Ant Colony Optimization With an Improved Pheromone Model for Solving MTSP With Capacity and Time Window Constraint

MIN WANG\textsuperscript{1}, TONGMAO MA\textsuperscript{2}, GUILING LI\textsuperscript{3}, XUE ZHAI\textsuperscript{4}, AND SIBO QIAO\textsuperscript{4}

\textsuperscript{1}College of Control Science and Engineering, China University of Petroleum, Qingdao 266580, China
\textsuperscript{2}Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid (UPM), Campus de Montegancedo, 28660 Boadilla del Monte, Spain
\textsuperscript{3}College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
\textsuperscript{4}College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China

Corresponding authors: Guiling Li (qd-liguiling@163.com) and Min Wang (1009070793@qq.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61572523, Grant 61873281, Grant 61672033, Grant 61873280, and Grant 61972416.

\section*{ABSTRACT}
The optimization of logistics distribution can be defined as the multiple traveling salesman problem (MTSP). The purpose of existing heuristic algorithms, such as Genetic Algorithm (GA), Ant Colony Algorithm (ACO), etc., is to find the optimal path in a short time. However, two important factors of logistics distribution optimization, including work time window and the carrying capacity of the vehicle in distribution system, have been ignored. In this paper, we consider the influences of time limitation of modern commercial logistics and carrying capacity of the vehicle on the logistics optimization, and then propose a MTSP with constraints of time window and capacity of each salesman. We design a novel hybrid algorithm by combining the minimum spanning 1-tree with ACO to find the optimal solution. In addition, we improve the pheromone update rules to increase the search efficiency of ACO algorithm. The experiments show that the novel hybrid algorithm achieves a shorter path than the other algorithms.

\section*{INDEX TERMS}
Multiple traveling salesman problem, ant colony optimization, time window, capacity.

\section*{I. INTRODUCTION}
Logistics distribution industry becomes more and more important due to the rapid development of the e-business. Nowadays, there are many challenges in logistics distribution industry. For example, Nov. 11\textsuperscript{th} is a peak date for online shopping that all the products online have a big discount and people go on a shopping spree, so there are billions of express items needed to be delivered during that time. Therefore, it is important to deliver the products to customer within a short time. Service composition [1]–[3] is a way which can combine all the logistics services processes efficiently. Another challenge is the balance between paths and items and it is more difficult. The logistics services process in China is to deliver all items from the head office to each brand. At the same time, the optimization of time cost and road distance, the amount of items, the number of transport vehicles as well as their carrying capacity are taken into account.

MTSP is a generalization of the well-known traveling salesman problem (TSP), where at least two salesmen could be used to deliver items in the solution. Obviously, the logistics distribution question is the same as a multiple traveling salesman problem with constraints. Moreover, MTSP could be extended to a wide variety of vehicle routing problems (VRP) according to incorporating a number of additional side-constraints. In addition, finding a quick and efficient way to transport goods and materials in earthquake relief work or military operation is significant.

There are many solutions to solve TSP like Tabu search, simulated annealing, genetic algorithms. The ant colony system (ACS) is improved by this paper which is also a well-behaved solution. It is a kind of the group of meta-heuristic methods. This idea was firstly proposed in the early 90s by Dorigo [4]. ACO is generated by simulating the real behavior of ants in nature and it is widely used to solve practical problems [5]–[7]. Many different transforms of ACO have been proposed. On the basic of these, a new optimal ACO with some improvements is proposed in the paper.
First of all, the minimum spanning 1-tree [8] can be applied to construct the path in MTSP. Secondly, the improved model and pheromone update rules make the algorithm more effective.

Our main contributions are as followings:
1. Redefining multiple traveling salesman problem with time window and capacity of each salesman, which comes from real life problem such as logistics industry;
2. Building an improved pheromone model with new pheromone update rules;
3. Combining the minimum spanning 1-tree with ACO based on the relationship between the optimal TSP tours and spanning tree.

The rest of the paper is organized as follows. Section II shows the related work about TSP and ACO on solving TSP. In the section III, the definition of MTSP and possible solutions for MTSP are discussed. Section IV discusses the algorithm of ant colony system. Section V proposes an optimal ACO algorithm on the basic of previous section. Section VI shows the evaluation and experiment of the improved algorithm. Section VII concludes the paper and outlines the future work.

II. RELATED WORK
A. RELATED WORK ON TSP
TSP is a typical branch of metaheuristic algorithm and there is a lot of research on it. We list a number of representative works recently, and there is an overview article by Panwar and Gupta [9], where earlier works are discussed in detail. They conclude some TSP for solving soft computing techniques.

1. Ant Colony Optimization has been widely used in proving the complexity of combinatorial problem. The movement of each ant is effected by the intensity of pheromones contained on the path. The ACO algorithm can calculate good results in a short time after a few iterations [10].
2. In [11], Ding et al. try to look for the shortest Hamiltonian cycle in all clusters, then remove a selected edge in every cycle to build an intra-cluster path. At last, attach the whole intra-cluster paths in a particular sequence to organize a complete tour. A novel method called Two-level Genetic Algorithm for the Cluster TSP is proposed, which is a good choice to alter an extensive TSP into a CTSP that can be settled by the TLGA. This method is well applied to large-scale TSP.
3. In [12], the conventional particle swarm optimization (PSO) is improved. This method takes the heuristic factor, crossover operator and adaptive disturbance factor into account. Then, the authors propose a novel hybrid discrete PSO algorithm which can improve the search performance in convergent speed and precision. This method can be applied to solve the problem of path optimization in TSP.
4. In [13], random keys are introduced to solve the coding problem of genetic algorithm. Random keys play an important role in the problems where permutations of the integers are required as well as feasibility problems caused by traditional one-or two-point crossover.

Also, there are some other algorithms [14]–[18] related to TSP. For example, in [18], Xu et al. redefine a general colored traveling salesman problem and propose a Delaunay-triangulation-based Variable Neighborhood Search algorithm. The new algorithm performs well in large-scale problems.

B. RELATED WORK APPLYING ACO TO TSP AND MTSP
In this section, we review the literatures that ant colony optimization is used to settle traveling salesman problem as well as multiple traveling salesman problem and figure out their pros and cons and clarify the remaining gaps and challenges for further investigations.

In [19], Mueller and Kiehne present a hybrid way by combining artificial ants with neurons intelligently. In the paper, they test diverse parameter combinations with the purpose of getting the best performing parameter settings. They subsume the obtained insights into an intelligent architecture including Ant Colony Optimization and Self Organizing Map. In [20], a resource allocation method (LET-ACO) is proposed and about 40% energy consumption can be avoided in LET-ACO.

In [21], Olivas et al. propose a method in which a fuzzy system is used to improve the parameter adaptation in ACO. The dynamical adaption of the parameter is responsible for the evaporation of the pheromone trails in the algorithm. Basically, the fuzzy system is responsible for the abilities of the algorithm for exploration and exploitation in the search space and the goal is to get better results. They compare their method with the original algorithm and the result shows that their method performs better than the original algorithm. But they do not have enough evidence to reject the null hypothesis in Burma14 problem. They also propose a fuzzy system for parameter adaptation in the Ant Colony Optimization (ACO meta-heuristic) in [22]. In this case, they use the dynamic adaptation parameters of Alpha and Rho to decide the performance of ACO whether in a global or in a local search.

In [23], Neyoy et al. explore a novel method of diversity control in ACO. To avoid or slow down full convergence, the authors use the dynamically varying alpha parameter. According to the performance of different variants of the ACO algorithm, they choose the best one as the basis of the proposed approach. Moreover, a convergence fuzzy logic controller with the purpose of maintaining diversity to avoid premature convergence is created.

In [24], Wang et al. propose an ant colony algorithm with neighborhood search called NS-ACO to solve the problem of dynamic TSP which is composed by random traffic factors. They take the advantage of the short-term memory to enhance the kinds of solutions. Also, to find the optimized solutions, they used the operations of swap, insertion and 2-opt moving. As a consequence, the experiments show that the method of NS-ACO performs better than conventional ACS and the ACO with random immigrants.
In [25], Elloumi et al. propose an improved PSO algorithm combined by ACO and apply it to TSP. According to the comparison experiment, the performance of improved PSO algorithm is perfect. Generally, ACO and PSO have some advantages in solving TSP. On the one hand, the original PSO algorithm can be applied well to solve the problem of moderate dimensions and search space. However, the original PSO algorithm cannot be applied to solve the problem when it is in a more complex search space. They improve the ACO algorithm and enhance the performance to solve complex problems.

As shown above, ACO and its improved algorithms perform well in TSP and MTSP. And there are many other nature-inspired methods to solve this kind of problem. Because of the lack of some constraints such as time and load for each salesman, these existing methods cannot be applied well to the actual industry. In this paper, we consider these constraints when applying MTSP into cost-effective fields and find the optimal path quickly to reduce the total cost.

Besides, there are also other algorithms that can inspire some ideas on solving MTSP [26], [27]. In [28], Huang et al. propose a niching memetic algorithm for Multi-solution Traveling Salesman Problem. In order to enable the parallel search of multiple optimal solutions, this algorithm is characterized by a niche preservation technique. They also use an adaptive neighborhood strategy to improve the search efficiency. In [29], Jiang et al. propose a new efficient hybrid algorithm for large scale multiple traveling salesman problems. They require the minimum and maximum number of cities that each salesman should visit. Then they combine partheno genetic algorithms and ant colony algorithms which is sufficiently effective. In [30], Vandermeulen et al. want to make the workload balanced so that they use average Hamiltonian partition problem. By using AHP, they improve the partitions of a graph and guarantee to improve the solution’s quality. This algorithm is good at solving large problems within a better run-time. In [31], Venkatesh et al. propose a multi-start iterated local search algorithm for the maximum scatter traveling salesman problem. They develop insertion and modified 2-opt moves on two local search algorithms.

III. PROBLEM DEFINITION AND FORMULATION

TSP is a typical combinatorial optimization problem that belongs to NP complete problem [10]. The aim is to find the shortest path by visiting every node once and then returning to the start node in a complete weighted graph $G$ with $n$ nodes and $n(n-1)$ edges. Furthermore, MTSP could use more than one salesman which is different from the TSP with only one salesman. When considering the carrying capacity and time windows, this problem becomes more meaningful in real life.

The MTSP with constraints can be defined as follows. First, given a set of nodes, arrange $m$ salesmen at a single depot node. Then, finding tours for all $m$ salesmen, and they all need to start and end at the depot. Meanwhile, the other nodes can be visited exactly once and the cost of visiting all nodes will be least. The two constraints are: every node is visited in a fixed time window and every salesman should take items as many as possible [32]. Specifically, for a city $i$, the salesman should visit it in a time window $E_i$, $L_i$, where $E_i$ is the earliest time that the salesman could visit and the salesman should visit city $i$ before $L_i$. And for every salesman, the maximum items they can carry is $M$.

Comparing the TSP with the MTSP, it is obvious that MTSP is more adequate to simulate real life situations, and it is capable of handling plenty of salesmen problem.

MTSP can be applied in the following context: print scheduling, workforce planning, transportation planning, mission planning, production planning, satellite systems, etc. Each context includes various types of real applications.

Consequently, there are various solutions proposed for MTSP. At first, some accurate solutions like cutting plane, branch and bound, and Lagrange relaxation were proposed. Then, heuristics are applied to this problem. Moreover, evolutionary algorithm, simulated annealing, tabu search, genetic algorithms and neural networks also have a good ability on working out the MTSP.

IV. ANT COLONY OPTIMIZATION

A. INTRODUCTION OF ACO

Ant colony optimization is a heuristic algorithm which is introduced into many combinatorial optimization problems due to it is one of the highest performance computing methods for MTSP [33]. Many efforts have been made on ant colony optimization techniques in different areas.

It takes inspiration from ants’ social behaviors of finding the best way between food and the ant nest. Ants can communicate with each other about amount of food and the distance of route. The process can be divided into two parts and a positive feedback mechanism is applied by the ants [34]. As shown in Figure 1, (a) and (b) are the first part, while the second part includes (c) and (d). At the first part, each ant chooses routes randomly and releases pheromone in the routes. A global pheromone updating rule is applied when all ants have finished their tours. Pheromone accumulates at a higher rate on the shorter path, so that more ants begin to choose the shorter path which has a higher intensity.
of pheromone. At the second part, ants affect each other by pheromone which is the main factor to lead the ants to find shorter route. Then the process is iterated to find the global optimal solution. In ACO, artificial ant replaces the real ant. They both can affect surrounding environment such as the intensity of pheromone in the path [35].

**B. ORIGIN ANT COLONY ALGORITHM**

Some definitions are introduced in Table 1, in order to describe the problem clearly.

**TABLE 1. Definitions of variable.**

| Variable | Definition |
|----------|------------|
| $b_i(t)$ | Amount of the ant in city $i$ at $t$ time |
| $\tau_{ij}(t)$ | The pheromone of the path $(i,j)$ at $t$ time |
| $N$ | The maximum number of the TSP |
| $d_{ij}$ | The distance between city $i$ and city $j$ |
| $m$ | The number of ants ($m$) in the problem and $m = \sum b_i(t)$ |
| $p_{ij}^k(t)$ | Selection probability of ant $k$ moves from city $i$ to city $j$ at $t$ time |

At first, there is no pheromone on every path and $\tau_{ij}(0) = \text{const}$. Ant $k$ $(k = 1,2,\ldots,m)$ chooses path by the intensity of pheromone on its way of moving. Set $\text{tabu}_k(k = 1,2,\ldots,m)$ to mark the visited cities and it will be updated according to the transfer of ants state. During the iterative process, $p_{ij}^k(t)$(the state transition probability) relies on both the amount of information and the existing heuristic information on the path [36].

$$p_{ij}^k(t) = \left\{ \begin{array}{ll} \frac{[\tau_{ij}(t)]^\alpha[\eta_{ij}(t)]^\beta}{\sum_{se J_k(i)}[\tau_{is}(t)]^\alpha[\eta_{is}(t)]^\beta}, & \text{if } j \in J_k(i), \\
0, & \text{otherwise.} \end{array} \right. \tag{1}$$

In the formula 1, $J_k(i) = \{C - \text{tabu}_k\}$ is a collection of cities that ant $k$ can visit in next step. $\alpha$ denotes the extent of the group cooperation of ants. That is to say, The bigger $\alpha$ is, the more likely an ant chooses a path passed away by its companion and the better group cooperation of the whole ant has. $\beta$ is expected heuristic factor which shows the relative importance of visibility and the value of choosing a path. The bigger $\beta$ is, the higher state transition probability closes to greedy criterion. $\eta_{ij}$ is a heuristic function, which corresponds to $\beta$.

$$\eta_{ij} = \frac{1}{d_{ij}}. \tag{2}$$

For each ant, a smaller $d_{ij}$ can decide a bigger $p_{ij}^k(t)$ that is the expected value of moving from $i$ to $j$.

Considering that too much accumulation of pheromone will cover up the heuristic information, so that after ants finish their tours, pheromone needs to be weakened. At $t + n$, set the pheromone on path $(i,j)$ as follows:

$$\tau_{ij}(t + n) = (1 - \rho)\tau_{ij}(t) + \Delta \tau_{ij}(t), \tag{3}$$

$$\Delta \tau_{ij}(t) = \sum_{k=1}^{m} \Delta \tau_{ij}^k(t), \tag{4}$$

$\rho$ is pheromone volatilization coefficients, and $1 - \rho$ is pheromone remaining factor. Avoiding too much pheromone left, set $\rho \in [0,1)$. $\Delta \tau_{ij}^k(t)$ is pheromone on path $(i,j)$ which released by ant $k$ every time it finished one route. Ant-Cycle Model applied in this method is as follows:

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } k \text{ passes path } (i,j) \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

$Q$ represents the concentration of pheromone, which can decide the rate of algorithm. $L_k$ is the distance between cities where ant $k$ passed.

Steps of Origin Ant colony algorithm:

1. Initialization Parameters
   - Divide $m$ ants into $n$ cities, initialize pheromone $\tau_{ij}(t) = \text{const}$ on every path $(i,j)$ and it is constant. Set $\Delta \tau_{ij}^0(0) = 0$, and other parameters like $N_c\text{ max}$, $\alpha$, $\beta$, $\rho$, $Q$ and so on.

2. Iteration process
   - While $(N_c < N_c\text{ max})$ do
     - for $i = 1$ to $n - 1$ do Traverse $n$ cities
     - for $k = 1$ to $m$ do For $m$ ants
         - for $j = 1$ to $n$ do For $n$ cities
             - ant $k$ chooses next city $j$ according to formula 1 then moves to city $j$ and add $j$ into tabu$_k$;
             - Calculate tour length of every ant, and update the pheromone by formula 3, 4 and 5 on every path;
     - end while.
   - (3) End and output the result.

**V. NEW ANT COLONY OPTIMIZATION**

**A. PROBLEM DESCRIPTION BY GRAPH THEORY**

In order to improve the search efficiency and obtain the shortest path. Thus, this part, we propose a new algorithm which uses the minimum 1 tree (1-MST) to construct the path. According to the relationship between 1-MST and optimal route, we establish the optimal search library by proximity is established and uses simplified pheromone diffusion model in this paper. In addition, time window and capacity constraints make it more practical significance.

The problem is described by using graph theory as followings. There is a weighted connected graph $G = (V,E,D)$. The set of vertex in the tree $T = \{v_0, v_1, \ldots, v_{n-1}\}$ that represents $n$ cities. Every point has two pairs of weights, and these are carrying capacity weight pair $w_{1i}$, $w_{2i}$ and time window weight pair $E_i$, $L_i$. In this problem, $w_1$ and $w_2$ denote the number of carrying items that needs to take away or bring; $E_i$ is the earliest time the salesman can arrive and $L_i$ is the latest time salesman must arrive. $E$ is a edge set that connects different vertexes. $D$ is nonnegative weights matrix that shows the distance between two cities. The capacity of a salesman is $W$ and the number of original salesman is $M$. By the total carrying items $M_{\text{total}}$, $W = M_{\text{total}}/M$ is obvious.

Tree $T$ is the 1-MST of Graph $G =< V,E >$, the total weights of which is $L(T)$. $T^+(i,j)$ denotes the 1-MST that includes the edge $e(V_i, V_j)$. So the proximity of edge $e(V_i, V_j)$...
is \( \alpha(i, j) = L(T^+(i, j)) - L(T) \). We can conclude that \( \alpha(i, j) \geq 0 \) only when \( e(V_i, V_j) \) is an edge of 1-MST. The smaller \( \alpha(i, j) \) is, the more likely it exists in the shortest route. As a result, we need to build a collection of best choice for each city. N cities of smallest proximity should be added into the original collection. When initializing pheromone matrix, we can set it in terms of the collection. Finally, it can weaken the bad influence of first random path that easily leads into a local optimal solution.

Now, we have to divide the MTSP into M TSP by means of copying start vertexes \( V^*_k (1 \leq k \leq M - 1) \). And the distance \( M \) between start vertexes is infinite and the distance between duplicate vertexes and other vertexes is the same as the first start vertex. It is easy to solve the problem by using the methods on TSP.

Next, we use the Kruskal method to get a minimum spanning tree. At the same time, we add the start vertex and two shortest edges connected with it into the tree. The new tree is called 1-MST. Then the \( T^+(i, j) \) and the proximity between edges are easily to get. When initializing the pheromone, each edge will be given the different original value by the proximity \( \alpha(i, j) \). In this way, the edges in the 1-MST are more easily to show up in the best solution, and it can accelerate convergence. Now, MTSP can be split into \( n \) TSP.

**B. PATH SELECTION**

Every ant starts from an empty solution and chooses cities judging by pheromone and heuristic information. The possibility of the next city chosen by ant \( k \) is

\[
P^k_{ij}(t) = \begin{cases} \left[ \frac{\tau_{ij}(t)}{\sum_{t \in \Delta_k} \tau_{ij}(t)} \right]^{\alpha} \left[ \frac{\eta_{ij}(S_k(t))}{\eta_{ij}(S_k(t))} \right]^{\beta}, & \text{if } j \in \Delta_k(t), \\ 0, & \text{otherwise.} \end{cases}
\]

(6)

\[
\eta_{ij}(S_k(t)) = (\frac{w_j}{\delta_i(k, t)}) + \frac{t}{\delta_j(k, t)}/2,
\]

(7)

where \( \delta_i(k, t) = \frac{\sum_{t \in \Delta_k} \delta_i(t, k)}{M} \) is the average degree of capacity, \( \delta_i(j, t) = \frac{\sum_{t \in \Delta_k} \delta_i(t, k)}{L_t + \Delta w_j} \) is the proportion of city \( j \)'s carrying items occupy in ant \( i \), \( \delta_j(k, t) = \frac{\sum_{t \in \Delta_k} \delta_i(t, k)}{M} \) is the average degree of time window, \( \delta_j(t, k) = \frac{\sum_{t \in \Delta_k} \delta_i(t, k)}{L_t - E_j} \) is the proportion time lag occupies in time window.

Furthermore, whether a city can be selected should depend on two constraints. First, the carrying items cannot be more than the capacity, at the same time, try to put more items.

\[
(W_i = \sum_{j \in \text{tab}_i} w_j) \leq W, \quad \text{(constraint 1)}
\]

\[
(W_i + \sum_{j \in (V^*_i - \text{tab}_i)} \Delta w_j) \leq W, \quad \text{(constraint 2)}
\]

where \( i = 1, 2, \ldots, M, j = 0, 1, 2, \ldots, n + M - 1 \).

Secondly, every ant has to pass a city at a special time window \([E_i, L_i]\). In order to make it easier, set penalty value \( D_1 \) for early arrival and \( D_2 \) for late arrival \( D_1 < D_2 \). Obviously, we can get the penalty value \( d \) and total penalty value \( D \). For every city, if the value is too big that is \( d_i > d \) or \( D > D_{\text{max}} \) (constraint 3), this city cannot be the next city.

**C. MODEL AND UPDATE RULES**

When an ant searches the path, it releases pheromones that directly have influence on two cities along the path. At the same time, the pheromones spread out from the edge, which also have impact on the behavior of the other ants in the vicinity, in order to reduce the interference when other ants elect the next city. So it can improve the convergence rate [37].

![Figure 2. Simplified pheromone diffusion.](image)

In Figure 2, \( X \) represents the distance between the source of pheromones; \( Y \) represents the amount of information; Point \( O \) shows the source location. The pheromone of point \( O \) is \( D_{\text{max}} \): \( \theta < 90^\circ \) which is fixed; The highness of cone: \( h \); Outward diffusion radius: \( r \), and \( r = h \tan \theta; \sigma \) is the distance between point \( C \) and \( O \); So the pheromone in \( C \) is

\[
D_c = D_{\text{max}} \frac{h \tan \theta - \sigma}{h \tan \theta}.
\]

(8)

When an ant passes an edge \((i, j)\), it will releases the pheromone which will affect the proximal cities. The city \( m \) is within the range of the diffusion, and the formula of pheromone on path \((i, j)\), and path \((i, m)\), are as followings:

\[
\Delta \tau^k_{ij}(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } k \text{ passed (i,j)}, \\ 0, & \text{otherwise,} \end{cases}
\]

(9)

\[
\Delta \tau^k_{im} = \frac{\gamma Q}{d_{im}(d_{im})^\omega} \cot \theta, \quad \text{where } d_{im} < \frac{d_{\omega + 1}}{d_{ij}},
\]

(10)

The pheromone in city \( j \) spreads to city \( k \) is the same as formula for city \( i \).
And set volatile coefficient as follows:

\[ Q(t) = \begin{cases} 0.5, & \text{if } \frac{t}{N_{\text{max}}} \leq 0.5, \\ 2 \frac{t}{N_{\text{max}}}, & \text{otherwise}. \end{cases} \]  

(11)

And set volatile coefficient as follows:

\[ \rho(t) = \begin{cases} 1, & \text{if } Q(t) > 0.5, \\ \rho, & \text{otherwise}. \end{cases} \]  

(12)

And now we can get the new rule for pheromone changes:

\[ \tau_{ij}^{\text{new}} = (1 - \rho(t))\tau_{ij}^{\text{old}} + \Delta \tau_{ij} + \alpha(i,j), \quad \text{where } \Delta \tau_{ij} = \sum_{k=1}^{m} \Delta \tau_{ik}^{k}. \]  

(13)

D. IMPROVED ALGORITHM

The proposed algorithm is combined with 1-MST, improved pheromone model and new update rules. More realistic conditions are taken into account. First of all, set up parameters. Maximum iterations \( N_{\text{max}} \), ant number is \( \text{Ant} = (n + M - 1)/1.5 \), capacity of ant \( W = 40 \), solution of ant is \( S_k \), best denotes the shortest route in one iteration, and set \( \alpha = 1, \beta = 2, \rho = 0.9 \). The algorithm is described as follows:

1. Read the data and process it. Prepare the required data.
2. Build 1-MST and get the proximity, then initialize pheromone matrix \( PM \), the original solution \( S_k \) and the best solution \( \text{best} \).
3. Initialize the ant colony, and calculate the heuristic information of every city.
4. While cycle times are less than \( N_{\text{max}} \)
   2. { 
      3. for \( k = 0, 1, \cdots, \text{Ant} - 1 \)/every ant 
      4. for \( k = 0, 1, \cdots, \text{city do}/every city 
      5. { 
      6. Calculate the probability of every city as formula 1; 
      7. if the city satisfies the constraint condition, add it into \( S_k \); 
      8. Get the time and weight and if ant passed \( v_0 \) or copying point, reset; 
      9. else search next point; 
      10. } 
      11. for \( k = 0, 1, \cdots, \text{Ant} - 1 \)/every ant 
      12. { Calculate the length and update the best; } 
      13. for \( k = 0, 1, \cdots, \text{city do}/every edge 
      14. { 
      15. Update the pheromone \( PM \); 
      16. Set unvisited vertex worst value; 
      17. } 
      18. }
5. output the best solution.

VI. EVALUATION

In order to evaluate the method we proposed, we took several MTSP problems from the TSPLIB website [38] (the best results are shown in Table 4) and did the test. Capacity information and time windows are added into them. The experiments results on modified figures are in Table 2.

**TABLE 2. The results of the new ACO for MTSP (+ means the re-modified figure).**

| Instance | Best | Worst | Average | Minimum iteration |
|----------|------|-------|---------|------------------|
| Berlin52+ | 7679 | 8324 | 7899.5 | 45 |
| Eil51+ | 437 | 466 | 449.6 | 26 |
| Eil76+ | 348 | 571 | 560.3 | 32 |

As we can see from Table 2, the best solution of Berlin52+, Eil51+ and Eil76+ are 7679, 437 and 548 respectively, the optimum solution of Berlin52, Eil51 and Eil76 with no constraints are 7542, 426, 538 respectively which are shown in Table 3. From this experiment we can see that three instances have been solved and the results are closer to best results without constraints, so our algorithm performs well when constraints are taken into consideration.

VI. EVALUATION

In order to evaluate the method we proposed, we took several MTSP problems from the TSPLIB website [38] (the best results are shown in Table 4) and did the test. Capacity information and time windows are added into them. The experiments results on modified figures are in Table 2.

**TABLE 3. The results and the comparison.**

| Instance | Optimum(1) | Algorithm | Best | Average | Relative error | Minimum iteration |
|----------|------------|-----------|------|---------|----------------|------------------|
| Berlin52 | 7542 | ACO | 7578 | 7645.6 | 130.4 | 75 |
|           | 7542 | New ACO | 7561.1 | 123.7 | 42 |
| Eil51 | 426 | ACO | 431 | 440.5 | 20.9 | 43 |
|           | 426 | New ACO | 431.1 | 8.3 | 21 |
| Eil76 | 538 | ACO | 544 | 557.4 | 26.6 | 45 |
|           | 538 | New ACO | 545.3 | 10.4 | 27 |

The relative error \( \sqrt{\sum(R_i - (1))²/n} \), where \( R_i \) is the length of every route.

As Table 3 shown, new ACO can get better results than normal ACO. Next, we compare our method with other algorithms such as improved genetic algorithms (IGA) [39], invasive weed optimization algorithm (IWO) [40], improved partheno genetic algorithms (IPGA) [40] and the instance in [29] and [41]. The results of this comparison are shown in the Table 4.

**TABLE 4. The results of comparing with other algorithms.**

| Instance | Eil51 | Eil76 |
|----------|-------|-------|
| TSPLIB | 426 | 538 |
| New ACO | 426 | 538 |
| IPGA | 428.98 | 557.71 |
| IWO | 652 | 857 |
| IGA | 443 | 568 |
| Instance in [29] | 492 | 616 |
| Instance in [41] | 444 | 358 |
As Table 4 shown, the results demonstrate that the New ACO can be well applied to find high quality solutions than the other algorithms.

VII. CONCLUSION
In this paper, we have presented a method for solving the multiple traveling salesman problem based on the improved ant colony optimization. The improved ACO has used the relationship between MTSP and 1-MST, and the simplified pheromone diffusion. The new pheromone update rules helped a lot to achieve a better solution. The experimental results have shown that the new method has quick convergence speed and can be well applied to find best solutions.

In future, we decide to combine multi-objective TSP with parallel processing [42], [43]. Parallel processing can improve algorithm speed and reduce algorithm execution time. This method may be greatly improve the speed of finding best solutions. Furthermore, we will implement an application that cannot only help the logistics distribution industry but also make the transportation in earthquake relief work or military operation easier.

REFERENCES
[1] X. Sun, S. Wang, Y. Xia, and W. Zheng, “Predictive-trend-aware composition of Web services with time-varying quality-of-service,” IEEE Access, vol. 8, pp. 1910–1921, Dec. 2020.
[2] S. Wang, D. You, and M. Zhou, “A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S 4PR,” IEEE Trans. Autom. Control, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.
[3] S. Pang, M. Wang, S. Qiao, X. Wang, and H. Chen, “Fault diagnosis for service composition by spiking neural P systems with colored spikes,” Chin. J. Electron., vol. 28, no. 5, pp. 1033–1040, Sep. 2019.
[4] M. Drigo, V. Maniezco, and A. Colomni, “The ant system: Optimization by a colony of cooperation agents,” IEEE Trans. Syst. Man, Cybern. B, Cybern., vol. 26, no. 1, pp. 29–41, Feb. 1996.
[5] Y. Feng, M. Zhou, G. Tian, Z. Li, Z. Zhang, Q. Zhang, and J. Tan, “Target disassembly sequencing and scheme evaluation for CNC machine tools using improved multiobjective ant colony algorithm and fuzzy integral,” IEEE Trans. Syst. Man, Cybern. Syst., vol. 49, no. 12, pp. 2438–2451, Dec. 2019.
[6] M. Boukens, A. Boukabou, and M. Chadli, “A real time self-tuning motion controller for mobile robot systems,” IEEE/C courtesy A. Ja. Autom. Syst., vol. 6, no. 1, pp. 84–96, Jan. 2019.
[7] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 2, pp. 601–614, Feb. 2019.
[8] T. Volgenant and R. Jonker, “A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation,” Eur. J. Oper. Res., vol. 9, no. 1, pp. 83–89, Jan. 1982.
[9] P. Panwar and S. Gupta, “Brief survey of soft computing techniques used for optimization of TSP,” Int. J. Comput. Sci., vol. 2, no. 1, pp. 54–59, Jan. 2013.
[10] I. Brezina and Z. Čičková, “Solving the travelling salesman problem using the ant colony optimization,” Manage. Inf. Syst., vol. 6, no. 4, pp. 010–014, Dec. 2011.
[11] C. Ding, Y. Cheng, and M. He, “Two-level genetic algorithm for clustered traveling salesman problem with application in large-scale TSPs,” Tsinghua Sci. Technol., vol. 12, no. 4, pp. 459–465, Aug. 2007.
[12] H. L. Fan and X. L. Lin, “Discrete particle swarm optimization for TSP based on neighborhood,” Appl. Rev. Comput., vol. 28, no. 2, pp. 511–513, Feb. 2011.
[13] L. V. Snyder and M. S. Daskin, “A random-key genetic algorithm for the generalized traveling salesman problem,” Eur. J. Oper. Res., vol. 174, no. 1, pp. 38–53, Oct. 2006.
[14] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, “A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems,” IEEE/Courtesy A. J. Autom. Syst., vol. 6, no. 4, pp. 904–916, Jul. 2019.
[15] L. Wang and J. Lu, “A memetic algorithm with competition for the capacitated green vehicle routing problem,” IEEE/Courtesy A. J. Autom. Syst., vol. 6, no. 2, pp. 516–526, Mar. 2019.
[16] X. Meng, J. Li, M. Zhou, X. Dai, and J. Dou, “Population-based incremental learning algorithm for a serial colored traveling salesman problem,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 48, no. 2, pp. 277–288, Feb. 2018.
[17] A. Misevičius, “Using iterated tabu search for the traveling salesman problem,” Inf. Technol. Control, vol. 3, no. 32, pp. 29–40, Jan. 2004.
[18] X. Xu, J. Li, and M. Zhou, “Delaunay-triangulation-based variable neighborhood search to solve large-scale general colored traveling salesman problems,” IEEE Trans. Intell. Transp. Syst., early access, Feb. 21, 2020, doi: 10.1109/TITS.2020.2972389.
[19] D. C. Mueller and N. Kiehne, “Hybrid approach for TSP based on neural networks and ant colony optimization,” in Proc. IEEE Symp. Ser. Comput. Intell., Cape Town, South Africa, Dec. 2015, pp. 1431–1435.
[20] L. Xu, “An overview and perspectives on bidirectional intelligence: Lmser duality, double IA harmony, and causal computation,” Math. Problems Eng., vol. 2017, no. 2, pp. 1–10, 2017.
[21] F. Olivas, F. Valdez, and O. Castillo, “Ant colony optimization with parameter adaptation using fuzzy logic for TSP problems,” in Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization (Studies in Computational Intelligence), vol. 601, Cham, Switzerland: Springer, 2015, pp. 593–603.
[22] F. Olivas, F. Valdez, and O. Castillo, “Dynamic parameter adaptation in ant colony optimization using a fuzzy system for TSP problems,” in Proc. Conf. Int. Fuzzy Syst. Assoc. Eur. Soc. Fuzzy Log. Technol., Gijon, Spain, 2015, pp. 765–770.
[23] H. Neyoy, O. Castillo, and J. Soria, “Dynamic fuzzy logic parameter tuning for ACO and its application in TSP problems,” in Recent Advances on Hybrid Intelligent Systems (Studies in Computational Intelligence), vol. 601, Berlin, Germany: Springer-Verlag, 2013, pp. 259–271.
[24] Y. Wang, Z. Xu, J. Sun, F. Han, Y. Todo, and S. Gao, “Ant colony optimization with neighborhood search for dynamic TSP,” in Proc. Conf. Int. Conf. Swarm Intell., Bali, Indonesia, Jun. 2016, pp. 434–442.
[25] W. Elloumi, H. El Abed, A. Abraham, and A. M. Alimi, “A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP,” Appl. Soft Comput., vol. 25, pp. 234–241, Dec. 2014.
[26] L. Li, H. Liu, “An overview and perspectives on bidirectional intelligence: Lmser duality, double IA harmony, and causal computation,” IEEE/Courtesy A. J. Autom. Syst., vol. 6, no. 4, pp. 865–893, Jul. 2019.
[27] W. Luan, G. Liu, C. Jiang, and M. Zhou, “MPTR: A maximal-marginal-relevance-based personalized trip recommendation method,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 11, pp. 3461–3474, Nov. 2018.
[28] T. Huang, Y.-J. Gong, S. Kwong, H. Wang, and J. Zang, “A niching memetic algorithm for multi-solution traveling salesman problem,” IEEE Trans. Evol. Comput., vol. 24, no. 3, pp. 508–522, Jun. 2020.
[29] C. Jiang, Z. Wan, and Z. Peng, “A new efficient hybrid algorithm for large scale multiple traveling salesman problems,” Expert Syst. Appl., vol. 139, Jan. 2020, Art. no. 112867.
[30] I. Vandermeulen, R. Grob, and A. Kolling, “Balanced task allocation by partitioning the multiple traveling salesperson problem,” in Proc. 18th Int. Conf. Auton. Agents MultiAgent Syst., Montreal, QC, Canada, May 2019, pp. 1479–1487.
[31] P. Venkatesh, A. Singh, and R. Mallipeddi, “A multi-start iterated local search algorithm for the maximum scatter traveling salesman problem,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jun. 2019, pp. 1390–1397.
[32] J.-Q. Li, P. B. Mirchandani, and D. Borenstein, “Real-time vehicle rerouting problems with time windows,” Eur. J. Oper. Res., vol. 194, no. 3, pp. 711–727, May 2009.
[33] M. Tuba and R. Jovanovic, “Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem,” Int. J. Comput. Inform. Commun., vol. 8, no. 3, pp. 477–485, Apr. 2013.
[34] J. M. Cecilia, J. M. García, A. Nisbet, M. Amos, and M. Ujaldón, “Enhancing data parallelism for ant colony optimization on GPUs,” J. Parallel Distrib. Comput., vol. 73, no. 1, pp. 42–51, Jan. 2013.
[35] G. Dong, X. Fu, and H. Xue, “An ant system-assisted genetic algorithm for solving the traveling salesman problem,” Int. J. Adv. Comput. Technol., vol. 4, no. 5, pp. 165–171, Mar. 2012.
[36] X. Shi, X. Wu, T. Song, and X. Li, “Construction of DNA nanotubes with controllable diameters and patterns using hierarchical DNA sub-tiles,” Nanoscale, vol. 8, no. 31, pp. 14785–14792, Aug. 2016.

[37] H. M. Botee and E. Bonabeau, “Evolving ant colony optimization,” Adv. Complex Syst., vol. 2, no. 3, pp. 149–159, Sept. 1998.

[38] TSPLIB Website. [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[39] X. Xu, H. Yuan, M. Liptrott, and M. Trovati, “Two phase heuristic algorithm for the multiple-travelling salesman problem,” Soft Comput., vol. 22, no. 19, pp. 6567–6581, Oct. 2018.

[40] H. Zhou, M. Song, and W. Pedrycz, “A comparative study of improved GA and PSO in solving multiple traveling salesmen problem,” Appl. Soft Comput., vol. 64, pp. 564–580, Mar. 2018.

[41] M. Hou and D. Liu, “A novel method for solving the multiple traveling salesmen problem with multiple depots,” Chin. Sci. Bull., vol. 57, no. 15, pp. 1886–1892, May 2012.

[42] S. Pang, T. Ding, A. Rodriguez-Paton, T. Song, and Z. Pan, “A parallel bioinspired framework for numerical calculations using enzymatic P system with an enzymatic environment,” IEEE Access, vol. 6, pp. 65548–65556, Oct. 2018.

[43] S. Pang, W. Li, H. He, Z. Shan, and X. Wang, “An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing,” IEEE Access, vol. 7, pp. 146379–146389, Oct. 2019.

MIN WANG received the master's degree from the China University of Petroleum, Qingdao, China, in 2019, where she is currently pursuing the Ph.D. degree in control science and engineering. Her current research interests include the theory and application of Petri Net and trusted computing.

TONGMAO MA received the degree from the China University of Petroleum, Qingdao, China, in 2015. He is currently pursuing the Ph.D. degree in artificial intelligence with the Universidad Politécnica de Madrid, Madrid, Spain. His current research interests include artificial intelligence, membrane computing, and bioinformatics.

GIULING LI was born in Shandong, China, in 1974. She received the Ph.D. degrees in control theory and control engineering from the Shandong University of Science and Technology, Qingdao, China, in 2009 and 2014, respectively. In 2005, she was with the Academy of Mathematics and Systems Science, Shandong University of Science and Technology. Her research interest includes the areas of stochastic control.

MIN WANG received the master’s degree from the China University of Petroleum, Qingdao, China, in 2019, where she is currently pursuing the Ph.D. degree in control science and engineering. Her current research interests include the theory and application of Petri Net and trusted computing.

GUILING LI was born in Shandong, China, in 1974. She received the Ph.D. degrees in control theory and control engineering from the Shandong University of Science and Technology, Qingdao, China, in 2009 and 2014, respectively. In 2005, she was with the Academy of Mathematics and Systems Science, Shandong University of Science and Technology. Her research interest includes the areas of stochastic control.

MIN WANG received the master’s degree from the China University of Petroleum, Qingdao, China, in 2019, where she is currently pursuing the Ph.D. degree in control science and engineering. Her current research interests include the theory and application of Petri Net and trusted computing.

XUE ZHAI received the B.S. degree from the Qingdao University of Technology, Qingdao, China, in 2018. She is currently pursuing the degree with the China University of Petroleum, Qingdao. Her current research interest includes Petri Net.

SIBO QIAO received the B.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2017. He is currently pursuing the degree with the China University of Petroleum, Qingdao. His current research interests include deep learning, person re-identification, and object detection.