmentation of histology slides of cortical bone using Pulse Coupled Neural Networks optimized by Particle Swarm Optimization", *Computerized Medical Imaging and Graphics*, vol.37, no.7-8, pp. 466–474.

7. Maryam Nasiri (2012), "Fetal electrocardiogram signal extraction by ANFIS trained with PSO Method", *International Journal of Electrical and Computer Engineering (IJECE)*, vol.2, no. 2, pp. 247–260.

8. Lakshmi Ravi (2012), "PSO based optimal power flow with hybrid distributed generators and UPFC", *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol.10, no. 3, pp. 409–418.

9. Yau King Lam and Peter W.M. Tsang (2012), "Exploratory K-Means: a new simple and efficient algorithm for gene clustering", *Applied Soft Computing*, vol.12, no.3, pp. 1149–1157.

10. MdAnisur Rahman and MdZahidul Islam (2014), "A hybrid clustering technique Combining a novel genetic algorithm with K-Means", *Knowledge-Based Systems*, vol.71, no. 11, pp. 345–365.

Мета. Цільова оптимізація – дуже важливий напрям у наукових дослідженнях і практичній діяльності. Робота присвячена вивченню комбінування прийомів оптимізації методом рою часток і кластеризації методом K-середніх.

Методика. Проведено всесторонній глибокий аналіз кластеризації методами рою часток і K-середніх, знайдені стратегії її поліпшення шляхом комбінування цих методів. Стосовно завдань як безперервної, так і дискретної оптимізації, продемонстровані гарні показники глобального пошуку, ефективне усунення передчасної конвергенції, властивій класичній оптимізації методом рою часток.

Результати. Розроблена динамічна багатороєва оптимізація методом рою часток на основі кластеризації методом K-середніх (KDMPSO), яка є гібридним алгоритмом кластеризації, що поєднує в собі кластеризацію методами рою часток і K-середніх, безвідмовно знаходить глобальний екстремум у різних завданнях. Поєднані переваги оптимізації методом рою часток і кластеризації методом K-середніх, вирішені проблеми сходження до локального оптимуму та низької ефективності класичного алгоритму оптимізації методом рою часток для складних завдань оптимізації. Підтверджена стабільність роботи алгоритму, підвищення його точності й здібність до підтримки різноманітності популяції, запобігання передчасній конвергенції.

Наукова новизна. Вивчена багатороєва оптимізація методом рою часток і кластеризації методом K-середніх (KDMPSO). У процесі ітерації, алгоритм оптимізації методом рою часток схильний застрявати в локальному оптимумі, наводячи до передчасної конвергенції, тоді як метод K-середніх широко використовується у кластеризації завдяки простоті в реалізації, будучи високоефективним алгоритмом лінійної тимчасової складності. Уперше розглянуто метод взаємодоповнюючого комбінування оптимізації методом рою часток і кластеризації методом K-середніх.

Практична значимість. Результати роботи застосовані на практиці в різних сферах, оскільки оптимізаційні заходи приймаються в управлінні підприємствами, дослідженні ринку, технічному проектуванні, наукових дослідженнях і так далі. Алгоритм KDMPSO не має недоліків, властивих класичному алгоритму оптимізації методом рою часток, і показує гарні результати.

Ключові слова: *цільова оптимізація, оптимізація методом рою часток, кластеризація методом K-середніх, гібридна кластеризація, мультирой, глобальний екстремум*

Цель. Целевая оптимизация – очень важное направление в научных исследованиях и практической деятельности. Работа посвящена изучению комбинирования приемов оптимизации методом роя частиц и кластеризации методом K-средних.

Методика. Проведен всесторонний глубокий анализ кластеризации методами роя частиц и K-средних, найдены стратегии ее улучшения путем комбинирования этих методов. Применительно к задачам как непрерывной, так и дискретной оптимизации, продемонстрированы хорошие показатели глобального поиска, эффективное устранение преждевременной конвергенции, свойственной классической оптимизации методом роя частиц.

Результат. Разработана динамическая многороевая оптимизация методом роя частиц на основе кластеризации методом K-средних (KDMPSO), которая является гибридным алгоритмом кластеризации, совмещающим в себе кластеризацию методами роя частиц и K-средних, безотказно находит глобальный экстремум в различных задачах. Совмещены преимущества оптимизации методом роя частиц и кластеризации методом K-средних, решены проблемы схождения к локальному оптимуму и низкой эффективности классического алгоритма оптимизации методом роя частиц для сложных задач оптимизации. Подтверждена стабильность работы алгоритма, повышение его точности и способность к поддержанию разнообразия популяции, предотвращению преждевременной конвергенции.

Научная новизна. Изучена многороевая оптимизация методом роя частиц и кластеризации методом K-средних (KDMPSO). В процессе итерации, алгоритм оптимизации методом роя частиц склонен застревать в локальном оптимуме, приводя к преждевременной конвергенции, тогда как метод K-средних широко используется в кластеризации благодаря простоте в реализации, будучи высокоэффективным алгоритмом линейной временной сложности. Впервые рассмотрен метод взаимодополняющего комбинирования оптимизации методом роя частиц и кластеризации методом K-средних.

Практическая значимость. Результаты работы применимы на практике в различных сферах, поскольку оптимизационные меры принимаются в управлении предприятиями, исследовании рынка, техническом проектировании, научных исследованиях и т.д. Алгоритм KDMPSO не имеет недостатков, присущих классическому алгоритму оптимизации методом роя частиц, и показывает хорошие результаты.

Ключевые слова: *целевая оптимизация, оптимизация методом роя частиц, кластеризация методом K-средних, гибридная кластеризация, мультирой, глобальный экстремум*

Рекомендовано до публікації докт. техн. наук В.В. Гнатушенком. Дата надходження рукопису 20.11.14.