近年来，一些平台提供了共享汽车服务，这符合低碳出行的要求。
低碳出行的要求，并为司机和客户节省成本。对于一个共享出行的场景。
一般来说，最多三名乘客同时乘坐一辆车。共享汽车涉及多个问题。
如订单调度、路线规划等。多路点路线规划问题（MWRP）
在共享出行场景中至关重要；也就是说，给定一辆车的位置和最多三名乘客在某个特定时刻的上下车位置，需要给出最短的路线，将司机和乘客串联起来。解决这个问题是实时调度的第一步。

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.

我自己的翻译大概就是对于车使用 pc 来表示其位置，使用p_0^i和p_d^i来表示第 i 个行人的位置；
对于两个乘客有路线序列，而且不止一条，不同的序列对应的路线长度不同。

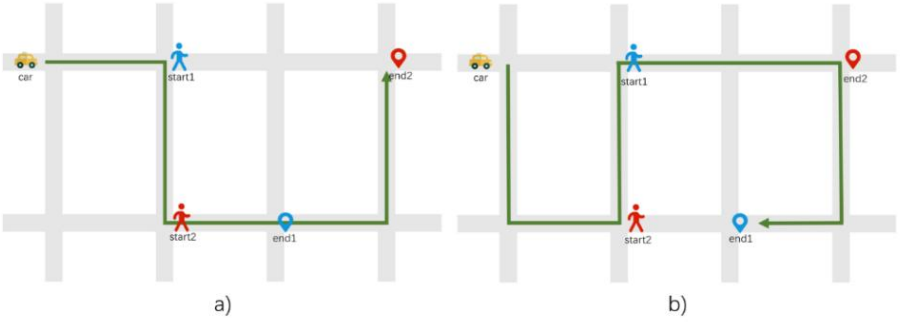
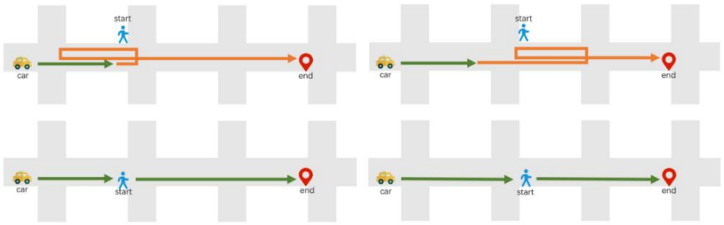


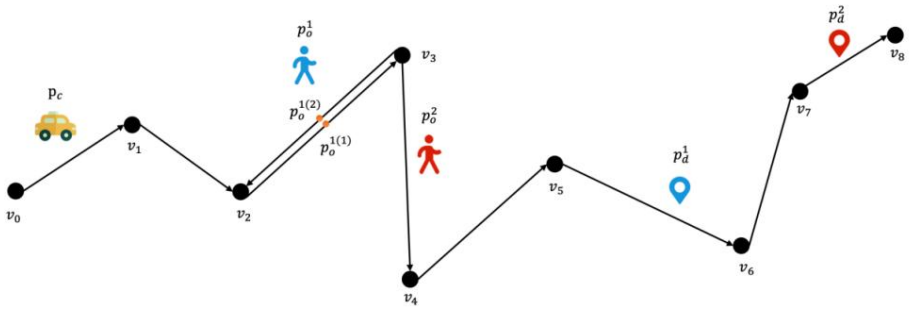
Figure 1 The order impact of pick-up and drop-off sequence with respect to route length.

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.

有些时候司机到达乘客附近时不意味着可以直接停车，因为司机有可能遇到双黄线（必须走到路口才能掉头接另一侧的乘客）。



To clarify this phenomenon, for a passenger i, we define a set of candidate passenger pick-up positions points $p_o^i = \{p_o^{i(1)}, p_o^{i(2)}, \ldots, p_o^{i(k)}\}$ and candidate passenger drop-off positions $p_d^i = \{p_d^{i(1)}, p_d^{i(2)}, \ldots, p_d^{i(k)}\}$. 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.



对这种情况的详细说明就是，如果是在（2）处上车，会比（1）处多走 2 个 v2-v3 路程

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 $d(S_1)$.

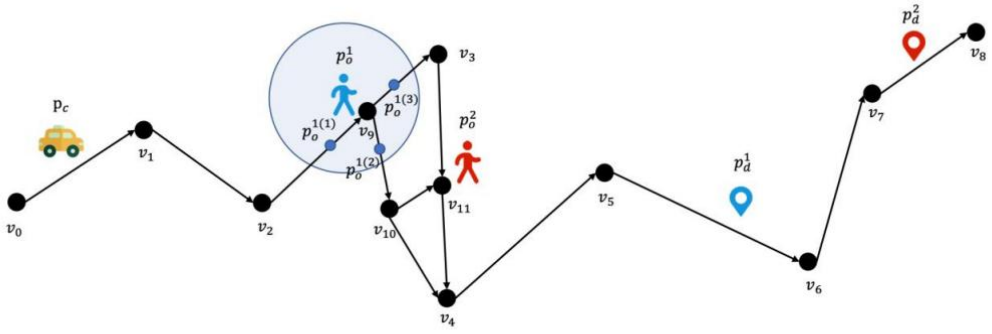


Figure 4 The impact of multiple candidate positions on the number of calculations of a pair-wise shortest path

乘客的预选上车点越多，需要考虑的情况就越多，比如此时三个 p1 点就对应着不同的从 pc 到 p1 的距离和从 p1 到 p2 的距离，这样就有 6 个不同的值需要求解。

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 $p_c p_o^1 p_0^2 p_d^1 p_d^2$, $p_c p_o^1 p_0^2 p_d^2 p_d^1$, $p_c p_o^2 p_0^1 p_d^1 p_d^2$, $p_c p_o^2 p_0^1 p_d^2 p_d^1$. 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?

公司希望你们解决如下问题：
问题一：如果乘客的上车点和下车点固定，对于两个乘客的可能的驾驶序列为（如上图），根据这个，请你正式定义什么是 legal carpool driving route sequence，然后列举出对于三乘客的 legal...sequence 以及计算其总数。

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.

问题二：基于第一问中的 legal sequence，计算在 case1 中的所有序列的路长，把它们都列举出来并且给出最短的那条。
实际上，在实践中，计算所有的列举案例是很浪费的，特别是当我们有有限的计算资源。因此，请分析哪些情况需要计算以找到最短的路径？请设计一些剪枝策略，在最小化的情况找到最短路径序列 s 算法的调用次数。将你的策略应用于案例 1，你需要计算多少个案例？你能找到 s 的可能性有多大？
提示：去除不合理的顺序根据观察球形的距离、距离角、或者其他的东西。上面是一个例子（没太理解在讲什么）。
问题 3：考虑到多个候选位置的影响（如图 2 和图 4）。第一步是确定我们需要为每个接送位置考虑多少个上车下车位置，以及如何找到它们。如果在 p0i,pdi 有 k（k≥1）个候选人 (k = 1,2,3)，什么是最合适的 k? 设计一个策略来确定 k 并找到这些位置。

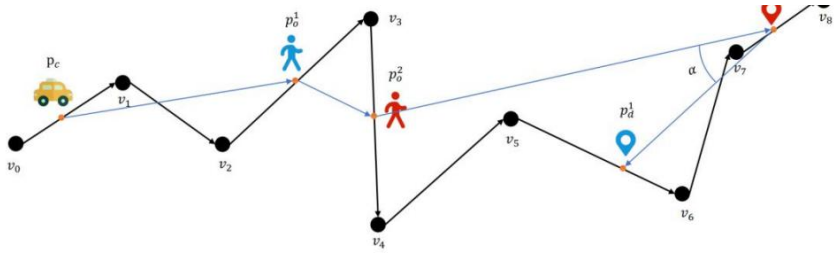


Figure 5 An example of pruning by direction angle.

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.

问题四：结合在问题二中的去除路径策略以及在问题三中的 k，来解决实际问题，此时请解决在文件 R3-case.csv 中的 10 个 case。

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).

根据你的结果，向 MCM 公司的顾客写一封信，解释为什么这个系统需要他们去走到附近的乘车点以及他们大概需要走多久（类似于下车问题）。

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.

都有的不翻译了。