8-2011

TaxiSim: A multiagent simulation platform for evaluating taxi fleet operations

Shih-Fen CHENG
Singapore Management University, sfcheng@smu.edu.sg

Thi Duong NGUYEN

Citation

CHENG, Shih-Fen and NGUYEN, Thi Duong. TaxiSim: A multiagent simulation platform for evaluating taxi fleet operations. (2011). *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (WI-IAT), 22-27 August 2011, Lyons, France: Proceedings*. 14-21. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/1405

TaxiSim: A Multiagent Simulation Platform for Evaluating Taxi Fleet Operations

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

Thi Duong Nguyen
School of Information Systems
Singapore Management University
Republic of Singapore
tdnguyen@smu.edu.sg

***Abstract*—Taxi service is an important mode of public transportation in most metropolitan areas since it provides door-to-door convenience in the public domain. Unfortunately, despite all the convenience taxis bring, taxi fleets are also extremely inefficient to the point that over 50% of its operation time could be spent in idling state. Improving taxi fleet operation is an extremely challenging problem, not just because of its scale, but also due to fact that taxi drivers are self-interested agents that cannot be controlled centrally. To facilitate the study of such complex and decentralized system, we propose to construct a multiagent simulation platform that would allow researchers to investigate interactions among taxis and to evaluate the impact of implementing certain management policies. The major contribution of our work is the incorporation of our analysis on the real-world driver's behaviors. Despite the fact that taxi drivers are selfish and unpredictable, by analyzing a huge GPS dataset collected from a major taxi fleet operator, we are able to clearly demonstrate that driver's movements are closely related to the relative attractiveness of neighboring regions. By applying this insight, we are able to design a background agent movement strategy that generates aggregate performance patterns that are very similar to the real-world ones. Finally, we demonstrate the value of such system with a real-world case study.**

***Keywords*-multiagent simulation; urban transportation; driver behavior; mobility pattern; taxi fleet**

I. Introduction

Taxi service is an important mode of public transportation in most metropolitan areas (e.g., in Singapore, taxi rides accounted for around 17% of public transports in 2007/08), since it provides door-to-door convenience in the public domain. Unfortunately, despite all the convenience taxis bring, taxi fleets are also extremely inefficient. In an ordinary city, a taxi can easily spend 50% of its operation time idling (waiting in queues or roaming around empty). For cities that are getting increasingly crowded, inefficient taxi fleet not only offers lower quality of service than its potential would grant, it also creates negative impacts on environment and road congestion. As such, improving the efficiency of the taxi fleet operation is an important issue for government agencies and taxi fleet operators alike.

Many past research efforts have been devoted to the modeling of the taxi fleet operations and also approaches that would improve the efficiency of taxi fleets. For example, advances in technologies like Global Positioning System (GPS) and communication networks enable advanced dispatch system to be deployed [1], [2]. On the other hand, a series of work conducted by Yang et al. [3], [4] provides a good framework for understanding the equilibrium properties of taxis in a network at the macro level. However, by reviewing these past works (which are mostly published in the transportation literature), we notice that there are very few attention paid to the decentralized nature of the taxi system. One exception is the design of taxi dispatch systems, where we do see the application of multi-agent technologies [5], [6]; nonetheless, taxi dispatch is only one possible mode of operations (the other more dominant modes being street pick-ups and queueing), and a comprehensive study that covers all modes of operations from a decentralized perspective is still not seen. Such decentralized perspective is critical in modeling taxi fleets because taxis can only be *incentivized* or *coordinated* and not centrally *controlled*. With proper models in place, not only can we improve the efficiency of current taxi fleets, a range of new services could be designed and evaluated as well (e.g., efficient cab-sharing service for serving last-mile travels between desired destinations and the closet public transport hubs).

In this paper, we propose to build a multi-agent-based simulation platform, TaxiSim, to simulate the operation of taxi fleets. TaxiSim is designed to be capable of modeling individual taxi driver's strategies at micro level, and it's also designed to be scalable so that it can simulate thousands of taxis simultaneously. Real-world operational data, if available, can also be imported to TaxiSim, and this allows us to construct a highly realistic simulation environment. This would allow researchers and policy makers to study and evaluate potential mechanisms, policies, and new services for improving taxi services.

II. Background

Since the early days of digital computers, simulations have played an important role in transportation research. In all major areas of transportation studies, be it traffic signal control, traffic assignment (routing), or even regional planning, simulations are all involved deeply. With rapid development of computing technology, simulations have now become even more powerful and ubiquitous. Some of the well-known transportation simulation platforms include TRANSYT [7], CORSIM [8], and MITSim [9]. Each of them is designed with different granularity (could be macroscopic, mesoscopic, or microscopic), and each might be used for different applications as well (e.g., vehicle routing, demand forecast, or traffic signal control). More recently, the advances in multi-agent technology have also motivated researchers to construct simulations that are capable of treating individual actors inside a transportation system as agents. Some notable open-source multi-agent simulation projects include MATSim [10] and SUMO [11]. We would like to emphasize that this is a very incomplete listing and it's not our intention to conduct a comprehensive reviews on available traffic simulation softwares available. Our purpose is to highlight the importance of simulation methods in conducting transportation studies, and also the growing trend of adopting multi-agent technology.

Despite all these efforts in building computer simulations for a wide range of studies, to the best of our knowledge, we cannot find any simulation platform that is capable of modeling realistic taxi fleet operations. Taxi fleet operation is special and cannot be modeled straightforwardly by using existing technologies for the following reasons:

- In most cities, taxi drivers pay a fixed rent and keep all remaining revenue. This revenue structure makes them naturally selfish, and to build a credible model, we need to understand how drivers make decisions empirically.
- Taxi drivers are subject to both voluntary and involuntary movements. Involuntary movements occur when customers board their vehicles. After a taxi reaches the destination specified by the boarded customer, it has to continue its voluntary movement from there. Such movement pattern is the most critical difference between taxis and ordinary passenger cars.

To address these unique requirements, we decided to develop our own multiagent simulation platform, TaxiSim.

III. System Architecture

TaxiSim is designed to be a decentralized discrete event simulation, focusing on modeling only taxi driver's behaviors. The traffic condition in the network is regarded as exogenous, and will not be modeled explicitly. This simplifies the design of TaxiSim, however, it should not have an adverse impact on the realism of the simulation, since taxis only constitute a small percentage of all vehicles. All taxi agents in TaxiSim are to be executed as individual threads in the simulation, and each agent maintains its own event queue. There are three major event types (the interactions among these events are illustrated in Figure 1):

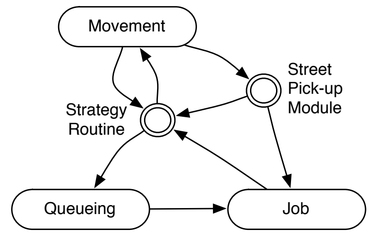

Figure 1. Flow of events.

- Movement event: move to a particular location. This event is generated by the main strategy routine. The expiry time of the event is the expected travel time from the current location to the destination. When the movement event expires, the street pickup module (to be described in detail later) will be invoked (if the service mode is roaming) to determine whether a street job can be picked up. If a job can be picked up, a job event will be spawned.
- Queueing event: join a particular queue. This event can only be generated if the current location is at the queue and queueing is chosen as the service mode. The expiry time of the event is the expected waiting time in the queue before picking up a job. When this event expires, the job event will be spawned.
- Job event: serve a client who intends to move to a particular location. The expiry time is the expected travel time from the current location to the destination. At expiry, revenue will also be generated.

With these event definition, the progression of the simulation can be described by the following steps:

1) (Initialization) Invoke main strategy functions in all threads, and one of the three events will be generated.
2) (Iteration) The main thread queries for the earliest event expiry time from all threads; that thread will be asked to pop and execute the event. For all other threads, their local clocks will be progressed to this earliest expiry time.
3) Step 2 will be executed until the stopping criterion is met (e.g., when the simulation clock exceeds 12 hours).

From Figure 1 we can see that the event structure of TaxiSim is relatively simple and the progression of the simulation depends heavily on the implementation of taxi agent's strategy. To simplify the design of agent strategy, we further decompose the agent strategy into components in Figure 2. The role of each component is briefly explained. A full-blown example on the strategy design is deferred to the next section.

Taxi Initialization Module. This component is responsible for *warming up* the simulation with an initial taxi distribution. This module has to generate two pieces of information: 1) geographical location and 2) agent type (which includes both the strategy and the strategy-related parameters). The

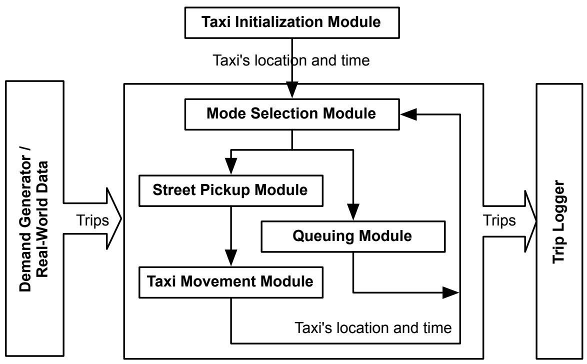

Figure 2. An agent-centric view of the system.

initialization module is controlled by a list of user-supplied parameters, and if necessary, users can supply their own initialization routines as well.

Time Keeping Module. This module repeatedly queries for the earliest event expiry time from all threads and also synchronizes the local clocks at all threads.

Mode Selection Module. This module decides what mode of operation should be used. This is the first major function required to define an agent's strategy.

Queueing Module. Suppose an agent chooses to join a particular taxi queue, the taxi queue will be simulated specifically by the queueing module. The most important queueing features to simulate is the arrival of customers at the queue according to demand data provided by the demand generator and the maintenance of currently queued agents.

Street Pickup Module. If an agent decides to roam the street, this module will determine which areas it should go towards. At each discrete epoch, it will also determine whether this agent can pick up a street job. The model we used in determining such pickups will be described in more detail in later section

Taxi Movement Module. This module dictates how a taxi moves from one point to another on the road network. This is the second major function required to define an agent's strategy. A default background strategy is provided, however, it can be replaced if necessary.

Demand Generator. Customer demands are generated by this module. The demand generated can be either based on real-world data or completely fictitious. No matter how a demand is generated, it must come with four parameters: a) origin, b) destination, c) time, and d) fare.

Logger. This module logs completed trips. If any additional information needs to be logged, user-defined loggers can be implemented based upon a common logging interface.

The simulation framework in TaxiSim allows a wide variety of servicing strategies to be implemented. A background strategy that closely resembles aggregate driver behaviors is included as the default strategy in TaxiSim. If necessary, user-defined strategy can also be designed easily by using the provided API.

IV. Designing the Background Strategy

As Gode and Sunder [12] have demonstrated in their work on the financial market simulation, simplistic and random *zero intelligence* (ZI) agents can collectively generate rational aggregate outcomes. When designing the background strategy for TaxiSim, we adopt similar design principles and try to identify a strategy that is simple yet capable of generating credible aggregate results.

In TaxiSim, taxi driver's behavior consists of two components: 1) service mode choice and 2) service strategy. For a taxi driver, the service mode choice refers to the choice of operation mode (roaming, queueing, or waiting for dispatch job). After the service mode is chosen, a taxi driver will then try to decide the best operational policy to use in that mode. For example, a taxi driver, on choosing the roaming mode, will have to decide which region to roam and what path to take. The implication of this design is that strategies can be built incrementally. For the background strategy, we initially include only street roaming (the dominant mode of operation), and if necessary, add additional modes later.

When designing the background strategy, our goal is to create a fleet of simulated taxis that is representative of real-world fleets. We do not intend to model taxi behaviors at micro level, instead, the macro-level regularity is what we are after. We made such modeling choice since micro-level real-world patterns are extremely noisy, and it's often difficult (if not impossible) to infer individual's intention from the observed data traces. Moreover, even when we have accurate behavioral models for certain individuals, it is still far from being representative of real-world fleet.

The macro-level regularity we are interested in is the revenue accumulation pattern over time. The revenue accumulation pattern is chosen as the target since it aligns with agent's objective function and the day-to-day pattern observed from the empirical data is consistently recurrent.

The design idea for the background strategy comes from our analysis of the empirical data, which is described in the following subsection.

A. *Quantifying Drivers' Behaviors*

The source of data that supports our analysis comes from a taxi fleet operator in Singapore. Our analysis on the dataset reveals a surprising resemblance of the aggregate driver's behaviors to a myopic strategy. More precisely, the myopic strategy refers to choosing zones to move to according to the relative density of trip originations. The definition of zones in our analysis is adopted from the official zoning defined by the Singapore Land Authority (illustrated in Figure 3).

Figure 3. Zones defined in Singapore.

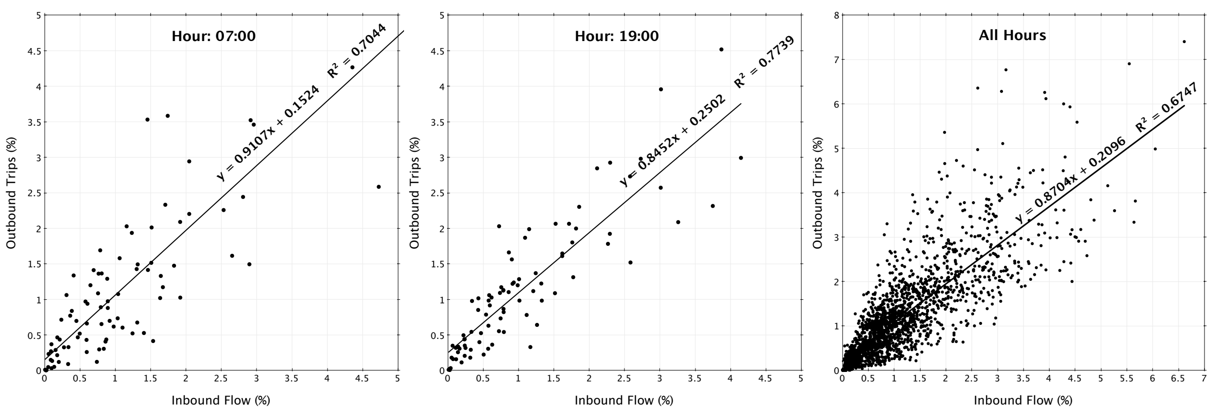

Figure 4. The correlations between outgoing trips and incoming flows at 7am, 7pm, and all hours.

The available data for our analysis includes trip information and movement logs. For each captured trip, the dataset contains fare, origin coordinate, destination coordinate, and times at departure and arrival. For movement logs, each log entry captures time of the log, latitude, longitude, and taxi status (free, hired, or others). The time-dependent density of trip origination out of each zone can be easily measured by accumulating trip counts based on origin coordinates and trip starting time. The free movement of drivers, on the other hand, is much more difficult to measure, due to the following two reasons:

- Some drivers might not have any strategy in mind when making movement decisions. For example, they might simply choose to arbitrarily roam the current neighborhood, with no specific destination in mind. Or they might even just park along the roadside and rest. These drivers are going to generate significant noises in the dataset.
- Even when drivers have specific destinations they would like to travel to, they are going to pass through a number of zones in between their current zones and the destination zones, and a lot of unintended log traces will be generated alone the way as a result.

To overcome these two difficulties, we adopt a simple threshold-based rule in filtering movement logs. From the movement logs, we first infer the amount of time each driver spent in the zones that s/he passes by. For each zone, we then define a size/distance dependent lower bound to filter out movements that are unlikely to be intentional (since they have not stayed long enough to show that they are indeed *interested* in that zone). After this filtering is done, we then aggregate all inbounding flows into every zone at each and every hour.

If drivers indeed adopt a myopic strategy in aggregate, we should see strong positive correlation between outgoing trips from a zone (which represent how attractive this zone is) and incoming flows to the same zone (which represent drivers' aggregate intentions to go to this zone). Figure 4 illustrates the analysis on the data we collected from the weekdays of July 2009. Each plotted data point represents the count from a zone in a particular hour, and all counts per hour (flows and trips) are converted to percentages of total counts over all zones per hour to ensure meaningful comparison.

We choose two most representative times to highlight our analysis. Two highlighted time frames, 7–8am and 7–8pm, are the morning and evening rush hours and their R^2 values are 0.7044 and 0.7739 respectively. For all time frames, the R^2 value is slightly lower at 0.6747. From this analysis we can see that a strong positive correlation indeed exists between the relative attractiveness of a zone and drivers' aggregate movement. Furthermore, we can see that such correlation is stronger when traffic patterns are more recurrent and predictable (e.g., the ones during rush hours). However, even the R^2 value of 0.6747 for all hours demonstrates sufficiently strong positive correlation.

B. *The Background Strategy*

Based on our empirical findings, we design a roaming strategy that makes probabilistic moves toward different zones according to their relative attractivenesses (precomputed based on historical demands). In other words, the taxi will make randomized movement weighted by the earning potentials. Such strategy should create aggregate movement patterns similar to the ones we observed from the real-world data.

Formally speaking, zones are predefined polygons that are mutually exclusive and collectively cover the whole area of interest (the main Singapore island in our case). The earning potential, or the *attractiveness* of zone j, denoted as a_j, can be quantified by

$$a_j \propto \frac{d_j}{r_j(p)^2}, \tag{1}$$

where d_j is the number of trips departing from zone j and $r_j(p)$ is the distance between the centroid of zone j and the location p. This definition is quite intuitive: a zone will be more attractive if it has more trips and is closer to the current location. a_j is inversely correlated to the square of $r_j(p)$ since we want to account for the fact that longer distance incurs both higher movement cost and also longer traveler time. The background strategy is designed to follow the attractivenesses computed in (1): the probability that the agent moves toward a particular zone is proportional to its attractiveness, i.e., $p_j = a_j / \sum_j a_j$, where p_j is the probability that zone j will be chosen.

V. Roaming Mode: Design and Calibration

The myopic background strategy introduced in the previous section provides the building block for the modeling of street roaming. However, the interaction of agent strategies still needs to be carefully designed and calibrated to achieve a realistic simulation. In this section, we discuss how to model the competition among taxi drivers for a limited pool of street jobs, and how we can calibrate the simulation with real-world data.

A. *Designing the Street Pickup Module*

The role of street pickup module is to determine what should happen when multiple agents are competing for the same pool of street jobs and it should be independent of the design of agent strategies.

To determine whether an agent can pick up a job at its current location, following factors need to be considered:

1) Spatial constraint: The agent must be within certain radius to the prospective job in order to pick it up.
2) Temporal constraint: The agent can only pick up jobs that are already revealed (i.e., for customers who have already revealed their needs for travels).
3) Competition: Even when an agent meets both the spatial and temporal constraints, it still has to compete with other agents (which may or may not be included in the simulation) for the revealed job.

Formally speaking, we define a job j to be the tuple $(p_s^j, p_e^j, t_s^j, t_e^j)$, where p_s^j, p_e^j, t_s^j, and t_e^j are respectively the job's origin coordinate, destination coordinate, departing time, and arrival time. With these notations, the set of feasible jobs at time t_c and location p_c can then be defined as:

$$J_p(t_c, p_c) \equiv \{j \mid t_s > t_c + T(p_c, p_s), T(p_c, p_s) \le \epsilon\}, \tag{2}$$

where $T(p_c, p_s)$ refers to the travel time from p_c to p_s and ϵ refers to the duration of one time unit in our simulation.

For jobs in $J_p(t_c, p_c)$, the probability for a job to be picked up should depend on the relative distance between a job's originating location and the taxi's current location. This comes from the intuition that taxis closer to a customer should have greater chances of getting the job. Similarly, the temporal difference between a job's reveal time and the time when the taxi spot that job should also follow similar intuition, i.e., a taxi comes closer to a job when it's revealed should be more likely to pick up that job than taxis that come later. In addition, the chance to pick up jobs also depends on how competitive an area is; i.e., all things being equal, it should be less likely to pick up a job in a more competitive area.

To summarize the influences of both the spatial and temporal distances, we define the normalized composite (NC) distance from a taxi to a job as:

$$\delta = \frac{1}{2} \times \left[\frac{\delta_d}{D_\epsilon} + (1 - \frac{\delta_t}{\epsilon}) \right], \tag{3}$$

where δ_d and δ_t refer to the spatial and temporal distances between the job and the taxi respectively and D_ϵ refers to the maximum distance the taxi can travel during one time unit. By construction δ should be a real number in the range of $[0, 1]$.

When $\delta = 1$, the chance of picking up a trip is 0.1. When $\delta = 0$, the chance of picking up a trip is 0.8. For $0 < \delta < 1$, the pickup probability follows an exponential function, and can be parametrized as $p(\delta) = \alpha e^{\beta\delta}$. α and β can be solved by using the above boundary conditions.

The level of competition in a zone is summarized by the *chance of retrying* and it represents how likely a taxi can keep looking for jobs in a time period. If a particular zone is very competitive, then there should be fewer such retrying opportunities, and the chance of retrying should have a smaller value.

To determine whether an agent can pick up a job at the current location and in the current time period, the following steps will be utilized (for the ease of presentation, assume that $J_p(t_c, p_c)$ is an ordered set with elements $j_1, j_2, \ldots$ and $\delta_{j_i} \leq \delta_{j_{i+1}}$ for all i):

1) Set the counter $i \leftarrow 1$.
2) For j_i, the probability that it will be picked up is $p(\delta_{j_i})$. Sample from $p(\delta_{j_i})$, if the result is positive, stop and return j_i (and j_i will be removed globally from all agents' considerations). If the result is negative, move to the next step.
3) With probability $(1 - q_{z,t})$, the street pickup module will terminate the search, return to the main strategy module and notify the agent to take next action. If the search is not terminated yet and $i + 1 \leq |J_p(t_c, p_c)|$ (implying that there is available job in set $J_p(t_c, p_c)$), increase the counter, $i \leftarrow i + 1$, and repeat step 2.

When the above procedures terminate, the agent will either be awarded a job (which will trigger the creation of a job event) or it will be ordered to move. The parameter $q_{z,t}$ introduced in step 3 is the retrying probability mentioned earlier, and it reflects the likelihood that the taxi will be granted an additional chance in trying to search for another job. In a more competitive area, such retrying will be less likely and it is reflected in having a lower $q_{z,t}$. As we should see in the next section, this parameter is our main focus in calibrating the simulation.

B. Calibrating the Street Pickup Module

The introduction of the background strategy allows us to put together a working simulation. However, to have a realistically useful simulation, we still need to calibrate it with the real-world dataset that we have. Of the three operational modes (roaming, queueing, and job dispatching), queueing and job dispatching need almost no calibration, as both modes follow deterministic rules and their operations are quite predictable. The most difficult mode to deal with is the roaming mode, since a lot of external factors (most of them uncertain) are involved. Therefore, when calibrating TaxiSim, our focus is on the roaming mode (handled by the street pickup module just introduced).

We propose a simple calibration process that is shown to be very effective in modeling unobservable competitions. As mentioned in the previous section, we have summarized the level of competition using the retrying probability. Therefore, the calibration of TaxiSim can be abstractly viewed as an optimization problem, with decision variables being the retrying probabilities and the objective function aiming to minimize the weighted sum of absolute differences in average revenues between the simulation and the real-world data. Expressed formally, the optimization problem is:

$$\begin{aligned} &\min \sum_{z,t} w_{z,t} \left| r_{z,t}(\{q_{z,t}\}) - R_{z,t} \right| \qquad (4) \\ &\text{s.t.} \\ &0 \leq q_{z,t} \leq 1, \forall z, t. \end{aligned}$$

In the above formulation, $r_{z,t}$ and $R_{z,t}$ are the average revenues obtained from the simulation and real-world data respectively, $q_{z,t}$ is the chance of retrying, and $w_{z,t}$ is the percentage of trips originating from zone z in time period t. The definition of $w_{z,t}$ straightforwardly incorporates the relative importance of different tuples.

Although Problem (4) looks simple, it's in fact very difficult to be solved exactly because $r_{z,t}(\{q_{z,t}\})$ can only be evaluated by executing multiple simulation runs. Due to the intractable nature of Problem (4), we have proposed a simple hill-climbing heuristic for the simulation calibration. This heuristic is described by the following steps:

1) Initialize retrying probability $q_{z,t}$ to 0.4 and step size $\epsilon_{z,t}$ to 0.1 for all (z, t) tuples.
2) Execute the simulation by using the current vector of $\{q_{z,t}\}$. Compute the average revenues $r_{z,t}$ for all tuples.
3) For each (z, t) tuple, if $r_{z,t} > R_{z,t}$, let $q_{z,t} \leftarrow q_{z,t} - \epsilon_{z,t}$, otherwise let $q_{z,t} \leftarrow q_{z,t} + \epsilon_{z,t}$. If the gap (i.e., $|r_{z,t}(\{q_{z,t}\}) - R_{z,t}|$) narrows over previous iteration, let $\epsilon_{z,t} \leftarrow 0.5\, \epsilon_{z,t}$.
4) Check if the stopping criterion is met; if not, go to step 2, otherwise, stop.

The stopping criterion can be either time-based, in which the process will execute a fixed number of iterations, or performance-based, in which the overall performance is monitored, and the process will terminate if it is not making sufficient improvement.

To calibrate TaxiSim, we load the simulation with 1,000 agents, each running the background strategy, and the calibration result is plotted in Figure 5. The result shows that after calibration, the average difference of revenue between our simulation and the real-world data is around \$0.46. Considering that the average revenue per hour is around \$12, this translates into a mere 3.8% error rate.

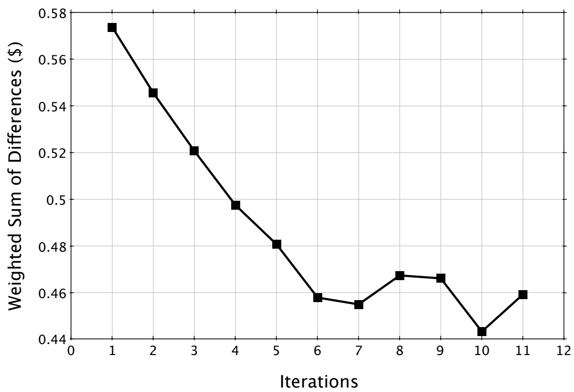

Figure 5. The evolution of weighted sum of differences over iterations. Note that the value stabilizes after iteration 5.

VI. Use Case: Evaluating a Strategy Profile

In this section, we evaluate the performance of a new strategy for taxi drivers and a potential information service from the fleet operator. We first describe the basis of the strategy, then our simulation setup and finally our analysis of the simulation result.

A. Optimal Service Choice Strategy

For individual agents, one of the difficulties in choosing service mode (and also service strategy) is that agents usually have very limited information regarding remote locations. Without such information agents will have to make their own predictions and this potentially can incur a lot of errors (and resulting in sub-optimal decisions). The inefficiency of such decision making process in real-world operations is quantified in an earlier study by [13], and one potential solution is for the taxi fleet operator to provide agents with necessary real-time information so that they can compute their "optimal service choices".

However, if we evaluate such proposal at the system level, it is doubtful such solution will actually improve driver's revenue. In fact, we suspect that due to the lack of strategic reasoning (recognizing the fact that other drivers also possess similar information), agents will end up clogging the queue when the expected revenue at the queue is high, and unnecessarily avoiding the queue when the expected revenue at the queue is low. This not only might cause drivers to suffer, it might also create adverse effects at the queue (the queue will constantly be either too crowded or too underserved). A formal analysis and studying of such phenomenon is beyond the scope of this paper, but we would like to present some initial simulation results from this study to demonstrate how TaxiSim can be used in assisting research agendas on quantifying driver's intentions as well as on designing incentive mechanism to better coordinate a fleet of selfish drivers.

B. Simulation Setup and Result

We set up a simulation with 2,000 taxis, and we try to experiment with different market penetration ratios of the "intelligent service choice technology". We start our experiment with only one agent equipped with such technology, and then gradually increase the ratio to 20%, 40%, 60%, 80%, and then finally ∼100% (all but one taxi adopt intelligent technology). For simplicity, we assume that there is only one major queue that agents can choose to go to. From the dataset, we recognize that the airport is the single largest queue (can hold several hundreds of taxis during the peak hours), thus it will be chosen as the designated queue. We also assume that agents by default will follow the background strategy, and it will only join the designated queue at random (a small probability derived from the real-world dataset). For agents holding the intelligent technology, they will query this technology for advices whenever they are choosing their service mode, and if the suggestion returned is to go to the airport queue, they will make a shortest-path travel there and join the queue.

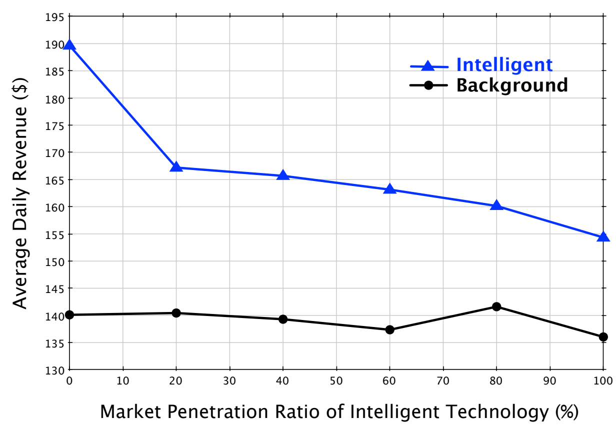

Figure 6. Background drivers v.s. intelligent drivers under different market penetration ratios.

The result of the simulation is illustrated in Figure 6, and it confirms our earlier conjecture: increasing market penetration ratios in our simulation study indeed causes the average performance of intelligent technology holders to deteriorate steadily. However, from the result we can also see that the drop in the performance is most significant when the ratio moves from ∼0% to 20% (a drop of around 12%), after which the decline is much gradual (all around 1% to 3%). Another interesting finding is that even in the case where intelligent technology floods the market and causes its holder's performance to drop, its average performance is still better than that of the background strategy (who only makes ad hoc visits to the queue).

VII. Conclusions

In recent years, more and more research efforts in AI are being introduced to transportation, a domain traditionally dominated by operations researchers and civil engineers. For example, an agent-based approach is recently demonstrated to be a more effective alternative to the traditional intersection control technology [14]. Also, traditional transportation researchers are increasingly more acceptable to the idea of multiagent technology; for example, the potentials of agent-based technologies and machine learning techniques in traffic control are highlighted in a recent review by [15].

This paper contributes to this increasingly promising line of research, and in particular, we contribute to the study of taxi fleet operations, an important yet overlooked area in the urban mobility research. Our primary contribution is the methodology used in creating TaxiSim, a highly realistic agent-based simulation platform dedicated to taxi fleets. In developing TaxiSim, we have successfully extracted a representative agent strategy from our analysis of the real-world dataset. We have also proposed a simple yet effective process for calibrating TaxiSim. Finally, we present an use case on how TaxiSim can be practically used to study complex strategic interactions in taxi fleet operations.

We show that TaxiSim is a good platform for evaluating and experimenting advanced strategies for taxi drivers, as well as new policies and mechanisms which affect the dynamics of the whole taxis eco-system. Taxi fleets in urban environments are agile and flexible. With proper coordinations and incentives, they can be utilized to improve the urban mobility eco-system. Realizing the full potential of these taxi fleets with AI techniques will remain one of our major research directions in the future.

Acknowledgment

This work was supported in part by the School of Information Systems, Singapore Management University, and also by the Future Urban Mobility group of the Singapore-MIT Alliance for Research and Technology.

References

[1] Z. Liao, “Real-time taxi dispatching using global positioning systems,” *Communications of the ACM*, vol. 46, no. 5, pp. 81–83, 2003.

[2] D.-H. Lee, H. Wang, R. L. Cheu, and S. H. Teo, “A taxi dispatch system based on current demands and real-time traffic information,” *Transportation Research Record*, vol. 1882, pp. 193–200, 2004.

[3] H. Yang and S. C. Wong, “A network model of urban taxi services,” *Transportation Research Part B: Methodological*, vol. 32, no. 4, pp. 235–246, 1998.

[4] H. Yang, M. Ye, W. H.-C. Tang, and S. C. Wong, “A multiperiod dynamic model of taxi services with endogenous service intensity,” *Operations Research*, vol. 53, no. 3, pp. 501–515, 2005.

[5] A. Alshamsi, S. Abdallah, and I. Rahwan, “Multiagent self-organization for a taxi dispatch system,” in *8th International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 21–28.

[6] K. T. Seow, N. H. Dang, and D.-H. Lee, “A collaborative multiagent taxi-dispatch system,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 607–616, 2010.

[7] D. I. Robertson, “TRANSYT method for area traffic control,” *Traffic Engineering & Control*, vol. 11, no. 6, pp. 276–281, 1969.

[8] A. Halati, H. Lieu, and S. Walker, “CORSIM – corridor traffic simulation model,” *Proceedings of the Traffic Congestion and Traffic Safety in the 21st Century Conference*, pp. 570–576, 1997.

[9] Y. Qi and H. N. Koutsopoulos, “A microscopic traffic simulator for evaluation of dynamic traffic management systems,” *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, 1996.

[10] B. Raney, N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen, and K. Nagel, “An agent-based microsimulation model of Swiss travel: First results,” *Networks and Spatial Economics*, vol. 3, no. 1, pp. 23–41, 2003, http://www.matsim.org/.

[11] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, “SUMO (Simulation of Urban MObility) — an open-source traffic simulation,” in *4th Middle East Symposium on Simulation and Modelling*, 2002, pp. 183–187, http://sumo.sourceforge.net/.

[12] D. K. Gode and S. Sunder, “Allocative efficiency of markets with zero-intelligence traders: Markets as a partial substitute for individual rationality,” *Journal of Political Economy*, vol. 101, no. 1, pp. 119–137, 1993.

[13] S.-F. Cheng and X. Qu, “A service choice model for optimizing taxi service delivery,” in *12th International IEEE Conference on Intelligent Transportation Systems*, St. Louis, MO, 2009, pp. 66–71.

[14] K. Dresner and P. Stone, “A multiagent approach to autonomous intersection management,” *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, March 2008.

[15] A. Bazzan, “Opportunities for multiagent systems and multiagent reinforcement learning in traffic control,” *Autonomous Agents and Multi-Agent Systems*, vol. 18, pp. 342–375, 2009.