The Ride Hailing Problem

Ali Yaddaden[1] and Sébastien Harispe[1]

EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France
{firstname.lastname}@mines-ales.fr

1 Problem definition

We consider a homogeneous fleet of vehicles $\mathcal{V} = \{0, 1, ..., V-1\}$ (with V being the number of vehicles) controlled by a central operator. The operator dispatches the vehicle to serve trip requests that arise randomly across the city of Manhattan. The goal is to maximize the total profit earned over a time horizon $T = 24$ hours calculated as the difference between the revenue earned from the service minus the travel costs.

Following the Markov Decision Process framework, we can distinguish the state, action, and reward.

1.1 State

The state is formulated as $s = (s_t, s_r, s_{\mathcal{V}})$ where :

- $s_t = (t, d)$ with $t \in [0, 86400]$ being the time of the day in seconds and $d \in \{0, 1, 2, 3, 4\}$ being the day of the week (0 for Monday).
- $s_r = (s_P, s_{rt})$ with $s_P = \{((o_x^i, o_y^i), (d_x^i, d_y^i)), \forall i \in P\}$ being the set of the pending requests' Cartesian coordinates of the Origin and Destination. $P = \{0, 1, ..., num\_pending\_requests - 1\}$ is the set of pending requests. Note that $num\_pending\_requests$ varies depending on the actions decided by the agent. $s_{rt} = \{rt_i, \forall i \in P\}$ represents the request time at which the requests were released.
- $s_{\mathcal{V}}$ represents the vehicles' state. For each vehicle $v \in \mathcal{V}$, it is composed of :
  - (x_v, y_v) the Cartesian coordinates of the vehicle location.
  - $\{jt_v^i \in \{0 : \text{idling}, 1 : \text{repositioning}, 2 : \text{setup}, 3 : \text{processing}, 4 : \text{null}\}, i \in \{1, 2, 3\}\}$ represents the job types. There are 3 jobs per vehicle.
  - (jo_v^i, jd_v^i) represents the job Origin and Destination Cartesian coordinates (i.e. $jo_v^i = (xo_v^i, yo_v^i)$ and $jd_v^i = (xd_v^i, yd_v^i)$).

Job types description :

- Idling (0): the vehicle is sitting at a designated idling location (lot).
- Repositioning (1): the vehicle is on its way to a lot.
- Setup (2): the vehicle is on its way to pick a customer.
- Processing (3): a vehicle is transporting a customer to their location.
- Null (4) : no job specified to the vehicle.

1.2 Action space

Agents specify for each vehicle and request :

- req_rejections $= \{0,1\}^{|P|}$ a boolean vector that indicates whether a request is rejected or not (0 : not rejected, 1: rejected)
- req_assgts $= [\mathcal{V} \cup \{|\mathcal{V}|\}]^{|P|}$ represents the vehicle v assigned to the request at index $i \in P$. If the agents does not want a request to be assigned and *wait*, it specifies the value $|\mathcal{V}|$ at index i.
- reposition $= [L \cup \{|L|\}]^{|\mathcal{V}|}$ represents where the vehicles will be repositioned. The array means that the vehicle at index i will be assigned to the idling location l. Vehicles can be assigned to any idling location $l \in L$ with $L = \{0, 1, ..., num\_lots{-}1\}$ or if the agents does not want to reposition the vehicle, then it specifies the value $|L|$.

Lots represent the idling locations. There are predefined lots that are represented by their Cartesian coordinates ($lots = \{(x_l^i, y_l^i), \forall i \in L\}$).

Note that the first state s_0 begins at $t = 0$, d initialized on some day, with no pending requests and vehicles idling at a lot ($jt_v^1 = 0$, $jt_v^2 = 4$, $jt_v^3 = 4$).

A decision epoch is triggered by two events :

1. Arrival of a new request.
2. A vehicle complete a request and has no more jobs.

If a *max_interdecision_time* is specified, a new decision epoch if triggered if more than *max_interdecision_time* seconds have elapsed.

1.3 Problem constraints

- When a vehicle is given a request assignment ($jt_v^1 = 2$) it immediately begins traveling to the customer, and then immediately begins serving the customer ($jt_v^2 = 3$). When done, it triggers a decision epoch.
- If a vehicle is serving a customer ($jt_v^1 = 3$) and has a pending request ($jt_v^2 = 2$ and $jt_v^3 = 3$), it does not trigger a decision epoch.
- A vehicle is eligible for assignment (for serving a new customer) if it can be dispatched immediately, or it is currently serving a customer ($jt_v^1 \in \{0, 1, 3, 4\}$).
- Repositioning instructions are ignored for vehicles that are serving requests ($jt_v^1 = 2, jt_v^2 = 3 \implies jt_v^3 \neq 1$ and $jt_v^1 = 3 \implies jt_v^2 \neq 1$).
- If a vehicle is processing a request ($jt_v^1 = 2$ and $jt_v^2 = 3$), then it is not allowed to be assigned a second request ($jt_v^3 \neq 2$).
- If a vehicle runs out of assignments ($jt_v^1 = 4$), it **must** receive a job assignment in the next decision epoch.
- Customers must be picked up withing 5 minutes of when they first request.
- If a vehicle assigned to a request and their expected arrival time to that request exceeds 5 minutes, then the assignment is ignored.
- If a request is marked as rejected, then any assignment of that request to a vehicle is ignored.

– It is not possible to cancel an assignment or assign it to another vehicle.

The diagram 1 below shows all possible jobs transitions based on the problem and constraints description. Arrows with $*$ stands for conditional transition. In this case, it refers to the time limit constraint to pick up a customer.

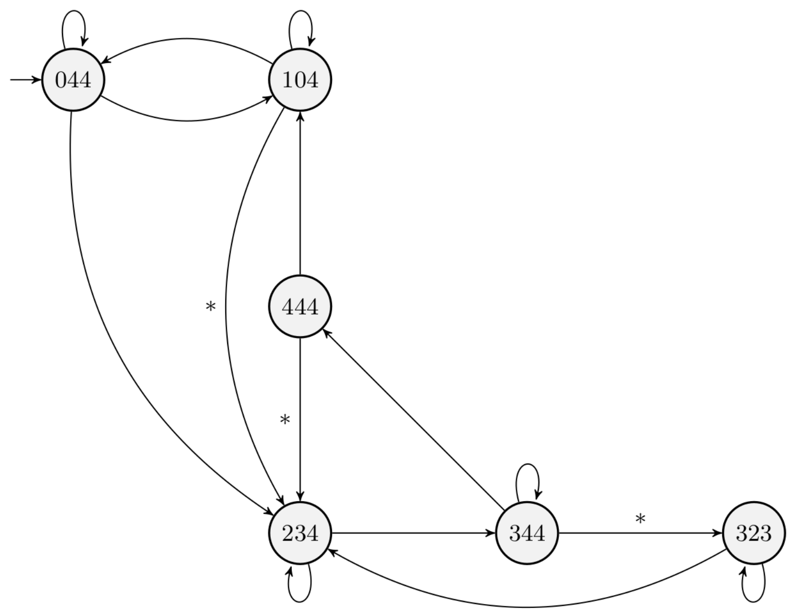

Fig. 1: Possible transitions. Idling (0), Repositioning (1), Setup (2), Processing (3), Null (4).

1.4 Agent constraints

– By default, agents are allowed unlimited wall-clock time to provide an action from a given observation. This is not especially realistic. For those looking for more of a challenge, try specifying a value for the *action_timelimit* argument. If the agent takes more than *action_timelimit* seconds (of wall-clock time) to specify an action, then the episode terminates with a reward of negative infinity (cf. the LTI setting).
– In evaluation, *max_interdecision_time* is set to 60 seconds (of simulated time).

1.5 Reward

As mentioned in the objective, the reward consists of the difference between the revenue earned from serving requests minus the travel costs. When serving a request, a vehicle earns a fixed reward $C_F = 10.75\$$ and a variable reward $C_D = 4.02\$/km$ that depends on the distance between the request's origin and

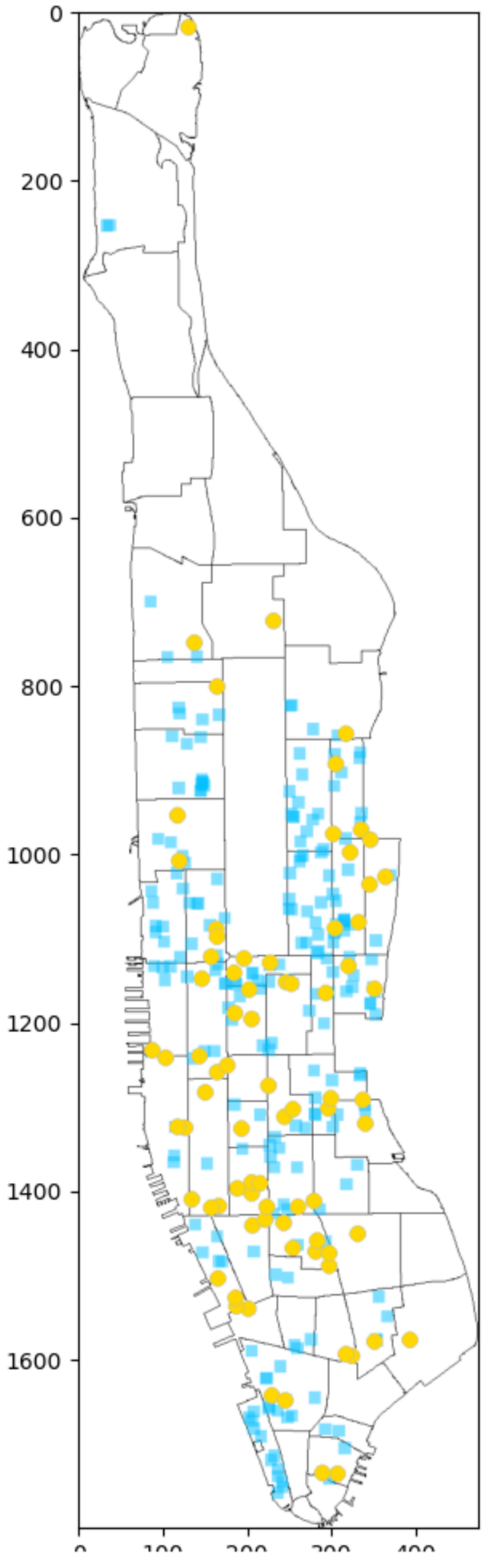

Fig. 2: Example of a Ride Hailing instance

destination. A travel cost $TC = 0.53\$/km$ is associated with the fleet. The total travel cost depends on the total distance traveled by the vehicles (either to reach or serve a client or to reposition).

Thus, the final reward function is given by the formula :

$$R = C_F + C_D \cdot \text{serving\_dist} - TC \cdot \ \text{total\_travel\_dist}$$