A Hybrid Search Scheduler for Dynamic Auto-driving Team Scheduling with Time Window under Cloud Plan

Ming Li
School of Software, East China Jiaotong University, NanChang, JiangXi, China, 330013
Email: mingbeyond@foxmail.com

Abstract. The car cloud plan means that the traveller in the cloud plan does not need to buy an automatic driving vehicle, only need to join the cloud plan, send the travel instruction to the cloud scheduler through the mobile device in hand, and then the cloud scheduler provides the travel service. With the development of actual travel, more and more problems are moving towards the direction of dynamic travel. In this paper, a hybrid solver based on ant colony optimization system search architecture is designed. In the ant colony algorithm, a new algorithm is constructed by combining local search heuristic algorithm with path problem. This paper combines the cloud planning platform with automatic driving vehicle scheduling to quickly collect customer orders and have rapid large-scale processing capabilities, using the order information sent by the traveller to the cloud, through an ant colony searcher to achieve rapid processing of orders.

1. Introduction
Cloud planning is the current trend of Internet development \cite{1,2}. Google has established a cloud computing platform \cite{3} provided by Google Search, which provides developers with Python application services that can manage infrastructure. IBM developed “Blue Cloud” \cite{4}.

In the research of automatic driving scheduling problem, researchers mainly focused on two aspects. The first one is local research \cite{5}, which only generates orders based on known information, regardless of the input of new orders. Regardless of any information about future orders, it is often used in cases where new orders are difficult to predict. The second method is called the forward-looking method \cite{6}, and in the generation of each order, future orders will be inserted into the probability form. In new orders, this method requires new information for future orders. Semet and Taillard \cite{7} proposed taboos to find the route of the vehicle and showed a good solution. Baker and Ayechew \cite{8} combine genetic algorithms with neighborhood search methods to give reasonable results for vehicle scheduling problems.

The online location model based on target scheduling is designed in this paper, and a hybrid search scheduler based on the ant colony algorithm for dynamic time window is proposed. Finally, the simulation experiment is carried out in the traffic network based on free flow and jammed flow \cite{9} in JADE, which verifies the advanced of the solver.

2. Hybrid Search Scheduler Based on Ant Colony Algorithm
2.1. Online Location Model Based on Target Scheduling
The automatic driving vehicle drives at a speed $v$ in the environment $Q$, and the traveller sends a travel instruction to the cloud scheduler through the mobile device in hand. The service operation starts
from 0, and the number of service requests is a random variable positive integer, in the area Q. The number of requests from the present is generated by a varying \( \lambda \) Poisson distribution, and the automatic driving vehicle travels at a constant speed in the area as time passes. When the automatic driving vehicle receives the service instruction, it will give the pedestrian service. In the cloud scheduler, the scheduler combines the a priori maps in the cloud to provide services in combination with the characteristics of beginners and automatic vehicles. This paper models the directed graph \( B = (C, F) \), \( C = (c_0, c_1, \cdots c_n) \) is the rendezvous point representing the customer \( F = \{ (c_i, c_j); i \neq j \} \), is an arc set, where \( (c_i, c_j) \) representing the customer \( i \) to the customer \( j \), \( c_0 \) represents the central distribution point of the vehicle. Among them, the traveller is assigned an automatic driving vehicle service through the travel characteristics of the traveller. Each automatic driving vehicle has a carrying capacity of \( Q \), a time window of \( [e_i, l_i] \), and a service duration of \( s_i \). The goal is to minimize the cost of travel, enabling automatic vehicles to serve travellers. In this paper, this model is defined as a binary math induction problem, if the vehicle \( k \) accesses the customer \( x_i \) immediately after accessing the customer \( x_j \), then \( \gamma_{ijk} = 1 \), otherwise \( \gamma_{ijk} = 0 \). The mathematical representation of the planning problems is shown as follows.

\[
 f(R) = \min \sum_{k=1}^{M} \gamma_{jk}
\]

Subject to:

\[
 \sum_{i=0}^{N} \gamma_{ijk} = 1, j = 1, \cdots, N
\]

\[
 \sum_{i=0}^{N} \gamma_{ijk} = 0, k = 1, \cdots, M, i = 0, \cdots, N
\]

\[
 a_i \leq c_Q
\]

The mathematical expressions indicate that the constraints inside can be constrained by the vehicles, and the detailed equations here are constrained as follows.

Equation (1) indicates that the system benefits are maximized; Equation (2) means that the starting point of each car is the yard; Equation (3) means that after each vehicle visits the traveller, the vehicle also leaves the traveller to serve other travellers; Equation (4) means that the demand of each customer is not greater than the carrying capacity of the automatic driving vehicle.

2.2. Automatic Vehicle Solving Algorithm with Time Window

This section proposes an algorithm for solving the online location model of the automatic driving vehicle based on the target scheduling given in section 2.1. The core part of the algorithm is the hybrid search scheduler based on ant colony algorithm. The following is a description of the solution algorithm.

This section is described based on the ant colony algorithm solver given in the directed graph in the previous section. The pheromone concentration between the automatic driving vehicle \( i \) and the automatic driving vehicle \( j \) is \( \tau_{ij} \) that the ant \( k \) is on the path connected through a node. The pheromone concentration selects the next node. The probability selection rule \( P_i^k(t) \) of the ant from
node $v_i$ to node $v_j$ at time $t$ is:

$$p_{ij}^t(t) = \begin{cases} 
\left[ \frac{\tau_{ij}(t)}{\sum \eta_{ij}(t)} \right]^\alpha \cdot \left[ \frac{\eta_{ij}(t)}{\sum \tau_{ij}(t)} \right]^\beta 
& \text{if } \sum \tau_{ij}^t(t) > 0 \\
0 & \text{otherwise}
\end{cases}$$

(5)

In the Equation (5), each symbol represents its meaning: $\tau_{ij}$ the pheromone level of the edge $(i, j)$; $\eta_{ij}$ the heuristic desirability of the edge $(i, j)$; $\alpha$ the influence on the probability value $\tau$; $\beta$ the influence on the probability value.

Each time the ant colony updates the pheromone through the pheromone update rule, the pheromone concentration after the ant colony passes the edge $(i, j)$ is:

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

(6)

The ant colony algorithm consists of three parts: Start operation, Integrated manager, and ACS. The core part of the algorithm is the Integrated manager, which reads the initial data, initializes the data structure, builds the initial solution, and starts ACS. When a suitable solution is found, ACS will start. The flow chart of the algorithm is shown in Figure 1.

---

**Figure 1.** Ant Colony Searcher Diagram

The Start operation represents start, and the algorithm is described as finding the shortest distance tour for each client node without violating the specified constraints. The pseudo code is expressed as Algorithm 1.

### Algorithm 1: Start operation

1. Let $L$ denote a collection of $n$ customers. Sort them by increasing the value of the earliest arrival time $e_i$, If the nodes have the same $e_i$, arrange them by increasing the value of the latest arrival time $l_i$;
2. Let $T$ be the patrol list, where $n_v$ is the length of the list. Initially, there was only one node per tour in $T$, the vehicle at the station;
3. $i=0$, $i<n$, Find the shortest index distance of $l_i$;
4. Insert the node into your travel.

Algorithm 2 is the pseudo code of the Integrated manager, which finds the best solution by controlling ACS.
Algorithm 2: Integrated manager
1. Start ACS in the thread;
2. If the time in T is less than the time in T*, insert T;
3. Loop to find the best T;

The pseudo code for the algorithm ACS is shown as Algorithm 3.

Algorithm 3: ACS
1. Enter the maximum number of vehicles $n_v$, given the pheromone concentration $\tau_0$;
2. Ant colony k accesses all nodes K;
3. Find the largest number of access nodes l, the nodes are $T^l$;
4. Update the local pheromone on the edge of $T^l$ use Equation (6);
5. Loop until the best T is found and the controller issues a stop control.

3. Simulation Experiments
In order to verify the effectiveness of the ant colony algorithm, this section uses JADE as a vehicle driving simulator. VanetMobiSim is used to simulate vehicle motion and experiments on artificial road network. Different road networks are configured in VanetMobiSim using scene extension marker statements. Vehicle behavior was simulated in JADE. In this section, the ant colony algorithm and the constructed heuristic algorithm are compared in the two scenarios of free flow and jammed flow, respectively. The rationality and efficiency of the ant colony algorithm are verified.

Figure 2. Artificial road map of 16 nodes

In this scenario, a certain area in the artificial road network is selected as the experimental object, and the site is selected as the commercial area of the city, which is shown in Figure 2.

(1) Free flow
In this section, traffic data is collected from the urban streets based on the free flow, and the ant colony algorithm and the constructed heuristic algorithm are compared. The experimental results are shown in Figure 3.

It can be obtained in the simulation experiment in free flow, and the vehicles are smoothly dispatched when the system burden is small. As shown in Figure 3, unscheduled customers in the flow are between 0 and 3. This fluctuation range is small, so it can be seen that under the free-flow road conditions, the ant colony algorithm and the constructed heuristic algorithm are all displayed. Besides, the ant colony algorithm is more advanced than the constructive heuristic algorithm.

(2) Jammed flow

Based on the congestion of urban road network, the following experiments were carried out on the ant colony algorithm and the constructed heuristic algorithm. The data were collected and the experimental results are shown in Figure 4.

In the road environment where the flow is jammed, a large number of customers have not been arranged for service. As can be seen from Figure 4, the unscheduled customers of the ant colony algorithm between 12 and 14 are 9, 10, 8, while the construction heuristics are 12, 13, 9. Then the two algorithms have some trend of data decline, and at 17, 18, there are no customers who have increased the trend. It can be seen that both algorithms have great optimization, but in general, the ant colony algorithm exhibits a more optimized feature than the constructed heuristic algorithm.

4. Conclusions

In this paper, an online location model based on target scheduling is established and a hybrid search scheduler based on the ant colony algorithm for time window is designed and simulated.

This paper also needs a lot of future research. With the development of vehicle scheduling problems, it is necessary to consider new ways to model vehicle scheduling, and also to consider the acquisition of future orders and predict the impact of future orders on the model.

5. References

[1] Hautiere N, N J T, N J L, et al. Automatic fog detection and estimation of visibility distance through use of an onboard camera[J]. machine vision applications, 2006, 17(1): 8-20.
[2] Zhang Y, Zhou Y. 4VP: A novel meta OS approach for streaming programs in ubiquitous computing[C]//21st International Conference on Advanced Information Networking and Applications (AINA'07). IEEE, 2017: 394-403.

[3] Jia X. Google Cloud Computing Platform Technology Architecture and the Impact of Its Cost[C]. wri world congress on software engineering, 2010: 17-20.

[4] Buyya R, Yeo C S, Venugopal S, et al. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities[J]. high performance computing and communications, 2008: 5-13.

[5] Chen Z L, Xu H. Dynamic column generation for dynamic vehicle routing with time windows[J]. Transportation Science, 2016, 40(1): 74-88.

[6] Larsen A, Madsen O, Solomon M. Partially dynamic vehicle routing—models and algorithms[J]. Journal of the operational research society, 2012, 53(6): 637-646.

[7] Powell W B. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers[J]. Transportation Science, 2016, 30(3): 195-219.

[8] Powell W B, Carvalho T A. Dynamic control of logistics queueing networks for large-scale fleet management[J]. Transportation Science, 2018, 32(2): 90-109.

[9] Yan LP, Hu WB, Wang H, et al. Dynamic Real-Time Algorithm for Multi-Intersection Route selection in Urban Traffic Networks [J]. Journal of Software, 2016 (9): 2199-2217.