Implementation of Refuelling Order Service Algorithm and New Order Allocation Model of Ride-sharing Platform

Congying Hu, Pingshan Liu*, Jiayue Su, Ruihong Qiu and Junli Fang
Business School, Guilin University of Electronic Technology, Guilin, China

*Corresponding author email: liup3@guet.edu.cn

Abstract. Aiming at the problem that online ride-hailing drivers often empty their cars to replenish fuel at gas stations when their cars run out of gas, a new order assignment algorithm is proposed. According to the vehicle fuel volume reported by the driver to the platform, the vehicle with low fuel volume will be automatically assigned to the destination close to the gas station orders, and the driver will be reminded to go to the nearby gas station to reduce the empty vehicle round-trip rate. First of all, orders are classified into ordinary orders and refuelling orders, so that when the vehicle is short of fuel, the driver can be assigned to the refuelling orders with destinations close to gas stations. Then, a new order allocation mathematical model is established. This paper solves the order assignment problem based on the heap-optimized Dijkstra’s algorithm and A* algorithm. Experimental results show that the algorithm can intelligently allocate orders for drivers with insufficient fuel to their destinations closer to the gas station, shorten the distance the driver drives empty to the gas station to replenish fuel, and reduce the empty vehicle round-trip rate. When using Dijkstra's algorithm and A* algorithm to solve the order allocation problem, there is no significant difference in their processing time.

Keywords: Path planning; Dijkstra’s algorithm; A* algorithm; Order classification; Ride-sharing.

1. Introduction
At the third meeting of the 12th National People's Congress (NPC) of the People's Republic of China, Premier Li Keqiang first proposed the idea of "Internet +". In recent years, as online ride-hailing platforms such as Didi and Uber have become more and more popular, the "Internet + Ride" industry has greatly changed our lives. The traditional ride is an economic activity in which people stand by the road and wait, beckon, and negotiate the price with the driver. In the booming "Internet + Ride" industry in recent years, users only need to provide information such as their current geographic location and personal mobile phone number on their mobile phones to reserve available vehicles via the Internet, greatly facilitating travel.[1] The development of this industry has not only improved people’s travel efficiency during peak hours of commuting and accelerated the pace of people’s lives, but also provided drivers with certain welfare feedback, which is attractive to both drivers and passengers.[2] However, there is a problem with the current online car-hailing platform. The platform cannot intelligently assign orders that are closer to the gas station for the drivers with low fuel volume, so as to shorten the driving distance for the drivers to replenish fuel volume by driving empty vehicles to the gas station. Today's online car-hailing platform drivers mostly rely on empiricism. They only go to the gas station to refuel when they find that the vehicle's fuel level is insufficient. This greatly increases the driver's unprofitable driving distance. Even if the driver does not notice that the fuel level
is too low, there may be situations where an order is received but there is not enough fuel to complete the order journey.

In response to the above problems, this article proposes a new online car-hailing driver service based on shared route percentage (SRP) [3] inspired by Ride-sharing[4]. This service can assign orders for vehicles whose fuel capacity is lower than the specified value to a destination close to a gas station. Hereinafter, this type of order is referred to as refueling orders. This service can reduce the empty vehicle round-trip rate of online car-hailing drivers and prevent drivers from going to gas stations to refuel on the basis of empiricism. Instead, the platform assigns refueling orders when their fuel volume level is low, so that drivers can travel to the gas stations within a short distance after completing the order. As the development of online car-hailing platforms is becoming more mature, this paper proposes a new order assignment mathematical model compatible with the current online car-hailing order assignment form, and uses Dijkstra’s algorithm[5] and Astar[6][7] algorithm to solve the newly proposed service order assignment problem.

The main contributions of this research are as follows:
1) A new type of online ride-hailing driver service based on SRP is proposed, namely, the refueling order service.
2) The mathematical model of order assignment of new online car-hailing vehicles including refueling order service is established.
3) This paper solves the problem of order classification and assignment in refueling order service based on the heap optimized Dijkstra’s algorithm and Astar algorithm respectively, and compares the planned path length, SRP and processing time (PT) according of the two algorithms.

2. Problem Description and Mathematical Model

2.1. Refuelling Order Service

Refueling order service is a kind of service for drivers proposed in this paper. The service automatically assigns orders for under-fueled vehicles to destinations close to gas stations. Such passenger destinations close to the gas station and meet certain conditions are classified as gas orders, and other orders are classified as ordinary orders.

2.2. Description of the Problem

This paper constructs a problem model of new online car-hailing order allocation. The problem can be described as: 1) The road network of a city is transformed into a directed graph \(G = (V, E)\), where \(V = \{v_1, v_2, ..., v_n\}\) is a node set composed of common point set \(I\) and gas station set \(J\), \(V = I \cup J\). \(E = \{(v_i, v_j) : v_i, v_j \in V, v_i \neq v_j\}\) represents the set of arcs, where the weight corresponding to the arc \((v_i, v_j)\) is \(w_{ij}\). 2) This article defines \(O\) as a collection of orders, \(O = \{o_1, o_2, ..., o_i, ..., o_n\}\). Orders are divided into ordinary orders and refueling orders. Refueling orders must have a route starting from the passenger's destination and ending at a nearby gas station, and the length of the route must be less than the distance threshold \(c_1\). 3) When passengers call for an order, use the Astar algorithm and Dijkstra’s algorithm respectively with the order destination as the starting point to search for whether there is a route from the order destination to the gas station within the distance threshold \(c_1\). If it exists, it means that the order is close to the gas station. Then, further judge whether the SRP of this order is less than the threshold value \(c_3\). If less, the order will be classified as a refueling order, otherwise, it will be classified as a common order. 4) If an order is successfully carpooled, three conditions must be met: a) the order is a normal order, b) the order allows carpoolings, c) the SRP of the order carpoolings is higher than \(c_2\). 5) There is a set of online car-hailing \(C = \{c_1, c_2, ..., c_i, ..., c_n\}\). Record the fuel volume of each vehicle, and assign fuel orders to vehicles with lower fuel levels. A vehicle must meet three conditions to be assigned to a fuel order: a) The fuel volume of the vehicle is below the fuel level threshold \(oil_0\), b) The distance from the initial position of the vehicle to the passenger boarding point is less than the distance threshold \(c_0\), c) The gas station near the destination is assumed to be the second passenger’s destination, and its SRP is required to be higher than the threshold \(c_3\).
$w_{ij}$ represents the weight of arc $e_{ij}$; $\gamma$ represents the fuel consumption per unit length of the car; $x_{ijk}$ is a 0-1 variable, which indicates whether the arc $e_{ij}$ is planned to enter the path from the passenger boarding point to the passenger destination in the ordinary order $k$, if there is one, it is 1; $x'_{ijk}$ is a 0-1 variable, which indicates whether the arc $e_{ij}$ is planned in the path from the initial position of the driver's vehicle to the passenger boarding point in the ordinary order $k$, if there is one, it is 1; $v_{ijk}$ is a 0-1 variable, which indicates whether the arc $e_{ij}$ is planned to enter the path from the passenger boarding point to the passenger destination in the refuelling order $k$, if there is one, it is 1; $v'_{ijk}$ is a 0-1 variable, which indicates whether the arc $e_{ij}$ is planned in the path from the initial position of the driver's vehicle to the passenger boarding point in the refuelling order $k$, if there is one, it is 1; $g_{ijk}$ is a 0-1 variable, which indicates whether the arc $e_{ij}$ is planned into the second passenger's path from the passenger's boarding point to the passenger's destination when the ordinary order $k$ is assembled. If there is one, it is 1; $g'_{ijk}$ is a 0-1 variable, which indicates whether the arc $e_{ij}$ is planned into the path of the second passenger from the initial position of the vehicle to the passenger boarding point when the ordinary order $k$ is assembled. If there is one, it is 1; $u_{ij}$ is a 0-1 variable, indicating whether the order $i$ is assigned to car $j$, if so, the value is 1; $z_i$ is a 0-1 variable, indicating whether the order $i$ belongs to a ordinary order; $y_i$ is a 0-1 variable, indicating whether the order $i$ belongs to a refueling order; $p_i$ represents whether the $i$ order is allowed to carpool; $q_i$ represents whether the $i$ order is successfully carpooling; $oil_{ik}$ represents the amount of fuel left when the driver receives the $k$ order. The schematic diagram of carpool order and refueling order is shown in figure 1.

![Figure 1. Carpooling order (left) with refuelling order (right) diagram.](image)

2.3. Description of the Problem

Based on the above conditions, the mathematical model is as follows:

$$MinZ = \gamma \sum_{i,j,k} \sum_{j,k} \sum_{k,0} w_{ij} (x_{ijk} + x'_{ijk} + g_{ijk} + g'_{ijk} + v_{ijk} + v'_{ijk} + v''_{ijk} - (x_{ijk} + x'_{ijk})(g_{ijk} + g'_{ijk}))$$

(1)
S.T.
\[ \sum_{j \in E} w_{ij}(x_{ij} + v'_{ij}) \geq c_2 q_i \quad \forall k \in 0 \]  
\[ \sum_{j \in E} v_{ij}x_{ij} + g'_{ij}x_{ij} \geq c_y y_k \quad \forall k \in 0 \]  
\[ y_k v_{ij} \sum_{j \in E} w_{ij}(v_{ij} + v'_{ij}) \leq a_0 l_k \quad \forall k \in 0 \]  
\[ \left| \sum_{j \in E} w_{ij}(v_{ij} + v'_{ij}) - \sum_{j \in E} w_{ij}(v_{ij} + v'_{ij}) \right| = \left[ \sum_{j \in E} w_{ij}(v_{ij} + v'_{ij}) \right] \forall k \in 0 \]  
\[ y_k \sum_{j \in E} (x_{ij} + x'_{ij}) - \sum_{j \in E} x_{ij} = \left[ \sum_{j \in E} x_{ij} + x'_{ij} \right] \forall k \in 0 \]  
\[ x_{ij} \sum_{j \in E} v_{ij}(x_{ij} + x'_{ij}) + g_{ij} + g'_{ij} - \sum_{j \in E} x_{ij} = \left( g_{ij} + g'_{ij} \right) \leq a_0 l_k \quad \forall k \in 0 \]  
\[ u_{k_1} z_h y_{k_2} x_{ij} v_{ij} x_{ij} v'_{ij} v'_{ij} g_{ij} g'_{ij}, p_k, q_k \in \{0, 1\} \quad \forall i, j \in E, k \in 0, l \in C \]  

Among them, the objective function (1) represents minimizing the total fuel consumption of online car-hailing, and the right formula represents the fuel consumption per unit length multiplied by the sum of the path lengths planned by all orders. It is worth mentioning that if for a section of path \( e_{ij} \), when \( x_{ij} = 1 \) and \( g_{ij} = 1 \), it means that the path is planned as the path of the first passenger and the second passenger at the same time, that is, the common road, the common road is actually only being calculated once; Similarly, if for a section of path \( e_{ij} \), when \( x_{ij} = 1 \) and \( g_{ij} = 0 \), it means that the path is only planned as the path of the first passenger, that is, a different path.

**Constraints:** Equations (2)-(3) indicate that an order can be divided into a normal order or a refueling order, and an order can only be allocated to one driver. Equations (4)-(5) indicate the prerequisites for allowing carpooling. The order is a normal order and the precondition for carpooling to be successful is to allow carpooling. Equation (6) indicates that the distance between the initial position of the driver's vehicle and the passenger boarding point cannot exceed the distance threshold \( c_0 \). Equation (7) indicates that in the refueling order, the distance between the passenger's destination and the gas station cannot exceed the distance threshold \( c_1 \). Equation (8) indicates that if the order is a successful carpooling order, the SRP of the order is greater than the threshold \( c_2 \). If the order is not a successful carpooling order, the formula is slack. Equation (9) means that if the order is a refueling order, the SRP of the order is greater than the threshold \( c_3 \). If the order is not a refueling order, the equation is slack. This formula indicates that the refueling order also needs to meet the constraints of SRP. The higher the value, the better the order at this time, thus reducing the empty round-trip rate. Equation (10) indicates that the route will be planned for the second passenger only when the carpool is successful. When the carpool order is unsuccessful, which means that \( q_i = 0 \), resulting in the right formula being 0 and the left formula must be 0 which means that the route isn’t planned for the second passenger. When the carpool order is successful, this formula becomes an identity and the formula is slack. Equations (11) and (15) indicate that the car needs to have enough fuel to complete the trip when receiving an order. Equation (12) indicates that only when the vehicle’s fuel level is lower than the fuel level threshold \( O \), the vehicle can receive a refueling order. Equation (13) means that when the
order is a normal order, the value of $v_{ijk}$, $v'_ijk$, and $v''_{ijk}$ must be 0. Equation (14) indicates that when the order is a refueling order, the value of $x_{ijk}$ and $x'_{ijk}$ must be 0. Equation (16) represents variable attributes. In this model, $c_0$, $c_1$, $c_2$, $c_3$, and $oil_0$ are parameters that need to be set manually.

3. Heap-optimized Dijkstra’s Algorithm and Astar Algorithm under Constraints

3.1. Introduction of Heap-optimized Dijkstra’s Algorithm and Astar Algorithm

The main idea of the Dijkstra’s algorithm implemented using arrays is to use arrays to calculate and store the shortest path from the starting point to other nodes until the shortest path from the starting point to the end is obtained. Each time a node is expanded, the dis array needs to be traversed. The heap-optimized Dijkstra’s algorithm uses a small root heap and uses a priority queue to maintain the points that have not been visited and are closest to the starting point, thereby greatly reducing the time complexity of the Dijkstra’s algorithm using arrays.

Astar algorithm is based on Dijkstra’s algorithm, but the difference between heap optimized Astar algorithm and Dijkstra’s algorithm is that it uses the heuristic function $f(n)$ to estimate the cost of each node in the graph to the end, so as to reduce the blindness when expanding nodes. The $f(n)$ function is shown in equation (17).

$$f(n) = g(n) + h(n)$$ (17)

$$h(n) = \sqrt{(x_n - x_{end})^2 + (y_n - y_{end})^2}$$ (18)

The g(n) function represents the actual cost from the start point to the node $n$, and the evaluation function $h(n)$ calculates the estimated cost from the node $n$ to the end point. It can be proved that Astar algorithm can find the optimal path if $h(n)$ is always less than the node $n$ to the end point. If $h(n)=0$, the Astar algorithm becomes the Dijkstra’s algorithm.[8] The evaluation function used in this paper is shown in formula (18), namely Euclidean distance, which can ensure that the Astar algorithm finds the optimal path.

3.2. Improved Dijkstra’s Algorithm and Astar Algorithm under the Order Classification Problem

This article will use the improved Dijkstra’s algorithm and Astar algorithm to solve the problem of order classification and allocation. Refueling orders need to meet the 6 constraint conditions of equations (6) (7) (9) (11) (12) (14) in the above mathematical model. As the current online car-hailing platform cannot obtain real-time fuel level information of vehicles, it has not been able to collect data on the fuel level of online car-hailing drivers. The vehicles in the experiment in this article are assumed to have fuel levels lower than $oil_0$ and the remaining fuel is enough to complete the order, and assuming that equations (11) and (12) are satisfied, in fact, when the algorithm is running, the service should determine whether the allocation of a certain order to a certain driver meets these two constraints. In the experiment of this article, two algorithms will be used to determine whether to allocate a certain order as a refueling order to a certain vehicle meets the other four constraint conditions, in order to achieve the purpose of reducing the vehicle idling rate.

The detailed process of the two algorithms is shown in figure 2. Node (start) is the passenger destination of the order, and at the same time as the starting point of the route planning in the gas order classification, Node (end) is the gas station near the passenger destination, q is the priority queue storing node. The SRP calculation method of the order is shown on the left of formula (9). The algorithm should output five kinds of results, but because the graph used in the experiment is a connected graph, only the results ①③④⑤ are output, and only the result ⑤ is that the order is successfully classified as a fuel order and assigned to the driver.
4. Experiments and Analysis

This article uses simulated data to conduct experiments, and the evaluation indicators adopted are the algorithm path length and processing time. In the simulation data used in this article, the road network data is obtained through the AutoNavi Map API (https://lbs.amap.com/demo/list/js-api). The data covers a road network of 125.72km² in Chengdu. The data of the pick-up point and destination of passenger orders are from the didi chuxing gaia initiative (https://gaia.didichuxing.com/). This article uses the 1441st to 1574th orders on November 1, 2016 in Chengdu as the experimental data. Pick-up point and destination of passenger orders form the nodes and the sections of the road between these nodes form the edges of graph. Part of the road network data near the railway gas station is shown in figure 3, where the location of the gas station is marked with a circle. This article will take the railway gas station in Hehuachi Street, Jinniu District, Chengdu, Sichuan Province as the experimental object. The latitude and longitude coordinates of GCJ-02 are: 104.066549, 30.691055; In this paper, the parameters $c_0$, $c_1$ in the model are set as 1000 / 2000 / 4000m, 1000 / 2000 / 4000m respectively, and $c_3$ is 0.8. As mentioned above, assuming that the experiment satisfies equations (11) and (12), the parameter $a_0$ will not be set.
In terms of path length, the length of the two algorithms is always equal and optimal. This is because Dijkstra’s algorithm can always find the optimal path, while A* algorithm can always find the optimal path when the evaluation function is always less than the actual cost. In terms of processing time, there is no significant difference between A* algorithm and Dijkstra’s algorithm. When the parameters \( c_0 \) and \( c_1 \) are both 1000m, A* algorithm runs faster than Dijkstra’s algorithm in 53.84% of cases, while when \( c_0 \) and \( c_1 \) are both 2000m, only A* algorithm runs faster than Dijkstra’s algorithm in 46.15% of cases. The reason for this phenomenon is that there are not too many nodes in the graph with 1000 / 2000m constraint range, which leads to the advantage of A* algorithm not to be brought into full play. Moreover, compared with Dijkstra’s algorithm, A* algorithm also needs to calculate the value of \( h(n) \) and \( f(n) \), which leads to the increase of its time.

The mean and standard deviation of the processing time of the two algorithms are shown in table 1. According to the experimental data, the p-value of the two algorithms is greater than 0.05, no matter whether \( c_0, c_1 \) are 1000m, 2000m or 4000m, so there is no significant difference in the processing time of the two algorithms in dealing with this problem. This is because there are too few nodes within 1000-4000m, and the advantage of A* algorithm in expanding nodes is not fully exerted. In this problem, \( c_0 \) and \( c_1 \) should not be too large, otherwise the passenger destination of the order will be too far away from the gas station, which does not facilitate the driver to go to the gas station to refuel. Figure 4 shows four successful cases of being classified as a refueling order when \( c_0 \) and \( c_1 \) are both 4000m and \( c_3 \) is 0.8. The four successful cases present in sub-figures (a), (b), (c) and (d) respectively. The blue marker represents the pick-up point, the first red marker represents the passenger destination, and the second red marker represents the gas station. It can be seen that the drivers only need to take a short distance to arrive at the gas station after completing the order so that drivers not only solve the problem of refueling, but also create economic benefits. The basic information of the four orders is shown in table 2.
Table 1. Indicators of the two algorithms at 1000/2000/4000m.

| Result | PT(s) of Dijkstra’s algorithm | PT(s) of Astar algorithm | P-value | Conclusion |
|---------|-------------------------------|--------------------------|---------|------------|
|         | Mean                          | STDev                    | Mean    | STDev      |            |           |
| C0, and |                               |                          |         |            | No significant difference |
| c1      |                               |                          |         |            |           |
| 1000m   | 4.2963E-06                    | 6.67818E-06              | 2.77037E-06 | 2.62588E-06 | 0.274270651 |
| 2000m   | 6.01538E-06                   | 5.27224E-06              | 4.67308E-06 | 3.80258E-06 | 0.297441623 |
| 4000m   | 1.19407E-05                   | 5.94327E-06              | 1.13037E-05 | 5.76044E-06 | 0.690845643 |

Table 2. Refuelling Order Details.

| Order ID | Longitude and latitude of boarding position | Latitude and longitude of the drop off location | Latitude and Longitude of Gas Station | The path length of the order | The length of the path from the destination to the gas station | SRP |
|----------|---------------------------------------------|-----------------------------------------------|--------------------------------------|-----------------------------|-------------------------------------------------------------|-----|
| 33345f751dc2a01a bea10df0db44b7141 | 103.94116,30.7043 | 103.9752,30.67325 | 104.066549,30.69105 | 18240 | 1510 | 0.923544 |
| 9098e69bd0d2d5e5c d9a5626e2be810 | 104.04703,30.62587 | 104.05866,30.67325 | 104.066549,30.69105 | 8670 | 1630 | 0.841748 |
| lee8ecb1ba8b68798 983736a1e1cb97 | 104.14061,30.5991 | 104.05793,30.67648 | 104.066549,30.69105 | 16440 | 3400 | 0.828629 |
| 751aa291e452606d93 cec6df08653d37 | 104.056435,30.55103 | 104.056435,30.55103 | 104.066549,30.69105 | 15960 | 2580 | 0.860841 |

Figure 4. Visualization of the planned path of the refuelling order.
The processing time curve of the two algorithms in this problem is shown in figure 5. We can see that in this problem, when $c_0$ and $c_1$ are both 1000 / 2000 / 4000m, Astar algorithm is slightly better than Dijkstra’s algorithm in processing time, but there is no obvious difference between the two algorithms. In terms of algorithm stability, Astar algorithm is obviously better than Dijkstra’s algorithm when $c_0$ and $c_1$ are both less than 4000m.

![Figure 5. Processing time graph.](image)

5. Conclusion
This paper proposes a refuelling order service based on the shared distance ratio, and establishes a new online car-hailing order assignment mathematical model that includes refuelling order services and is compatible with existing services. Then use the heap optimized Dijkstra’s algorithm and Astar algorithm to solve the new online car-hailing order allocation problem, divide the orders into ordinary orders or refuelling orders and assign them to drivers, and calculate the various indicators of the two algorithms. It is concluded that there is no significant difference in the processing time of the two algorithms in realizing the service, and the length of the path plan is equal and optimal. The shortcomings of this article: Because the author used manual measurement when building the topology map, I could not obtain a large amount of data to carry out the experiment. In order to simplify the problem, the gas station used in the experiment was Jinniu District Hehuachi Street Railway Gas Station, Chengdu, Sichuan Province, but in the actual situation, we should first traverse all the gas stations in the city to find the nearest gas station close to the order destination. However, the author thinks that the principle implemented here is similar, so I omitted this step, and proceeded straight to the subject to classify and allocate orders.

Acknowledgments
This research was supported by the National Natural Science Foundation (61762029, U1811264), Guilin Science and Technology Development Program(20190211-20)

References
[1] R. S. Thangaraj, K. Mukherjee, G. Raravi, A. Metrewar, N. Annamaneni and K. Chattopadhyay, "Xhare-a-Ride: A Search Optimized Dynamic Ride Sharing System with Approximation Guarantee," 2017 IEEE 33rd International Conference on Data Engineering (ICDE), 2017, pp. 1117-1128, doi: 10.1109/ICDE.2017.156.
[2] Giovanni Maria Carломagno and Andrea Ianiro. Thermo-fluid-dynamics of submerged jets impinging at short nozzle-to-plate distance: A review[J]. Experimental Thermal and Fluid Science, 2014, 58: 15-35.
[3] J. Angelo Escalona, B. Manalo, W. J. R. Limjoco and C. C. Dizon Electrical, "A Ride Sharing System based on An Expansive Search-Based Algorithm," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 870-874, doi: 10.1109/TENCON50793.2020.9293943.

[4] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma and Z. Gong, "An Efficient Ride-Sharing Framework for Maximizing Shared Routes," 2018 IEEE 34th International Conference on Data Engineering (ICDE), 2018, pp. 1795-1796, doi: 10.1109/ICDE.2018.00255.

[5] Maximilian Schreieck et al. A Matching Algorithm for Dynamic Ridesharing[J]. Transportation Research Procedia, 2016, 19: 272-285.

[6] František Duchoň et al. Path Planning with Modified a Star Algorithm for a Mobile Robot[J]. Procedia Engineering, 2014, 96: 59-69.

[7] Z. Xu, X. Liu and Q. Chen, "Application of Improved Astar Algorithm in Global Path Planning of Unmanned Vehicles," 2019 Chinese Automation Congress (CAC), 2019, pp. 2075-2080, doi: 10.1109/CAC48633.2019.8996720.

[8] Z. Xu, X. Liu and Q. Chen, "Application of Improved Astar Algorithm in Global Path Planning of Unmanned Vehicles," 2019 Chinese Automation Congress (CAC), 2019, pp. 2075-2080, doi: 10.1109/CAC48633.2019.8996720.