MCM Training Contest, Problem B

Ride-Sharing: A Combinatorial Magic

In recent years, some platforms have provided ride-sharing service, which meets the requirement of low-carbon travel, and is cost-saving for drivers and customers. For a ride-sharing scenario, generally, up to three passengers take a car at the same time. Ride-sharing involves multiple problems, such as order dispatching, route planning, etc. The Multi-Waypoint Route Planning Problem (MWRP) is essential in the ride-sharing scenario; that is, given the location of a car and the pick-up and drop-off positions of at most three passengers at a specific moment, it is required to give the shortest route that connects the driver and the passengers in series. Solving this problem is the preliminary for real-time dispatching.

For the MWRP, we define a "matchup" of a vehicle and multiple passengers as a multi-waypoint match $\{c, U\}$, where c represents the vehicle and U represents the passenger set, which can be expressed as $U = \{u_i\}$, where u_i is the i-th passenger. The platform can obtain the location information p_c of the vehicle and the starting and ending points of each passenger $\{p_o^i, p_d^i\}$. As shown in Figure 1(a), The taxi c starts from the location p_c and picks up two passengers, u_1 and u_2, from the boarding location p_o^1 and p_o^2 (starting point). Then, let two passengers get off at p_d^1 and p_d^2 (endpoint), respectively, thus forming a driving route sequence $S_1 = \{ p_c, p_o^1, p_o^2, p_d^1, p_d^2\}$. However, according to the geographic information of the road map and the relative positions of passengers and cars, different pick-up and drop-off sequences can cause huge route length differences. For instance, if the driving route sequence is changed as Figure 1(b) and forms the route sequence $S_2 = \{p_c, p_o^2, p_o^1, p_d^2, p_d^1\}$, then its total route length is longer than that of S_1.

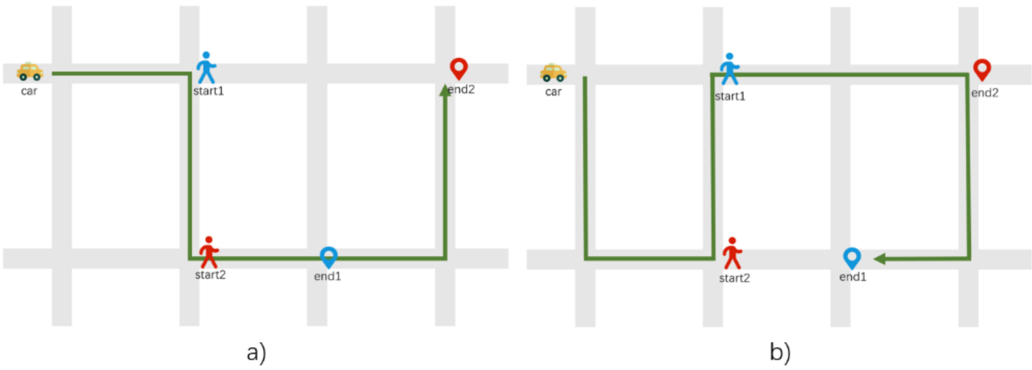

Figure 1 The order impact of pick-up and drop-off sequence with respect to route length.

Additionally, the problem becomes more complicated when passengers can “fine-tune” their pick-up and drop-off positions within walking distance. As shown in Figure 2, the passenger’s pick-up location may have 4 alternatives at a crossroad. Figure 2 shows the car's route when different candidate positions are selected as boarding points. It illuminates that different pick-up positions will bring different route lengths.

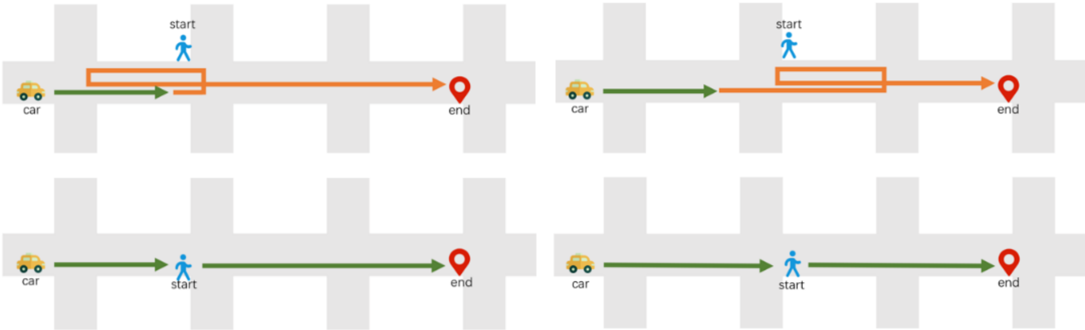

Figure 2 The impact of choosing different boarding candidate positions on driving route planning.

To clarify this phenomenon, for a passenger i, we define a set of candidate passenger pick-up positions points $p_o^i = \{p_o^{i(1)}, p_o^{i(2)}, \dots, p_o^{i(k)}\}$ and candidate passenger drop-off positions $p_d^i = \{p_d^{i(1)}, p_d^{i(2)}, \dots, p_d^{i(k)}\}$. As shown in Figure 3, For sequence $S_1 = \{p_c, p_o^1, p_o^2, p_d^1, p_d^2\}$, if $p_o^{1(2)}$ is selected to get on the train, it will cause the driver to drive along (v_2, v_3) first, turn head after arriving at point v_3, then drive along (v_3, v_2) to pick u_1, turn around after arriving at point v_2, and then drive forward. However, choosing to get on at point $p_o^{1(1)}$ can save the total path length and effectively optimize the result of path planning.

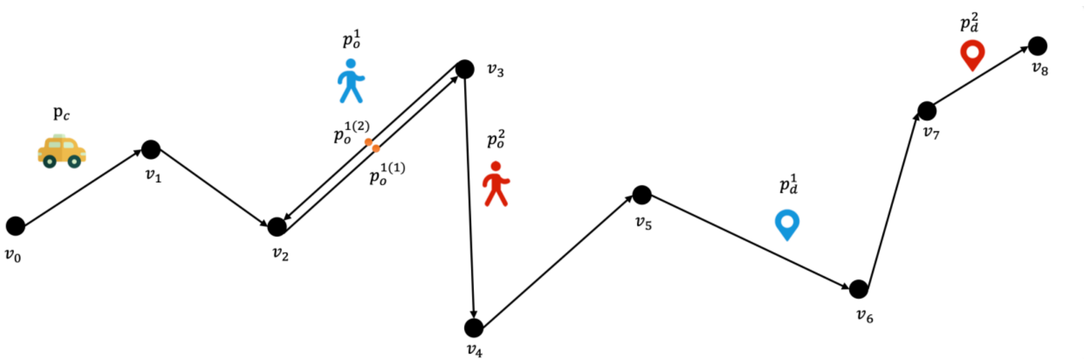

Figure 3 The impact of "fine-tuning" pick-up candidate positions on route length

However, multiple candidate positions will increase the calculation of the shortest path sequence S^*. As shown in Figure 4, to compute the route length $d(S_1)$ of sequence $S_1 = \{p_c, p_o^1, p_o^2, p_d^1, p_d^2\}$, we need to compute the pair-wise shortest path four times ($p_c \to p_o^1$, $p_o^1 \to p_o^2$, $p_o^2 \to p_d^1$, $p_d^1 \to p_d^2$) and sum up the results together. While if passenger 1 has three candidate pick-up positions $p_o^1 = \{p_o^{1(1)}, p_o^{1(2)}, p_o^{1(3)}\}$, because the distance from p_c to $p_o^{1(1)}, p_o^{1(2)}, p_o^{1(3)}$ and the distance from them to p_o^2 need to be calculated separately, we need to compute the pair-wise shortest path eight times to obtain $d(S_1)$.

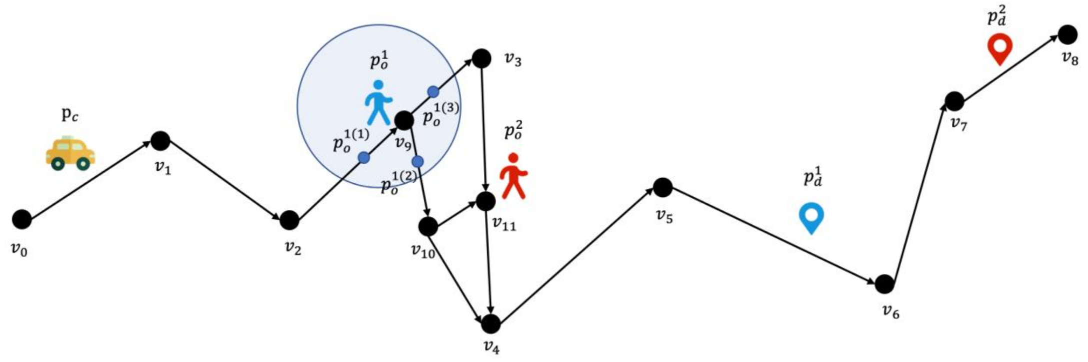

Figure 4 The impact of multiple candidate positions on the number of calculations of a pair-wise shortest path

Computing shortest paths in an actual large-scale road map are time-consuming. Magic Car-hailing Management (MCM) company hopes your team will solve the following problems to reduce the computation workload.

Problem 1: If a passenger has fixed pick-up and drop-off positions. All four possible driving route sequences of a carpool two-passenger case are $p_c p_o^1 p_o^2 p_d^1 p_d^2$, $p_c p_o^1 p_o^2 p_d^2 p_d^1$, $p_c p_o^2 p_o^1 p_d^1 p_d^2$, $p_c p_o^2 p_o^1 p_d^2 p_d^1$. According to this, please formally define what a legal carpool driving route sequence is. Then enumerate all legal driving route sequences of a carpool three-passenger case and give the total number. How many legal sequences?

Problem 2: Based on the possible legal sequence that you find in Problem 1, calculate the path length of all the sequences of Case 1. List them all and give the shortest path sequence S^*.

Case1 (You can also refer to "R3-case.csv"): $p_c = (104.0416,\ 30.66072)$ at link 2798

$p_o^1 = (104.0352,30.64502)$ at link 1022; $p_d^1 = (104.1657,30.66395)$ at link 745

$p_o^2 = (104.0295,30.62556)$ at link 1641; $p_d^2 = (104.1773,30.69224)$ at link 793

$p_o^3 = (104.0275,30.63736)$ at link 1855; $p_d^3 = (104.1648,30.66162)$ at link 745

In practice, it is wasteful to calculate all enumeration cases, especially when we have limited computation resources. Therefore, please analyze which cases need to be calculated to find a shortest path? Please design some pruning strategies to find the shortest path sequence S^* under minimizing the number of calls to the shortest path algorithm. Apply your strategy to Case 1, how many cases do you need to calculate? How likely can you find S^*?

Hint: You can prune unreasonable order by observing spherical distance, direction angle, or other indicators. Figure 5 is a toy example. Suppose $S_1 = [p_c, p_o^1, p_o^2, p_d^2, p_d^1]$, $angle(p_o^2, p_d^2, p_d^1)$ ($\angle\alpha$) is less than 45°, which indicates a detour. When there is a sequence with a better direction angle, we can prune S_1 away.

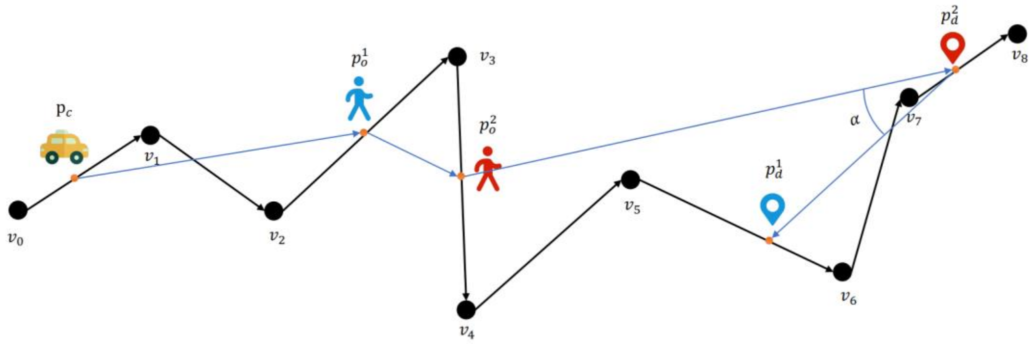

Figure 5 An example of pruning by direction angle.

Problem 3: Consider the influence of multiple candidate positions (as discussed in Figure 2 and Figure 4). The first step is determining how many positions we need to consider for each pick-up and drop-off position, and how to find them. If there are k $(k \geq 1)$ candidates at p_0^i, p_d^i $(i = 1,2,3)$, what is the most suitable k? Design a strategy to determine k and find these positions.

Problem 4: Combine the pruning strategy of Problem 2 with k candidate positions from Problem 3 to give the final strategy that can find out the shortest path sequence S^*. Solve the 10 cases in "R3-case.csv" by your strategy.

According to your results, please also write a letter (non-technically) to the customers of MCM, explaining why the system requires them to go to the nearby pick-up point and how far they need to go (similar to the drop-off service).

Your PDF solution of no more than 25 total pages should include:

- One-page Summary Sheet.
- Table of Contents.
- Your complete solution.
- One-page Article to MCM Corporation.
- Reference List.

Note: The contest has a 25-page limit. All aspects of your submission count toward the 25-page limit (Summary Sheet, Table of Contents, Reference List, and any Appendices). You must cite the sources for your ideas, images, and any other materials used in your report.

Appendix.

This problem includes three files:

1. File "R1-link.csv" provides a simplified road map of Chengdu, China, as a link list. Each row in the file represents a directed "road unit" (also referred to as a "link"), whose direction is always from the "Node_Start" to the "Node_End" with its actual route length. The "road unit" represents the minimum measurement of a road with no intersection with other roads in the middle. Note that this map includes both one-way and two-way roads. For example, link 1 (0, 48) and link 3052 (48, 0) indicate a two-way road, and link 173 (115, 61) indicates a one-way road.

2. File "R2-distance.csv" provides the length of the pairwise shortest path between every two nodes.

3. File "R3-case.csv" provides 10 ride-sharing data instances in Chengdu, where each row represents a ride-sharing order, including p_0^i, p_d^i $(i = 1,2,3)$, p_c. Each position is represented by its longitude, latitude, and neighbor link Id.

<u>Acknowledgement:</u> This problem was proposed by Engineer Jun Fang from DiDi Global Inc. (DIDI GAIA Research Collaboration Plan), modified and formalized by Prof. Xiaofeng Gao from the Department of Computer Science and Engineering, Shanghai Jiao Tong University. It was then integrated, reorganized, and finalized by Ph. D. Students Yucen Gao and Jiale Zhang. Master Students Yulong Song and Yang Luo collected the related data and provided valuable comments and suggestions. Teaching Assistant Le Huang from the MCM/ICM Training Camp drew up the first version of the draft document. They all come from Shanghai Jiao Tong University.