APPLICATION OF ANT COLONY OPTIMIZATION ALGORITHM BASED ON CLOUD MODEL IN ROUTING PROBLEMS

Xu LI[1], Zhengyan LIU[2*]

A cloud model-based ant colony optimization algorithm was proposed in this research to address the problem of difficult balance between exploring the search domain and developing known solutions in ant colony optimization algorithm. Deep analysis of the characteristics of cloud models could provide reasonable settings of cloud model parameters. Through simultaneous search for excellent solutions to the problem to replace the global optimal solution, the global search mode of the algorithm was reconstructed and ultimately an adaptive adjustment mechanism was developed to maintain the balance of the algorithm. In simulation experiments, a typical routing problem was adopted for solution and the obtained experimental results revealed that the proposed algorithm had excellent characteristics.

Keywords: Cloud mode; Adaptive mechanism; Ant colony optimization; Routing problem

1. Introduction

Routing problems, such as vehicle routing and traveling salesman problems, typically refer to the problem of generating a path or a set of paths in a network or a part of a network under certain constraints and optimization objectives. As a typical combinatorial optimization problem, routing problem has been extensively investigated and applied both theoretically and practically [1-3].

As a typical NP hard problem, routing problems become exceptionally complex to solve as problem size is increased, resulting in a “combinatorial explosion”. Therefore, most traditional optimization methods fail to effectively find the optimal solutions. Several intelligent optimization algorithms that combine evolutionary strategies and heuristic information have been proposed, which can effectively solve such problems due to their high performance and convergence in global search [4-7]. Among them, several researchers have introduced constructive ant colony algorithms and achieved favorable results in solving routing problems. Wu et al. [8] developed a game mechanism, selection evolution mechanism, and

[1] School of Computer and Information Engineering, Fuyang Normal University, Fuyang, China, e-mail: 16556793@qq.com
[2] School of Computer and Information Engineering, Fuyang Normal University, Fuyang, China, e-mail: zhyliu@fynu.edu.cn

dynamic jitter mechanism in ant colony algorithm. These strategies have revealed good performance in solving large-scale TSP problems. Elhassania [9] modified ant colony algorithm to solve path problems with capacity constraints, developed a new mathematical model, and achieved good experimental results. Marco [10] applied exploratory movement of ants to modify the probability law establishing a new version of ant colony algorithm, which achieved good results in solving traveling salesman problem.

However, ant colony algorithm has a key issue, which is maintaining the balance between developing known optimal solution and exploring unknown feasible solutions. This research has focused on this issue and explored the mechanism of combining cloud model with ant colony algorithm.

2. The theory of cloud model

Deyi proposed the theory of cloud model for the first time, which could transform uncertainties among qualitative concepts and their numerical representations [11]. In the cloud model, randomness and fuzziness could be integrated and unified, quality and quantity could be transformed into each other, and membership functions had new interpretations and innovations.

2.1 Digital features

The cloud model mainly used three numerical features (*Ex*, *En*, and *He*) to express the overall qualitative and quantitative conversion characteristics. *Ex* represents the expected value, reflecting the information center value of the qualitative concept, *En* denotes entropy showing the number of elements in the domain space that can be accepted by qualitative concepts, i.e. ambiguity, and *He* is the entropy of *En*, presenting the degree to which each numerical value belongs to a qualitative concept.

2.2 Definition of cloud and normal cloud

Cloud: Assume X to be a regular set defined by numerical values, i.e. $X=\{x\}$, and denoted as the domain space and A to be the qualitative concept corresponding to X. For any x, there exists a relatively stable random number $\mu(x)$ between 0 and 1, representing the certainty degree of x with respect to concept A, i.e. membership degree; here, $\mu(x)$ is called membership function, x is called cloud droplet, and the distribution of cloud droplets in the entire domain space is called the cloud model, which is shortly denoted as cloud [12].

Normal Cloud: If a cloud droplet presents a normal distribution on the domain, i.e. $x \sim N(Ex, En'^2)$, in which En' also follows a normal distribution, i.e.

$En' \sim N(En, He^2)$, and the membership value $\mu(x)$ satisfies Eq. 1, then the cloud model is a normal cloud.

$$\mu(x) = e^{-\frac{(x-Ex)^2}{2En'^2}} \tag{1}$$

2.3 Example of normal cloud

Normal cloud enjoys from excellent mathematical features and is an essential cloud model. This research also relied on normal cloud. Fig. 1 illustrates the normal cloud distribution of 1000 cloud droplets generated for different digital features values. It was seen from the figure that if *En* was large, the coverage area of cloud droplets was also large and when *He* was large, cloud droplets were more scattered.

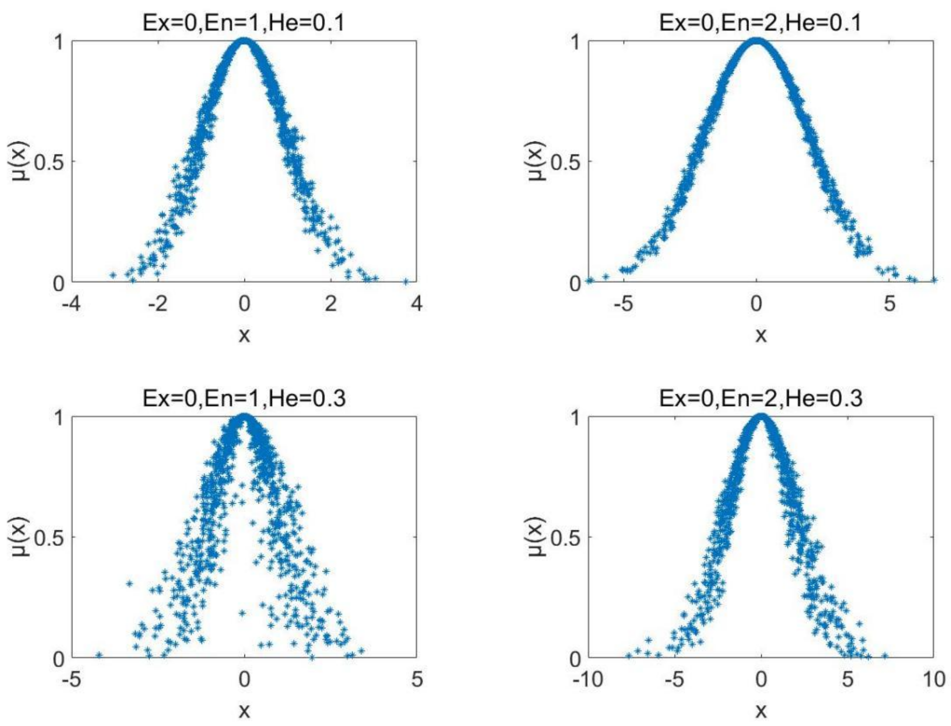


Fig. 1. Normal cloud distribution values of different digital features

3. The proposed algorithm: CMACOA

3.1 Algorithm flowchart

By integrating the ideas of cloud model into ant colony algorithm, this research proposed a new algorithm: CMACOA. The developed algorithm first used ants to establish paths, obtained feasible solutions, and updated local pheromones for each feasible solution path. Then, pheromone distribution status was determined and investigated, and some relevant parameters in the algorithm were adaptively adjusted based on the obtained evaluation calculation results. Finally, membership degree for feasible solutions was calculated, excellent solutions were determined, and global pheromones for each excellent solution path were updated. Fig. 2 illustrates the flowchart of CMACOA.

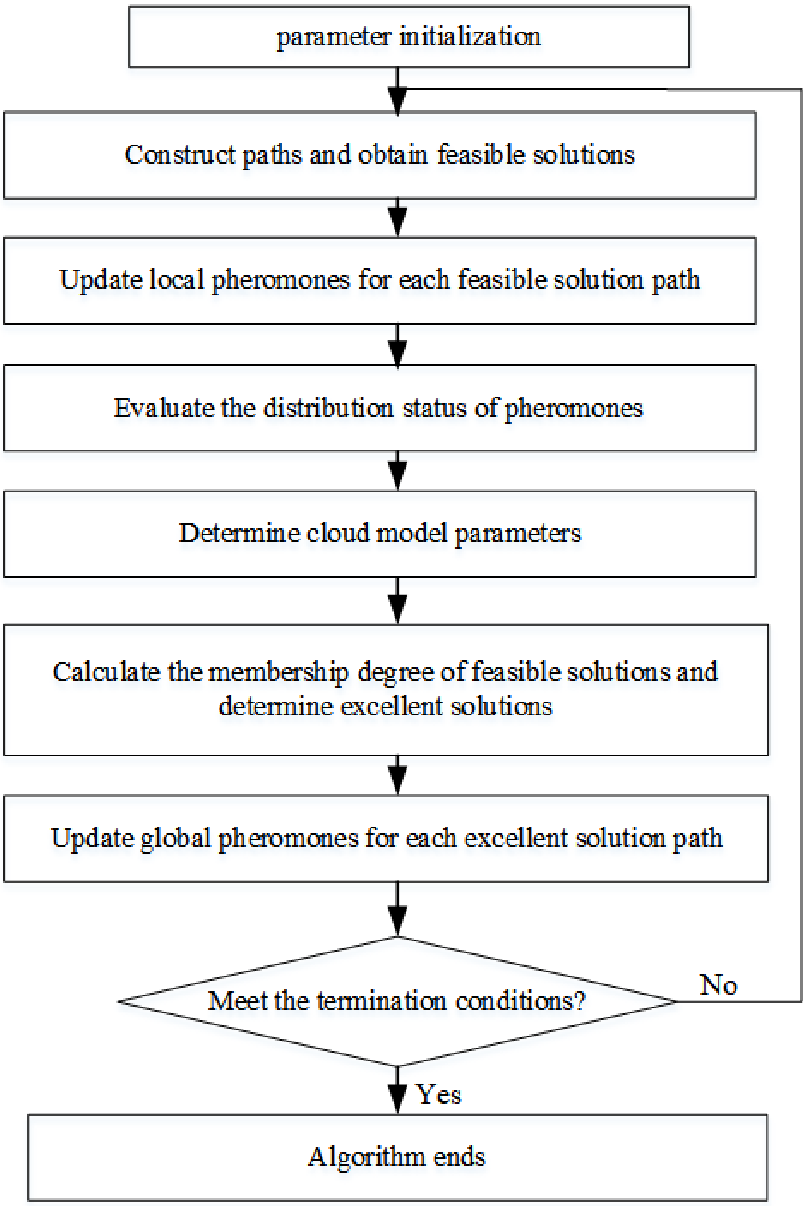


Fig. 2. Flowchart of the developed CMACOA algorithm

Below, we have provided a detailed explanation of the new ideas, specific implementation strategies, and major operations involved in the developed algorithm.

3.2 Path construction method

In routing problems, the process of employing ants to establish paths is actually the process of ants selecting the next node from the current node in the network [13-14]. In this research, ants first calculated transition probability from the current node to other nodes when establishing a path. Eq. 2 was applied to calculate transition probability. Then, the rule of Eq. 3 was used for the specific selection of the next node.

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in allowed_k}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, j\in allowed_k \\ 0, j\notin allowed_k\end{cases} \tag{2}$$

where $allowed_k$ is the set of nodes that ants have not yet passed through, τ_{ij} and η_{ij} are the intensity and visibility of pheromone on the (i, j) line, respectively, and α and β are heuristic factors.

$$j=\begin{cases}\arg\max\limits_{s\in allowed_k}\left\{[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}\right\}, q\le q_0 \\ J, q>q_0\end{cases} \tag{3}$$

where q is a randomly generated number and q_0 is a set parameter. The value of J was derived from Eq. 2. This rule allowed ants to adopt the current possible optimal path with a probability of q_0, while evaluating other paths with a probability of $(1\text{-}q_0)$.

3.3 Parameter setting of cloud model

The parameters that needed to be set in the cloud model mainly included *Ex*, *En* and *He.* Based on the description provided in introduction in Part 2, *Ex* is the global optimal solution found by the algorithm and *En* is derived from the 3-σrule [15]. Assuming $Ex+3En=Len$, where *Len* was the length of the path found using the nearest neighbor method, $En=(Len\text{-}Ex)/3$. Since *He* is the entropy of *En*, the value of *He* is related to *En*. To present the uncertainty of membership degree and adaptively adjust algorithm operation, it also needed to be related to pheromone distribution state. Therefore, we defined *He* as $He=En/St$, where *St* is obtained by evaluating pheromone state, as detailed in the following section.

3.4 Evaluation of pheromone state

Every time the algorithm ran, it needed to explore pheromone state and derive a pheromone state evaluation value. Then, cloud model parameters and membership threshold could be adaptively adjusted based on this value [16]. Hence, pheromone state evaluation was the most important part of constructing an algorithm adaptive mechanism. The pseudocode of the algorithm for the evaluation of pheromone state was described as follows:

For each node *i*,

Firstly, the average pheromone value from node *i* to other nodes was calculated;

Then, the number of edges exceeding the average pheromone value was counted, which was denoted as B_i;

End For

Finally, the average B_i value of all *n* nodes was obtained to derive pheromone state evaluation value *St*, as stated in Eq. 4.

$$St = \sum_{i=1}^{n} B_i / n \tag{4}$$

3.5 Setting of membership threshold

The size of membership threshold determined the number of excellent solutions and regulated the adaptive mechanism of the algorithm. Therefore, the setting of membership threshold was very important. This research adaptively adjusted membership threshold according to pheromone state evaluation results, and threshold setting is expressed in Eq. 5.

$$r = 1 - 1/St \tag{5}$$

3.6 Calculation of membership value

This research used normal cloud to obtain membership values via Eq. 1. Here, *x* was the feasible solution obtained by the algorithm, cloud model parameters were obtained from section 3.3, and a normal distribution *En'* was generated according to corresponding parameters. The number of cloud droplets generated in the algorithm should be 1, so that each feasible solution could obtain a membership value through calculation.

3.7 Determination of excellent solutions

After calculating the membership values of feasible solutions, it was necessary to identify the excellent feasible solutions. Determination method was to check whether the membership value exceeded the set threshold; that is, $\mu(x)>r$. If so, the feasible solution was considered as an excellent solution.

3.8 Updating of pheromone values

The updating of pheromone values in this research included the updating of local and global pheromones.

The updating of local pheromones referred to updating the pheromones on each edge that each ant passed through after constructing a path [17]. The updating approach is stated in Eq. 6.

$$\tau_{ij} = (1-\xi)\tau_{ij} + \xi\tau_0 \tag{6}$$

where ξ satisfies $0<\xi<1$ and τ_0 is initial pheromone.

The updating of global pheromones did not only update one optimal path, but also updated the pheromones of all paths with excellent solutions. The updating method is presented in Eqs. 7 and 8.

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau'_{ij} \tag{7}$$

$$\Delta\tau'_{ij} = \mu / L(T') \tag{8}$$

where $L(T')$ is the path length of the excellent solution T' and $\Delta\tau'_{ij}$ is the value-added of pheromones.

3.9 Algorithm flow

To make the understanding of CMACOA algorithm easier, we provided specific process for its execution.

```
Algorithm 1: CMACOA Algorithm
Parameters were initialized and initial pheromone values τ0 were calculated;
while t < max iterations
    for each ant k
        ant k randomly selects a starting city;
        construct path according to formulas (2) and (3);
        calculate path length L_k;
    end for
     // Local pheromone update
    Update local pheromone according to formula (6);
    // Pheromone state evaluation
    Calculate St according to section 3.4;
    // Determine cloud model parameters and membership threshold
    calculate Ex, En, and He according to section 3.3;
    Obtain the membership threshold r according to formula (5);
```

```
 // Evaluate feasible solutions
for each ant k
        μ_k = MembershipCalculation(L_k, Ex, En, He)
        if μ_k > r then
        mark path k as excellent solution;
        end if
end for
// Global pheromone update for excellent solutions
Update global pheromone according to formula (7) and (8);
t = t + 1;
end while
Return optimal solution and its length.
```

3.10 Adaptive mechanism

The most important part of the adaptive mechanism in CMACOA algorithm was the determination and evaluation of pheromone states. According to the obtained evaluation result for the value of *St*, membership threshold *r* and super entropy *He* were adaptively adjusted [18].

When pheromone distribution was relatively even, the value of *St* was larger, membership threshold *r* was also larger, and super entropy *He* was smaller. This way, membership condition was improved and only a few high-quality feasible solutions were selected as excellent solutions. The algorithm tended to develop the optimal path. When pheromone distribution was relatively concentrated, the values of *St* and membership threshold *r* were decreased and super entropy *He* was increased. Hence, membership degree condition was reduced, membership degree randomness was increased, and more feasible solutions were selected as excellent solutions and the algorithm tended to explore unknown paths.

Fig. 3 illustrates the schematic diagram of adaptive mechanism. When threshold was $r=r_1$, the optimal solution interval was [Ex, X_1] and when threshold was $r=r_2$, the interval of excellent solutions was [Ex, X_2]. With the increase of the threshold *r*, the number of excellent solutions was decreased, which guided the algorithm to develop the optimal path. At the same time, *He* was decreased, the distance between cloud boundaries was reduced, algorithm randomness was weakened, and certainty was enhanced. When the threshold *r* was decreased, however, the number of excellent solutions was increased, which guided the algorithm to explore more paths. At the same time, *He* was increased, the distance between cloud boundaries was increased, algorithm randomness was enhanced, and certainty was weakened.

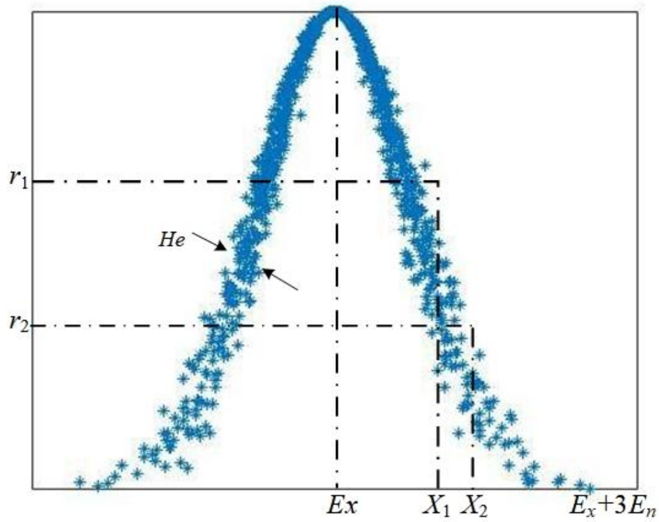


Fig. 3. Schematic diagram of adaptive mechanism

4. Simulation experiments

The routing problem adopted in this research was traveling salesman problem (TSP). We selected three TSP problems (att48, eil51, and ch150) from the TSPLIB dataset for algorithm application and problem solving. We compared CMACOA algorithm with ACS [19] and MMAS [20] algorithms. MATLAB R2021b software was used in the experiments. Table 1 lists the values of the algorithm parameters.

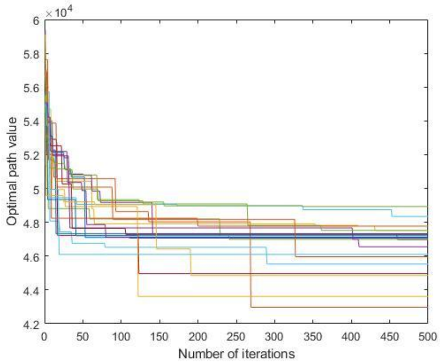
Table 1

Parameters used

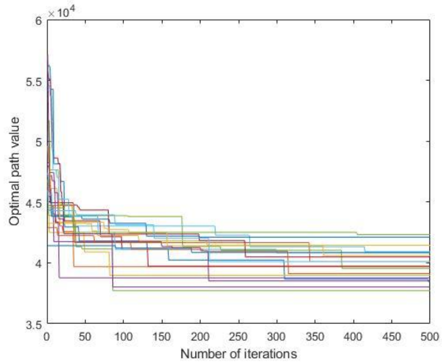
Iteration	Ant No.	α	β	ρ	ξ	q_0
500	100	1	2	0.1	0.1	0.95

Considering potential randomness in the experiment, we performed 20 independent experiments on att48, eil51, and ch150 problems using ACS, MMAS, and CMACOA algorithms. Figs. 4 to 6 present the experimental results obtained, from which it was seen that the optimal values evolved during each iteration search. At the same time, statistical analysis was performed on the results obtained from the three algorithms after 20 independent runs. Tables 2 to 4 summarize statistical results. The best, worst, and average values obtained from each algorithm after 20 independent runs were recorded. To demonstrate the stability of each algorithm, the standard deviation of the best results was calculated after 20 independent runs. Meanwhile, the earliest iteration to find the best value was recorded.

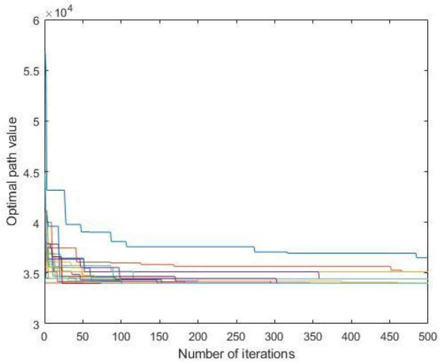
To prove the superiority of CMACOA algorithm to ACS and MMAS algorithms and to demonstrate that its performance advantage was not due to random factors, we adopted Wilcoxon rank-sum test—a non-parametric statistical method—to perform inferential statistical testing on the results obtained from 20 independent runs, thereby performing a statistical significance analysis of the performance of the algorithm. The experimental results obtained are presented in Table 5.



(a) ACS

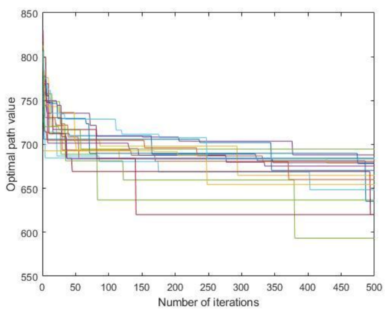


(b) MMAS

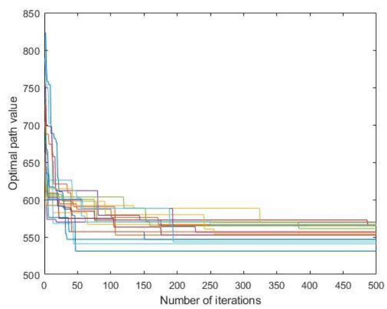


(c) CMACOA

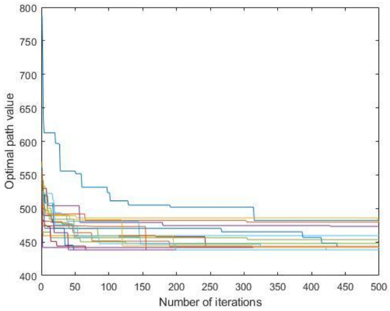
Fig. 4. Solution situation of att48 problem



(a) ACS

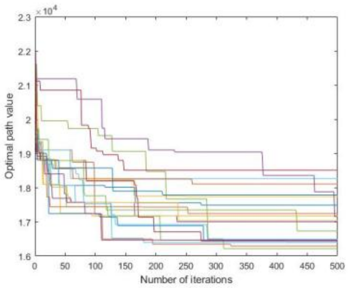


(b) MMAS

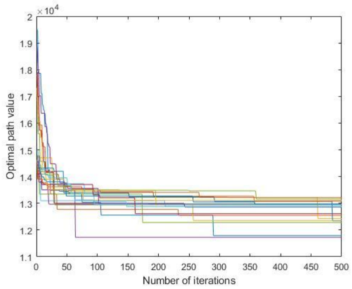


(c) CMACOA

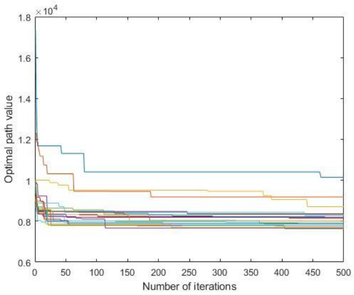
Fig. 5. Solution situation of eil51 problem



(a) ACS



(b) MMAS



(c) CMACOA

Fig. 6. Solution situation of ch150 problem

Table 2

Statistical results of att48 problem

Algorithm	Best Value	Worst Value	Average Value	Standard Deviation	The earliest iteration to find the best value
ACS	42967.63	48935.53	46529.62	1508.5963	269
MMAS	37716.04	42326.9	39990.13	1300.9365	86
CMACOA	33981.71	36506.8	34295.18	631.4005	42

Table 3

Statistical results of eil51 problem

Algorithm	Best Value	Worst Value	Average Value	Standard Deviation	The earliest iteration to find the best value
ACS	593.2597	694.8794	661.5905	27.6319	380
MMAS	531.1524	570.0863	556.3387	21.2910	47
CMACOA	438.3607	482.0383	451.058	15.7271	272

Table4

Statistical results of ch150 problem

Algorithm	Best Value	Worst Value	Average Value	Standard Deviation	The earliest iteration to find the best value
ACS	16211.56	18509.9	17100.52	693.3650	312
MMAS	11727.34	13214.34	12704.29	429.9902	64
CMACOA	7621.73	10143.34	8170.46	602.8339	404

Table5

Results of statistical significance analysis

Test Instance	Compariso n Group	Sample Size (A/B)	U Statistic	Z Statistic	One-Tailed p-Value	Significance(α=0.05)
att48	CMACOA vs ACS	20/20	0	-5.238	9.47×10^{-8}	Extremely Significant
att48	CMACOA vs MMAS	20/20	0	-5.547	1.46×10^{-8}	Extremely Significant
eil51	CMACOA vs ACS	20/20	0	-5.238	9.52×10^{-8}	Extremely Significant
eil51	CMACOA vs MMAS	20/20	78	-3.589	1.64×10^{-4}	Extremely Significant
ch150	CMACOA vs ACS	20/20	0	-5.238	9.47×10^{-8}	Extremely Significant
ch150	CMACOA vs MMAS	20/20	0	-5.238	9.47×10^{-8}	Extremely Significant

Figs. 7 to 9 compares the optimization performance of various algorithms in a certain experiment, from which we could make clear judgments of the optimization quality and convergence performance of each algorithm.

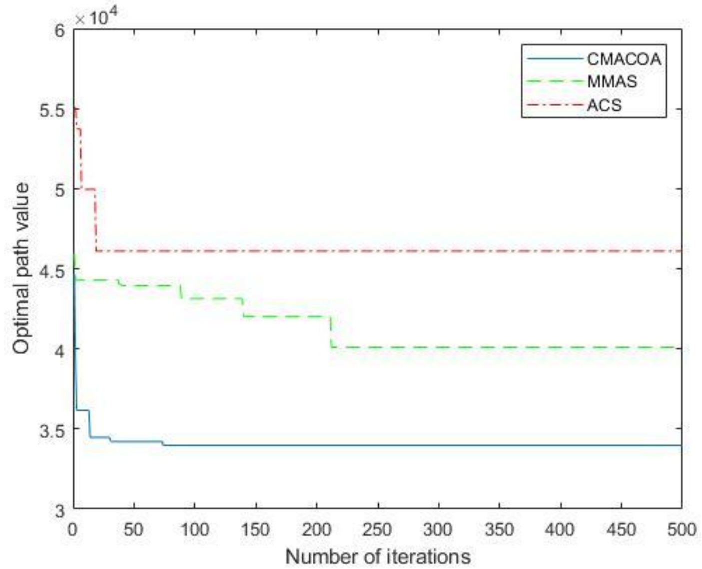


Fig. 7. Comparison diagram of various algorithms for att48 problem

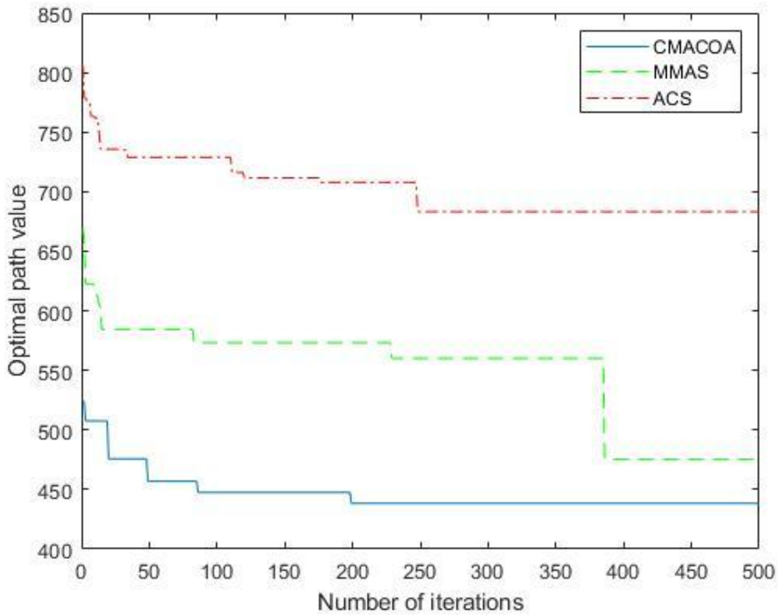


Fig. 8. Comparison diagram of various algorithms for eil51 problem

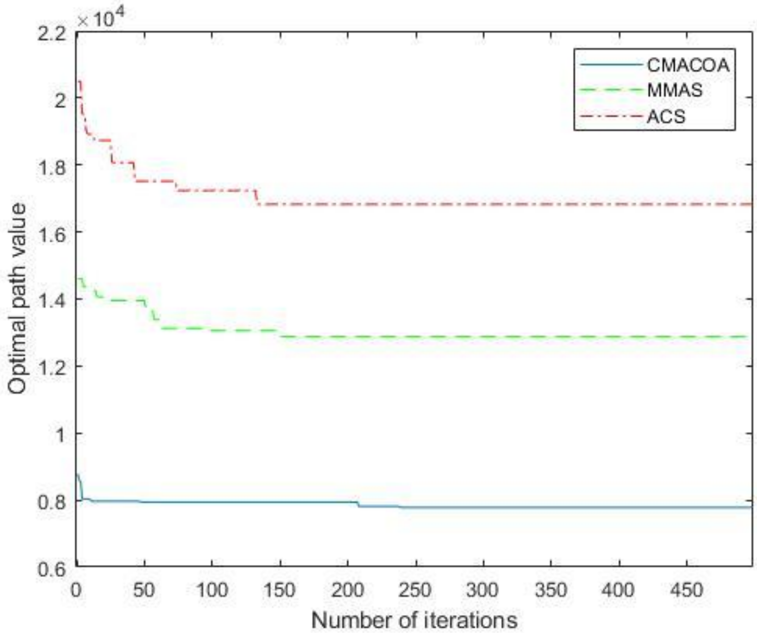


Fig. 9. Comparison diagram of various algorithms for ch150 problem

It was easily seen from Tables 2 to 4 that regardless of whether the algorithm found the best, worst, or average value, CMACOA algorithm solution quality was very high. It was clear that ACS and MMAS algorithms were much inferior. The worst values found by CMACOA algorithm after 20 independent runs were better than the best values obtained by ACS and MMAS algorithms. It was also clearly seen from the Wilcoxon rank-sum test results in Table 5 that the one-tailed p-value was much lower than 0.001, indicating that the performance of CMACOA algorithm was much higher than those of ACS and MMAS algorithms. From Figs. 4 to 6, it was be observed that CMACOA exhibited relatively stable and significantly superior characteristics in solving the three TSP problems.

For the convenience of observation and comparison, the variations of the optimal path values obtained by the three algorithms with the number of iterations were plotted in a graph. Therefore, Figs. 7 to 9 intuitively present the superior performance of CMACOA algorithm. CMACOA algorithm could not only search for shorter paths and find better solutions, but also more rapidly converged to the best solution. This once again demonstrated that the new algorithm introduced in this research was efficient.

All above descriptions indicated that the adaptive mechanism based on cloud model and pheromone update mechanism for excellent solutions had good characteristics. These strategies could not only make full use of excellent solutions to search for the optimal solution but also broaden search area and enhance the optimization quality of the developed algorithm. Furthermore, by solving mediumscale TSP instances, the stability, effectiveness, and good adaptability of the algorithm proposed in this research was also verified.

5. Conclusions

This research used the idea of cloud model to enhance ant colony algorithm and applied it to solve routing problems. Firstly, cloud model principle was proposed and the characteristics of normal cloud were evaluated. Then, the mechanism of combining cloud model characteristics with ant colony algorithm was investigated. An improved algorithm was proposed to achieve a dynamic balance between searching for unknown solutions and using known solutions. Firstly, traveling salesman problem was adopted for the application and solution of the developed algorithm. In the future, we will continue to explore the cloud model, further investigating the parameter settings and adaptive mechanisms of the algorithm. The integration of cloud model with other intelligent optimization algorithms will also be actively explored. Furthermore, we will continue to modify the algorithm, expand its scope of application, and investigate other routing problems such as network routing and vehicle routing.

Acknowledgment

This research work has received support from the key projects of Anhui Provincial Department of Education (Fund No. 2024AH051466 and No. 2022AH051325), for which we express our gratitude.

REFERENCES

[1]. *Oliveira F.D., Martins S.M. and Sousa M.J.*, “Divide-and-conquer initialization and mutation operators for the large-scale mixed Capacitated Arc Routing Problem”, in European Journal of Operational Research, **vol. 321**, no. 2, Nov. 2025, pp. 383-396

[2]. *Santana E.A., Onieva L. and Muñoz C.C.J.*, “A cuckoo search algorithm to improve a routing problem adapted to last mile e-commerce logistics”, in Central European Journal of Operations Research, Nov. 2024, (prepublish): pp.1-15

[3]. *Aksit B., Gurcan O.G. and Saricicek I.*, “Flying Sidekick Traveling Salesman Problem in Truck–Drone Team Logistics with Energy Issues”, in Transportation Research Record, **vol. 2678**, no. 11, Nov. 2024, pp. 1235-1249

[4]. *Liu J., Tong L. and Xia X.*, “A genetic algorithm for vehicle routing problems with time windows based on cluster of geographic positions and time windows”, in Applied Soft Computing, **vol.169**, 2025, pp. 112593-112593

[5]. *Li K.J., Li Q.J. and Xu Y.*, “An improved multiobjective evolutionary algorithm for time-dependent vehicle routing problem with time windows”, in Egyptian Informatics Journal, **vol. 28**, Nov.2024, pp. 100574-100574

[6]. *Chen Y., Chen M., Yu F.. and Lin H.*, “An Improved Ant Colony Algorithm with Deep Reinforcement Learning for the Robust Multiobjective AGV Routing Problem in Assembly Workshops”, in Applied Sciences, **vol. 14**, no. 16, Aug. 2024, pp. 7135-7135

[7]. *Tong M., Peng Z. and Wang Q.*, “A hybrid artificial bee colony algorithm with high robustness for the multiple traveling salesman problem with multiple depots”, in Expert Systems With Applications, **vol. 260**, 2025, pp. 125446-125446
[8]. *Lisheng W., Xiaoming Y. and Sheng L.*, “Heterogeneous ant colony algorithm based on selective evolution mechanism and game strategy”, in The Journal of Supercomputing, **vol. 80**, no. 6, Nov. 2023, pp. 7171-7206
[9]. *Elhassania M.*, “A hybrid ant colony optimization algorithm to solve a transport problem with stochastic customer demands and electric vehicles of limited load capacity”, in Evolutionary Intelligence, **vol. 17**, no. 4, Jan. 2024, pp. 2537-2553
[10]. *Marco S.*, “The AddACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems”, in Mathematics and Computers in Simulation,**vol. 218**, Jan. 2024, pp. 357-382
[11]. *Li D.Y.*, “Uncertainty in knowledge representation”, in Chinese Engineering Science, **vol. 2**, no. 10, Feb. 2000, pp. 73-79
[12]. *Anjali and Gupta A.*, “A novel Shannon entropy-based backward cloud model and cloud K-means clustering”, in The Journal of Supercomputing, **vol. 81**, no. 1, Oct. 2024, pp. 65-65
[13]. *Xu Lixia, Zhong Jilong, Wu Shaoshi, et al*. “Indicator Transfer Learning Based on Cloud Model and Maximum Mean Discrepancy”, in Journal of System Simulation, **vol. 36**, no. 9, Oct. 2024, pp. 2004-2015
[14]. *Fang W., Liao Z. and Bai Y.*, “Improved ACO algorithm fused with improved Q-Learning algorithm for Bessel curve global path planning of search and rescue robots[J]. Robotics and Autonomous Systems, **vol. 182**, Oct. 2024, pp.104822-104822
[15]. *Ma X. and Liu C.*, “Improved Ant Colony Algorithm for the Split Delivery Vehicle Routing Problem”, in Applied Sciences, **vol. 14**, no.12, Jun. 2024, pp. 5090-5090
[16]. *Meng J W, You X M and Liu S*, “Heterogeneous ant colony optimization based on adaptive interactive learning and non-zero-sum game”, in Soft Computing, **vol. 26**, no. 8, Feb. 2022, pp. 1-18
[17]. *Lisheng W., Xiaoming Y. and Sheng L.*, “Multi-ant colony optimization based on bidirectional induction mechanism and cooperative game”, in Soft Computing, **vol. 27**, no. 20, Jun. 2023, pp. 15075-15093.
[18]. *Danesh M. and Danesh S.*, “Optimal design of adaptive neuro-fuzzy inference system using PSO and ant colony optimization for estimation of uncertain observed values”, in Soft Computing, **vol. 28**, no. 1, Jan. 2024, pp. 135-152
[19]. *Dorigo M. and Gambardella L.M.*, “Ant colony system: a cooperative learning approach to the traveling salesman problem”, in IEEE Transactions on Evolutionary Computation, **vol. 1**, no. 1, Oct. 1997, pp.53-66
[20]. *Stützle T. and Hoos H.H.*, “MAX-MIN ant system” in Future Generation Computer Systems, **vol. 16**, no. 8, May. 2000, pp. 889-914