International Journal of Computer Applications Technology and Research
Volume 14–Issue 09, 23 – 27, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1409.1004

An Improved K-means Clustering-Based Co-evolutionary Genetic Algorithm

Zehua Lv
Chengdu University of Information Technology
College of Communication Engineering
ChengDu, China

Wei Ma
Chengdu University of Information Technology
College of Communication Engineering
ChengDu, China

Chengyu Hou*
Chengdu University of Information Technology
College of Communication Engineering
ChengDu, China

Abstract: This article proposes an improved k-means clustering-based co-evolutionary genetic algorithm, which preserves the individuals closest to the center in each cluster, performs traditional genetic algorithm operations (selection, crossover, mutation) on the original population, selects the optimal subset of individuals from the new population generated by the genetic algorithm, merges the individuals retained by K-means with the optimal individuals generated by the genetic algorithm, and forms a new generation population. This hybrid algorithm combines the global search capability of K-means with the local fine search capability of genetic algorithm. Finally, this article uses the Alpine and other function to test and analyze the optimization of the final algorithm. The improved algorithm can quickly jump out of local optima and converge to the global optimum.
Keywords: Co-evolutionary Genetic Algorithm;Improved K-means Clustering;genetic algorithm;population;local optima

1. INTRODUCTION

Co-evolutionary Genetic Algorithms (CGA) is an intelligent optimization method that integrates evolutionary algorithms and coevolutionary theory. Its core mechanism simulates the process of different species (or populations) evolving together through interactions in nature. Unlike traditional genetic algorithms that only optimize a single population, CGA achieves optimization through dynamic interaction of multiple subpopulations - these populations may be in a competitive, cooperative, or symbiotic relationship, and the fitness of individuals is no longer directly calculated by a fixed function, but depends on the interaction results with other populations (for example, in a competitive relationship, the fitness of a predator is determined by the performance of its prey, while in a cooperative relationship, the population needs to collaborate to complete tasks in order to achieve high fitness). This design makes it more suitable for handling complex, multi-objective, and dynamically changing problems, especially in scenarios where traditional algorithms are difficult to model[1].

In practical applications, the advantages and challenges of CGA coexist. Its advantage lies in strong robustness, which can handle dynamic problems and multi constraint scenarios, and reduces the risk of premature convergence through group interaction; But at the same time, it also faces the problems of high complexity and high evaluation cost, requiring careful design of interaction rules to avoid one group excessively suppressing other groups. At present, CGA has been widely applied in fields such as game theory (such as chess AI strategy optimization), multi-objective engineering design (such as cost and efficiency balance), and distributed systems (such as multi robot collaboration), and its combination with deep learning and reinforcement learning is becoming a new research direction, further expanding its application boundaries[2].

The K-means clustering algorithm is a classic unsupervised learning method that divides a dataset into K clusters, resulting in high similarity within clusters and low similarity between clusters. It is widely used in fields such as data mining, pattern recognition, and image segmentation[3]. However, traditional K-means has drawbacks such as sensitivity to initial cluster centers, susceptibility to local optima, and the need to manually specify the number of clusters K. The improved K-means clustering algorithm optimizes for these problems, improving clustering performance and stability.

This article implements a hybrid optimization algorithm that combines K-means clustering and Co-evolutionary Genetic Algorithms. This hybrid optimization strategy is particularly effective in dealing with high-dimensional complex optimization problems, and can improve convergence speed while maintaining global search capability.

2. RELATED WORK

As an effective optimization algorithm, the co-evolutionary genetic algorithm has demonstrated promising application prospects in multiple fields. Many scholars have proposed distinctive solutions based on this algorithm for different practical problems. Here, we will introduce some relevant research results.Yin Jing et al. proposed an optimization model for scheduling services of multiple tower cranes with overlapping areas, while achieving collision free approaches and methods. A collaborative co evolutionary genetic algorithm (CCGA) was proposed to solve the model[4].Zhang Xinyuan et al. solved the above-mentioned problems related to a small number of lens datasets and introduced the Hilbert encoded co evolutionary genetic algorithm (HCCGA), which is a novel image enhancement algorithm designed for automatic generation of 3D lookup tables (3D LUTs)[5].Zhang Xinyuan et al. proposed a tree structured CC genetic algorithm (T-CCGA) specifically designed for our reconstruction task. Our goal is to overcome the limitations of current reconstruction algorithms and pave the way for more accurate and efficient fragment reconstruction methods[6].Kumari Monika and his team implemented a co evolutionary genetic algorithm in CloudSim simulation tool to find the optimal cost per configuration[7].Niwa et al. proposed an algorithm that effectively searches for dependency relationships between variables by introducing link trees into the CC method[8].Diniz et al. proposed a method for joint optimization of transmitter in-phase, quadrature, and internal polarization time deviations, amplitude mismatches, and bias voltages. This method is based on collaborative co evolutionary genetic algorithm, and

its fitness function is extracted from the directly detected reference quadrature amplitude modulation (QAM) signal generated at the transmitter[9].Li, Yuanzhang et al. proposed a quality of service (QoS) based co evolutionary genetic algorithm for web service composition, which fully considers the individual relationships between populations[10].

3. SYSTEM ALGORITHM

3.1 Co-evolution

Co evolution refers to the biological phenomenon in which two or more species in an ecosystem exert selection pressure on each other through long-term interactions, leading to the coordinated adjustment of their adaptive characteristics[11]. This interaction is widely present in relationships between predators and prey, parasites and hosts, symbiotic organisms, etc., manifested as adaptive matching between species in terms of morphology, physiology, behavior, and other aspects. For example, in the interaction between plants and pollinating insects, plants may evolve specific floral structures and chemical signals to attract specific insects, while insects simultaneously develop adapted feeding organs and behavioral patterns, which not only ensure their own resource acquisition but also promote plant reproduction; In the relationship between hosts and parasites, hosts may evolve immune defense mechanisms, while parasites may develop strategies to evade or inhibit these mechanisms. Co evolution is an important driving force for species adaptive evolution and is of crucial importance in maintaining the diversity and stability of ecosystems.

3.2 K-means Clustering

The K-means algorithm uses Euclidean distance as a metric to cluster samples, and its core idea is to achieve efficient data partitioning by minimizing the sum of squared distances from all sample points within the cluster to the corresponding cluster center. The goal of this algorithm is clear: to minimize the sum of squared distances between sample points within each cluster and its center[12]. The essence of this goal is to maximize the similarity of data within the same cluster, thereby achieving optimal clustering results. The mathematical expression formula is as follows:

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| \mathrm{x}_i^{(k)} - \mu_k \right\|^2 \quad (1)$$

Among them, J is the total cost function, K is the number of clusters, $X_i^{(K)}$ is the i -th data point in class k , u_i is the cluster center of class k , n_k is the number of samples in class k .

K-means clustering is a commonly used unsupervised machine learning algorithm, mainly used to divide a dataset into K clusters with similar features. The core idea is to maximize the similarity of data points within a cluster and minimize the similarity of data points between clusters through iterative optimization[13]. The specific process is as follows: first, randomly select K data points as the initial cluster centers; Then calculate the distance between each data point and the center of each cluster (usually using Euclidean distance), and assign the data points to the nearest cluster; Then recalculate the new cluster center (i.e. the mean of all data points within the cluster) based on the data points within each cluster; Repeat the above allocation and update steps until the change in cluster center is less than the preset threshold or reaches the maximum iteration number, and finally obtain stable clustering results. K-means clustering is widely used in fields such as data mining, pattern recognition, and image processing due to its simple implementation and high computational efficiency.

3.3 Genetic Algorithm

Genetic algorithm (GA) is a meta heuristic algorithm based on the natural selection process, which is a subcategory of evolutionary algorithm (EA), which is a broader category. Genetic algorithms are commonly used to develop high-quality solutions for optimization and search problems by relying on biologically inspired operators such as mutation, crossover, and selection[14].

The design process of GA is shown in the following Figure 1:

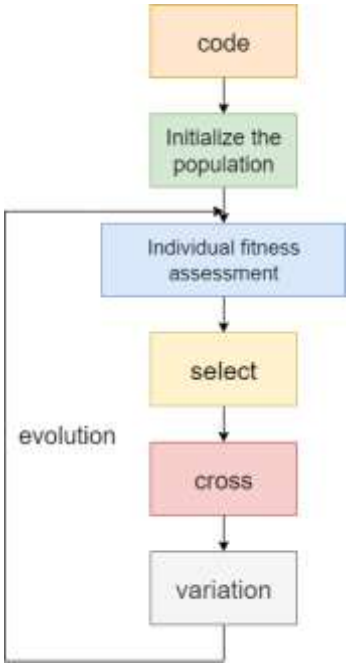

Figure 1 Genetic Algorithm Diagram

The algorithm steps of GA are as follows：

- Population initialization:Design appropriate initialization operations based on the characteristics of the problem to initialize N individuals in the population
- Individual evaluation:Calculate the fitness value of individuals in the population based on the optimized objective function
- Iterative settings:Set the maximum iteration count for the population and set the current iteration count to 1
- Individual selection:Design appropriate selection operators to select individuals in population P (g), and the selected individuals will enter the mating pool to form the parent population FP (g) for cross transformation to generate new individuals.
- Crossover operator:Determine whether the parent individual needs to perform a crossover operation based on the crossover probability pm (pre specified, usually 0.9). The crossover operator should be designed based on the characteristics of the optimized problem. It is the core of the entire genetic algorithm, and its design directly determines the performance of the entire algorithm.

International Journal of Computer Applications Technology and Research
Volume 14–Issue 09, 23 – 27, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1409.1004

- Mutation operator:Determine whether the parent individual needs to undergo mutation operation based on the mutation probability pc (pre specified, usually 0.1). The main function of mutation operators is to maintain population diversity and prevent the population from falling into local optima, so they are generally designed as a random transformation.

Following the crossover mutation process, the parent population gives rise to a new offspring population, with the number of population iterations increasing by 1. The next round of iterative operations then proceeds (skipping to Step 4) and continues until the maximum number of iterations is reached. Through the crossover operation, the original two individual combinations produce two new ones, which is comparable to conducting a search within the solution space — each individual within this space represents a feasible solution.

Once a round of genetic mutation is completed, these new offspring are evaluated using a fitness function. If the function confirms they possess adequate fitness, they will replace chromosomes in the population that have insufficient fitness. The loop terminates when any of the following conditions are met: first, after X iterations, there has been no significant overall change; second, the algorithm has completed the pre-defined number of evolutions; third, the fitness function has reached a pre-specified value[15].

3.4 CO-EVOLUTIONARY-GENETIC ALGORITHM

The steps of the algorithm are shown in the figure below：

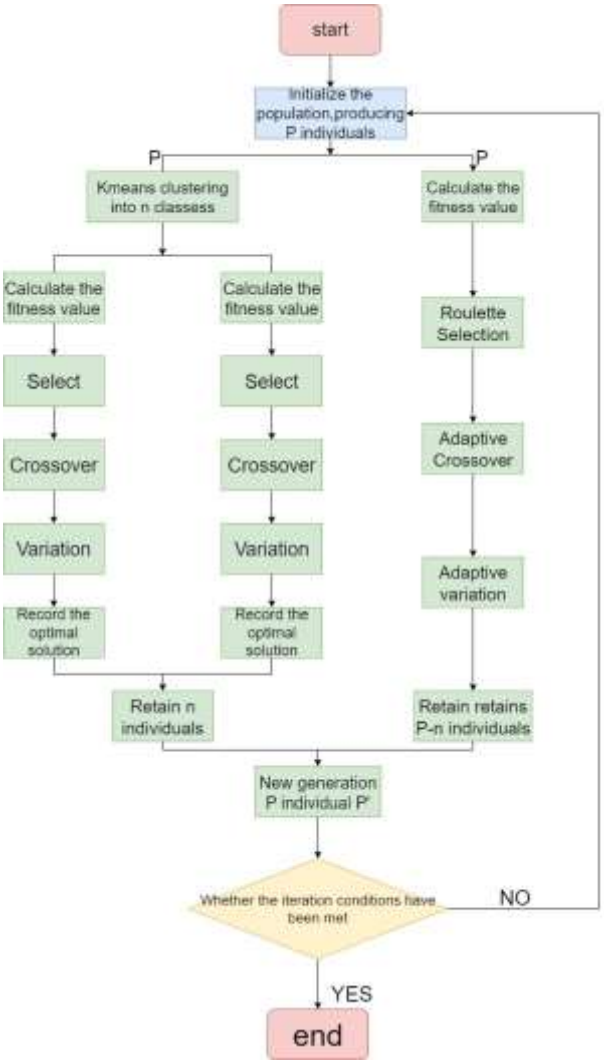

Figure 2 Algorithm flow chart

(1) Initial population construction: The real number coding strategy is used to randomly generate an initial population composed of N particles, providing basic data samples for subsequent algorithm iterations.

(2) Dual-path parallel processing: Two core operations are performed simultaneously for the generated initial population - k-means cluster analysis and adaptive genetic operation to achieve multi-dimensional optimization of the population.

(3) Clustering optimization and optimal value screening: The population is divided into n independent clusters by the k-means clustering algorithm, and the adaptation calculation is performed for each cluster, and the optimal values in each cluster are screened out and retained, and finally n high-quality individuals are obtained.

(4) Genetic operation and individual screening: The roulette selection mechanism is used to complete adaptive genetic operations with adaptive cross operators and mutation operators, and P-n individuals that meet the optimization goals are screened and retained from the population.

(5) Generation of new generation populations: Integrate the n optimal values retained in step (3) and the P-n

International Journal of Computer Applications Technology and Research
Volume 14–Issue 09, 23 – 27, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1409.1004

individuals screened in step (4) to construct a new generation population with better structure.

(6) Iterative termination judgment: Check whether the preset termination conditions are met (such as the number of iterations reaching the standard, optimal value convergence, etc.): if so, output the final optimization result and terminate the algorithm. If it is not satisfied, the new generation population is used as the new initial population, and the next round of iteration process is started by returning to step (3).

4. EXPERIMENTS AND ANALYSIS

To further confirm how the improved co-evolutionary genetic algorithm based on k-means clustering performs in more complex and varied optimization scenarios, a number of supplementary typical test functions can be chosen for simulation validation. These supplementary validation functions are grouped into three types: multi-peak complex functions, high-dimensional nonlinear functions, and discontinuous/noise-containing functions. These categories not only add to the single-peak and multi-peak functions already applied in the paper but also aim at the potential optimization difficulties the algorithm might encounter in real-world use.

The specific supplementary test function is as follows:

(1) Alpine Function

$$f(x)=\sum_{i=1}^{n}|\, x_i \sin x_i + 0.1x_i \,| \tag{2}$$

(2) Michalewicz Function

$$f(x)=-\sum_{i=1}^{n}\sin(x_i)*\left[\sin(\frac{ix_i^2}{\Pi})\right]^{2m} \tag{3}$$

(3) Dixon-Price Function

$$f(x)=(x_i-1)^2+\sum_{i=2}^{n} i*(2x_i^2-x_{i-1})^2 \tag{4}$$

(4) Noisy Rastrigin Function

$$f(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\Pi x_i)+10\right]+\varepsilon*randn \tag{5}$$

The algorithm parameters for this experiment are set as follows: population size of 100, iteration number of 100, initial function dimension of 10, K-means clustering number of 30, crossover probability of 0.8, mutation probability of 0.01, genetic algorithm retains the number of individuals and calculates 70 through "population size cluster number".The test results are shown in the following Figure 3-4:

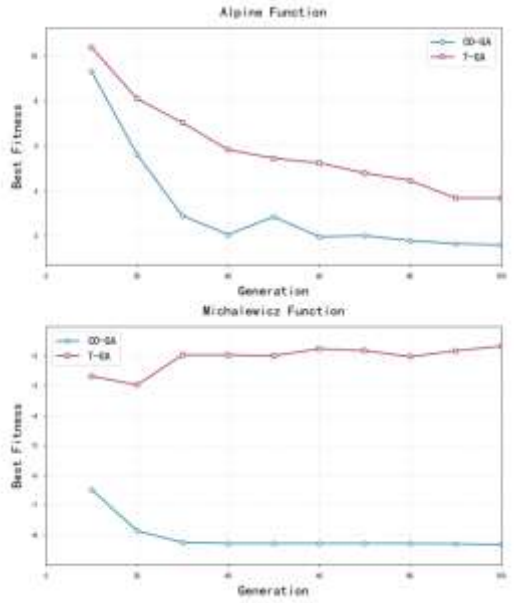

Figure 3 Best Fitness about Generation

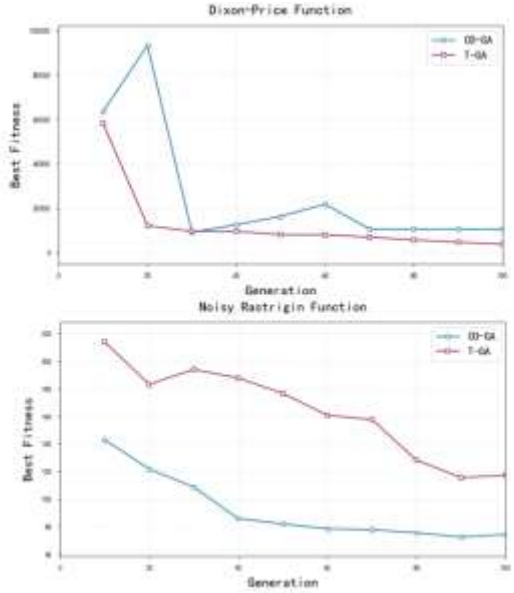

Figure 4 Best Fitness about Generation

It can be seen that compared to traditional genetic algorithms, the optimal fitness of an improved k-means clustering-based co-evolutionary genetic algorithm has been greatly improved.

To further validate how well the enhanced algorithm performs across various generations, this study additionally compared the optimal search outcomes against those of the standard genetic algorithm. The parameters were configured as follows: a population size of 100, generation counts of 100 and 500 respectively, and a dimensionality of 50. These comparisons were conducted using different test functions, and the results of the optimal solution comparisons are presented in Table 1:

Table 1. Best Fitness Comparison Table

functions	Population size	genetic algebra	Dime-nsion-ality	Standard Genetic Algorithm	Algorithm of this paper
Alpine Function	100	100	50	10.258	4.624
Michalewicz Function	100	100	50	-4.253	-8.525
Dixon-Price Function	100	100	50	1025	424
Noisy Rastrigin Function	100	100	50	8.547	3.223

5. CONCLUSION

To enhance population diversity, the algorithm first employs k-means clustering on existing populations to generate new individuals. Concurrently, it integrates adaptive genetic operations—enabling individuals to make corresponding adjustments based on their current fitness values—which effectively mitigates the risk of premature population maturity. After each evolutionary generation, further interaction is introduced between the two populations; this mechanism empowers the population to break free from local optima efficiently. Ultimately, these synergistic design choices lead to a notable enhancement in the algorithm's search capability.

REFERENCES

[1] Alcaraz-Herrera,,Hugo,Cartlidge,,& John.(2021).Substitution of the fittest: A novel approach for mitigating disengagement in coevolutionary genetic algorithms.arXiv.

[2] Zaki,,Tomas,Zeitrag,,Yannik,Neves,,Rui,Figueira,,Jose, & Rui.(2024).A cooperative coevolutionary genetic programming hyper-heuristic for multi-objective makespan and cost optimization in cloud workflow scheduling.COMPUTERS & OPERATIONS RESEARCH,172.

[3] Ponnuswamy,,Priya,Palaniappan,,Shabariram,& Chokkalingam.(2024).Prediction of wind energy location by parallel programming using MPI-based KMEANS clustering algorithm.ENERGY SOURCES PART A-RECOVERY UTILIZATION AND ENVIRONMENTAL EFFECTS,46(1),5451-5473.

[4] Yin,,Jing,Li,,Jiahao,Yang,,Ahui,Cai,,& Shunyao.(2024).Optimization of service scheduling problem for overlapping tower cranes with cooperative coevolutionary genetic algorithm.ENGINEERING CONSTRUCTION AND ARCHITECTURAL MANAGEMENT,31(3),1348-1369.

[5] Zhang,,Xinyuan,Yang,,Boda,Ou,Hu,,& Yue.(2024).Hilbert-encoded Cooperative Coevolutionary Genetic Algorithm for Few-shot Learning.

[6] Zhang,,Xin-Yuan,Yang,,Jin-Hao,Gong,,Yue-Jiao,Zhan,,Zhi-Hui,Zhang,,& Jun.(2025).Tree Structured Cooperative Coevolutionary Genetic Algorithm for Fragment Reconstruction.IEEE Transactions on Evolutionary Computation.

[7] Kumari,,Monika,Sahoo,,& Gadadhar.(2019).Cost-Effective Resource Provisioning in Cloud Using Cooperative Coevolutionary Genetic Algorithm.

[8] Niwa,,Takatoshi,Ihara,,Koya,Kato,,& Shohei.(2020).Cooperative coevolutionary genetic algorithm using hierarchical clustering of linkage tree.

[9] Diniz,,Julio,Cesar,Medeiros,Da,Ros,,Francesco,da,Silva,,Edson,Porto,Jones,,Rasmus,Thomas,Zibar,,& Darko.(2018).Optimization of DP-M-QAM Transmitter Using Cooperative Coevolutionary Genetic Algorithm.JOURNAL OF LIGHTWAVE TECHNOLOGY,36(12),2450-2462.

[10] Li,,Yuanzhang,Hu,,Jingjing,hujingjing@***,Wu,,Zhuozhuo,Liu,,Chen,Peng,,Feifei,Zhang,,& Yu.(2018).Research on QoS service composition based on coevolutionary genetic algorithm.Soft Computing - A Fusion of Foundations, Methodologies & Applications,22(23),7865-7874.

[11] Li,,Guangpeng,Li,,Li,Cai,,& Guoyong.(2025).A two-stage coevolutionary algorithm based on adaptive weights for complex constrained multiobjective optimization.APPLIED SOFT COMPUTING,173.

[12] Ponnuswamy,,Priya,Palaniappan,,Shabariram,& Chokkalingam.(2024).Prediction of wind energy location by parallel programming using MPI-based KMEANS clustering algorithm.ENERGY SOURCES PART A-RECOVERY UTILIZATION AND ENVIRONMENTAL EFFECTS,46(1),5451-5473.

[13] Li,,Zhanjiang,Yuan,,Yixiao,Sun,,Tianning,Li,,& Pengfei.(2023).Early warning model of credit risk for family farms and ranches in Inner Mongolia based on Probit regression-Kmeans clustering.MATHEMATICAL BIOSCIENCES AND ENGINEERING,20(5),8546-8560.

[14] Sabry,& Fouad.(2023).Genetic Algorithm.One Billion Knowledgeable.

[15] Alkafaween,,Esra'a,Hassanat,,Ahmad,Essa,,Ehab,Elmougy,,& Samir.(2024).An Efficiency Boost for Genetic Algorithms: Initializing the GA with the Iterative Approximate Method for Optimizing the Traveling Salesman Problem-Experimental Insights.APPLIED SCIENCES-BASEL,14(8),3151.