Adaptive Operating Hours for Improved Performance of Taxi Fleets

Rajiv Ranjan Kumar
School of Computing and Information Systems, Singapore Management University
rajivrk.2017@phdcs.smu.edu.sg

Pradeep Varakantham
School of Computing and Information Systems, Singapore Management University
pradeepv@smu.edu.sg

Shih-Fen Cheng
School of Computing and Information Systems, Singapore Management University
sfcheng@smu.edu.sg

ABSTRACT

Taxi fleets and car aggregation systems are an important component of the urban public transportation system. Taxis and cars in taxi fleets and car aggregation systems (e.g., Uber) are dependent on a large number of self-controlled and profit-driven taxi drivers, which introduces inefficiencies in the system. There are two ways in which taxi fleet performance can be optimized: (i) Operational decision making: improve assignment of taxis/cars to customers, while accounting for future demand; (ii) strategic decision making: optimize operating hours of (taxi and car) drivers. Existing research has primarily focused on the operational decisions in (i) and we focus on the strategic decisions in (ii).

We first model this complex real world decision making problem (with thousands of taxi drivers) as a multi-stage stochastic congestion game with a non dedicated set of agents (i.e., agents start operation at a random stage and exit the game after a fixed time), where there is a dynamic population of agents (constrained by the maximum number of drivers). We provide planning and learning methods for computing the ideal operating hours in such a game, so as to improve efficiency of the overall fleet. In our experimental results, we demonstrate that our planning based approach provides up to 16% improvement in revenue over existing method on a real world taxi dataset. The learning based approach further improves the performance and achieves up to 10% more revenue than the planning approach.

KEYWORDS

Equilibrium Solution, Game Theory, Optimization

ACM Reference Format:
Rajiv Ranjan Kumar, Pradeep Varakantham, and Shih-Fen Cheng. 2021. Adaptive Operating Hours for Improved Performance of Taxi Fleets. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

In recent years, various advances in mobile and information technologies have transformed many sectors of economy, allowing workers to participate in the workforce flexibly using mobile Apps. This form of worker engagement, commonly called the *digital gig economy*, has brought benefits and challenges for both firms and workers. For workers in the digital gig economy, the critical decisions are when and how to work. These decisions are challenging to derive since the performance of a worker depends on not just his/her own efforts, but also on other worker's decisions.

In this paper, we propose to study how workers should supply their labor in the digital gig economy, in anticipation of competitions from other workers. To provide a concrete context, we focus on taxi drivers in the transport gig economy, since we have collected a rich set of proprietary data logging locations and statuses of all taxi drivers in a major Asian city over a span of multiple years. Although our numerical examples focus on taxi drivers, we believe our models and solution approaches are general enough to be applied to other similar settings (e.g., ride-hailing, logistics, or food-delivery services).

To facilitate game-theoretic modeling, we discretize an agent's strategy space by dividing the city into zones of reasonable size. The time horizon is also discretized into periods with uniform length. The strategic decision of a driver (agent) consists of two stages: 1) the agent decides when to enter the system and begin working; and 2) after entering the system, the agent will work for a pre-determined duration, during which the agent has to decide how to move around the city in order to maximize his utility. For a particular agent, his utility depends on not just his own strategy, but also other agents' strategies. This motivates the use of game-theoretic models in our formulation. Structurally speaking, our game-theoretic model has the congestion game property, as an agent's expected reward in staying in a zone is non-increasing in the number of competing agents in the same zone. However, agents in our formulation are subject to *involuntary movements* when they are hired and have to move to the destinations specified by the passengers. This class of model was first proposed by [20], and subsequently used extensively in explaining taxi driver's strategic behaviors. Our first contribution lies in the expansion of an agent's strategy space to include entry time. This expansion allows us to model the fluctuation of active agent population in the second stage as a direct consequence of agents' respective strategic decisions in the first stage.

An agent's strategy is thus composed of the choice of the time period to begin working and a sequence of zones to visit in time periods when the agent is active. This game-theoretic model is intractable in practice, as the size of the game is exponential in the number of agents, and it is common to have thousands of agents. To make the model tractable, we follow [20] and assume agents to be homogeneous and anonymous, implying that we only need to consider agent counts in each zone and time period. With these assumptions, we could define the solution to our problem as a symmetric Nash equilibrium where all agents should follow (since agents are assumed to be homogeneous and anonymous).

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

The symmetric Nash equilibrium should contain strategies for agents to adopt for both stages of their decision making process. For the first stage, we should provide agents with a distribution over their starting time periods. For the second stage, the equilibrium strategy should contain a policy that instruct them on which zones to move to for each (zone, time) tuple.

To compute a symmetric Nash equilibrium efficiently, we have developed two variants of the fictitious play algorithm. The first variant, Fictitious Play-Based Planning for Dynamic Population (FP-P-DP), is based on multi-agent Markov decision process (MDP), which formulates individual agent's best response computation problem as a MDP, and the fictitious play process is used as a mechanism to coordinate responses from all agents. The second variant, Fictitious Play-Based Learning for Dynamic Population (FP-L-DP), approximate the best response computation by applying deep learning technique. In our numerical experiments, the FP-L-DP approach could achieve performance similar to that of the FP-P-DP approach, but with much shorter computational time.

To demonstrate the effectiveness of our approaches, we use a real-world dataset derived from over 20,000 taxis in a major Asian city. Besides algorithmic comparison of our approaches, we also compare the identified equilibrium policy against the actual driver's policy extracted from the dataset. From the comparison, we can see clear advantage of adopting the equilibrium-inspired policies for drivers.

In summary, our paper makes the following methodological and practical contributions:

(1) We introduce a two-stage game-theoretic formulation to directly capture the dynamics of agent population changes over time, which results from agents' strategic choices on the work starting time. The model also captures important congestion game features that are commonly seen in transport gig economy.
(2) We introduce two fictitious play-based algorithms to search for a symmetric Nash equilibrium in our model. The major difference of these two variants is on how we compute the best response strategies. While one relies on MDP-based planning method, another one relies on deep learning approach to approximate the best response policy.
(3) We demonstrated the effectiveness of our approach and the importance of using equilibrium policies by using a large-scale real-world dataset containing more than 20,000 drivers. Our real-world comparison baseline is taxi driver's actual work schedule choices extracted from the dataset.

The rest of the paper is organized as follows: In the next section we provide related work. After that we provide a motivation section, where we provided details of the problem setting (taxi fleet optimization problem) that we are solving in this work. In the later section we introduce a general framework called the Selfish Routing with Transition uncertainty (SRT) for modeling problems such as taxi fleet optimization. We provide details of this framework before we move to next section where we a provide solution approach section to solve SRT model (fictitious play-based equilibrium computation) where we provide two approaches (a FP based planning approach and a FP based learning approach) to solve our SRT model. Finally we provide experimental results on a real world taxi data set from a large Asian city in the experimental section.

2 RELATED WORK

The optimization of fleet operations in transport gig economy has been well-studied from the time when taxi is the only service mode. During early years, the focus has mostly been on taxi dispatch (e.g., [18]). With ride-hailing services gaining popularity globally, focus has gradually shifted to approaches that could increase driver-level or fleet-level performances.

At the driver-level, there are efforts that focus on finding optimal driving strategies [7, 15, 16]. This line of work aim to maximize driver's profits by providing routing suggestions based on currently available passengers and drivers.

To account for mutli-period considerations, a number of researchers have recently used various Markov Decision Process (MDP) approaches to optimize a single agent's sequential decision-making process over a time horizon [6, 17, 21, 22, 25]. Compared to earlier efforts, these MDP-based approaches aim to optimized long-term expected rewards for drivers, given different assumptions about the environment and driver states.

As these efforts are mostly single-agent based, when adopted universally, it might lead to suboptimal results. Some researchers have thus adopted game-theoretic approaches to optimize fleet-level performance, considering all taxi drivers as agents explicitly. An example of such approach is mentioned in the introduction [20], however, it does not consider dynamic driver population as we do. The closest effort in the literature is by [5], which explicitly considers driver's operating hours as constraints. Compared to [5], our approach incorporates starting time period as part of agent's *strategies* (not constraints). And our work also integrate service time decisions with roaming decisions (the second-stage strategies).

Methodologically speaking, our game-theoretic formulation is most relevant to the modeling of dynamic agent population. In the game theory literature, dynamic agent population can be achieved by modeling agent activation as a stochastic process: deciding whether individual agents would become active following certain stochastic processes [1, 2, 9]. Alternatively, dynamic agent population can also be achieved by using Poisson games (a form of population games), where the number of active players at any time step in the game is supposed to be drawn from a random variable, whose probability distribution is known [12, 13]. Inter-time-period dependency in the number of agents could also be established by assuming the the number of active agents follows a conditional probability distribution on the number of active agents from the previous time period [8].

In all these approaches with finite number of players, the dynamics of agent population are determined by exogenous parameters/distributions, and not controlled by agents. In contrast, our approach models dynamic agent population explicitly, not as exogenous parameters, but as part of agent's strategy space.

3 MOTIVATING DOMAIN: TAXI FLEET OPTIMIZATION

In urban cities such as Singapore, New York, Hong Kong, taxis are considered an important mode of public transportation. However,

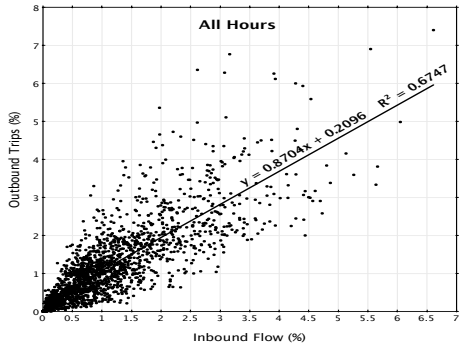

Figure 1: Correlation between incoming flow and outgoing trips over all zones.

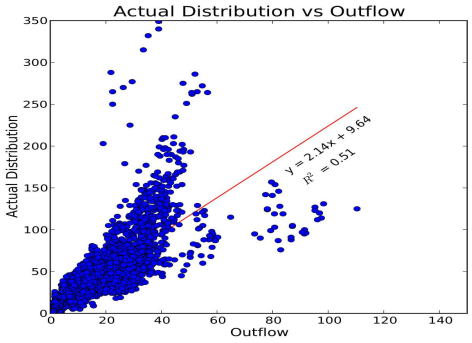

Figure 2: Correlation between actual taxi distribution and outflow.

because of independent and myopic optimization of taxi drivers (Figure 1 hints to this myopic optimization[1]), the positioning and movement of taxis are not in synchronization with customer demands arising in various parts of a city (see Figure 2). For instance, high-demand zones (such as airports, popular attractions) typically get many more taxis than required because of this independent optimization. Our research is motivated by the problem of figuring out how many taxis should start at which time of the day, while coordinating the movement of a fleet of self-interested taxis[2]. More specifically, we are interested in generating advice for taxi drivers so as to improve their revenue distribution and also better correlate movements of taxis with customer flow.

A fleet of taxis P is serving a city divided into M zones and the goal is to provide decision support for taxi drivers on their movement decisions at each time step. The flow of customers between any two zones i and j starting at time step t is given by $fl^t(i,j)$. The demand for taxis in a zone i at time step t is given by $\sum_j fl^t(i,j)$. If the number of taxis in a particular zone during a time period is fewer than the number of customers in that zone, all taxis will be hired (the determination of their destinations is described in the following paragraph). Otherwise, only a fraction of the taxis (equal to the number of customers) will be hired. We discretize the time horizon into time intervals and provide decision support at each of these time points.

The movements of taxis between zones depends on whether they are hired by a customer or not. If a taxi is hired in a zone, the movement is involuntary (decided by the customer onboard) and is governed by the probability distribution computed from the outgoing flows of customers from that zone to other zones. Therefore, if a taxi is hired by a customer in zone i, during a time interval starting with t, the probability of moving to zone j is $\frac{fl^t(i,j)}{\sum_j fl^t(i,j)}$. Furthermore, a hired taxi in such a case will receive a revenue of $r^t(i,j)$ and incur a cost of $c^t(i,j)$ (we will defer the formal definitions to later sections, however, do note that the value of $r^t(i,j)$ is dependent on other agent's actions, while $c^t(i,j)$ is a constant that is independently determined).

On the other hand, if a taxi is not hired, its movement is voluntary and it will receive no revenue but incur a cost of $c^t(i,j)$. While previous papers [11, 16, 20, 23, 24] have focussed on providing decision support for each taxi on which zone to move to at each time step, so that there is no incentive (with respect to expected revenue) for individual taxis to deviate from the suggested decisions, our focus is on identifying the number of taxis that should be active at different times of the day, given a constraint on the duration each taxi driver can work.

4 MODEL: SRT

We now introduce a general framework called the *Selfish Routing with Transition uncertainty* (SRT) for modeling problems such as taxi fleet optimization . SRT is an extension of the Distributed Decision model for Agent Populations (DDAP) introduced by [20]. Informally, SRT represents decision problems in congestion scenarios under movement uncertainty and can be viewed as a combination of stochastic games and selfish routing. More specifically, SRT generalizes the notion of resources and movement uncertainty to states and transition functions respectively.

SRT represents a subset of problems represented by the generic stochastic game model [10, 14, 19]. In SRT, the transition and reward functions for an agent are dependent only on the aggregate distributions of other agent states, whereas in stochastic games the transition and reward function for an agent can be dependent on specific state and action of every other agent. SRT represents problems with selfish agents and hence is different to cooperative models such as Decentralized POMDPs (DEC-POMDP) [3].

An SRT instance is the tuple:

$$\left\langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \{\mathcal{R}_\tau\}_{\tau\in\Gamma}, \boldsymbol{d}^0, \delta, H \right\rangle,$$

$\mathcal{S}$ corresponds to the set of states encountered by each agent. $\mathcal{A}$ is the set of actions executed by each agent. The transition and reward models for specific state action pairs at a decision epoch are dependent on the distribution of agent states at that decision epoch.

$\mathcal{T}$ models the involuntary movements of every agent and more specifically, $\mathcal{T}^t(s,a,s',\boldsymbol{d})$ represents the probability that an agent of in state $s \in \mathcal{S}$ after taking action $a \in \mathcal{A}$ would transition to state

[1]We provide a correlation between incoming taxis to zones and the outgoing hired taxis from those zones over a 3 month period for a major taxi company in Singapore. In other words, higher correlation implies a preference for zones with higher demand. As we see from Figure 1, the R^2 value is close to 0.7 suggesting a high correlation.
[2]In our definition, we assume that each taxi is driven by an independent driver (thus we use *taxi* and *taxi driver* interchangeably).

s', when the state distribution of all agents is $\boldsymbol{d}$ at time t. Similarly, $\mathcal{R}^t(s, a, s', \boldsymbol{d})$ is the reward obtained by an agent of type τ when in state s, taking action a and moving to state s' when the state distribution of other agents is $\boldsymbol{d}$ at time t. δ is the maximum number of time steps any agent can be active. Finally H represents the time horizon of the decision making process.

The objective in solving a SRT is to compute (a) a policy π_i for each agent i; (b) a starting state and time α_i for each agent i; such that there is no incentive to unilaterally deviate from its policy and/or its starting state and time , in terms of the expected value (i.e., $\mathcal{V}(\vec{\pi}_i^0, \alpha_i, \vec{\pi}_{-i}^0, \alpha_{-i})$). Individual agent policy, π_i, has the same definition as the one used for MDPs . That is to say, $\pi_i^t(s, a)$ indicates the probability of agent i taking action a in state s at time t and $\sum_a \pi_i^t(s, a) = 1, \quad \forall i, t, s$.

Initial distribution (starting state and time) has the same definition as the one used in MDPs except, in MDPs every agent starts at time 0 in some fixed states, where this is something SRT model need to optimize. That is to say, $\alpha_i(t, s)$ indicates agent i becomes active in SRT model in state s at time t. And $\sum_{t,s} \alpha_i(t, s) = 1, \quad \forall i$

4.1 Taxi Fleet Optimization as a SRT

The taxi fleet optimization problem can be represented as a SRT with the following mapping: $\mathcal{P}$ is the set of taxis in the fleet, $\mathcal{S}$ is the set of zones a taxi could move to, $\mathcal{A}$ is the set of zones to which a taxi driver wishes to move. The transition function, $\mathcal{T}$, depends on the involuntary movements between zones. The involuntary movement between any two zones i and j at a time step t is determined by the number of customers (customer flow), $fl^t(i, j)$ moving between those zones. Since the movement of taxis and the revenues received for serving customers are dependent only on the customer flows, the type index is not necessary and hence is dropped for purposes of this section. Equation (1) provides the expression for computing the transition probabilities between states.

Intuitively, if the number of taxis is less than the number of customers in a zone, then all taxis will be hired. It should be noted that this assumption is valid as long as the size of the zone is small enough for a taxi to quickly and fully cruise a zone for customers. The exact transition probabilities depend on normalized demands to other zones from the current zone (**C1** represents this case). On the other hand, if the number of taxis is more than the number of customers in a zone, the transition probabilities should depend on whether the action (intended zone) coincides with the destination zone (**C2** and **C3** represent these two cases). The condition **C2** is only possible when the taxi agent is hired by a customer heading towards any s' that is not a. For condition **C3**, the taxi agent can either be free or hired by a customer heading towards the agent's intended zone.

$$\mathcal{T}^t(s, a, s', \boldsymbol{d}) = \begin{cases} \frac{fl^t(s,s')}{\sum_{\hat{s}} fl^t(s,\hat{s})} & \text{if } \sum_{\hat{s}} fl^t(s, \hat{s}) \geq d_s (\mathbf{C1}) \\ \frac{fl^t(s,s')}{d_s} & \text{if } \sum_{\hat{s}} fl^t(s, \hat{s}) < d_s \text{ and } a \neq s' (\mathbf{C2}) \\ 1 - \frac{\sum_{\hat{s} \neq s'} fl^t(s,\hat{s})}{d_s} & \text{if } \sum_{\hat{s}} fl^t(s, \hat{s}) < d_s \text{ and } a = s' (\mathbf{C3}) \end{cases} \quad (1)$$

Similar to the transition probabilities, the reward function $\mathcal{R}^t(\cdot)$ is defined differently under these three conditions:

$$\mathcal{R}^t(s, a, \boldsymbol{d}) = \begin{cases} \sum_{s'} \mathcal{T}^t(s, a, s', \boldsymbol{d}) \cdot \left(r^t(s, s') - c^t(s, s')\right) & \mathbf{C1} \\ \sum_{s' \neq a} \mathcal{T}^t(s, a, s', \boldsymbol{d}) \cdot \left(r^t(s, s') - c^t(s, s')\right) & \mathbf{C2} \\ \frac{fl^t(s,a)}{d_s} \cdot r^t(s, a) - \mathcal{T}^t(s, a, a) \cdot c^t(s, a) & \mathbf{C3} \end{cases} \quad (2)$$

It should be noted that taxis are hired in conditions **C1** and **C2**; therefore, the expected rewards in these two cases are the sum of expected rewards to all feasible destinations. For **C3**, cost is incurred for sure, but revenue can only be earned if the taxi is hired.

Our goal in solving taxi fleet optimization problem as a SRT is to maximize expected revenue for individual taxi drivers who are perfectly rational and follow computed policies. As taxi drivers can only increase their earnings by serving more customers, increasing the average expected revenues for all taxi drivers implies that the number of unserved customers will also decrease. Therefore, both the global and individual objectives are aligned and can be optimized without involving multi-objective reasoning. In the following numerical example, we illustrate how transition probabilities and the rewards are computed for a small problem:

Example 1. *Consider a map with three zones, $\mathcal{S} = \{s_0, s_1, s_2\}$. For simplicity we set flow values to 1 across all time periods; i.e., one passenger goes from each zone to an adjacent zone in all time periods. We also set rewards and costs to be fixed at $r^t(s, s') = 1$ and $c^t(s, s') = 0$ for all time periods t and all zones $s, s' \in \mathcal{S}$. If the distribution of taxis at a given time period t is $\boldsymbol{d}^t = (1, 1, 4)$, then:*

- *The transition function $\mathcal{T}^t(s, a, s', \boldsymbol{d}^t)$ is specified by matrix $m(s)$, in which the row label represents action a, and the column label represents destination zone s'. Transition function for s_0 and s_2 are:*

$$m(s_0) = \begin{pmatrix} 0.0 & 0.5 & 0.5 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.5 & 0.5 \end{pmatrix} \quad m(s_2) = \begin{pmatrix} 0.75 & 0.25 & 0.0 \\ 0.25 & 0.75 & 0.0 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$$

- *Similarly, the reward function $\mathcal{R}^t(s, a, \boldsymbol{d})$ is specified as a matrix, in which the row label represents current zone s, and the column label represents action a:*

$$\begin{pmatrix} 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 0.5 & 0.5 & 0.5 \end{pmatrix}$$

*Consider state s_2: $fl^t(s_2, s') = 1$ for $s' \in \{s_0, s_1\}$ and $d^t(s_2) = 4$. The transition and reward functions are computed using (1) and (2). For $a = s_2$ and $s' \in \{s_0, s_1\}$, $\mathcal{T}^t(s_2, a, s', \boldsymbol{d}^t) = 1/4$ by **C2**. For $a = s_2$ and $s' = s_2$, $\mathcal{T}^t(s_2, a, s', \boldsymbol{d}^t) = 1 - 2/4 = 1/2$ by **C3**. Rest of the transition and reward function values can be computed similarly.*

Unlike in other game, in this model number of players playing the game at different time steps are different. At any time step t, it is bounded by: $\sum_s \sum_{k=t-\delta}^{k=t} I^k(s)$. Where $I^k(s) = \sum_i I_i^k(s)$ and $I_i^k(s)$ is indicator variable for if agent i is starting its operation at time k

in state s. Thus,

$$\text{Active players count at time t} \leq \sum_{s} \sum_{k=t-\delta}^{k=t} I^k(s) \tag{3}$$

In this game $\mathbb{G}$, there are total N players, but all of them (may) not be active (playing game) at every time step. Total time horizon of the game is T, and any player can only play the game for maximum of α time steps. Each time agent play the game will get some reward and will transit to a new state according to its transition function.

5 SOLUTION APPROACH FOR SRT

To solve the SRT model where we want to optimize the distribution of agents at each epoch as well as their policies, we convert SRT to a k-symmetric game. In this game k represents the type of agents and each type shares the same initial distribution. We provide two techniques to solve SRT with dynamic populations (different number of drivers at different decision epochs).

5.1 Fictitious Play-Based Planning For Dynamic Population (FP-P-DP)

In order to achieve the above objective we design a single fictitious play (FP) process (Algorithm 1) that optimizes both (a) policy of every agent; and (b) initial distribution of every agent; as a single equilibrium computation process. Algorithm 1) provides this fictitious play approach, where each agent starts with a random policy as well as a random initial distribution. We then find the best response assuming all but one agent are following their current policy and initial distribution. It should be noted that best response for any agent is a set of "policy and its initial distribution". Using the best response, we compute average policy and average initial distribution and this process is repeated until policy and initial distribution converges.

Best Response Computation: Algorithm 2 provides the best response computation method employed within the FP-P-DP process. To compute the best response for an agent, Algorithm 2 uses the average policy and average initial distribution and simulates all but one agent. Using this simulated information it computes the best response for an agent which contains both a policy and an initial distribution. Since best response computation is not specific to any agent, we don't use agent index in policy and initial distribution. In the best response computation, to handle dynamic population setting, we do the following: (a) Introduce a sink node; (b) Treat initial distribution as a optimization variable. A sink state, s_{sink} is an absorbing state that represents the state of taxis when they are not active.

In a normal MDP, initial distribution (α) of agents is known. However, in the best response computation we are interested in finding the best initial distribution and therefore, we treat the initial distribution (α) as a variable. s.t, $\sum_{t,s} \alpha^t(s) = 1$ Given maximum number of time steps (δ) any agent can be active in the game, the flow of agents (x) at some time steps t+δ should not exceed $[\alpha^{t+1} + \alpha^{t+2} + ... + \alpha^{t+\delta}]$ i.e, $\sum_{s,a} x^{t+\delta}(s,a) \leq \sum_s \sum_{j=1}^{\delta} \alpha^{t+j}(s)$ By utilizing the algorithm 2 as its best response dynamics, FP process aims to find an approximate equilibrium solution for all agents.

Algorithm 1 Finding ϵ-Equilibrium

$\pi_0 = getRandomPolicy()$
$\alpha_0 = getRandomInitialDistribution()$
For empirical,
$\alpha_0 = getRandomInitialDistribution\_closeTohistoricalData()$
$x_0 = getAgentFlow(\pi_0, \alpha_0, N)$
i = 1
converged = false
while converged = false **do**
  $probability = getAgentCountProbability(\pi_0, \alpha_0, N-1)$
  $< x_1, \alpha_1 >= solveMDP(probability)$
  $x_1^t(s,a) = (x_0^t(s,a) \cdot i + x_1^t(s,a))/(i+1), \quad \forall t, s, a$
  $\alpha_1^t(s) = (\alpha_0^t(s) \cdot i + \alpha_1^t(s))/(i+1), \quad \forall t, s$
  ${\pi_1}^t(s,a) = x_1^t(s,a)/\sum_a x_1^t(s,a), \quad \forall t, s, a$
  if $\mid \pi_0 - \pi_1 \mid \leq \epsilon_\phi$ & $\mid \alpha_0 - \alpha_1 \mid \leq \epsilon_\alpha$ **then**
    converged = TRUE
  else
    $\pi_0 = \pi_1$
    $\alpha_0 = \alpha_1$
  end if
  $i + 1$
end while

Algorithm 2 solveMDP(p)

$$\max \sum_{t,s,a,i} x^t(s,a) \cdot p_i^t(s) \cdot R^t(s,a,i+1) \tag{4}$$

s.t (5)

$$\sum_a x^t(s,a) - \sum_{s', a \neq a_{sink}} x^{t-1}(s',a) \cdot p_i^t(s') \cdot \phi_{i+1}^t(s',a,s) = \alpha^t(s) \quad \forall s, t \tag{6}$$

$$\sum_{t,s} \alpha^t(s) = 1 \tag{7}$$

$$\sum_{s,a} x^{t+n}(s,a) \leq \sum_s \sum_{j=1}^{n} \alpha^{t+j}(s), \quad \forall s \tag{8}$$

5.2 Fictitious Play-Based Learning For Dynamic Population (FP-L-DP):

Due to the presence of a large number of agents, solving FP-P-DP is very time consuming, which takes from few hours to few days on different data sets (on different days of taxi data of a large Asian city). To provide a high quality solution in a reasonable amount of time, we provide a deep learning based fictitious play method (Algorithm 3), which achieves comparable results in significantly smaller training time (details in the experimental section).

Naively applying deep learning technique on such large scale can create many problems. To achieve a good solution and at the same time keep the entire training process simple and efficient, we do the following: (a) we create a neural network (basically a neural network to learn an average value) that takes policy as input and provides average best response as output. (b) we assume greedy

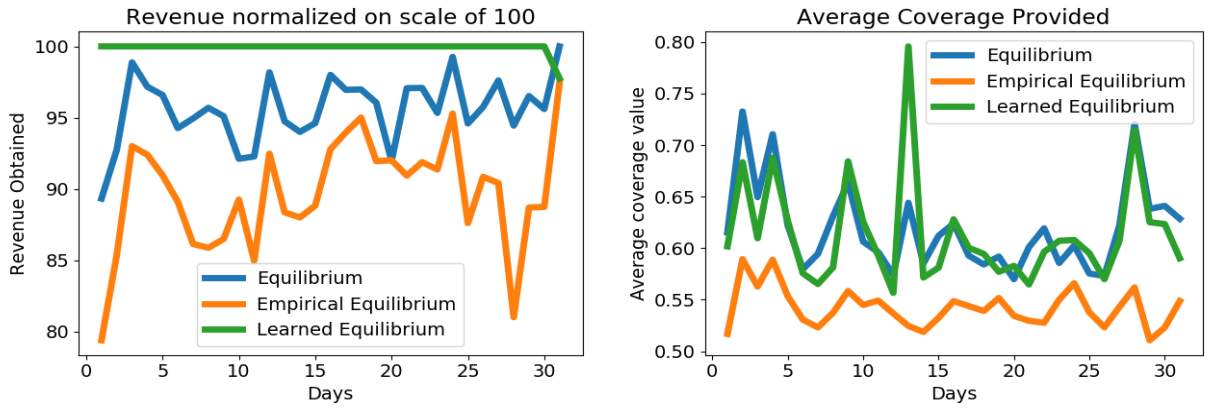

Figure 3: (a) Comparison of equilibrium, empirical equilibrium and learned equilibrium solution quality (averaged over 2000 simulation instances); (b) Average coverage provided by equilibrium, empirical equilibrium and learned equilibrium solutions.

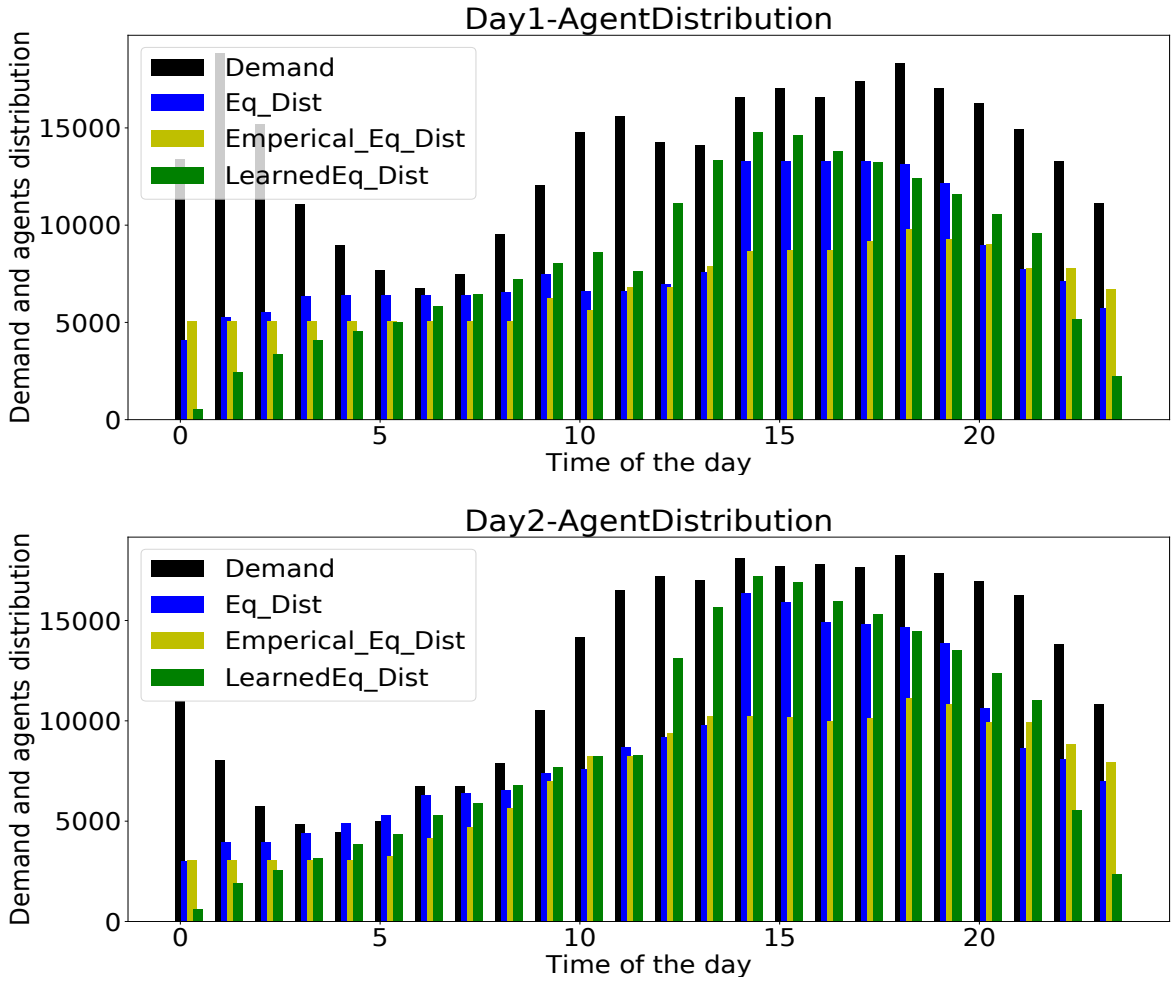

Figure 4: Distribution on agents on day 1 and 2

policy for every agent; (c) we simulate all agents (assuming greedy policy) and based on average reward obtained in different starting states, we compute the best response (starting state providing the highest average reward will be considered as best response); (d) once we have the best response from simulator we store it in the best response buffer; (e) and as the final step we train our neural network to learn a average value over stored experiences (i.e, learn the average best response); To learn the average best response we take inspiration from maximum likelihood estimation and we achieve it in our neural network by optimizing the log loss value over (sampled) stored experiences.

Even with the greedy policy assumption, this approach faces the following issues:

- Policy space is too large: Even though we assume greedy policy, its scale is large due to large state/action space and longer time horizon.
- Best response computation is time consuming: Entire training relies on getting the best response so it is very important to compute the best response in a faster and efficient way.
- Biased samples for training: Efficient training required balances samples, but due to operating hour constant, biased samples will be generated, that will lead to inefficient results.

To handle above 3 issues we do the following:

Map policy to a lower dimension: The neural network takes policy as input, since scale of the problem is large, so does the policy of agents. In the taxi problem setting, policy have $24x10^4$ entries [24 time steps, 100 states, 100 actions]. Providing such a large input to neural network makes is very complex in terms of training, and it requires a very long training cycle before it can produce meaningful outputs. So we reduce the dimension of policy before we pass it as input to neural network, In order to do so we take advantage of "principal component analysis" method.

The main idea of principal component analysis (PCA) is to reduce the dimension of a data set consisting of many variables correlated with each other, while retaining the variation present in the dataset, in simpler words its a method of summarizing data.

Using PCA we reduce the policy dimension to 2400, which is same dimension as dimension of initial distribution that we need to learn (2400 dimension: 24 time steps, 100 states).

Getting best response in an efficient way: In order to train our average initial distribution network, we need best response experiences. We can train a deep Q network to achieve this, this requires a separate training for such Q network. Given the large problem setting this Q network will require longer training time before it can be used for generating meaningful best response. To keep the process simple and fast, we directly extract best response from simulator (instead of using the simulated data to train a Q network to predict the best response), we directly use best response information from simulator to train the average initial distribution network.

Handling bias in best response data: In this problem of interest, any agent can only operate for certain hours, say 10 hrs in case of taxi optimization problem. Therefore best starting state for any taxi will not be towards the end of horizon, where it gets to operate for less than 10 hrs. Therefore almost all the experiences collected from simulations, will only have data in the range of first 15 hrs (assuming 24 hrs of operation). This create bias experience buffer which would lead to poor solution quality. Therefore (to not fine tune too much) we create 2 experience buffer, one stores the experiences generated in the first half of operation (i.e, first 12 hrs) and second experience buffer to store the remaining experiences. In the training process we sample from both buffers equally and train the neural network with it.

Algorithm 3 Symmetric fictitious play: for initial distribution

1: Initialize initial distribution network(θ^α),
2: **while** Not Converged **do**
3: policy = greedy policy
4: Simulate all agents
5: Compute average reward for each possible starting state
6: Store best response (starting state) experience in buffer: M_{first} if its in first half of operation, otherwise store in M_{second}
7: Sample from M_{first}, train policy network: $\mathcal{L}(\theta^\alpha) = \mathbb{E}_{(t,s)}\Big[-log(\alpha(t,s|\theta^\alpha)\Big]$
8: Sample from M_{second}, train policy network: $\mathcal{L}(\theta^\alpha) = \mathbb{E}_{(t,s)}\Big[-log(\alpha(t,s|\theta^\alpha)\Big]$
9: **end while**

6 EXPERIMENTS

In this section we provide experimental results on the real world taxi data set provided by a taxi company that operates in a large Asian city. Data set contains approximately 20000 taxis operating every day. We compared the results for 31 days of data (Jan 2017). We compared performance of different policies, i.e., "Equilibrium policy" – obtained by the FP-P-DP approach, "Empirical Equilibrium policy:" obtained by the FP-P-DP approach but where agents have to follow the historical distribution and "Learned Equilibrium" – obtained by the FP-L-DP approach

6.1 Evaluation against historical data based baseline

We simulated the policy generated by FP-P-DP and compared it against the historical policy (referred to as the Empirical equilibrium solution). For fair comparison:

- We computed the distribution of agents "as close as possible" to historical distribution using Algorithm 4.
- We find the equilibrium policy using same fictitious play method, but in best response MDP (Algorithm 5) agent tries to find its best response where its initial distribution at every time step is fixed same as historical distribution. It is allowed to change states though, i.e it can start from different location but must start at the same time as historical distribution.
- We computed this equilibrium policy under same maximum operating hour constraint [which is 10 hr in our experiments].

Experimental setup is as follows: state (zone for taxis) size is 100, number of possible actions (to move between zones) is 100,

maximum operating hour is 10 hrs, we extracted number of taxi from historical data (we removed taxis operating for 16 hrs or more), time horizon is 24 i.e we are computing policy for every hour.

Once we compute the equilibrium solutions generated by FP-P-DP as well as historical distribution based empirical equilibrium solution, we compared them as follows:

- We simulate both policies every 60 seconds, where taxis can serves demand generated in last 60 seconds.
- Since policy generated is on an hourly basis, we reuse the policy in simulation for every decision in that hour.

On all 31 days of data Figure(3(a)), FP-P-DP provided better results as compared to historical equilibrium solution. These results are averaged over 2000 simulations. Results are normalized on a scale of 100, with the best approach getting 100.

Algorithm 4 *getDistribution*(e^t: Distribution from historical data)

$$\min \sum_{t} |e^t - \beta_t| \tag{9}$$

s.t

$$\beta_t = \sum_{t'=0}^{n-1} \alpha_{compute}^{t-t'}.$$

Algorithm 5 *solveMDP_Empirical*(p)

$$\max \sum_{t,s,a,i} x^t(s,a) \cdot p_i^t(s) \cdot R^t(s,a,i+1), \tag{10}$$

s.t

$$\sum_{a} x^t(s,a) - \sum_{s',a \neq a_{sink}} x^{t-1}(s',a) \cdot p_i^t(s') \cdot \phi_{i+1}^t(s',a,s), = \alpha^t(s) \qquad \forall s,t, \tag{11}$$

$$\sum_{t,s} \alpha^t(s) = 1, \tag{12}$$

$$\sum_{s,a} x^{t+n}(s,a) \leq \sum_{s} \sum_{j=1}^{n} \alpha^{t+j}(s), \qquad \forall s,$$

$$\sum_{s} \alpha^t(s) = \alpha_{compute}^t. \tag{13}$$

To understand the availability and coverage that both solutions are providing, we also analyzed how taxis are operating through out the day under both solutions (once policies along with their distributions are simulated). In Figure 4 we provide the comparison for day 1 and 2 (other days had similar results). Here we observe that equilibrium solutions obtained by FP-P-DP and FP-L-DP are providing better availability and coverage. Similar results are observed on other days as well. Figure(3(b))) provides average coverage provided by all agents under different methods, it suggests that equilibrium solutions obtained by our methods provide much better coverage than the empirical equilibrium on all 31 days of data. Average coverage shown is averaged over all time and states.

6.2 Evaluation of FP-L-DP approach:

Parameter setting: We used the following parameters in the neural network. Learning Rate = 10^{-3}, exploration parameter, $\epsilon = 0.9$ and reduced it by a factor of 0.9 after every 50 iterations. We simulated in a manner similar to the FP-P-DP method. Our average initial distribution network has 2 hidden layers with 2400 nodes in each layer. We used layer norm after each layer. We used relu activation function in hidden layers. Input size is 2400. We used batch size of 12. We trained it for 1000 iteration which took approximately 1 hr. We compared our deep learning based approach (FP-L-DP) with equilibrium and empirical equilibrium, solution quality wise as well as average coverage provided by this method.

We are able to show that FP-L-DP method (within an hour of training) outperformed FP-P-DP method on 30 out 31 days of data, and outperformed empirical equilibrium on all 31 days of data. Using FP-L-DP we can achieve better solution in a significantly lower amount of time. Overall, FP-P-DP provides up to 16% improvement in revenue over empirical equilibrium. The learning based approach FP-L-DP further improves the performance and achieves up to 10% more revenue than FP-P-DP approach.

7 CONCLUSION AND FUTURE WORK

In many real world problems agent population changes over due to agents' strategic choices. We introduce a model (SRT) to directly capture this dynamics of agent population change. Our model also captures important congestion game features that are commonly seen in transport gig economy. We introduced two fictitious play-based approaches (FP-P-DP and FP-L-DP) to solve this model. We provided experimental results on a real world taxi data from a large Asian city and demonstrated the effectiveness of our approach and the importance of using equilibrium policies. In our experimental results, we demonstrate that our planning based approach (FP-P-DP) provides up to 16% improvement in revenue over existing method. The learning based approach (FP-L-DP) further improves the performance and achieves up to 10% more revenue than the planning based approach.

As the final remarks, below we list two streams of future work under consideration: 1) Improve the efficiency of the planning-based approach. 2) The implementation mechanism for fleet operators to encourage drivers to follow the desired entrance distribution. It could be in the form of incentives or information disclosure.

In practice, we envision the "time of entry" decision to be delivered in two potential ways: (a) Via a recommendation system. As demonstrated by past researchers ([4]), recommendation system could be implemented for a general taxi fleet and significantly improve individual taxi driver's productivity. Similarly for our case, we could recommend the "time of entry" based on agent's stated preference (e.g., day-shift or night-shift). To encourage drivers to follow the recommendation, we could even nudge them by displaying the potential drop in revenue if they choose not to follow the recommendation (i.e., the cost associated with the deviation from the equilibrium). (b) Via incentive. For a fleet operator who aims to maximize its service level (number of passengers the fleet can serve), it can offer incentives to achieve the recommended distribution on the "time of entry".

REFERENCES

[1] Pierre Bernhard and Marc Deschamps. 2016. Dynamic equilibrium in games with randomly arriving players. (2016).

[2] Pierre Bernhard and Marc Deschamps. 2017. On dynamic games with randomly arriving players. *Dynamic Games and Applications* 7, 3 (2017), 360–385.

[3] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research* 27, 4 (2002), 819–840.

[4] Shih-Fen Cheng, Shashi Shekhar Jha, and Rishikeshan Rajendram. 2018. Taxis strike back: A field trial of the driver guidance system. (2018).

[5] Jiarui Gan and Bo An. 2017. Game-theoretic considerations for optimizing taxi system efficiency. *IEEE Intelligent Systems* 32, 3 (2017), 46–52.

[6] Yong Gao, Dan Jiang, and Yan Xu. 2018. Optimize taxi driving strategies based on reinforcement learning. *International Journal of Geographical Information Science* 32, 8 (2018), 1677–1696.

[7] Yan Huang and Jason W Powell. 2012. Detecting regions of disequilibrium in taxi services under uncertainty. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 139–148.

[8] Ioannis Kordonis and George P Papavassilopoulos. 2015. LQ Nash games with random entrance: an infinite horizon major player and minor players of finite horizons. *IEEE Trans. Automat. Control* 60, 6 (2015), 1486–1500.

[9] Dan Levin and Emre Ozdenoren. 2004. Auctions with uncertain numbers of bidders. *Journal of Economic Theory* 118, 2 (2004), 229–251.

[10] J-F Mertens and Abraham Neyman. 1981. Stochastic games. *International Journal of Game Theory* 10, 2 (1981), 53–66.

[11] Luis Moreira-Matias, João Gama, Michel Ferreira, and Luís Damas. 2012. A predictive model for the passenger demand on a taxi network. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 1014–1019.

[12] Roger B Myerson. 1998. Extended Poisson games and the Condorcet jury theorem. *Games and Economic Behavior* 25, 1 (1998), 111–131.

[13] Roger B Myerson. 1998. Population uncertainty and Poisson games. *International Journal of Game Theory* 27, 3 (1998), 375–392.

[14] Abraham Neyman, Sylvain Sorin, and S Sorin. 2003. *Stochastic games and applications*. Vol. 570. Springer Science & Business Media.

[15] Jason W Powell, Yan Huang, Favyen Bastani, and Minhe Ji. 2011. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *International Symposium on Spatial and Temporal Databases*. Springer, 242–260.

[16] Meng Qu, Hengshu Zhu, Junming Liu, Guannan Liu, and Hui Xiong. 2014. A cost-effective recommender system for taxi drivers. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 45–54.

[17] Huigui Rong, Xun Zhou, Chang Yang, Zubair Shafiq, and Alex Liu. 2016. The rich and the poor: A Markov decision process approach to optimizing taxi driver revenue efficiency. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2329–2334.

[18] Kiam Tian Seow, Nam Hai Dang, and Der-Horng Lee. 2009. A collaborative multiagent taxi-dispatch system. *IEEE Transactions on Automation science and engineering* 7, 3 (2009), 607–616.

[19] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39, 10 (1953), 1095–1100.

[20] Pradeep Varakantham, Shih-Fen Cheng, Geoff Gordon, and Asrar Ahmed. 2012. Decision support for agent populations in uncertain and congested environments. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[21] Tanvi Verma, Pradeep Varakantham, Sarit Kraus, and Hoong Chuin Lau. 2017. Augmenting decisions of taxi drivers through reinforcement learning for improving revenues. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

[22] Xinlian Yu, Song Gao, Xianbiao Hu, and Hyoshin Park. 2019. A Markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological* 121 (2019), 114–134.

[23] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. 2012. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on knowledge and data engineering* 25, 10 (2012), 2390–2403.

[24] Kai Zhang, Zhiyong Feng, Shizhan Chen, Keman Huang, and Guiling Wang. 2016. A framework for passengers demand prediction and recommendation. In *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 340–347.

[25] Xun Zhou, Huigui Rong, Chang Yang, Qun Zhang, Amin Vahedian Khezerlou, Hui Zheng, M Zubair Shafiq, and Alex X Liu. 2018. Optimizing Taxi Driver Profit Efficiency: A Spatial Network-based Markov Decision Process Approach. *IEEE Transactions on Big Data* (2018).