Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Decision Support for Agent Populations in Uncertain and Congested Environments

Pradeep Varakantham[†], Shih-Fen Cheng[†], Geoff Gordon[‡],Asrar Ahmed[†]
[†]School of Information Systems, Singapore Management University, Singapore 178902
[‡]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[†]{pradeepv,sfcheng,masrara}@smu.edu.sg [‡]ggordon@cs.cmu.edu

Abstract

This research is motivated by large scale problems in urban transportation and labor mobility where there is congestion for resources and uncertainty in movement. In such domains, even though the individual agents do not have an identity of their own and do not explicitly interact with other agents, they effect other agents. While there has been much research in handling such implicit effects, it has primarily assumed deterministic movements of agents. We address the issue of decision support for individual agents that are identical and have involuntary movements in dynamic environments. For instance, in a taxi fleet serving a city, when a taxi is hired by a customer, its movements are uncontrolled and depend on (a) the customers requirement; and (b) the location of other taxis in the fleet. Towards addressing decision support in such problems, we make two key contributions: (a) A framework to represent the decision problem for selfish individuals in a dynamic population, where there is transitional uncertainty (involuntary movements); and (b) Two techniques (Fictitious Play for Symmetric Agent Populations, FP-SAP and Softmax based Flow Update, SMFU) that converge to equilibrium solutions. We show that our techniques (apart from providing equilibrium strategies) outperform "driver" strategies with respect to overall availability of taxis and the revenue obtained by the taxi drivers. We demonstrate this on a real world data set with 8,000 taxis and 83 zones (representing the entire area of Singapore).

Introduction

Research on understanding and controlling dynamic and large scale flow of agents (e.g., humans, industries, vehicles) between different states spans various domains such as urban transportation (Wardrop 1952; Yang and Wong 1998) (e.g., movement of vehicles between different regions of an area), industry dynamics (Weintraub, Benkard, and Roy 2006) (e.g., strategizing on marketing investments by different companies selling the same product), labor mobility between cities (Harris and Todaro 1970) (e.g., analyzing individuals search for jobs in new locations), advertising and others (Alpern and J.Reyniers 2002). The main challenge in these problems is accounting for the implicit interaction that exists between agents. For example, vehicles trying to get on the same road are implicitly competing. These difficulties in analyzing/controlling road traffic are dealt with by the concept of *user equilibrium*, which was first proposed by Wardrop (Wardrop 1952). Unfortunately, the solution concept of user equilibrium does not apply to our analysis directly due to the presence of involuntary movements.

We are focused on similar problems, except in cases where agents can have involuntary (or forced) movements. One key problem of interest is with respect to the operation of a taxi fleet. Taxi drivers are subject to both voluntary (at driver's own decision) and involuntary (when customers board taxis) movements. Different regions might have different demands for taxis (both in terms of numbers and revenues) and due to this an implicit competition exists between taxis. The goal here is to improve the operational efficiency of the fleet and the revenues obtained by taxi drivers. Similar problems exist in analyzing industry dynamics (where different companies strategize to maintain their competitive advantage)and labor mobility (where individuals reason about their movement to different geographical regions). Unfortunately, due to the presence of involuntary movements and the scale of problems involved, the user equilibrium solution concept is not applicable.

To account for large-scale problems (e.g., 8,000 taxis in the taxi problem) encountered in domains of interest, we provide techniques that exploit symmetry or identical nature of agents in our problems. The first set of techniques is based on an iterative update of best response in a fashion similar to Fictitious Play for normal form games. Secondly, we provide a suite of greedy randomization techniques which rely only on immediate rewards.

The primary example we will base our discussion on throughout the paper is the analysis of a taxi fleet. In most metropolitan areas, taxi is an important class of public transportation (e.g., in Singapore, taxi accounted for 17% of public transportation in year 2007/08). However, even after decades of improvement, the operational efficiency of a typical taxi fleet is still not very satisfactory (based on our analysis, a taxi on average spends more than 50% of time idling or roaming to find customers). From the policy maker's point of view, understanding how to improve the efficiency of the taxi service is of vital importance, since large amount of taxis constantly roam the city area, and even an improvement of few percentage points would mean savings of millions of man hours per year (for both drivers and customers). The incentive structure of taxi drivers in most cases allows taxi drivers to pay a fix rent and keep the rest of the revenue. This implies that taxi drivers are typical selfish agents that would

only react to incentive and cannot be controlled centrally. In DDAP model of the taxi problem, we employ an objective function that optimizes the taxi driver's revenue directly and the operational efficiency of the fleet indirectly (through its transition function).

We were able to illustrate that our equilibrium approaches and one of the greedy approaches provide solutions that improved significantly over real world taxi driver policies. This improvement was with respect to both the (a) operational inefficiency, characterized as starvation in our results; and (b) the minimum revenue obtained by any taxi driver and the average revenue of all the taxi drivers. These results emphasize the utility of our equilibrium approaches and greedy techniques in solving DDAP problems.

Motivating Domain: Taxi Fleet Optimization

In a M zone region to be covered by a taxi fleet, $\mathcal{P}$, the goal is to provide decision support to individual taxis on their next zone so that the performance (with respect to driver revenues and availability of taxis) is improved. Movement of taxis between zones is controlled by the underlying customer flow (people moving between zones) between zones and the number of taxis in the current zone.

More specifically, for two zones i and j at time t, there is an (average) underlying customer flow $fl^t(i,j)$. For zone i, if there are more number of customers in the zone ($\sum_j fl^t(i,j)$) than the number of taxis in that zone (d_i), then all taxis will get customers and get sent to a zone depending on the the relative ratio of $\frac{fl^t(i,j)}{\sum_j fl^t(i,j)}$. In other words, their zone transition will follow the underlying probability; also, all of them will receive revenue $Re^t(i,j)$ and incur cost $Co^t(i,j)$. Similarly, there is an underlying probability if there are not enough taxis in a zone for all customers.

Given the movement probability, the goal is to increase revenues for individual drivers and the availability of taxis across all zones.

Model: DDAP

We now describe a general representation called the Decentralized Decision model for Agent Populations (DDAP) as a model to represent the decision problems for individual agents in a population operating in dynamic domains (like the taxi fleet optimization problem, labor mobility and strategizing in industry dynamics). It is represented using the tuple: $\langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \phi, \mathcal{R}, H, d^0 \rangle$, where $\mathcal{P}$ represents the agent population. $\mathcal{S}$ corresponds to the set of states encountered by every agent in the population. $\mathcal{A}$ is the set of actions executed by each agent. This would be equivalent to a repeated congestion game with transition uncertainty.

Before defining the transition and the reward function, we first define the set of state distributions,
$D = \{d | d = \langle d_1, d_2, \cdots, d_{|\mathcal{S}|} \rangle, \sum_{s \in \mathcal{S}} d_s = |\mathcal{P}|\}$, where d_s represents the number of agents in state s. ϕ models the involuntary movements of agents and more specifically, $\phi_d^t(s,a,s')$ represents the probability that an agent in state $s (\in \mathcal{S})$ after taking action $a (\in \mathcal{A})$ would transition to state s', when the current state distribution is d and time is t.

$\mathcal{R}_d^t(s,a,s')$ is the reward obtained by an agent when in state s, taking action a and moving to state s' when the distribution of agents is d at time t. H is the time horizon for the decision process. d^0 represents the starting distribution of agent states. Expected value for an agent starting from state s with a starting distribution, d^t is defined as

$$V^t(s,d^t) = \max_{\pi^t} \{ \sum_a \pi_{s,a}^t \sum_{s'} \phi_{d^t}(s,a,s') \\ [R_{d^t}(s,a,s') + V^{t+1}(s', d^{t+1})] \} \quad (1)$$

The objective is to compute a policy,$\{\pi^t\}_1^H$ for each agent, so that no agent has an incentive to deviate with respect to expected value over the horizon.

DDAP for the Taxi Problem

$\mathcal{P}$ represents the set of taxis in the fleet. $\mathcal{S}$ refers to the set of all zones in which a taxi could be present. $\mathcal{A}$ refers to the set of next zones ($\mathcal{A} \equiv \mathcal{S}$). Equation 2 provides the expression for computing the transition probability between states. The intuitions behind the expression are three fold as formalized by the conditions **C1**, **C2** and **C3**: (a) The taxi is always hired if the flow of customers out of a zone is higher than the number of taxis currently in that zone. Hence, the probability of moving to a specific zone s' is directly proportional to the customer flow between s and s'. (b) When the intended zone (the action) is not the destination zone, then it implies the taxi was hired. (c) When the intended zone is same as the destination zone, then a part of the probability is due to the taxi getting hired and a part of the probability is due to the intended movement of the taxi.
C1: if $\sum_{\hat{s}} fl^t(s,\hat{s}) \geq d_s$
C2: if $a \neq s'$, $\sum_{\hat{s}} fl^t(s,\hat{s}) < d_s$
C3: if $a = s'$, $\sum_{\hat{s}} fl^t(s,\hat{s}) < d_s$

$$\phi_d^t(s,a,s') = \left\{ \begin{array}{ll} \frac{fl^t(s,s')}{\sum_{\hat{s}} fl^t(s,\hat{s})} & \textbf{C1} \\ \frac{fl^t(s,s')}{d_s} & \textbf{C2} \\ 1 - \frac{\sum_{\hat{s} \neq s'} fl^t(s,\hat{s})}{d_s} & \textbf{C3} \end{array} \right\} \quad (2)$$

$\mathcal{R}$ is the revenue obtained by taxi drivers while accounting for the cost of driving between zones. $\mathcal{R}^t(s,a,d) =$

$$\left\{ \begin{array}{ll} \sum_{s'} \phi_d^t(s,a,s') \cdot (Re^t(s,s') - Co^t(s,s')) & \textbf{C1} \\ \sum_{s'} \phi_d^t(s,a,s') \cdot (Re^t(s,s') - Co^t(s,s')) & \textbf{C2} \\ \frac{fl^t(s,s')}{d(s)} \cdot Re^t(s,s') - \phi_d^t(s,a,s') \cdot Co^t(s,s')) & \textbf{C3} \end{array} \right\} \quad (3)$$

H corresponds to the number of discrete intervals of a day that are being considered. Finally D^0 is obtained by aggregating taxi distributions at the starting time in the dataset.

In solving a DDAP, each agent maximizes expected revenue for the individual taxi drivers. However, the transition function depends on the number of taxis in the zone and maximizing expected revenue implies minimizing starvation as well. In this problem domain, both the welfare metrics (revenue and starvation) are optimized at once, however, in other domains there could be multiple objectives that are not in alignment and multi-objective reasoning might be required.

Solving the DDAP Model

We now describe two techniques for solving the DDAP model to obtain a policy for each agent. Our approach is to

convert DDAP to a symmetric game and compute a symmetric equilibrium. While this may or may not be the ideal Nash equilibrium (in terms of social welfare), we show that it provides better performance than existing human strategies and benchmarks.

The policy for an agent in DDAP can be formally defined as $\pi = \langle \pi^0, \pi^1, \cdots, \pi^{H-1} \rangle$, where $\pi^t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{[0,1]}$ is a mapping from states and decision epochs to a probability distribution over actions. The probability of executing an action a at time t according to policy π is given by $\pi^t(s, a)$. This policy can be both randomized (non-zero probabilities for multiple actions) and non-stationary (different action probabilities for different decision epochs).

An agent in DDAP can be considered to have initial state distribution where an agent starts in s is $\frac{d^0(s)}{\sum_{s'} d^0(s')}$). Then, all agents have identical strategy space, with no type, and have utilities depending only on π and joint strategy from other agents. An implication of this is that the DDAP can be represented as a symmetric repeated game.

Given agent state distribution over the horizon, $d = \langle d^0, d^1, \ldots, d^{H-1} \rangle$, the single agent's optimal planning problem can be modeled as a Markov decision process (MDP), and the standard dual LP formulation (Puterman 1994) for computing a policy that maximizes the expected reward is:

$$\begin{aligned} &\textsc{SolveMDP}(ddap, d): \\ &\max \sum_{t,s,a} \mathcal{R}^t(s,a,d) \cdot x^t_{s,a} \\ &\text{s.t.} \quad \sum_a x^t_{s',a} - \gamma \sum_{s,a} x^t_{s,a} \cdot \phi^t_d(s,a,s') = \delta^t(s'), \forall t \\ &\qquad x^t_{s,a} \geq 0, \forall s, a, t; \;\; 0 \leq \gamma < 1, \end{aligned} \tag{4}$$

where $x^t_{s,a}$ represents the number of times action a has been chosen in time t and state s, and $\delta^t(s) = \frac{d^t(s)}{\sum_{s'} d^t(s')}$ is the likelihood that agents are in state s in time t. The agent policy can then be obtained by normalizing $\{x^t_{s,a}\}$:

$$\pi = \{\pi^t(s,a) : \pi^t(s,a) = \frac{x^t_{s,a}}{\sum_{a'} x^t_{s,a'}}, \forall t, s\}.$$

Similarly, given d^0 and π, the agent state distribution over the whole horizon can be computed as follows:

$$\begin{aligned} &\textsc{GetDist}(ddap, \pi): \\ &\delta^0(s) \leftarrow \frac{d^0(s)}{|\mathcal{P}|}, \\ &\delta^{t+1}(s) \leftarrow \sum_{s' \in \mathcal{S}} \delta^t(s') \sum_{a \in \mathcal{A}} \pi^t(s', a) \phi^t_{d^t}(s', a, s), \; t < H-1, \\ &d^{t+1}(s) \leftarrow \delta^{t+1}(s)|\mathcal{P}|, \; \forall t. \end{aligned}$$

Given a particular agent distribution assumption d, all agents will generate identical best responses, which can be computed by $\textsc{SolveMDP}(ddap, d)$. This suggests an intuitive iterative procedure for computing the equilibrium:

1. Generate an arbitrary initial policy, π_0. Assume that all agents adopt π_0, and compute the resulting agent state distribution $d_0 \leftarrow \textsc{GetDist}(ddap, \pi_0)$.
2. Let current iteration be $i+1$. Assume that the agent state distribution is d_i, solve $\textsc{SolveMDP}(ddap, d_i)$ for the best response π_{i+1}. The new agent state distribution by adopting π_{i+1} can be computed by $d_{i+1} \leftarrow \textsc{GetDist}(ddap, \pi_{i+1})$.
3. If $\pi_i = \pi_{i+1}$, the process converges, stop. Otherwise, $i \leftarrow i+1$, repeat the previous step.

If the above procedure converges, we know that the obtained π will be a symmetric equilibrium, since π is the best response onto itself.

In game-theoretic terms, the above procedures are what researchers called *best response dynamics*; since an agent repeatedly computes best response assuming all other agents play the strategies in the previous iteration. Although the best response dynamics is simple and intuitive, the fact that it disregards all past plays makes it vulnerable to non-convergence or cyclic plays for most problems. In the next two subsections, we describe two techniques that are inspired from an iterative procedure that is well studied and commonly used for learning in games called the fictitious play (FP) algorithm, which was introduced by Brown (Brown 1951) and Robinson (Robinson 1951).

Fictitious Play for Symmetric Agent Populations (FP-SAP)

Algorithm 1 FP-SAP($ddap$)

$\pi_0 \leftarrow \textsc{GetRandomPolicy}()$
$i \leftarrow 0$
repeat
  $d_i \leftarrow \textsc{GetDist}(ddap, \pi_i)$
  $\{\tilde{x}^t_{s,a}\} \leftarrow \textsc{SolveMDP}(ddap, d_i)$
  $x^t_{s,a} \leftarrow \frac{(i \cdot x^t_{s,a} + \tilde{x}^t_{s,a})}{i+1}, \forall\, s, a, t$
  $\pi^t_i(s,a) \leftarrow \frac{x^t_{s,a}}{\sum_a x^t_{s,a}}, \forall\, s, a, t$
  $i \leftarrow i+1$
until $\pi_i = \pi_{i-1}$

The fictitious play (FP) algorithm (Brown 1951) is one of the oldest learning algorithm for computing Nash equilibrium in games. The key idea in FP algorithm is that it repeatedly computes best response by assuming that other agents are to play according to their historical distributions (i.e., other agents will pick one of their past plays uniformly). The FP algorithm in general does not converge, however, for certain classes of games (most notably, zero-sum games, identical interest games and potential games (Monderer and Shapley 1996)), it does converge. A key drawback of FP algorithm has been their limited applicability to large scale problems (with lots of agents).

Since DDAPs are used to represent large agent populations (e.g., for modeling domains like taxi operations, labor mobility, or crowded theme park), a key result desired of any algorithm used to solve DDAPs is its scalability with respect to number of agents. The implementation of a DDAP-specific FP algorithm (which we call it FP-SAP, the Fictitious Play for Symmetric Agent Populations) is illustrated in Algorithm 1. There are two distinct features in FP-SAP

which allow for scalability to large agent populations:
(1) The strategy profile to be searched is symmetric (all agents are identical and the game is symmetric), thus only one best response computation is needed in each iteration.
(2)The best response is computed by solving a MDP, while the impact of all other agent's strategies is limited to the aggregate agent population distribution, d.

To see the significance of the above two features, consider a generic stochastic game, in which agents can have global states and tight transition/reward dependence on each of the other agent actions. This exponentially increases the complexity of the best response computation, since the best response computation for an agent would require accurate tracking of other agent's individual states and actions at each time step (instead of an aggregate distribution).

Proposition 1 *Fictitious play algorithm converges to Nash Equilibrium if* GETDIST() *can be used for updating agent state distributions.*

Proof. We prove this proposition by showing that there exists a potential function, ϕ for a DDAP game with one type. Since a fictitious play algorithm converges to Nash Equilibrium for potential games, therefore SoFA will converge to Nash Equilibrium if there exists a potential function.

The potential function, ϕ for a horizon t DDAP problem is defined as follows:

$$\phi^t(\{\pi_i\}_{i\in\mathcal{P}}) = \sum_k \mathcal{V}_k^t(\pi_k, \{\pi_i\}_{i\neq k, i\in\mathcal{P}}) \tag{5}$$

To show that this is a potential function for a given DDAP, we need to show that for any arbitrary agent k and two of its policies, $\pi_{(k,1)}$ and $\pi_{(k,1)}$:

$$\begin{aligned}&\phi^t(\{\pi_i\}_{i\neq k,i\in\mathcal{P}} \cup \pi_{(k,1)}) - \phi^t(\{\pi_i\}_{i\neq k,i\in\mathcal{P}} \cup \pi_{(k,2)})\\ &= \mathcal{V}_k^t(\pi_{(k,1)}, \{\pi_i\}_{i\neq k,i\in\mathcal{P}}) - \mathcal{V}_k^t(\pi_{(k,2)}, \{\pi_i\}_{i\neq k,i\in\mathcal{P}})\end{aligned} \tag{6}$$

We show that this is a potential function for a DDAP game by using mathematical induction on horizon t.

Base case: t = 0

The value function for an agent is defined as follows:

$$\mathcal{V}_k^0(\pi_{(k,1)}, \{\pi_i\}_{i\neq k,i\in\mathcal{P}}) = \sum_{s,a} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)$$

Since the one step rewards are only dependent on d^0 and it is the same for all agents, the difference in potential functions is equal to difference in value functions for the different policies of agent k.

Therefore, **let us assume that the ϕ is a potential function for horizon $t = m$**, i.e.

$$\begin{aligned}&\phi^m(\{\pi_i\}_{i\neq k,i\in\mathcal{P}} \cup \pi_{(k,1)}) - \phi^m(\{\pi_i\}_{i\neq k,i\in\mathcal{P}} \cup \pi_{(k,2)})\\ &= \mathcal{V}_k^m(\pi_{(k,1)}, \{\pi_i\}_{i\neq k,i\in\mathcal{P}}) - \mathcal{V}_k^m(\pi_{(k,2)}, \{\pi_i\}_{i\neq k,i\in\mathcal{P}})\end{aligned} \tag{7}$$

Now, we will prove that it holds for $t = m + 1$

$$\begin{aligned}&\mathcal{V}_k^{m+1}(\pi_{(k,1)}, \{\pi_i\}_{i\neq k,i\in\mathcal{P}})\\ &= \sum_{s,a} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)\\ &\quad + \sum_{s,a,s'} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\phi_k(s,a,s',d^0)\cdot\\ &\quad \cdot\mathcal{V}_k^m(s',\pi_{(k,1)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}})\end{aligned}$$

From equations used in GETDIST()

$$\begin{aligned}&= \sum_{s,a} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)\\ &\quad + \sum_{s'} p_k^1(s')\cdot\mathcal{V}_k^m(s',\pi_{(k,1)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}})\\ &= \sum_{s,a} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\mathcal{R}_k(s,a,d^0) + \mathcal{V}_k^m(\pi_{(k,1)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}})\end{aligned} \tag{8}$$

Using Equation 8, we have

$$\begin{aligned}&\mathcal{V}_k^{m+1}(\pi_{(k,1)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}}) - \mathcal{V}_k^{m+1}(\pi_{(k,2)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}})\\ &= \sum_{s,a} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)\\ &\quad - \sum_{s,a} p_k^0(s)\cdot\pi_{(k,2)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)\\ &\quad + \mathcal{V}_k^m(\pi_{(k,1)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}}) - \mathcal{V}_k^m(\pi_{(k,2)},\{\pi_i\}_{i\neq k,i\in\mathcal{P}})\end{aligned}$$

From the assumption of Equation 7

$$\begin{aligned}&= \sum_{s,a} p_k^0(s)\cdot\pi_{(k,1)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)\\ &\quad - \sum_{s,a} p_k^0(s)\cdot\pi_{(k,2)}(s,a)\cdot\mathcal{R}_k(s,a,d^0)\\ &\quad + \sum_{\{i\neq k,i\in\mathcal{P}\},s,a} p_i^0(s)\cdot\pi_i(s,a)\cdot\mathcal{R}_i(s,a,d^0)\\ &\quad - \sum_{\{i\neq k,i\in\mathcal{P}\},s,a} p_i^0(s)\cdot\pi_i(s,a)\cdot\mathcal{R}_i(s,a,d^0)\\ &\quad + \phi^m(\{\pi_i\}_{i\neq k,i\in\mathcal{P}}\cup\pi_{(k,1)}) - \phi^m(\{\pi_i\}_{i\neq k,i\in\mathcal{P}}\cup\pi_{(k,2)})\end{aligned}$$

Combining terms using the definition of potential function (Equation 5), we have

$$= \phi^{m+1}(\{\pi_i\}_{i\neq k,i\in\mathcal{P}}\cup\pi_{(k,1)}) - \phi^{m+1}(\{\pi_i\}_{i\neq k,i\in\mathcal{P}}\cup\pi_{(k,2)})$$

Hence proved. ■

Soft-Max-based Flow Update (SMFU)

FP-SAP presents an algorithm for solving DDAPs, where the best response part of the algorithm is scalable. However, there is no guarantee on the number of iterations it will take until convergence to the equilibrium and as we show in our experimental results, it can be very slow. To address this, we introduce our second contribution called the soft-max-based fictitious play. It is a standard result in MDP literature (Puterman 1994) that solving an MDP yields a deterministic policy, i.e. all the flow out of any state is due to only one action. Intuitively, it means that there is a sharp increase or decrease in the aggregate flows of one action after each iteration of FP and hence the aggregate flows can potentially take many iterations to converge.

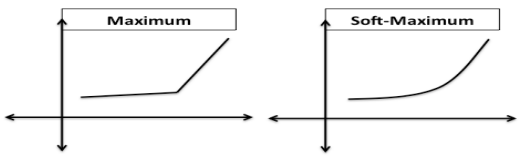

Figure 1: Soft-max and Max of two linear functions

To address this, we employ *soft-max value iteration* (SMVI) introduced by Ziebart *et al.* (Ziebart 2010) for solving MDPs. Soft-max or soft maximum approximates the

hard maximum and is a convex function similar to the hard maximum. The key difference is that while hard maximum has sharp edges, soft maximum is smooth. Softmax of two numbers x, y is defined as $\log(e^x + e^y)$. To get a better picture of the difference between max and soft-max, please refer to the picture of Figure 1.

Algorithm 2 SMVI(S, A, P, R)

1: $\mathcal{V}(s) \leftarrow 0, \forall s$
2: **while** not converged **do**
3: $\mathcal{V}(s) \leftarrow \mathcal{V}'(s), \forall s$
4: $\mathcal{V}'(s) \leftarrow \text{softmax}_a\{R(s,a) + \sum_{s'} P(s,a,s')\mathcal{V}(s')\}, \forall s$

SMVI computes a policy that maximizes the causal entropy. The only difference between SMVI and standard value iteration is the use of soft-maximum instead of maximum in the computation of $\mathcal{V}^t(s)$. It computes policy at a state as follows:

$$\pi(s,a) = e^{\mathcal{V}(s,a)-\mathcal{V}(s)}$$
$$\mathcal{V}(s,a) = R(s,a) + \sum_{s'} P(s,a,s')\mathcal{V}(s')$$
$$\mathcal{V}(s) = \text{softmax}_a \mathcal{V}(s,a)$$

SMVI computes policies that have the highest entropy possible while minimally (near optimal solution quality) reducing the overall expected value. Algorithm 2 (Ziebart 2010) provides the algorithm for SMVI. Once the policies are computed using SMVI, we then compute the $x_{s,a}$ values by executing the policies. SMFU has the same algorithm as Algorithm 1, except that it employs SMVI for solving the MDP in line 5. In our experimental results, we show that there is a marked improvement in the performance of SMFU over FP-SAP while converging to ϵ-nash equilibrium.

Experimental Results

To demonstrate the effectiveness of our approaches, in particular, the applicability of our approaches in a real-world environment, we prepared two sets of data in our experiments. The first data set is synthetic, and we have full control over how flows, rewards, and costs are generated. We use this data set to explore the performance of our algorithms under different settings. The second data set is a month-long operational data from a real-world taxi fleet. We use this large-scale data set to demonstrate the real-world scalability and the effectiveness of our approaches. In both studies, we look at the two performance metrics that we aim to optimize earlier: (a) Revenue (both minimum and average) of taxi drivers; and (b) Starvation for customers across all zones.

To understand how our approaches fare against straightforward, myopic reasoning, we introduce a family of heuristics based on the *quantal response equilibrium* (to be formally introduced later). We also make comparison against policy based on the one-step best response computation.

Experimental Benchmarks

Quantal response equilibrium (QRE) is a solution concept first proposed by McKelvey and Palfrey (McKelvey and Palfrey 1995). QRE allows bounded rationality to be incorporated in equilibrium seeking process (computationally speaking, it implies errors are allowed in best reply computation). One of the most common specification for QRE is *logic equilibrium* (LQRE); i.e., agents assign probability to each action according to the proportion of the expected payoff of this action over the sum of payoffs of all actions. In LQRE, the parameter λ controls how *rational* agents are: $\lambda = 0$ implies completely irrational, and agents simply choose their actions uniformly, while $\lambda \to \infty$ implies perfectly rational, and agents play according to Nash equilibrium.

In real-world taxi cruising, we observe that the percentage of drivers flowing into zone z during time t to be closely correlated to the percentage of total out-bound trips from zone z in time t (based on traces of about 8,000 taxis, the R^2 value is 0.6747 as illustrated in Figure 2(a)). If we view *the percentage of total out-bound trips* as an approximation to the expected payoff for choosing zone z in time t, this close correlation implies that real-world cruising patterns of taxi drivers are very similar to the one predicted by LQRE. In subsequent graphs, we use $L(\lambda)$ to represent the approximated LQRE that follows the above insights.

Besides LQRE, we also define simpler greedy heuristics that assign uniform probabilities to the top g actions (represented as $G(g)$ in our graphs). Conceptually speaking, $G(g)$ is a limited version of $L(\lambda = 0)$, in which agents can only recognize the top g actions. These two classes of greedy approaches are outlined in Algorithm 3.

Finally, we also look at an online greedy approach (denoted as BR in graphs) that computes one-step best response against the known state distribution of other agents.

Algorithm 3 Greedy(ddap, g, type, λ)

1: $\tilde{D} \leftarrow \mathbf{0}, \mathcal{R} \leftarrow$ UPDATEREWARDSTRANSITIONS($\tilde{D}$)
2: **for all** $t \leq H$ **do**
3: **for all** $s \in \mathcal{S}$ **do**
4: $\{O, R\} \leftarrow$ SORTA($\{\sum_{s'} \phi_{\mathbf{0}}(s,a,s')\mathcal{R}_{\mathbf{0}}(s,a,s')\}_{a \in \mathcal{A}}$)
5: **for all** $i < g$ **do**
6: **if** $type = quantal$ **then**
7: $\pi^t(s, O(i)) \leftarrow$ GETQUANTAL(O, R, λ)
8: **else**
9: $\pi^t(s, O(i)) \leftarrow \frac{1}{g}$
10: **return** π

Experimental Setup and Results

We experiment with two sets of data. Firstly, we created a synthetic data set to test our approaches on different combinations of flows and rewards/costs. We generated our synthetic data set as follows: (a) Customer flows are randomly generated based on different types of zones (office, residential, entertainment etc.). (b) Revenues and costs are generated according to structure of the map. We varied the overall customer population and the number of zones.

The second data set is the real world data set obtained by filtering and aggregating data over six months from a leading taxi company in Singapore. In this data set we have 8,000 taxis (agents) traveling through 83 zones in Singapore. The

flows represented in this problem are the actual customer flows aggregated over the span of a month. Here, we have a horizon of 48 (corresponding to 30 minute intervals on a 24 hour weekday). The flows cannot be changed, so we run with 10 different starting state distributions.

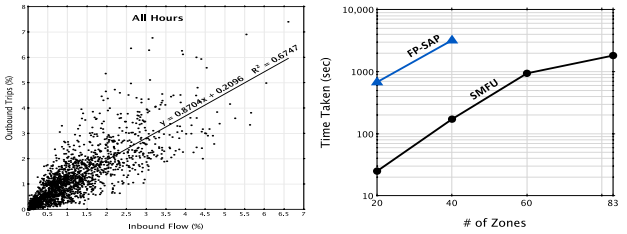

Figure 2: (a) Correlation of inbound flow and outbound trips; (b) Run-time performance of SMFU and FP-SAP.

FP-SAP and SMFU provide ϵ-equilibrium policies, i.e., drivers would not have an incentive more than ϵ to deviate from the computed policy. This is a strong result especially in large-scale problems with thousands of agents. However, the policies are only useful if they are also shown to provide tangible benefits . This is the main theme of our experimental results section – to demonstrate the usefulness of our equilibrium approaches in large scale problems such as the taxi problem. In the taxi problem, we optimize revenue (directly for taxi drivers) and starvation (indirectly for the management) and these are our tangible metrics.

We have two sets of results in this section. Firstly, we will provide the run-time performance of the FP-SAP and SMFU algorithms on both the data sets . Figure 2(b) provides the results of this experiment. X-axis denotes the scale of the problems represented using the number of zones. Y-axis denotes the amount of time taken in seconds on a logarithmic scale. As expected, the FP-SAP approach took a lot more iterations to converge and consequently it took a lot more time. There is at least an order of magnitude speed up provided by SMFU over FP-SAP for all the cases. For instance in the 40-zone case, SMFU took 171 seconds on average , while FP-SAP took 3216 seconds. Furthermore, FP-SAP was unable to finish within our cut-off time of 3 hours from problems with more than 40-zones. We have also included the result for the real world taxi data set as part of this graph (83 zone problem). For all instances of the real world problem, SMFU finished within 30 minutes. In the second set of results, we compare SMFU[1] with all the greedy heuristics. Figure 3(a)-(c) are results on synthetic data set with 40 zones[2] and Figure 3(d)-(f) are results on real world data set with 83 zones and 8,000 taxis. Y-axis is the revenue obtained in graphs (a),(b),(d),(e) (higher the better), while on (c), (f) it indicates the number of people who did not get a taxi (over the entire horizon) and hence lower is better. X-axis is the number of people in the island (representative complexity of the problem) who use taxis.

Here are some of the key conclusions that can be made from the graphs:

- SMFU obtains higher minimum revenue than all greedy heuristics on most cases (even considering the variance). However, when population size is 4,000 on synthetic data set, some of the greedy approaches obtain a higher minimum revenue. This could be due to the fact that most of the taxis are idle when population is low and since we optimize over average revenue (and not minimum revenue).
- With respect to average revenue and starvation, we outperformed greedy heuristics by a significant margin on all the cases. For instance, a SMFU driver makes about S$40 more than the next best greedy policy. Similarly, there are about 5,000 more people served with our SMFU policies than the next best greedy policy.

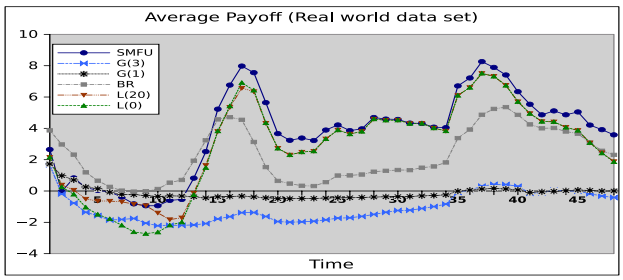

Figure 4: Average revenue plot for 24 hours.

Figure 4 provides the average pay-off accrued by SMFU in comparison with other greedy heuristics over the 30 minute intervals during the 24 hours. The peaks are during the business hours 7:00 AM - 10:00 AM (14-20) and 5:00 PM - 8:00 PM (34-40). It can be noted that SMFU obtains higher average pay-off across most times during the day.

These results indicate that symmetric Nash equilibrium solutions can provide much better solutions than human policies and typical benchmarks employed in the literature.

Related Work

In this section, we briefly describe research related to the contributions made in this paper. The first thread of related research is in the field of transportation. User equilibrium (UE) is a classical and powerful equilibrium concept in transportation explaining individual route choices in face of competition for road usages from other users. Originally proposed in static setting (Wardrop 1952), it was later expanded to dynamic cases (where temporal choices are also important) (Friesz et al. 1993). In either format, static or dynamic, the concept of UE provides a way to infer and to predict the behaviors of individual drivers; such ability helps not just individual drivers to identify better routes, but it can also help policy makers to properly design road network in anticipation of driver's responses. The most widely adopted approach in computing UE is proposed by (LeBlanc, Morlok, and Pierskalla 1975). Their approach is based on the Frank-Wolfe algorithm, which essentially is an iterative process that minimizes a function by repeating: 1) identify a movement direction from the current point, 2) search for the step size, 3) move to the next point. The iteration stops when certain stopping criterion is met. While user equilibrium is

[1]We obtained similar results with FP-SAP on problems where we finished within the cut-off limit and hence are not shown here.

[2]We have results for the 20 zone and 60 zone cases. Those results look very similar to the ones for 40 zones.

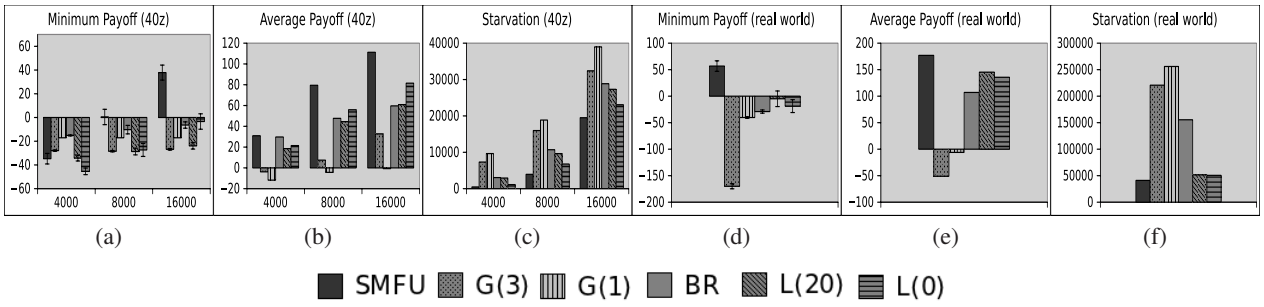

Figure 3: Comparison of SMFU with Greedy heuristics on 40 zone synthetic data set and real world data set.

relevant, it does not account for the presence of in-voluntary movements for agents.

The next thread of relevant research is due to Weintraub *et al.* (Weintraub, Benkard, and Roy 2006; 2008). This research introduces the concept of oblivious equilibrium for large scale dynamic games. They provide a mean field approximation to solve problems where there is stochasticity in state transitions. While, the problem is similar to DDAPs, the assumption of mean field (or a stationary distribution of taxis in our case) is not applicable in the context of taxi problems. In fact, there is a huge variance in the set of possible distributions at each decision epoch and hence oblivious equilibrium is not directly applicable in our context.

DDAP model represents a subset of problems represented by the generic Partially Observable Stochastic Games (POSG) model. There are many approaches (Seuken and Zilberstein 2007; Velagapudi et al. 2011; Varakantham et al. 2009) provided for solving identical payoff stochastic games (also referred to as Decentralized POMDPs or DEC-POMDPs). However, because the approaches typically assume a unique identity to each of the agents, they cannot scale to problems with 8,000 agents.

References

Alpern, S., and J.Reyniers, D. 2002. Spatial dispersion as a coordination problem. *Theory and Decision* 1(53):29–59.

Brown, G. W. 1951. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*. John Wiley, New York. 374–376.

Friesz, T. L.; Bernstein, D.; Smith, T. E.; Tobin, R. L.; and Wie, B. W. 1993. A variational inequality formulation of the dynamic network user equilibrium problem. *Opers. Res.* 41(1):179–191.

Harris, J. R., and Todaro, M. P. 1970. Migration, unemployment and development: A two-sector analysis. *Amer. Econ. Rev.* 60(1):126–142.

LeBlanc, L. J.; Morlok, E. K.; and Pierskalla, W. P. 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Trans. Res.* 9(5):309–318.

McKelvey, R., and Palfrey, T. 1995. Quantal response equilibria for normal form games. *Games and economic behavior* 10:6–38.

Monderer, D., and Shapley, L. 1996. Potential games. *Games and economic behavior* 14:124–143.

Nash, J. F. 1951. Non-cooperative games. *The Annals of Mathematics* 54(2):286–295.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Robinson, J. 1951. An iterative method of solving a game. *Annals of Mathematics* 54(2):296–301.

Seuken, S., and Zilberstein, S. 2007. Improved memory-bounded dynamic programming for decentralized POMDPs. In *UAI*.

Varakantham, P.; Kwak, J. Y.; Taylor, M.; Marecki, J.; Scerri, P.; and Tambe, M. 2009. Exploiting coordination locales in distributed POMDPs via social model shaping. In *ICAPS*.

Velagapudi, P.; Varakantham, P.; Sycara, K.; and Scerri, P. 2011. Distributed model shaping for scaling to decentralized pomdps with hundreds of agents. International Joint Conference on Autonomous Agents and Multiagent Systems.

Wardrop, J. G. 1952. Some theoretical aspects of road traffic research. In *Proc. of Inst. of Civil Engrs. Part II*, volume 1, 325–378.

Weintraub, G. Y.; Benkard, L.; and Roy, B. V. 2006. Oblivious equilibrium: A mean field approximation for large-scale dynamic games. In *NIPS*.

Weintraub, G.; Benkard, C.; and Roy, B. V. 2008. Markov perfect industry dynamics with many firms. *Econometrica* 76(6):1375–1411.

Yang, H., and Wong, S. C. 1998. A network model of urban taxi services. *Trans. Res.* 32B(4):235–246.

Ziebart, B. D. 2010. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. Dissertation, Carnegie Mellon University.