Routing Design and Optimization Based on the Improved Ant Colony Algorithm

Jingxuan Yang

Data Science, Nottingham University, NG7 2QL, UK
psxjy9@nottingham.ac.uk

Abstract. This study aims to address the practical problem of providing an optimal and cost-effective short-distance route for tourists by employing an enhanced version of the Ant Colony Optimization algorithm (ACO). The research begins by presenting a comprehensive overview of the evolution of ACO algorithms. Subsequently, the focus shifts towards the application of Ant System (AS) and Max-Min Ant System (MMAS) to solve the Traveling Salesman Problem (TSP) and ACO to address the Quadratic Assignment Problem (QAP). Through a comparative analysis conducted on multiple datasets, it is determined that MMAS outperforms AS in terms of convergence speed, stability, and the attainment of optimal solutions. The algorithm is validated using publicly available TSP and QAP datasets, thereby confirming its feasibility. Subsequently, the TSP and QAP problems are integrated, and the improved algorithm is applied to the merged problem. Through the practical scenario of visiting the scenic spots in Zhuhai City in China, the effectiveness of the enhanced algorithm in solving the merged TSP and QAP problem is demonstrated. Experimental results demonstrate the capability of the improved algorithm to effectively solve the merged TSP and QAP problem. Furthermore, when applied to the practical problem of visiting Zhuhai's scenic spots, the improved algorithm proposes a rational route that fulfills the objective.

Keywords: Route design, colony algorithm, optimization

1 Introduction

The surge in travel demand has given rise to various challenges related to traveling and the play of visitors. For example, the traffic jam and the attraction have many visitors, especially on holidays. According to statistics, the number of people traveling during the May Day holiday in 2023 was as high as 374 million in China. The National Bureau statistics estimate that tourists will reach 4.55 billion in 2023. In the context of tourism, the presence of time constraints often hinders tourists from fully experiencing all desired attractions within a limited duration. In this case, it is essential to have a reasonable travel route, which can be considered as a path planning issue that is an optimization task. The optimal path planning is to find the most suitable route from the start point to the endpoint under the criteria of time, distance, and work costs.

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_27

In the last decades, numerous algorithms have been devised for path planning, including genetic algorithms, simulated annealing, etc. Genetic algorithms have strong global searchability but slow convergence and low efficiency. The simulated annealing algorithm has a solid local search ability and a short running time. However, the global search ability is poor, so it cannot guarantee to find the global optimum. They are greatly affected by parameters. Consequently, a more suitable algorithm, characterized by superior convergence and pronounced global searchability, is sought. The Ant Colony Optimization algorithm (ACO) can yield rapid convergence compared with the algorithms mentioned above, which is also considered in this study.

The advancement of simulation technology has facilitated significant contributions to any colony optimization algorithms. For instance, Marco Dorigo et al. first proposed an algorithm called Ant System (AS) by simulating ant colony foraging behavior in 1991 [1]. They proposed Ant Colony System Algorithm (ACS) in 1996 and Ant Colony Optimization Algorithm (ACO) in 1999. ACO is a heuristic search algorithm based on population optimization, and it can solve complex combinatorial optimization problems such as the Traveling Salesman Problem (TSP) and Quadratic Assignment Problem (QAP). By setting the range of pheromone values to improve the speed and avoid falling into sub-optimal too early, German scholar Thomas Stutzle proposed Max-Min Ant System (MMAS) in 1996 [1].

The application of ACO is widely used, except for routing problems and assignment problems. ACO is also used in scheduling problems. In general, scheduling problems are the behavior of allocating limited resources to a task center with time passing by. Single Machine Weighted Tardiness Scheduling (SMTWTS) is a kind of single scheduling problem due to the task finish time exceeding the delivery time, so the aim is to minimize the weighted delay cost [2]. When the instance of SMTWTS has more than 50 processes, it cannot be solved by the classical branch and bound [3]. So, Besten, Merkle, and Middendorf improved ACO to make it can be used in SMTWTS. The feasibility of ACO has been verified, so it will be applied to the routing problem and assignment problem in this research.

This research endeavors to devise a short-distance and low-cost route for travelers to help them plan a suitable route. First, some improvements are carried out to the defects of the existing algorithm. After that, the public data is employed to test this algorithm. The results demonstrated that the improved algorithm performs well in convergence speed and accuracy. Finally, the improved algorithm is applied to the real situation.

2 Method

2.1 Data Collection and Preprocessing

This research employs the Traveling Salesman Problem (TSP) and Quadratic Assignment Problem (QAP) public datasets to evaluate the effectiveness of the algorithm. This dataset is specifically designed for TSP and QAP testing, respectively, and they contain the optimal solutions. Therefore, it facilitates comparison with the results of the improved algorithms. The algorithm is applied to the real city path data in the final practical application.

2.2 Ant Colony Algorithm on Traveling Salesman Problem

Introduction of Ant Colony Algorithm. Ant Colony Algorithm shown in Fig. 1 is a bionic optimization algorithm. The algorithm simulates the foraging behavior of ants. When ants forage, they will release the pheromone on the path. Research shows that ants deliver information using pheromones. When an ant encounters the pheromone left by the ant before, it is more likely to decide to follow compared with the trace no ants passed. And then, the following ants use their pheromone to enhance the pheromone concentration on the trace. The pheromone will disappear over time, so if no other ants follow and put the pheromone, this road will lose its appeal to ants. The thicker the path pheromone concentration is, the more attractive it is for the ants to follow this path. Through this positive feedback, an optimal path will eventually be found.

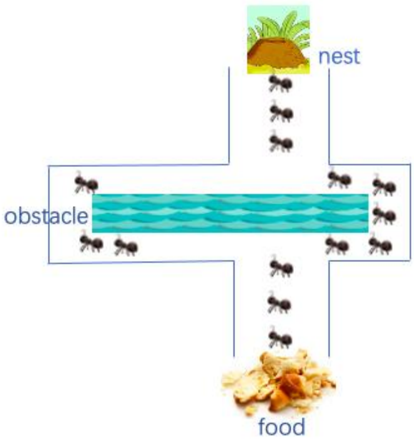

Fig. 1. A simple example of the Ant Colony Algorithm (Photo/Picture credit: Original).

It is evident that the duration of time taken by ants is inversely related to the length of the route traversed. Because of the shortest distance of the optimal route at this time, more ants pass through the optimal route in unit time. Therefore, if the residual pheromone on this path is higher than on other paths, the subsequent ants tend to follow this path. After repeated iterations, the pheromone concentration on the optimal route is the highest. As a consequence, the optimal route becomes prominently distinguished from alternative paths within the system.

Ant Algorithm. The TSP is a typical problem in combinatorial optimization. The TSP aims to find the shortest route, which is to go through several cities from one city and return to the original city, and each city is traveled once [4]. To prevent ants from repeating the same path, set up a tabu for each ant to record the paths it has taken. The following Table 1 is a specific definition and detailed description of the algorithm model for TSP.

Table 1. Variable declaration

Variable Name	Variable Explanation
m	number of ants
n	number of nodes(cities)
α	the importance of pheromone
β	the importance of distance
ρ	the evaporation rate of pheromone
t	time
τ	the pheromone concentration ($\Delta\tau$: pheromone increment)
p_{ij}	the possibility of route selecting from i to j
η	the reciprocal of the distance matrix

Based on the performance of the pheromone, the Ant Colony system shown in Fig. 2 can be formed. The process of ACO to solve TSP can be divided into 6 steps: 1) Initializing the number of ants at different nodes and the pheromone with different random values. 2) Considering the distance and pheromone concentration, calculate the probabilities from the current node to node other nodes shown in equation (1). 3) Making the roulette selection based on the calculation results. 4) Updating the pheromone concentration as shown in equation (2). 5) Looping this process until all nodes have been visited. 6) Iterating the whole process until it reaches the maximum iteration or falls into convergence. Finally, the shortest route is the optimal solution at this time.

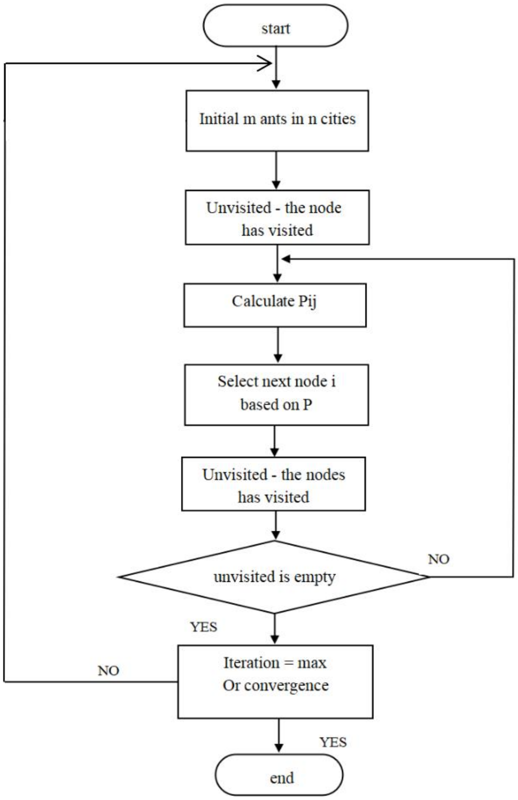

Fig. 2. Flow chart on TSP use AS (Photo/Picture credit: Original).

$$p_{ij}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}}{\sum_{j=1}^{n} \left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}} \tag{1}$$

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \tag{2}$$

AS algorithm has strong robustness in solving performance and the ability to search for better solutions, which means it can be applied to other problems with a little modification. It is easy to combine a variety of heuristic algorithms to improve the performance of the algorithm. However, AS algorithm still has many shortcomings, such as slow convergence, and the final result cannot be guaranteed to be the global optimal solution. Based on this, scholars have developed the elite strategy [5]. The elite strategy only adds the pheromone to the shortest route (the shortest one in the path selection or the whole iteration). Then the MMAS algorithm was delivered to improve the AS algorithm [6].

Max-Min Ant System. In order to increase iterative convergence speed, numerous studies used the elite strategy. Then the pheromone concentration gap between the optimal and other paths increases, so it will converge faster. Nevertheless, in this case, fewer different routes may appear in the algorithm iteration. In order to mitigate this issue and ensure convergence towards the optimal solution, researchers have introduced a restriction on pheromone concentration. Then all paths might be selected even with negligible pheromone concentration. There are three evident differences between MMAS and AS [6].

In terms of the first difference, only one ant (Elite ant) is allowed to update the trajectory in each iteration. One path is updated pheromone increment in each iteration in the MMAS algorithm, that is, the shortest path. Use $\frac{1}{L_{best}}$ as the pheromone increment ($\Delta\tau$), which has three definitions: iteration optimal, solution optimal, and mixed-use of the two.

In terms of the second difference, to limit the search stagnation, the allowable range of possible tracking strength is limited to some maximum and minimum values in equation (3) and (4). The equation (2) is still used to calculate the updating value of pheromones. However, except for the optimal path, the other paths only evaporate and do not increase. However, because pheromone concentration is limited to the upper and lower limits, the pheromone concentration will not exceed this range. When it is lower than the lowest pheromone concentration, it will be redefined as the lowest pheromone. When it is higher than the highest pheromone concentration, it will be redefined as the highest pheromone. Moreover, the definition of maximum and minimum was different. As long as the value is appropriate, it can be used in the algorithm.

$$\tau^{max} = \frac{1}{\rho L_{best}} \tag{3}$$

$$\tau^{min} = \frac{\tau^{max}}{2n} \tag{4}$$

In terms of the third difference, the trajectory is initialized in a specific way. MMAS usually initializes pheromone concentration to be much larger than the maximum pheromone value. This can increase the exploratory ability of the algorithm in order to avoid the condition appearing that a super individual appears in the ant colony in the early stage of the algorithm. The super individual has a pheromone concentration far more than the average pheromone concentration of the current ant colony. This measure makes the algorithm avoid converging to the local optimal solution earlier.

2.3 Ant Colony Optimization on Quadratic Assignment Problem

The quadratic assignment problem is that there are a number of production tasks and a number of production units in the production process. QAP aims to minimize the production cost when allocating each task to the appropriate production unit.

When realizing this problem, the core is which point ants distribute the goods to minimize the total cost. There are some variables and formulas to help understand QAP.

$$\begin{aligned} &\min \left({}_{i,j=1}^{\;n} d_{ih} f_{jk} x_{ij} x_{hk} \right) \\ &\text{s.t.} \quad {}_{i-1}^{\;n} x_{ij} = 1, \quad j = 1, 2, \ldots, n \\ &{}_{j-1}^{\;n} x_{ij} = 1, \quad i = 1, 2, \ldots, n \\ &x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \ldots, n \end{aligned} \tag{5}$$

Where $D = [d_{ih}]_{n*n}$ represents the distance between position i and h, $F = [f_{jk}]_{n*n}$ represents the traffic between task j and k

It can be defined that when task j is assigned to position i, x_{ij}=1, otherwise x_{ij}=0. The ants assign high-traffic tasks to a position close to the center. The cost and the pheromone are placed on nodes, and each node is assigned a two-dimensional coordinate, representing the task i is completed by unit j [7].

The process of ACO to solve QAP can be divided into 5 steps: 1) I Initialize iteration and $\tau(i, j)$, make $\tau_0 = 1$. 2) Let ant_k's current point be i, and ant_k will assign task j to position i, selecting the next point by probability shown in equation (5). 3) The kth ant uses a local pheromone to update each arc when it passed shown in equation (6) and (7). 4) In order to prevent premature convergence, this study optimize the algorithm locally, searching the solution of the neighbor to find whether there is a better solution [8]. The solution S(k) is according to L_k increment order. 5) It will be terminated if the algorithm reaches the set number of iterations. The flow of local optimization is shown below Fig. 3.

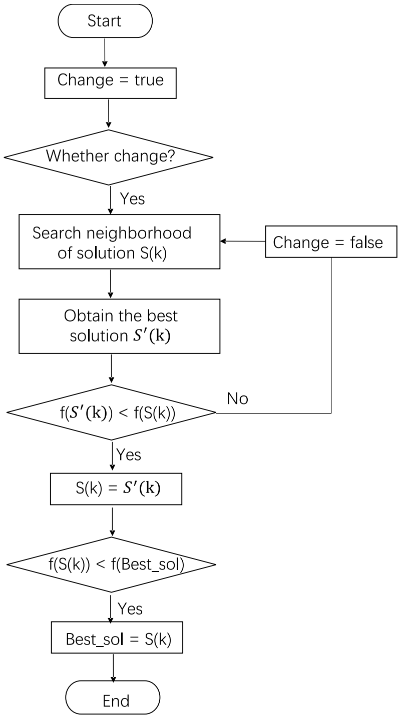

Fig. 3. The flow of local optimization (Photo/Picture credit: Original).

2.4 Improved the Ant Colony Optimization Algorithm

Realize Record. Random numbers implement the initialization of ants and pheromones. Furthermore, using a list to record the nodes that have not been visited. As soon as initialization, move the node they are currently at from the list. Create a matrix to record ants and nodes, and then use the index of each number in the matrix to represent one ant at one node.

Realize the Second Stop Condition. Although there are two conditions to stop iteration in theory: one is to reach the maximum number of iterations, and the other is algorithm convergence, almost no code implements the second stop condition. Because the concept of convergence is abstract, this study embodied it as the optimal solution that does not change in multiple iterations shown in Fig. 4.

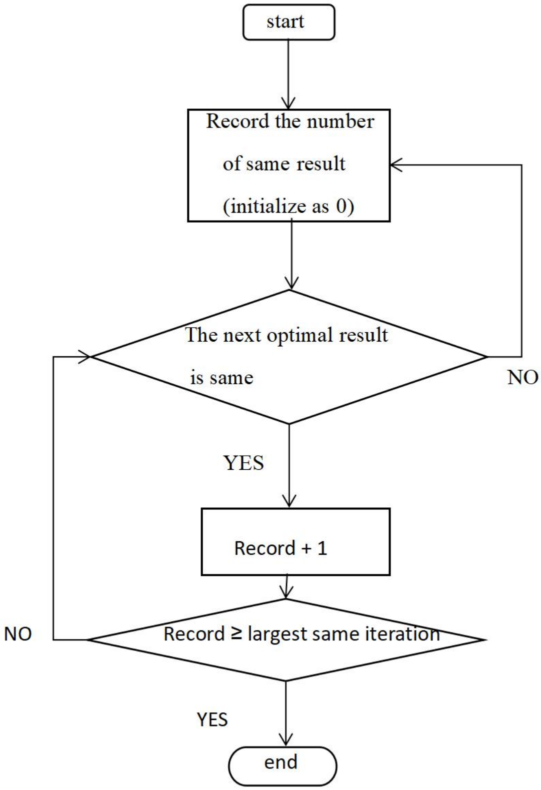

Fig. 4. The second stop condition flow (Photo/Picture credit: Original).

Roulette Algorithm. The roulette algorithm is often realized through those probabilities calculated, subtracting a random number, and then selecting the path corresponding to the minimum calculation result in greater than 0 in previous studies.

However, this method deviates from the principle of roulette. Moreover, because of randomness, errors may occur. The roulette algorithm was realized based on the theory shown in Fig. 5.

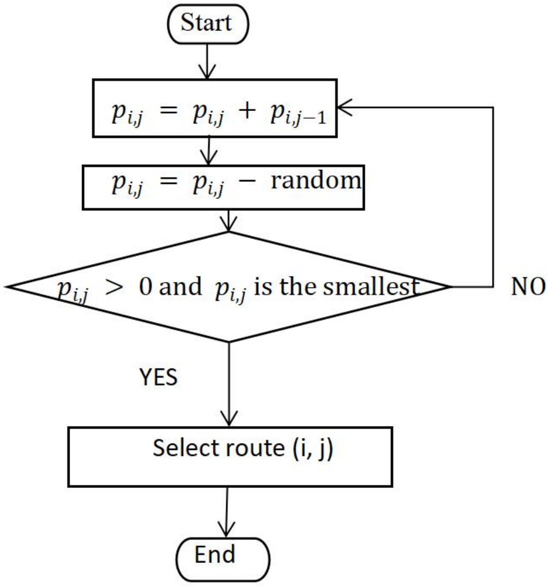

Fig. 5. Independent roulette algorithm (Photo/Picture credit: Original).

3 Results and Discussion

3.1 Parameter Adjustment

Model parameters are configuration variables within the model, and the values of model parameters need to be estimated with data; The model hyper-parameter is an external configuration of the model, and the hyper-parameter value needs to be set manually. Adjust hyper-parameters through different standards. The measurement criteria in TSP are time and the path length; The measurement criteria in QAP are time and cost. In Dorigo and this experiment, by adjusting the parameters α, β and ρ, the results are shown below Table 2.

(1). α : A small value of α causes slow convergence. When the value of α is lower than or higher than the critical value, the optimal solution can not be found most of the time.

(2). β : When β is higher and higher will lead to faster convergence. However, when β is larger than a certain critical value, it may fall into the sub-optimal solution soon.

(3). ρ: When ρ is lower than a certain value, the convergence speed can be slowed down.

These critical values are tested, and different values are used in different data, but each parameter has a roughly appropriate range. Therefore, when using the algorithm, adjust the parameters to achieve the best state of the algorithm.

Table 2. Experiment results by M. Dorigo and this study's data.

	Best α	Best β	Best ρ
Ant-cycle (by M.Dorigo)	1	5	0.5
Ant-cycle (by ourselves)	1.1	3	0.6

The parameters, ants' number, and maximum iteration number of Ant Colony Optimization in the Travel Salesman Problem and Quadratic Assignment Problem can also be adjusted by comparing the result. Use the same dataset to test many times. In TSP, compare which has a shorter distance of route solution and time used, while in Quadratic Assignment Problem, compare the cost (Fig. 6) and time used (Fig. 7).

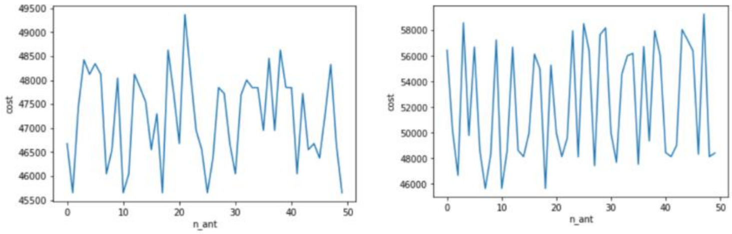

Fig. 6. Repeatedly test using cost (Photo/Picture credit: Original).

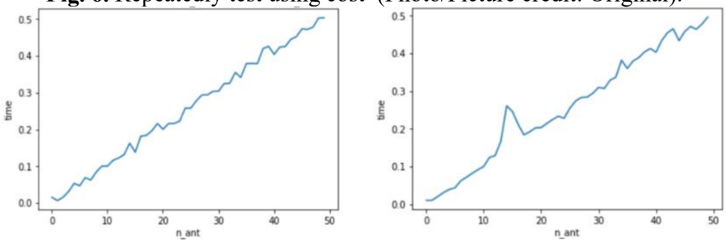

Fig. 7. Repeatedly test using time (Photo/Picture credit: Original).

Under one example, do the test. From the numerical results and figure, when the number of ants is 10, 18, or 30, the costs are relatively small (Fig. 6). At the same time, the time increases as the number of ants increases. Therefore, the best choice of the number of ants is 10 in this example.

3.2 The Results of Traveling Salesman Problem

After introducing AS and MMAS, this study will compare the two algorithms in TSP with a different dataset. In this Table 3, "30 city coordinates" is a dataset that is created to build and tune the algorithm initially. The other dataset is the public dataset, which has the best distance values.

Table 3. Two algorithms in TSP with different dataset

Dataset		AS		MMAS	
TSP instance	Best distance	Common results	Iterations	Common results	Iterations
30 city coordinates	423-430	428	90	423	40
Eil51	426	448	290	443	140
Berlin52	7542	7681	200	7628	160
St70	675	703	540	696	200
Pr107	44303	46063	300	46011	100

Next, the path map (Fig.8) and iteration graph (Fig. 9) of the "eil51" dataset will be used as an example for analysis. After comparing, the iteration of MMAS is less than AS, which is only 140. After iterating to 140, MMAS has become stable, and AS has always floated. For AS, the iteration is 290, it is much larger than MMAS. The MMAS (448) shows a better optimal distance than AS (443).

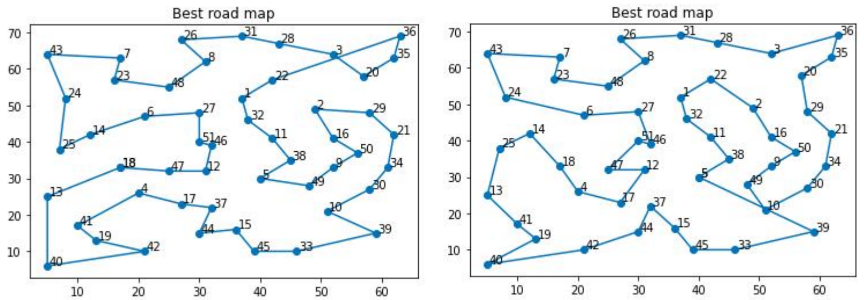

Fig. 8. The best road map in AS (left) and MMAS (right) (Photo/Picture credit: Original).

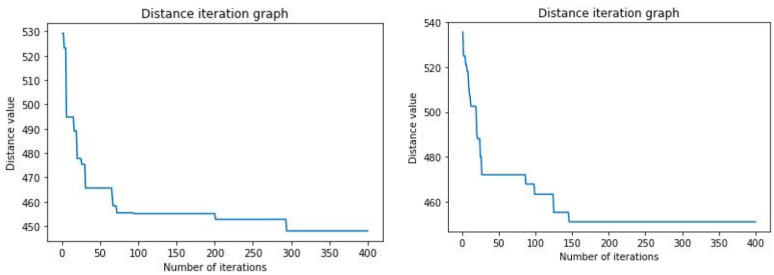

Fig. 9. The best road map in AS (left) and MMAS (right) (Photo/Picture credit: Original).

From the result of Table 2, make a basic conclusion. Compared with AS, MMAS has the following advantages. The MMAS is faster convergence, more relatively stable, and more likely to be optimal. In general, MMAS is the more appropriate algorithm for TSP because it has relatively good results and faster convergence.

3.3 The Results of the Quadratic Assignment Problem

The QAPLIB will be used to test the performance of ACO on QAP shown in Table 4, and it is a library of secondary allocation problems with the optimal solution. Compared with the result of ACO and the optimal solution, ACO performs relatively well. Thus, ACO can deal with QAP effectively.

Table 4. ACO in QAP with different dataset.

QAPLIB			ACO
Name	size	Optimal solution	Common results
Chr12b	12	9742	9746
Had16	16	3720	3722
Nug20	20	2570	2587
Chr22a	22	6156	6169
Kra30b	30	91420	91532

3.4 The Results of Practical Application

After comprehensive and operating ACO on TSP and QAP, the two problems can be combined. They can be considered an assignment problem if making distance is one of the parameters when calculating the cost in QAP. Some changes will be made to the original QAP algorithm to realize this thinking.

Firstly, fixing the beginning and ending points. Because in practical path planning, the starting and ending points are often clearly defined. In QAP, the formula of cost is flow times distance. Noticeably, to consider the actual situation, use Amap to measure the actual distance instead of the straight-line distance between two points. The flow matrix is the flow of people between two points. However, considering when it is actually implemented, the flow is difficult to define. So, changing the flow to 1 divided by satisfy, the satisfied range is from 0 to 100. If a point has a higher satisfaction value, it represents people having a greater desire to go there.

In addition, add a new element, "time," in the cost calculation. Because cost is affected by many factors, one of the important aspects is time. For example, if a person visit attractions during commuting hours or holidays, the spending time may differ. Moreover, the time is the actual spending time to take the feasible transportation recommended by Amap. For example, if this person wants to visit an island, he or she may need to take a car and then go there by ship, so the total time is car spending time plus ship spending time. Finally, the formula becomes cost = $\frac{1}{\text{satisfaction}}$ * distance * time. Also, setting the weight of three important factors, if the user thinks which aspect is more important, the weight of it can be improved.

Moreover, normalize the input matrices to eliminate the dimensional influence between different indicators. Distance, satisfaction, and time metrics have different ranges, so they need to be normalized. To avoid the denominator being zero during the operation between matrices, change the range of processed data between 1 and 2.

After introducing changes to the original formula, a practical example will be given to show how to realize it. Now, imagine a person is a tourist visiting Zhuhai, which is a city in China. Furthermore, this person has decided on some places he or she wants to visit. Because this person is unfamiliar with Zhuhai, he or she do not know the access order of these points. At this time, the model may help visitors to solve this problem. If there are 10 points this person wants to visit are shown below (Table 5).

Table 5. 10 visiting points in Zhuhai

Number	Point name
1	JinWan airport
2	Zhuhai Grand Theatre
3	Chimelong Ocean Kingdom
4	Yuan Ming New Garden
5	Yang Ming Square
6	Dong AO Island
7	Hui Tong Village
8	Feng Huang Mountain
9	Gong Bei
10	Ruins of St.Paul

Input some arguments, like the distance, the satisfaction, and access time between two points. As mentioned before, the spending time may be affected by the selected visiting time, so the results shown afterward are only at one point in time. Lastly, defining the first and last points are the airport and the weight of the three important factors are equal. Use the latitude and longitude of each point on the map as their coordinates. The visiting points distribution map (Fig. 10) and the final path map (Fig. 11) are shown below.

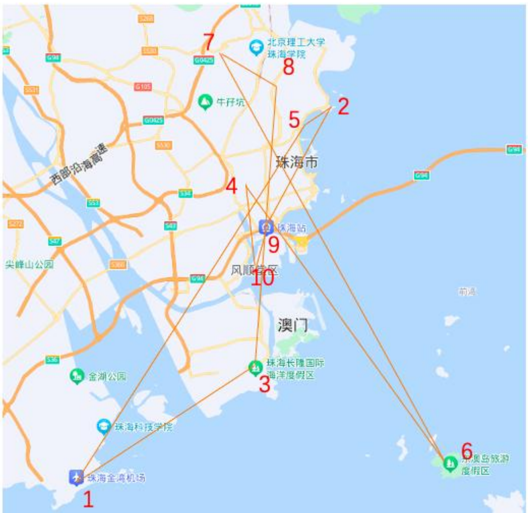

Fig. 10. Visiting points on Zhuhai (Photo/Picture credit: Original).

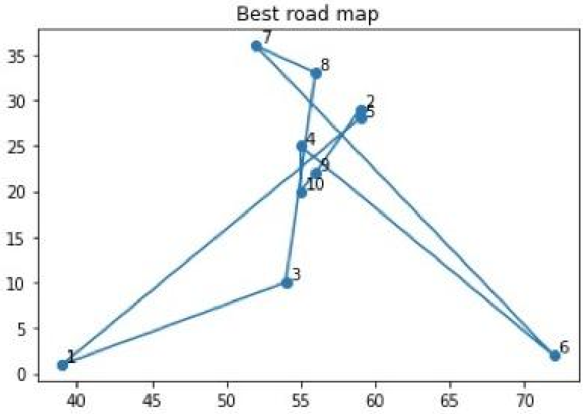

Fig. 11. The best visiting path on Zhuhai (Photo/Picture credit: Original).

From the output, the visit index is: [1, 5, 2, 9, 10, 4, 6, 7, 8, 3]. Now, let analyzing this result. Suppose a person is a tourist after arriving in Zhuhai. In that case, this perons wants to put the luggage first and set the high satisfy to point 5, so the model advises people to go from point 1 (JinWan airport) to point 5 (Yang Ming Square), where it is convenient to get accommodation. After that, visit point 2 (Zhuhai Grand Theatre), which is close to point 5. In addition, points 9 (Gong Bei) and 10 (Ruins of St.Paul) are accessed together due to the way to Macau through Gong Bei. Similarly, points 7 (Hui Tong Village) and 8 (Feng Huang Mountain) are also visited together because of their proximity.

Another interesting phenomenon is that Dong AO Island's visiting order is from point 4 (Yuan Ming New Garden) to 6 (Dong AO Island). However, point 6 is closer to point 1 (JinWan airport) and point 3 (Chimelong Ocean Kingdom). Why not connect points 1 and 6 or 3 and 6 when visiting the route? There is a reason for this. When the order is from 4 to 6, the total cost is less than the other two options. Although the three points have similar distances, the model must consider time and satisfaction comprehensively. When inputting the satisfy matrix, this study sets the satisfy from point 1 to 6 and 3 to 6 very low, which means the willingness to go from point 1 to 6 or 3 to 6 is low, and it will cause the total cost to become high, and the model will avoid choosing these routes.

After analyzing, the visit index may satisfy the actual situation. Therefore, the improved algorithm can be reasonably applied to practical problems. Additionally, more advanced algorithms such as neural network will be also considered for further improving the performance of the model due to their successful application in routing optimization problem [9, 10].

4 Conclusion

The ACO is reviewed in this research, and the improved ACO is tested on TSP and QAP. Finally, the improved algorithm is applied to the practical problem. By combining TSP and QAP, the improved ACO can be applied to both problems. Applying the improved ACO to multiple data sets of TSP and QAP proves that the algorithm can solve these two kinds of problems. The improved ACO is applied to a practical problem combining TSP and QAP, and the results show that the proposed route is consistent with the expected situation. In the future, the plan is to adapt the algorithm to handle larger amounts of data.

References

1. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE transactions on systems, man, and cybernetics, part b (cybernetics), 26(1): 29-41 (1996).
2. Qiao, D., Pei, J., Wen, X., et al.: An Improved Ant Colony Algorithm for Solving Single Machine Total Weighted Delay Scheduling Problem. China Mechanical Engineering, 29(22): 2703 (2018).
3. Naidu, J. T., Gupta, J. N. D., Alidaee, B.: Insights into two solution procedures for the single machine tardiness problem. Journal of the Operational Research Society, 53: 800-806 (2002).
4. Marco. D., et al.: Ant Colony Optimization Artificial Ants as A Computational Intelligence Technique. Ieee Computational Intelligence Magazine, pp. 28-39 (2006).
5. Stutzle, T., Hoos, H.: MAX-MIN ant system and local search for the traveling salesman problem. Proceedings of 1997 IEEE international conference on evolutionary computation (ICEC'97). IEEE, 309-314 (1997).
6. Stützle, T., Hoos, H. H.: Improving the Ant System: A detailed report on the MAX–MIN Ant System. FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12, (1996).
7. Zhao, H. P., et al.: Evolutionary Stable Strategy Ant Colony Algorithm for Solving Graph Colouring. Computer Applications and Software, 22-25 (2013).
8. Chun, H., et al.: Study of Ant Colony Algorithm for Second Assignment Issue. Railway Transport and Economy, 68-70 (2005).
9. Liu, X., Zhang, D., Zhang, J., et al.: A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. Cluster Computing, 1-15 (2021).
10. Liu, B.: Logistics distribution route optimization model based on recursive fuzzy neural network algorithm. Computational Intelligence and Neuroscience, (2021).