1. Markov Decision Process

In order to model the taxi service in New York City, we adopt the framework of MDPs. This framework allows us to deal with the uncertain demand over the different periods in the grid, and to model them explicitly. The MDP is a stochastic decision process with a set of states (S) and a set of possible actions (A) that transition the states from one to another. Each action will correspond to the process of the current state to the new state with a probability transition function and a reward function. The collection of optimal actions for each state is called the policy, which maximizes the total reward over several numbers of steps. The objective of our model is to minimize the average seeking time for the taxi to maximize the expected revenues.

System States. The state for a taxi is described by its current locations and the current time. The details are explained as follows.

Location: $(x, y) \in L = \{1, \ldots, 50 \times 1, \ldots, 50\}$: the area is divided into grid 50×50 grid cells;

Time: $t \in T = \{1, \ldots, 60\}$: we use minutes as the interval of a time slot, and a total of one hour as time horizon.

Each pick-up and drop-off location is assigned to a grid cell. We remove the records that contain 1) incomplete data information, 2) trip distance over 100 kilometers, 3) trip duration over 60 minutes, 4) pick-up and drop-off locations with the same coordinates, 5) pick-up and drop-off locations outside the grid, and 6) shifts that are shorter than six hours and longer than nine hours.

We denote the system state of our MDP model as $s = (x, y, t)$, and the collection of all admissible states is denoted by S.

Actions. The admissible actions from a given state s have nine possibilities to choose from. We use numbers $1, \ldots, 9$ to index the directions. We express them formally as:

$$A = \begin{array}{|c|c|c|} \hline 7 & 8 & 9 \\ \hline 4 & 5 & 6 \\ \hline 1 & 2 & 3 \\ \hline \end{array}$$

where, e.g., action 9 moves the taxi to the neighboring North-East location.

Parameters of the MDP model. In this subsection, we state the parameters used in the rest of MDP model.

The probability parameters are defined as:

- $P_{\text{find}}(x, y, t)$ describes the probability of successfully picking up a passenger in grid cell (x, y, t). We can calculate the probability of picking up a passenger in the cell by dividing the number of successful pick-ups in the cell ($n_{\text{find}}(x, y, t)$) by the total number of times this cell is visited by a vacant taxi in particular time t. The vacant taxi includes the number of taxis that drop off passengers in grid cell (x, y, t) ($n_{\text{drop-off}}(x, y, t)$) and also the taxis that are seeking for passengers ($n_{\text{OSRM}}(x, y, t)$). To locate the vacant taxi every minute during the seeking trip, we use the API provided by Open Source Routing Machine [**?**], to estimate the coordinates. We use one hour time slots between 12:00 to 13:00 for the day shift model and 0:00 to 1:00 for the night shift model. In our overall model, we took the average of the

day time and night time models to estimate the number of vacant taxis at each grid during the month of January in 2013. Thus,

$$P_{\text{find}}(x,y) = \frac{n_{\text{find}}(x,y)}{n_{\text{find}}(x,y) + n_{\text{drop-off}}(x,y) + n_{\text{OSRM}}(x,y)}.$$

- $P_{\text{dest}}(x',y',x'',y'')$ describes the probability of a passenger travelling from grid cell (x,y) to the grid cell (x'',y''). To estimate the destination probability for a time slot, we calculate the number of trips between each pair of source and destination locations in that time slot and get a 50×50 matrix. The value is divided by the sum of the entire number of trips of the grid cells. Therefore, P_{dest} has the empirical probability distribution of a passenger choosing destination location (x'',y'') when he is picked up at location (x',y').

The time parameters are defined as:

- $T_{\text{seek}}(x,y)$: The duration required for a taxi to traverse the grid cell denoted by (x,y) is 1 to travel to a neighboring location and 2 to travel to the diagonal movement.
- $T_{\text{drive}}(x,y,x',y')$: The driving time from (x,y) to (x',y'). We can calculate the total driving time from grid cell (x,y) to grid cell (x',y') and then divide by the number of trips from grid cell (x,y) to grid cell (x',y'). A similar approach is applied to calculate $T_{\text{drive}}(x,y,x'',y'')$. We calculate T_{drive} individually for all models. Similarly for $T_{\text{drive}}(x,y,x'',y'')$: The driving time from (x,y) to (x'',y''). From the calculation, there is approximately +15.67% driving time difference between the day shift model and the night shift model, and there is a +4.14% difference between the weekend and the weekday.
- We assume that there is no waiting time for passengers to get in and out of the vehicle.

The reward is defined as:

- $r(x,y,x'',y'')$: The reward from grid cell (x,y) to grid cell (x'',y''). Similar to T_{drive}, we calculate the average fare of the number of trips between each pair of source and destinations as the expected fare. Note that due to this definition, we reward does not depend on the action of the taxi driver. We calculate r separately for all models. Similarly to T_{drive}, there is approximately +6.21% reward difference between the day shift model and the night shift model, and there is a +1.21% difference between the weekend and the weekday.

State transition function. The state transition function describes the probability that one moves from state (x,y,t) after taking decision a moves to state (x',y',t'). As a result, the taxi will cruise to the next destination (x',y') in $T_{\text{seek}}(x',y')$ Assuming the current state is $S = (x,y,t)$ and action a is taken, there are two possible outcomes of the transition:

(1) The taxi successfully finds a passenger in grid (x',y') after cruising $T_{\text{seek}}(x,y)$ minutes. The taxi with the passenger goes to destination (x'',y'') with probability $P_{\text{dest}}(x,y,x'',y'')$. The taxi arrives at location (x'',y'') with $T_{\text{drive}}(x',y',x'',y'')$ as the total time used to travel from (x',y') to (x'',y''). The taxi driver receives $r(x',y',x'',y'')$ as the expected reward.

Then the taxi will start seeking for a passenger from grid cell (x'', y''). In this case, the new state becomes $s' = (x'', y'', t+T_{\text{seek}}(x,y)+T_{\text{drive}}(x', y', x'', y''))$.

(2) The taxi does not find a passenger after $T_{\text{seek}}(x,y)$ being in grid cell (x,y) with the probability $(1 - P_{\text{find}}(x,y))$. The taxi driver does not receive a reward and adds the driving time T_{drive}. The taxi driver starts to make the next action at grid cell (x', y'). Hence, the state of the taxi becomes $s' = (x', y', t + T_{\text{seek}}(x,y) + T_{\text{drive}}(x,y,x',y'))$.

The objective function. The objective function of the MDP model is the total average rewards starting from an initial state. The terminal states are the states with $t = 60$. No more actions can be taken once the system reaches the terminal states. The maximal expected reward for an action a in state $s = (x,y,t)$ is expressed as $V(s,a)$ shown below.

(1.1)

$$\begin{aligned} V(s,a) =& (1 - P_{\text{find}}(x,y,t)) \times \max_{a' \in A} \left[V(x', y', t + T_{\text{seek}}(x,y) + T_{\text{drive}}(x,y,x',y'), a') \right] + \\ & \sum_{(x'',y'') \in L} P_{\text{find}}(x,y,t) \times P_{\text{dest}}(x,y,x'',y'') \times \\ & \left[r(x,y,x'',y'') + \max_{a' \in A} V(x'', y'', t + T_{\text{seek}}(x,y) + T_{\text{drive}}(x,y,x'',y''), a') \right]. \end{aligned}$$

The optimal policy π^* is defined as:

$$\pi^*(s) = \arg\max \{V(s,a)\}, \tag{1.2}$$

and the optimal value function is given by

$$V^*(s) = V(s, \pi^*(s)). \tag{1.3}$$

Markov Decision Process solution. In order to solve the Markov Decision Problem to derive the optimal policy, we employ dynamic programming to maximize the expected rewards. The algorithm starts from time $t = 60$ and then traces backward to time $t = 1$. The algorithm is listed in Algorithm 1.

Algorithm 1 Solving MDP using Dynamic Programming

InputInputOutputOutput $L, A, T, P_{\text{find}}, P_{\text{dest}}, r, T_{\text{drive}}, T_{\text{seek}}$ The best policy π^* V is a $\mid L \mid \times \mid T \mid$ matrix; $V \leftarrow 0$

for $t = \mid T \mid 1$ **do**

for all $(x,y) \in L$ **do** $\triangleright s = (x,y,t)$ $a_{max} \leftarrow a$ that maximizes $V(s,a)$

$\pi^*(s) \leftarrow a_{max}$ $V^*(s) \leftarrow V(s, a_{max})$ **return** π^*

TABLE 1. "New" Revenue Efficiency E_{rev}.

	Weekday dayshift	Weekday nightshift	Weekend dayshift	Weekend nightshift	Overall
Top 10%	0.6339133	0.666667	0.6242991	0.6441137	0.6209029
P_{find}	0.52267	0.50915	0.51463	0.45475	0.50030
Bottom 10%	0.4386435	0.4389722	0.3954553	0.4127084	0.4049057