A dynamic shortest path algorithm based on an improved ant colony algorithm

S J Zhang
Department of Logistic and Information Engineering, Huzhou Vocational and Technical College, Huzhou 313000, China
E-mail: zsjsouth@163.com

Abstract. The shortest path problem is the key problem in intelligent transportation systems. In this paper, the shortcomings of classical shortest path algorithms in solving the dynamic shortest path problem are analyzed. A dynamic traffic network model is constructed, and the characteristics of an ant colony algorithm are analyzed. According to the characteristics of the traffic network, a dynamic shortest path algorithm is proposed based on an improved ant colony algorithm. A simulation experiment showed that the algorithm proposed in this paper could effectively find the shortest path in a dynamic traffic network.

1. Introduction
The shortest path problem can be divided into a static shortest path problem and a dynamic shortest path problem according to the characteristics of the traffic information. The essence of the static shortest path problem is to find the shortest path between two nodes in a deterministic network. There are many classical algorithms, such as the Dijkstra [1], Dreyfus algorithms [2]. These algorithms are on the basis of the Bellman condition [3], that is, in the shortest path from origin node to destination node, the path from origin node to intermediate node is also the shortest path for arriving at this intermediate node; so, the sub-path of each shortest path is also the shortest path. However, the classical algorithm is unable to acquire the shortest path in a dynamic network because although the sub-path obtained at one certain time is the shortest path, at another time, this path may not be the shortest path due to the change of traffic conditions, so the classical deterministic shortest path algorithms are not suitable for solving the shortest path problem in a dynamic traffic network.

In a traffic network, the traffic condition in the traffic network usually changes with time. The dynamic shortest path algorithm seeks for optimal paths for the travelers based on dynamic traffic information. The dynamic shortest path problem is now facing a new challenge due to the increasing scale of the urban traffic network. In a complex urban traffic network, it is not wise for the shortest path algorithm aggressively to pursue the precise solution. The higher the algorithm precision, the more complexity of algorithm time there will be, therefore, it is more difficult to meet dynamic requirements. Therefore, some scholars have studied how to use intelligent algorithms to solve the shortest path problem in complex networks, for example by using the ant colony algorithm (ACO) [4-6]. The purpose of this paper is to study further the shortest-path problem in traffic networks and to develop a novel ACO algorithm to solve the problem efficiently.

2. Symbols and Models

2.1 Dynamic traffic network model
A traffic network whose traffic condition changes with time can be mapped into a connected weighted network \( G: \{V, E, W(e, t)\} \), where \( V = \{(1, 2, \cdots, n) | n \text{ is the node number}\} \) is a set of the nodes of the traffic network, shown in Figure 1; \( E \subset V \times V \) is a set of roads of the traffic networks; \( W(e, t) \) is a set of the impedance functions of the roads of the network at time \( t \), where the values are measured using the travel time of the road. The traffic conditions of the roads always change over, which can lead to a change in the travel time of the road. This kind of traffic network is called as dynamic traffic network (DTN).

![Figure 1. Example of a traffic network](image)

### 2.2 Dynamic Shortest Path Model

In DTN, using \( p_{ij}^t(k) \) to indicate a path from node \( i \) to \( j \) at time \( t \); assuming set \( Q = \{p|p_{ij}^t(k), k = 1, 2 \cdots m\} \) is all paths from node \( i \) to \( j \) at time \( t \) in a certain time region \([t_0, t_n]\). If there is a path \( p_{ij}^t(l) \in Q \), and the following formula is workable:

\[
F(P_{ij}^t(l)) \leq F(p) \tag{1}
\]

Where \( F(\cdot) = \sum w(t) \) represents the sum of travel time of all roads of a path; \( p \) represents any one path in set \( Q \), then \( P_{ij}^t(l) \) is called as the dynamic shortest path from node \( i \) to \( j \) at time \( t \).

### 3. The Basic Ant Colony Algorithm

As a kind of biomimetic intelligent algorithm, the ant colony algorithm simulates the phenomenon where numerous ants find the shortest path from the ant nest to the food location during the foraging process \([7, 8]\).

During the foraging process, the ants select the heading direction based on a certain probability calculated according to the pheromone amount in the path; namely, ant \( k \) selects a path from position \( i \) to next location \( j \) based on a transition probability, which is calculated as follows:

\[
P_{ij}^k(t) = \begin{cases} 
\frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{n \in \text{nallowed}_k} \tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}, & j \in \text{nallowed}_k, \\
0, & \text{otherwise}
\end{cases} \tag{2}
\]

Where, \( P_{ij}^k(t) \) denotes the probability of ant \( k \) choosing road arc \( e_{(i,j)} \) in location \( i \) at moment \( t \), \( \tau_{ij}(t) \) represents the amount of pheromones present on road arc \( e_{(i,j)} \) at moment \( t \), \( \eta_{ij} \) is the visible coefficient from location \( i \) to \( j \); parameter \( \alpha \) denotes the significance of pheromones for the ant selecting the travel direction, and parameter \( \beta \) denotes the significance of the heuristic information; the set \( \text{nallowed}_k = \{0, 1, \cdots, m\} \) represents the next position that ant \( k \) is allowed to reach from position \( i \), it is a subset of the nodes of the network. It can be seen from formula (2) that the transition probability \( P_{ij}^k(t) \) is directly proportional to \( \tau_{ij}^\alpha \eta_{ij}^\beta \), when the amount of pheromones on a road arc is more and the heuristic information is more important, then the ant more likely chooses that road arc.
After all ants finish finding paths once, the amount of pheromones on each road arc will be updated as follows:

\[ \tau_{ij}(t + 1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t, t + 1), \quad (3) \]

\[ \Delta \tau_{ij}(t, t + 1) = \sum_{k=1}^{m} \Delta \tau_{ij}^k(t, t + 1), \quad (4) \]

where \( \Delta \tau_{ij}^k(t, t + 1) \) represents the amount of pheromones that the \( k \)th ant left on road arc \( e_{(i,j)} \); \( \Delta \tau_{ij}(t, t + 1) \) represents the total amount of pheromones that the ant colony left on road arc \( e_{(i,j)} \) within one loop; the pheromones on a road arc volatilize at a certain rate; \( (1 - \rho) \) is the volatility coefficient, where \( \rho \), the residual coefficient, is \( \rho < 1 \).

4. Dynamic Shortest Path Algorithm

The abovementioned content shows that it is workable to simulate the change of traffic condition in the traffic network using the change of the pheromones in the ant colony algorithm. For example, if a certain road in the traffic network is unimpeded, more pheromones can be placed on this road, then the probability of ants selecting this road is higher. When a road is blocked, then zero pheromones can be placed on this segment, and ants will not select this road at this moment. For this purpose, we propose one dynamic shortest path algorithm based on an ant colony algorithm; the following is the formalized description of the algorithm:

Begin  
Step 1: \( t = 0 \).  
Step 2: initialize the parameters of the ant colony  
Step 3: The value of pheromone on the road is assigned  
Step 4: seek the shortest path  
Do  
Step 5: \( t = t + 1 \), if the traffic condition changes, update the pheromone value and execute the operations in step 4; if the traffic condition does not change, maintain the original path.  
While (travelers arrive at destination)  
End

The basic flow chart is shown in Figure 2:
In the foraging process, ants search for their next walking direction following a random principle without guidance. The basic ant algorithm simulates this process to select the next location in the process of marching forward also based on a certain probability that is calculated according to the pheromone amount in the path, and this will greatly affect the algorithm’s search efficiency. To solve this problem, we improved the transition rule to guide the ants’ heading direction. As analyzed above, a traffic network is a geographical network, whose shortest path between a pair of origin–destination nodes distributes along the connecting line between the origin node and the destination node. Therefore, we can use the following approach to improve the transition rule. As shown in Figure 3, given A is the origin node, B is the destination, and an ant has moved to node $i$ from A, and given that there are nodes of $E$ and $F$ to choose to walk to, line $ij$ is parallel to the line $AB$; it can be seen from the figure that the value of the included angle $\theta$ between line $ij$ and $iF$ is less than the included angle $\phi$ between line $iE$ and $ij$, which means that line $iF$ is more parallel to the line $AB$. If the amount of pheromones on road $e_{(i,F)}$ is approximately equal to that on road $e_{(i,E)}$ and the road conditions of the two roads are similar, then the road $e_{(i,F)}$ will be selected to walk along with larger probability.

Figure 3. Example of the search direction

Therefore, in the improved ant colony algorithm, when calculating ants’ transition probability, the coefficient $\eta_{ij}$ in formula (2) can be calculated according to the following formula:

$$\eta_{ij} = \frac{1}{\theta},$$

where the coefficient $\theta$ is described above. It is known from formula (11) that $\eta_{ij}$ will be larger as the road inclines toward the direction to the destination.

5. Experiment

A simulation experiment was conducted to evaluate the algorithm proposed above. An urban traffic network was taken as the experimental object. The experiment simulated the traffic information issuing platform to update traffic information in real-time. For example, for traffic congestion that occurred on a certain road, the estimated congestion duration time was provided, and if the planned path passed the congested road, then the pheromone level of the road was updated, and the algorithm was triggered to recalculate the shortest path.

The algorithm was implemented using the programming language C#, based on a digital map that was constructed on the platform of the geographic information system software ArcGIS10.2. The algorithm parameters were set as follows: the population size of the ant colony was $pop\_size = 30$; the amount of initial pheromone was $\tau_0 = 10$; the amount of pheromones that the ants leave in the path after each iteration was $Q = 100$; the residual coefficient of pheromone was $\rho = 0.8$, and parameters $\alpha = 1$ and $\beta = 2$. Three origin–destination pairs were selected from the experimental traffic network to conduct the experiment. The shortest paths of each origin–destination pair were calculated each time the traffic conditions changed. The experimental results are shown below in Table 1 (the solution was the shortest time to arrive at the destination). The results showed that the algorithm proposed in this paper had improved astringency and high efficiency. This is shown in Table 1, which
reveals that the improved algorithm had a clear improvement in performance compared with the basic ant colony algorithm.

Table 1. Experimental results

| Origin–Destination | Basic Ant Colony Algorithm | Improved Algorithm |
|--------------------|---------------------------|--------------------|
|                    | Shortest path (min) | CPU Time (s)     | Shortest path (min) | CPU Time (s)     |
| 16–535             | 26.3                  | 26.6              | 23.7                  | 3.2              |
| 20–568             | 27.7                  | 28.4              | 23.6                  | 2.5              |
| 77–288             | 24.1                  | 27.2              | 21.2                  | 1.9              |

6. Conclusion

The traffic conditions in a traffic network often change with time. The traditional shortest path algorithms cannot adapt to the dynamic nature of a traffic network. It has important real-world significance to study the dynamic shortest path problem under the situation that traffic networks often congest. In this paper, the spatial distribution nature and dynamic nature of a traffic network were studied. The characteristics of the ant colony algorithm were analyzed, and a dynamic shortest path algorithm based on the ant colony algorithm was proposed. To evaluate the performance of the proposed algorithm, an experiment was conducted; the experimental result shows the new algorithm is practical and feasible.

The ant colony algorithm is an intelligent algorithm with fine properties. In the ant colony algorithm, the ant colony utilizes the pheromone on the roads to search for the shortest path. The change in the pheromone on the roads can be used to simulate the change of situation of the road condition on the traffic network with time. This can ensure that the algorithm meets the dynamic requirements. The dynamic shortest path algorithm proposed in this paper made full use of the characteristics of the ant colony to make the algorithm adapt to the dynamic situation of the traffic network.

Acknowledgment

This work has been supported in part by Natural Science Research Project of Education Department of Zhejiang Province (Y201432450).

References

[1] Dijkstra E W 1959 Numerische Mathematik vol 1 pp 269-271
[2] Dreyfus S 1969 Operations Research vol 17 pp 395-412
[3] Bellman E 1958 Quarterly of Applied Mathematics vol 16 pp 87-90
[4] Basha M, Kumar M S, Pranav V S and Rehman B K 2018 Intelligent Communication, Control and Devices (Springer, Singapore) pp 1243-1252
[5] Song M X, Li J Q, Li L, Yong W and Duan P Y 2018 International Conference on Intelligent Computing (Springer, Cham) pp 831-840
[6] Bell J E and McMullen P R 2004 Advanced engineering informatics vol 18 pp 41-48
[7] Dorigo M, Birattari M and Stutzle T 2006 IEEE computational intelligence magazine vol 1 pp 28-39
[8] Liu J, Yang J, Liu H, Tian X and Gao M 2017 Soft Computing vol 21 pp 5829-5839