Real-time and Large-scale Fleet Allocation of Autonomous Taxis: A Case Study in New York Manhattan Island

Yue Yang$^1$, Wencang Bao$^2$, Mohsen Ramezani$^3$, Zhe Xu$^4$

$^1$ Beihang University, 37 Xueyuan Road, Beijing, 100191, P.R.China
$^2$ Georgia Institute of Technology, North Avenue, Atlanta, GA 30332, U.S.A
$^3$ University of Sydney, Camperdown, Sydney, NSW 2006, Australia
$^4$ Didi Chuxing, Compound 8, Dongbeiwang Road, Beijing, 100000, P.R.China

yang_yue@buaa.edu.cn, bomsun@gatech.edu, mohsen.ramezani@sydney.edu.au, xuzhejesse@didiglobal.com

Abstract

Nowadays, autonomous taxis become a highly promising transportation mode, which helps relieve traffic congestion and avoid road accidents. However, it hinders the wide implementation of this service that traditional models fail to efficiently allocate the available fleet to deal with the imbalance of supply (autonomous taxis) and demand (trips), the poor cooperation of taxis, hardly satisfied resource constraints, and on-line platform’s requirements. To figure out such urgent problems from a global and more farsighted view, we employ a Constrained Multi-agent Markov Decision Processes (CMMDP) to model fleet allocation decisions, which can be easily split into sub-problems formulated as a Dynamic assignment problem combining both immediate rewards and future gains. We also leverage a Column Generation algorithm to guarantee the efficiency and optimality in a large scale. Through extensive experiments, the proposed approach not only achieves remarkable improvements over the state-of-the-art benchmarks in terms of the individuals efficiency (arriving at 12.40%, 6.54% rise of income and utilization, respectively) and the platforms profit (reaching 4.59% promotion) but also reveals a time-varying fleet adjustment policy to minimize the operation cost of the platform.

Introduction

Autonomous vehicle industry is going through a rapid development right now. GM CRUISE, Tesla, Waymo and Daimler have tested their autonomous taxis on road, and more than ten global automakers have launched or planned programs regarding self-driving cars. Autonomous taxi, as a result, becomes an emerging topic in urban transportation. On one hand, it alleviates the burden of city congestion and increases road capacity (Greenblatt and Saxena 2015); on the other hand, it is also capable of reducing traffic accidents by preventing human errors. Furthermore, as a means of public transportation, the driver-less taxi is able to effectively hinder the spread of the COVID-19 for passengers, which makes it more attractive in such a pandemic crisis. Plenty of predictions and simulations showed the high possibility that the self-driving taxi could come into reality in the foreseeable future (Litman 2014).

Equipped with sensing and communication tools, each self-driving taxi can periodically upload real-time positions and status (i.e. vacant or occupied) to the platform. In turn, during each time slot (i.e. one minute), the platform will collect the arriving trips and make assignments. Under this circumstance, developing optimal control algorithms plays an important role for these self-driving platforms to balance the demands and supplies. One representative is order dispatching, where various methods (Lowalekar, Varakantham, and Jaillet 2018; Xu et al. 2018) have been widely studied to minimize the supply-demand mismatch, reduce the passengers’ waiting time, lower the fuel cost; another key application is redistributing available vehicles ahead of time to ameliorate the differences between demand and supply (Nair and Miller-Hooks 2011). If available taxis are precisely directed to locations with high demand, it will consequently improve the order fulfillment rate, and thus benefit for all aspects of the system: increasing the utilization of mobility, promoting the efficiency of each taxi and boosting the Gross Merchandise Volume (GMV) of the platform. As an example illustrated in Figure 1, five vacant taxis of current time step (A) will be guided to the potential pick-up locations to fulfill the demands in the next step (B).

![Figure 1: Fleet allocation examples for autonomous taxis.](image-url)
where a long-term optimization strategy is needed. Besides, the demand dramatically changes with hours in a day, so it is crucial to determine how many vehicles should be actually invested in advance. Resource restriction, always ignored by traditional models, also affects agents’ decisions significantly. Finally, allocating thousands of autonomous taxis of one round may pose a great challenge to the computational efficiency and optimality of optimization approaches.

To address these challenges, we demonstrate a novel Constrained Multi-agent Markov Decision Process (CMMDP) model, and design a Column Generation algorithm to accelerate the solving speed and retain tractability. The extensive experiments on simulation system further shows the proposed algorithm remarkably improves the individual’s efficiency and the platform’s profit. The highlights of this work are summarized as follows:

- We formulate the large-scale fleet allocation problem in a CMMDP setting, which considers the long-term payoffs and the immediate resource constraints simultaneously.
- To accommodate the real-time requirements, the CMMDP can be further split into dynamic assignment subproblems, and each decision is endowed with a combinatorial function of instant and learned future rewards.
- Facing the complexity of large-scale allocation problem, we employ the Column Generation algorithm to achieve a computational efficiency and optimality, which shows a great potential to be deployed in industrial application.
- Last but not least, the proposed model significantly outperforms the state-of-the-art fleet allocation methods with regard to individual’s efficiency and platform’s profit.

The remainder of the paper is structured as follows. Section 2 reviews existing ideas related to fleet allocation and CMMDP. In section 3 We define important concepts and a computational efficiency and optimality, which shows a model-free deep RL framework. Still, it hasn’t been solved efficiently that how to relieve the computational burden in hyper-scale cases.

- **Constrained Multi-agent Markov Decision Process**

Constrained multi-agent Markov Decision Process refers to the multi-agent MDP with resource constraints which always affect agents’ decisions in practical use. Beynier and Mouaddib assess the interdependence of agents under resource and temporal constraints. Matignon, Jeanpierre, and Mouaddib utilize a sequential decision making of multi-robot collaboration under communication constraints. Erwin Walraven considers road capacity constraints in taxi repositioning plan. Different from analyzing expectation, Nijs et al. measure the probability of resource constraint violation. Fowler et al. also discuss the performance of MDP with action, reward and probabilistic satisfaction constraints.

The main challenge in CMMDP is how to lower the computational complexity. One idea is to decompose multi-agent decisions making to individuals’. Beynier and Mouaddib allow the independence of decision making for each agent and consider the lost value stimulated by local decision on other agents. Another idea, which is also applied in our paper, harness column generation algorithm. Instead of solving individual agent’s policy or action per time, the column generation outputs the solutions for all agents at a time. Erwin Walraven illustrate the column generation algorithm for CMMDP. De Nijs and Stuckey combine column generation and CMMDP to investigate the balance of efficiency and safety in policy relaxation. However, few of CMMDP methods in fleet allocation field precisely contemplate real-time decision making which is of high importance for allocation efficiency, profit and customer satisfaction. In light of the fact, CMMDP is divided into subproblems to get a better and quicker real-time allocation in our model.

**Related Works**

This section is an overview of related works in general and specific levels. In the general level, we summarize some approaches for the resource allocation problem, the generalized version of fleet allocation. In the specific level, we focus on one methodology, CMMDP, and analyze its variants.

- **Resource Allocation**

  Autonomous taxi fleet allocation plan essentially is a resource allocation problem, that is, a decision of available resources to demands with time, cost and other limitations. In taxi allocation plan, a traditional way is generating recommendations of immediate next movements for each driver through models with fixed structures. Further details are discussed from the perspectives of route recommendation and taxis competing (Qian et al. 2015), increasing profitablity by reducing cruising miles (Powell et al. 2011), analysis high-revenue location factors (Hwang, Hsueh, and Chen 2015). Nevertheless, the long-run global-optimal solution is easily missed in the aforementioned studies because of uncertainties, especially in large-scale instances.

Recently some network-based models have emerged to solve that problem, such as MDP and Reinforcement Learning (RL). These models make real-time decisions under risks of uncertain features, and their flexible structure allows MDP and RL models to be effortlessly extended. David J. Musliner analyzes the interactions when MDP is extended to a multi-agent model. Robbel, Ollehoek, and Kochenderfer introduce a property called “anonymous influence” to measure the benefits of computational efficiency for variable elimination. Xu et al. formulate customer-taxi matching as a sequential decision-making problem to optimize instant and long-term rewards. Mao, Liu, and Shen estimate policy and value functions in a model-free deep RL framework. Still, it hasn’t been solved efficiently that how to relieve the computational burden in hyper-scale cases.

**Real-time Allocation Framework**

To model sequential decision making under uncertainty, we employ a Markov Decision Process (MDP) (Bellman 1957) in this section. Specifically, each autonomous taxi is defined as an agent with a spatial action set (movements); the whole operating area is formulated as a network \( G = (S, E) \), where \( S \) denotes the set of location \( s \) and \( E \) is the set of edge \( e(l, l') \) connecting two locations \( s \) and \( s' \).
After that, we address the resource limitation problem in multi-agent environments (Boutilier 1996) by CMMDP, in which the resource constraints are the maximum location capacity. To meet the on-line execution requirements, we split CMMDP into multiple subproblems combining both immediate rewards and future gains.

**Single-agent Setting**

As a stepping stone, a multi-agent MDP consisting of \( N \) independent agents can be defined through the single agent formulation \( (S, A, P, R, T) \) where only one agent interacts with the environment:

- **Time horizon** \( T \), operating time steps set \( \{1, ..., T\} \).
- **State space** \( S \), a finite set of local states, and each state \( s, \forall s \in  S \) is the current location of the agent.
- **Action subspace** \( A(s) \), candidate action set for agents at location \( s \). Here, we develop a searching scheme for computing an approximately feasible actions set for \( s \):

\[
A(s) = \{ s' \mid \tau(s, s') < \delta, \forall s' \in S \} \tag{1}
\]

where \( \tau(s, s') \) denotes the shortest travel time from \( s \) to \( s' \), which can be solved directly through Dijkstra’s algorithm (Dijkstra et al. 1959) in the network \( G \); \( \delta \) is the travel time threshold limiting the searching space of candidates. For a better understanding, we set an example in figure 2 to search for the actions of \( s_0 = l_0 \), consequently, the action space \( A(s_0) \) will be set to \( \{l_0, l_1, l_2, l_3\} \) derived from our scheme.

- **Global action space** \( A \), the joint space for all \( A(s), \forall s \in S \).

**Matching probability**, \( p_m(t, s) \), estimates the probability that the vacant taxi is matched to a trip. According to other scholars’ work (Buchholz 2018; Yu et al. 2019), the matching probability from a taxi’s view is:

\[
p_m(t, s) = 1 - e^{-\frac{n_{\text{trip}}(t, s)}{\theta}} \tag{2}
\]

where \( n_{\text{trip}}(t, s) \), \( n_{\text{taxi}}(t, s) \) denote the number of trips and taxis in \( s \) at \( t \). In addition, we will introduce a parameter \( \theta \in (0, 1] \) to describe how matching probability changes with demand-supply ratio \( \frac{n_{\text{taxi}}(t, s)}{n_{\text{trip}}(t, s)} \).

**Destination probability**, \( p_d(t, s, s') \), measures the likelihood of the destination of the trip being the state \( s' \) when the passenger is picked up at state \( s \) at step \( t \). This parameter can be estimated from the abundant historical trip data:

\[
p_d(t, s, s') = \frac{n_{\text{dest}}(t, s, s')}{n_{\text{trip}}(t, s)} \tag{3}
\]

where \( n_{\text{dest}}(t, s, s') \) represents the number of trips starting from \( s \) at step \( t \) to \( s' \).

**State transition probability** \( P(t, s, a, s') \) is the transition probability model that state \( s' \) will be reached when action \( a \) is taken in state \( s \) at step \( t \).

Suppose there is a taxi locating at state \( s = l \) of step \( t \) and it takes action \( a = l' \) to state \( l' \), then:

- If it successfully finds a trip from \( s' = l' \) to \( l'' \), it will directly transfer to state \( s'' = l'' \), then, the transition probability \( P(t, s, a, s'') \) can be defined as:

\[
P(t, s, a, s'') = p_m(t + \tau(l', l''), s') \cdot p_d(t + \tau(l', l''), s', s'') \tag{4}
\]

- Unfortunately, if it fails to find a trip, it will continue repositioning from \( s' = l' \), here the transition probability can be formulated bellow:

\[
P(t, s, a, s') = 1 - p_m(t + \tau(l', l''), s') \tag{5}
\]

To illustrate the state transition process, we present a transition scenario of \( s_0 \rightarrow s_1 \) in Figure 2. In this case, the transition probability \( P(t, s_0, l_1, s_1) \) can be obtained as

\[
p_m(t_0 + \tau(l_0, l_1), s_1) \cdot p_d(t_0 + \tau(l_0, l_1), s_1, s_2)
\]

In particular, we should note that if the taxi takes an action (i.e. \( l_2 \) in Figure 2) which is not directly linked to his current location in the network, it will take the shortest route composed of at least two locations (i.e. if taxi takes \( l_2 \) as an action, it means that it will take the shortest route \( l_0 \rightarrow l_1 \rightarrow l_2 \) in figure 2). Under this circumstance, the state transition process will be rather complicated since we need to take each passing location into consideration. For simplicity, the probable transitions of each intermediate location can be negligible because we assume there will be no trips assigned to the taxi until it arrives at the assigned destination, and we only discuss the state transitions happening in the final location.

**Immediate reward** \( R(t, s, a) \), is a reward model that maps steps \( t \), states \( s \) and actions \( a \) to rewards of the agent. Intuitively, the immediate reward can be set as the payment of a trip, by which the goal of the system is to maximize the Gross Merchandise Volume (GMV) of the platform.

**Optimal expected reward** \( V^*(t, s) \) is the maximal collective expected reward of the agent after \( T \) time steps. Hence, the optimal expected reward \( V^*(t, s) \) starting from state \( s \) at step \( t \):
where $Q(t, s, a)$, in the Bellman form (Bellman 1957), is the expected reward that a taxi can earn by starting from $s$ and specifying an action $a$ at step $t$:

$$Q(t, s, a) = R(t, s, a) + \sum_{s'} \gamma^r(s, s') [P(t, s, a, s')V^*(t + \tau(s, s'), s')]$$

Here, $\gamma$ is discount factor, set as a constant slightly smaller than 1 to ensure the existence of finite optimal expected payoff. Given the optimal value $V^*(t, s)$, an optimal policy $\pi^*$ of a single-agent is to simply act greedily:

$$\pi^*(t, s) = \arg \max_{a \in A(s)} \{Q(t, s, a)\}$$

Algorithm 1 Dynamic programming algorithm

**Input**: State set $S$, Action set $A$, Transition Probabilities $P$, Immediate reward $R$, Terminate step $T$  

**Output**: Optimal value $V^*$, $Q_{\pi^*}$, and Optimal policy $\pi^*$

1: Initialize a hash table $V^*$ to store optimal state values.
2: $V^*(t, s) = 0$ for all states $s$ where $t = T$.
3: for $t = T - 1$ to 1 do
4: for $l \in L$ do
5: $s \leftarrow l$
6: Compute $V^*(t, s)$ and $Q(t, s, a)$ based on Eq 6
7: Update the optimal policy $\pi^*(t, s)$ based on Eq 8
9: end for
10: return $V^*$, $Q_{\pi^*}$, $\pi^*$

Constrained Multi-agent Planning

So far we discussed MDP from the angle of one individual agent without any constraints. However, the aforementioned single-agent model falls short of the competition among multiple agents or the resource constraint on each action. Take an example similar to our problem, the number of vehicles taking a particular road at any given time is limited by the capacity of the road, i.e., a 3-lane road segment will fit at most 3 cars side by side. Back to our setting, the number of taxis heading for a particular location at any given time is limited by the trip amounts generated in this location. To model such a problem, we consider a large class of decision making problems involving multiple independent agents with resource constraints, which imposes restrictions on the actions that the agents could execute.

In the first place, We construct a Multi-agent MDP (MMMDP) problem $\langle N, S, A, P, R, T \rangle$ by extending the single-agent framework to a set of $N$ agents.
The Linear Program (LP) in Eq. [11] computes a solution that maximizes the expected value of agents joint policy under the resource constraints, and they will require optimizing a single large centralized program. Nevertheless, the LP can quickly grow very large, having \( O(N \cdot T \cdot |S'| \cdot |A'|) \) variables, which means that finding optimal policies is intractable in general. Also, CMMDP in Eq. [11] inevitably has a fundamental drawback: meeting the resource constraints only in expectation. To downscale the optimization variables and guarantee a strict adherence to the resource constraints, we further introduce a binary variable \( y(i, t, s, a) \rightarrow \{0, 1\} \), which indicates whether or not agent \( i \) takes \( a \) from \( s \) in \( t \), to reformulate the CMMDP as a Mixed-Integer Linear Programs (MILP) formalism in the following section.

**On-line Assignment**

From the perspective of the platform, real-time situations and on-line implementation should be considered in the optimization framework, which naturally falls into the category of ‘Dynamic assignment problem’ (Spivey and Powell 2004). Therefore, the LP of Eq. [11] will be split into subproblems as a ‘General assignment problem’ (Fisher and Jaikumar 1981) in every time step \( t \) to downscale the variable size and adapt to the on-line implementation:

\[
\begin{align*}
\max & \sum_{i} \sum_{a} y(i, t, s', a) \cdot Q_{\pi^*}(t, s', a) \\
\text{s.t.} & \sum_{i} y(i, t, s', a) \leq \Delta, \forall a \\
& \sum_{a} y(i, t, s', a) \leq 1, \forall i; \\
& y(i, t, s, a) \in \{0, 1\}
\end{align*}
\]

In each time step \( t \), the current state \( s' \) of each agent \( i \) will be collected and the goal of the online optimization algorithm is to determine the best matching between agents and candidate actions to maximize the sum of expected rewards \( Q_{\pi^*}(t, s', a) \). Specifically, the first constraint ensures the resource constraint \( \Delta \) imposed on a given action \( a \) at step \( t \). The second constraint guarantees that each agent will be assigned at most one action at a time.

Note that \( Q_{\pi^*}(t, s', a) \) represents an expected reward in multi-agent system with each agent having its unique policy \( \pi \). It is still an open issue on how to tackle large-scale self-interest agents planning problems. However, since all autonomous taxis in our setting are entirely subject to the centralized platform, we can restrict all agents to have a common-interest in maximizing the global gain, resulting in them sharing the same policy \( \pi^* \) of single-agent setting.

Intuitively, \( Q_{\pi^*}(t, s', a) \), a learned expected reward function, can be computed in Algorithm [1] based on historical data. In many previous researches on MDP-based methods (Yu et al. 2019; Shou et al. 2020), the optimal policy derived from historical episodes is directly used to guide taxis in the current episode, which naturally misses and overlooks the particularities and characteristics of the real-time data.

To capture more information in real-time planning, we design a modified reward value \( U(i, t, s', a) \) combining both learned reward function \( Q_{\pi^*}(t, s', a) \) and the real-time reward \( \hat{R}(t, s', a) \) as below:

\[
U(i, t, s', a) = \hat{R}(t, s', a) + Q_{\pi^*}(t, s', a)
\]

where \( \hat{R}(t, s', a) \) is the instant reward estimated at step \( t \), equivalent to average payment earned by each taxi from action \( a \).

Figure 3 illustrates the proposed procedure combining off-line learning step with on-line assignment step. Therefore, the objective of Eq. [12] can be further modified as:

\[
\max \sum_{i} \sum_{a} y(i, t, s', a) \cdot U(i, t, s', a) \quad (14)
\]

**Solving Approaches for Large-scale Setting**

Although problem in Eq. [12] decouples the constraints and enables us to solve small-scale problems in a reasonable time (i.e. by employing available algorithms (Bang-Jensen and Gutin 2008)) of minimum-cost flow problem), it becomes computationally expensive as the overwhelming number of variables are introduced. To solve larger-scale problems efficiently, we develop a column generation procedure that can be easily adapted to real-world cases. Conceptually, Column Generation is an effective technique for decomposing combinatorial optimization problems, such as stock cut problem (Gilmore and Gomory [1961]), graph coloring problem (Mehrotra and Trick [1996]), or the vehicle routing problem (Desrochers, Soumis, and Desrosiers [1984]).

The basic framework is to reformulate the assignment problem as a Restricted master problem (RMP), then choose entering columns by solving the pricing subproblems. Let

\[
\mathcal{Y} = \{Y_{a_1,1}^{1}, ..., Y_{a_1,K_{a_1}}^{1}, Y_{a_2,1}^{1}, ..., Y_{a_2,K_{a_2}}^{1}, ..., Y_{a_j,1}^{K_{a_j}}, ..., Y_{a_j,K_{a_j}}^{K_{a_j}}\}
\]

be the set of all feasible assignments(columns) to all actions(locations), where \( K_{a_j} \) is the number of feasible solutions of action \( a_j \). For any \( Y_{a_j}^{k}, 1 \leq k \leq K_{a_j} \), we assume
to be a group of feasible solutions satisfying the resource constraint of \( a_j \). Let \( \lambda_{a_j} \in \{0, 1\} \) be a binary variable indicating whether an assignment \( Y_{a_j}^k \) is selected for action \( a_j \). The assignment problem now is formulated as the Restricted master problem (RMP) in Eq. (17).

\[
Y_{a_j}^k = \{y^k(1, t, s^1, a_j), \ldots, y^k(N, t, s^N, a_j)\} \tag{16}
\]

Algorithm 2

**Input:** CMMDP \( \langle \mathbb{N}, S, A, \mathbb{P}, R, \mathbb{T} \rangle \)

**Output:** Optimal policies \( y^* \) for each agent.

1. \( Q_{x_0} \leftarrow \) Employing dynamic programming algorithm.
2. \( R, \Delta \leftarrow \) Estimated from the on-line step.
3. \( U \leftarrow \) Modifying reward function in Eq. (14).
4. Initialize Sub-objective \( \text{Obj} \leftarrow \infty \) and Solution \( \Psi \).
5. while \( \text{Obj} \geq 0 \) do
   6. Constructing the RMP in (17).
   7. \( \langle \mu_j, \nu_j \rangle \leftarrow \) Solving the RMP.
   8. for \( a_j \in A \) do
      9. Constructing the Pricing problem in (18).
      10. \( \text{Obj}, Y_{\text{new}} \leftarrow \) Solving the Pricing problem.
      11. if \( \text{Obj} > 0 \) then
          12. \( \Psi \leftarrow \) Adding \( Y_{\text{new}} \) into \( \Psi \)
          13. Break
      14. end if
   15. end for
5. end while
17. return \( y^* \) of RMP

where the first set of constraints enforce that each agent takes at most one action and the second set of constraints enforce that at most one assignment is selected for each action.

RMP can not be solved directly due to the exponential number of columns, so we start from a small subset of the columns. Whether other columns for the RMP join the subset is decided by solving the pricing problem of each action \( a_j \), which can be reformulated as a knapsack problem. Given \( \mu_j, \nu_j \) being the optimal dual price of the first and second constraints in the RMP, respectively, the pricing problem of a specific action \( a_j, \forall a_j \in A \) can be defined as:

\[
\begin{align*}
\max & \sum_{i} y(i, t, s^i, a_j) \cdot (U(i, t, s^i, a_j) - \mu_i) - \nu_j \\
\text{s.t.} & \sum_{i} y(i, t, s^i, a_j) \leq \Delta_{t,a_j} \\
& y(i, t, s^i, a_j) \in \{0, 1\}
\end{align*} \tag{18}
\]

Consequently, if the objective values of any pricing problems are less or equal to zero, the current optimal solution for RMP is also optimal for the unrestricted master problem (Savelsbergh 1997). Algorithm 2 summarizes the steps of our approach to deal with large-scale fleet allocation problem.

### Case Study of Manhattan

To provide a more complete understanding of the effectiveness of our model, we utilize a large-scale historical taxi data in Manhattan island to evaluate our proposed method.

### Basic Setting

The data used in this work were collected from taxi data-sets in Manhattan island, New York from November 1st-20st, 2019 on Yellow Cab’s website[1]. The data mainly includes trip’s location (origin and destination), generating time and payments etc. The study area is comprised of 67 locations (\(|L| = 67\)) and 164 edges (\(|E| = 164\)) according to the road network of Manhattan. Trip requests falling outside the spatial boundary are disregarded, and origins, destinations, taxis’ positions are converted to the locations in \( L \). Shortest paths and travel time among all locations are pre-calculated and stored in a look-up table.

To obtain the performance of our planning framework, a Monte Carlo simulation is implemented in this work. In the simulation phase, 3000 autonomous taxis are randomly generated on 00:00 am per day, and each day is treated as an episode segmented into 10-minute windows (\( T = 144 \)).

We use the first 15 episodes for learning and conduct evaluation on the following 5 episodes. All the quantitative results of fleet allocation presented in this section are averaged over ten runs. We adopt three evaluation metrics below:

- **Income Per Hour (IPH).** The average IPH represents the efficiency of taxis allocation as a whole.
- **Occupied rate (OR).** The occupied rate of a taxi is defined as the ratio of the time spent on carrying a trip to the total operating time of the taxi.
- **Gross Merchandise Volume (GMV).** GMV records the total payment of all trips served by taxis in the system.

Additionally, in this phase we set some hyper-parameters: the action searching threshold \( \delta = 10 \) min, the matching parameter \( \theta = 0.94 \) by logistic regression on real-world data, and the lower-bound probability \( \hat{b}_{m} \) is presented as time-dependent forms in Table 1 empirically.

[1]https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
Table 1: The range of the lower-bound probability $\hat{p}_m$.

| Hours          | 00:00am | 02:00am | 07:00am | 06:00pm |
|----------------|---------|---------|---------|---------|
| $\hat{p}_m$    | 0.5     | 0.4     | 0.7     | 0.8     |

Evaluation

To evaluate the performance of the optimal policy from our real-time allocation model, there are two baseline heuristics:

1. **Hotspot walk**, essentially defines a stochastic policy $\pi^i$ for each agent $i$, where the actions are chosen according to the probability $Pr^i(t, s^i, \pi^i(t, s^i) = a)$ computed by trips quantity at each step $t$:

   $$Pr^i(t, s^i, \pi^i(t, s^i) = a) = \frac{n_{trip}(t, a)}{\sum_{a'} n_{trip}(t, a')}$$  \hspace{1cm} (19)

2. **DiDi repositioning** [Xu et al., 2020], employs a centralized optimization framework to determine the repositioning tasks (next pick-up location) of each taxi. Each task will be assigned by solving the Mixed-integer linear programming (MILP) model similar to Eq. (12) while the reward function $U(i, t, s^i, a)$ will be set to the partial derivative with respect to $n_{taxi}$ in Eq. (19):

   $$U(i, t, s^i, a) = \frac{\partial p_m(t + \tau(s^i, a), a)}{\partial n_{taxi}(t + \tau(s^i, a), a)}$$  \hspace{1cm} (20)

Table 2 compares average IPH, OR and GMV of Hotspot Walk, DiDi Repositioning and Optimal policy from our approach. Averagely our policy provides the best solution for taxis with 12.40% higher average IPH, 6.54% larger average OR and 4.59% better GMV compared to Hotspot Walk. DiDi repositioning ranks next, leading 5.27% average IPH, 3.07% average OR, 3.56% GMV than Hotspot Walk. In summary, the proposed method showed a better performance on all evaluation metrics, indicating amelioration on both individual’s efficiency and platform’s profits. In detail, Figure 4 and 5 show the concrete IPH and OR probability distributions of the three models, respectively. Consistent with aforementioned analysis, the optimal policy of our approach achieves the best average IPH and OR, reflecting that our approach provides a better utilization rate of taxi resource and presents a great potential in practical use.

Dynamic Fleet Adjustment

Having obtained encouraging results of optimal policies in simulation, we are also curious about adjusting the fleet rather than maintaining a fixed size during the whole day. Because making a dynamic adjustment of fleet could reduce unnecessary cost and improve some social welfare like mitigating traffic congestion and conserving fuel.

Figure 6 depicts the heatmaps of taxis’ matching probability between 05:00am and 06:00pm, we can observe that the matching probability at 05:00am is significantly weaker than 06:00pm. The main factor resulting in this conspicuous difference is the oversized fleet serving such scarce demands in 05:00am. Therefore, it’s essential for the platform to dynamically adjust the fleet size to avoid waste.

According to the simulation results, we further compare the available fleet and actually required fleet of each step $t$ in Figure 7. The green curve shows the available fleet size and the red indicates the required size, providing a guideline to control the fleet size. In particular, the platform need
Conclusions and Future Works

In this paper, we propose a novel framework of a real-time fleet allocation approach providing an effective redistribution for autonomous taxis on self-driving platforms. To do so, we model the allocation problem as a CMMDP and then divide it into multiple assignment problems by online implementation, in which the value of assignment pair is computed as the sum of an instant reward and a long-term expected value learned from historical data. Furthermore, the real-world large-scale assignment is solved more efficiently with the column generation algorithm. Through extensive experiments on the simulator, the proposed algorithm remarkably improves individual’s efficiency and platforms profit, and also reveals a time-varying fleet adjustment policy to minimize the operation cost.

In this work, we assume taxis transfer to next-step destinations immediately, whereas the routing paths are ignored. Investigating path decisions possesses superior appliance value [Yu et al. [2019]; Tang et al. [2020]]. Also, in the fleet adjust experiment, it appears that dynamic fleet size delivers lower cost and higher utilization, while dynamic fleet size adjustment models such like determining when, where and which taxis should join and exit the operation [Ye et al. [2020]] are missed in this work. Including such appealing models in the future would be more beneficial.

Acknowledgments

The author Yue Yang would like to appreciate his precious and impressive experience of being an intern in Didi Chuxing, and all the data and codes of this work were shared on the GitHub.

References

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Bang-Jensen, J.; and Gutin, G. Z. 2008. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media.

Bellman, R. 1957. A Markovian decision process. *Journal of mathematics and mechanics* 679–684.

Bertsekas, D. P.; Bertsekas, D. P.; Bertsekas, D. P.; and Bertsekas, D. P. 1995. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA.

Beynier, A.; and Mouaddib, A.-I. 2005. A polynomial algorithm for decentralized Markov decision processes with temporal constraints. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 963–969.

Beynier, A.; and Mouaddib, A.-I. 2006. An iterative algorithm for solving constrained decentralized Markov decision processes. In *AAAI*, volume 6, 1089–1094.

Boutilier, C. 1996. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, 195–210.

Buchholz, N. 2018. Spatial equilibrium, search frictions and dynamic efficiency in the taxi industry. Technical report, Working Paper.

David J. Musliner, Edmund H. Durfee, J. W. D. A. D. R. P. G. M. S. B. 2006. Coordinated Plan Management Using Multiagent MDPs. *the Association for the Advancement of Artificial Intelligence (AAAI)* 4–11.

De Nijs, F. 2019. *Resource-constrained multi-agent Markov decision processes*. Ph.D. thesis, Delft University of Technology.
De Nijs, F.; and Stuckey, P. J. 2020. Risk-Aware Conditional Replanning for Globally Constrained Multi-Agent Sequential Decision Making. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, 303–311.

Desrosiers, J.; Soumis, F.; and Desrochers, M. 1984. Routing with time windows by column generation. Networks 14(4): 545–565.

Dijkstra, E. W.; et al. 1959. A note on two problems in connexion with graphs. Numerische mathematik 1(1): 269–271.

Erwin Walraven, M. T. J. S. 2018. Column Generation Algorithms for Constrained POMDPs. Journal of Artificial Intelligence Research 62: 489–533.

Fisher, M. L.; and Jaikumar, R. 1981. A generalized assignment heuristic for vehicle routing. Networks 11(2): 109–124.

Fowler, M.; Tokekar, P.; Charles Clancy, T.; and Williams, R. K. 2018. Constrained-Action POMDPs for Multi-Agent Intelligent Knowledge Distribution. In 2018 IEEE International Conference on Robotics and Automation (ICRA), 3701–3708.

Gilmore, P. C.; and Gomory, R. E. 1961. A linear programming approach to the cutting-stock problem. Operations research 9(6): 849–859.

Greenblatt, J. B.; and Saxena, S. 2015. Autonomous taxis could greatly reduce greenhouse-gas emissions of US light-duty vehicles. Nature Climate Change 5(9): 860–863.

Hwang, R.-H.; Hsueh, Y.-L.; and Chen, Y.-T. 2015. An effective taxi recommender system based on a spatio-temporal factor analysis model. Information Sciences 314: 28–40.

Litman, T. 2014. Autonomous vehicle implementation predictions. Victoria Transport Policy Institute 28(2014).

Lowalekar, M.; Varakantham, P.; and Jaillet, P. 2018. Online spatio-temporal matching in stochastic and dynamic domains. Artificial Intelligence 261: 71–112.

Mao, C.; Liu, Y.; and Shen, Z.-J. M. 2020. Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. Transportation Research Part C: Emerging Technologies 115: 102626.

Matignon, L.; Jeanpierrre, L.; and Mouaddib, A.-I. 2012. Coordinated Multi-Robot Exploration Under Communication Constraints Using Decentralized Markov Decision Processes. In AAAI 2012, p2017–2023. Toronto, Canada. URL https://hal.archives-ouvertes.fr/hal-00971744

Mehrotra, A.; and Trick, M. A. 1996. A column generation approach for graph coloring. informs Journal on Computing 8(4): 344–354.

Nair, R.; and Miller-Hooks, E. 2011. Fleet management for vehicle sharing operations. Transportation Science 45(4): 524–540.

Nijs, F. D.; Walraven, E.; de Weerdt, M. M.; and Spaan, M. T. J. 2017. Bounding the Probability of Resource Constraint Violations in Multi-Agent MDPs. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, 3562–3568. AAAI.

Powell, J. W.; Huang, Y.; Bastani, F.; and Ji, M. 2011. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In International Symposium on Spatial and Temporal Databases, 242–260. Springer.

Qian, S.; Cao, J.; Mouël, F. L.; Sahel, I.; and Li, M. 2015. SCRAM: A Sharing Considered Route Assignment Mechanism for Fair Taxi Route Recommendations. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 15, 955964. New York, NY, USA: Association for Computing Machinery. ISBN 9781450336642. doi:10.1145/2783258.2783261. URL https://doi.org/10.1145/2783258.2783261

Robbel, P.; Oliehoek, F. A.; and Kochenderfer, M. J. 2016. Exploiting Anonymity in Approximate Linear Programming: Scaling to Large Multiagent MDPs. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI16, 25372573. AAAI Press.

Savelsbergh, M. 1997. A branch-and-price algorithm for the generalized assignment problem. Operations research 45(6): 831–841.

Shou, Z.; Di, X.; Ye, J.; Zhu, H.; Zhang, H.; and Hampshire, R. 2020. Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning. Transportation Research Part C: Emerging Technologies 111: 91–113.

Spivey, M. Z.; and Powell, W. B. 2004. The dynamic assignment problem. Transportation Science 38(4): 399–419.

Tang, J.; Wang, Y.; Hao, W.; Liu, F.; Huang, H.; and Wang, Y. 2020. A Mixed Path Size Logit-Based Taxi Customer-Search Model Considering Spatio-Temporal Factors in Route Choice. IEEE Transactions on Intelligent Transportation Systems 21(4): 1347–1358.

Xu, Z.; Li, Z.; Guan, Q.; Zhang, D.; Li, Q.; Nan, J.; Liu, C.; Bian, W.; and Ye, J. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 905–913. ACM.

Xu, Z.; Men, C.; Li, P.; Jin, B.; Li, G.; Yang, Y.; Liu, C.; Wang, B.; and Qie, X. 2020. When Recommender Systems Meet Fleet Management: Practical Study in Online Driver Repositioning System. In Proceedings of The Web Conference 2020, 2220–2229.

Ye, X.; Li, M.; Yang, Z.; Yan, X.; and Chen, J. 2020. A Dynamic Adjustment Model of Cruising Taxicab Fleet Size Combined the Operating and Flied Survey Data. Sustainability 12(7): 2776.

Yu, X.; Guo, S.; Hu, X.; and Park, H. 2019. A Markov decision process approach to vacant taxi routing with e-hailing. Transportation Research Part B: Methodological 121: 114–134.