A Service Choice Model for Optimizing Taxi Service Delivery

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

Xin Qu
School of Information Systems
Singapore Management University
xinqu@smu.edu.sg

Abstract— **Taxi service has undergone radical revamp in recent years. In particular, significant investments in communication system and GPS devices have improved quality of taxi services through better dispatches. In this paper, we propose to leverage on such infrastructure and build a service choice model that helps individual drivers in deciding whether to serve a specific taxi stand or not. We demonstrate the value of our model by applying it to a real-world scenario. We also highlight interesting new potential approaches that could significantly improve the quality of taxi services.**

I. Introduction

Taxi service is of significant importance in metropolitan areas, however, even after decades of studies and improvements, its operational efficiency is still not quite satisfactory (e.g., a taxi fleet would spend over 50% of time idling in a typical day). The reason behind this inefficiency is how the taxi service is organized: in most cities, taxi services are delivered either by pre-arranged pick-ups (e.g., Dial-a-Cab service in UK and Singapore), street pick-ups, or taxi stand pick-ups. Although modern taxi-dispatch systems (e.g., see [1] and [2]) can satisfy pre-arranged pick-up requests very efficiently, significant portion of demands still have to be delivered by the latter two choices and they are highly dynamic and variable. This is one of the main reasons why taxi services are still delivered very inefficiently.

To counter this uncertainty and come up with sensible strategy, taxi drivers have to rely on their own experiences and very limited real-time information. In other words, most services not pre-arranged are delivered by drivers using local heuristics without global information. Thus, if we take a bird's-eye view over the urban area, there are always imbalances of supplies and demands in different sub-areas. Some areas might have far more taxis than potential demand while some areas might experience shortage in taxis and a surge in customer waiting time. This implies that simply by reducing these imbalances, the quality of taxi service could be improved without policy interventions (e.g., increase the quota for taxi licenses or modify taxi fares).

To perfectly balance taxi supplies and demands across the traffic network, we need to come up with a model that correctly describes taxi operations. Unfortunately, building such model would be infeasible even for networks of moderate sizes. The major difficulties come from the fact that the model needs to incorporate thousands of autonomous taxis and also stochastic demands at numerous locations (for street pick-ups, even demand locations are stochastic). Any attempt to build an analytical model would be hopeless, and significant investment and efforts in continuous data collection would be necessary if a serious simulation model is to be built.

In this paper, we propose a highly simplified model as the first step in addressing the above described issue. The model is simplified in terms of both number of players and number of choices. We assume that the model will be utilized by a single decision maker and he will choose between two alternatives: 1) serve the general network, and 2) serve a major taxi stand. Despite its simpleness, the model is adequate in demonstrating the weakness of the current system. This provides a good starting point for further studies in more complicated networks and also the design of mechanisms that would resolve the paradox faced by current system design.

The data required to drive the model is assumed to be provided by the Global Position System (GPS) that is installed on each and every individual taxi. To infer model parameters from the noisy and sometimes inaccurate GPS signals, we also propose a practical filtering and post-processing procedure. The applicability of our methodology is demonstrated by a real-world scenario.

The paper is organized as follows. In Section II, we describe the problem and its background, some related works are also discussed. In Section III, we formally introduce the service choice model for making service decision at a busy taxi stand. In Section IV, we discuss how to incorporate GPS traces into the abstract model we just described. In Section V, we present a real-world application based on GPS data obtained through our partner. Finally, we conclude the paper in Section VI.

II. Background

Taxi service has long been studied in the queuing and transportation literature. However, in almost all past research, a single operation mode is assumed. For taxi dispatching, Liao [1] and Lee et al. [2] provided recent reviews on how technology innovations like GPS and real-time communication infrastructure can be incorporated to provide high-quality services. For services delivered via taxi stand, Kendall [3] proposed a *double queue* model in which both customers and taxis form separate queues. Sasieni [4] later extended the model such that either customers or taxis might leave the queue if they have waited too long. For services delivered via street pickups, Yang and Wong [5] was the first

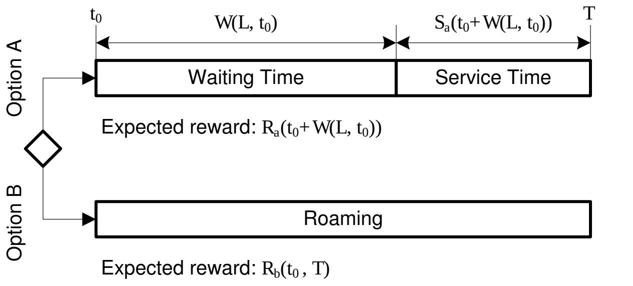


Fig. 1. The service choice model with two options. Option A serves the taxi stand, while Option B serves the general network.

to incorporate network topology into the analysis. This model was later extended in various different ways to consider previously neglected simplifications. For example, Wong et al. [6] extended the model to include congestion and demand elasticity; more recently, efforts have also been put in to consider multiple user classes and vehicle modes [7].

Despite these past efforts in coming up with individual models for different taxi operation modes, it is still not clear how one could integrate these models to obtain an unified model for making optimal servicing decisions involving multiple operation modes. To more realistically model the taxi service delivery, we need to build a model where taxi drivers could freely switch servicing modes between roaming the street, waiting for dispatch, or serving a taxi stand.

In the next section, we would propose one such integrated model that considers two operation modes: roaming the traffic network and serving a taxi stand.

III. The Service Choice Model

As mentioned in the previous section, we attempt to create a model that includes multiple modes of operations. More specifically, we assume that taxi drivers can either choose to serve the general traffic network by roaming, or alternatively, they can choose to wait at one particular taxi stand.

Generally speaking, serving a taxi stand could provide more stable income since demand pattern at a particular taxi stand is usually more recurrent and predictable than roaming on the road. However, the benefit of serving the taxi stand would vanish quickly with the length of the queue. Following this intuition, if the waiting time at the queue is monotonically decreasing in the queue length, the optimal policy for drivers should then be threshold-based. The main advantage of having a threshold-based policy is that it is very easy to understand (and thus easy to implement) for taxi drivers. For taxi fleet operators, the threshold-based policy also allows quick and efficient assessment of supply and demand balances at many locations simultaneously. This information could help them improve the quality of taxi service.

The service choice model is illustrated in Fig. 1. Following the earlier stated assumption, there are two options available in our service choice model: (a) serve the taxi stand, and (b) serve the general network. The reward for serving option A at time t is denoted as $R_a(t)$.The reward for serving option B from time t_1 to t_2 is denoted as $R_b(t_1, t_2)$. Note that for option A, the ending time of the service is beyond driver's control, therefore, to make a fair comparison, the starting and the ending times of the service at option B will be set to be the same as option A.

For each taxi that chooses option A at time t_0 and observes queue length L, we denote its waiting time as $W(L, t_0)$. At the end of the waiting period, a customer will be served, with expected revenue $R_a(t_0 + W(L, t_0))$ and expected service time $S_a(t_0 + W(L, t_0))$.

For option B, the expected reward $R_b(\cdot)$ can be viewed as the opportunity cost for choosing option A. Therefore, $R_b(\cdot)$ should be computed as the expected revenue from time t_0 to time T, where T is the expected service termination time for choosing option A. Suppose time-dependent expected revenue for the general network could be characterized as $r(t)$, we can then compute $R_b(\cdot)$ as:

$$R_b(t_0, T) = \int_{t_0}^{T} r(t)dt.$$

Without loss of generality, assume that $W(.)$ is a non-decreasing function in L. We can then obtain the threshold policy by finding the largest L that satisfies $R_a(\cdot) \geq R_b(\cdot)$. This problem can be formally defined as an optimization problem:

$$\begin{aligned} \max \quad & L & (1)\\ \text{s.t.} \quad & & \\ & R_a(t_0 + W(L, t_0)) \geq R_b(t_0, T) & \\ & L \geq 0 & \end{aligned}$$

In some cases, Problem (1) might have no solution. This simply implies that option B dominates option A, and serving taxi stand is never recommended.

To solve Problem (1) properly, we need expressions for $R_a(\cdot)$, $R_b(\cdot)$, and $W(\cdot)$. Since these expressions are usually scenario dependent, we will defer the discussion to Section V, when we explore a real-world case study.

IV. Incorporating GPS Traces

In Section III, a service choice model is proposed to provide a conceptual framework in reasoning about the optimal policy within a taxi service network. A critical piece that is missing in the model, as stressed in Section III, is the lack of information for characterizing important model parameters.

As GPS devices become more and more common in taxi fleets, they could be utilized to provide information regarding real-time taxi locations as well as paid trips. Unfortunately, there are at least two issues we need to address before using these GPS traces:

1) Civilian GPS devices, in particular, the ones installed in most taxis, are not very accurate. And sometimes these errors might be quite significant. Appropriate filtering and processing is necessary if we would like to use it as main source of information.

2) GPS devices are good for providing location information. However, important information like queue length at a taxi stand or elapsed waiting time at a queue cannot be sensed directly and need to be inferred.

To handle the first issue, we adopt a filter that could infer network topology from GPS traces and subsequently drop traces that are irrelevant to the studied taxi stand. The handling of the second issue is based on the same filtering procedure: after all traces are marked to the network topology, the boundary of the queue can be identified and the queue build-up and the waiting time can then be measured accordingly.

The above mentioned filtering and post-processing procedures are best illustrated with a numerical example. These techniques will be described in detail in the next section.

V. Case Study

A. Case Background

To illustrate how the service choice model could be implemented in the real-world scenario, we worked with a taxi fleet operator in Singapore and acquired its operational data. Each and every taxi in it fleet is equipped with a specially-designed mobile data terminal (MDT) that embeds GPS and allows bi-directional text communications. The MDT is designed to provide basic GPS functionalities and transmit back (to the central server) current vehicle positions every 30 to 60 seconds; for each paid trip, trip-related information is also sent back and recorded on the central server (e.g., type of job, starting and ending location, total fare). This data set thus allows us to perform the study on service choice at virtually any location in Singapore.

Since the focus of this paper is on how to make optimal decisions at a taxi stand, we pick a popular yet relatively isolated tourist spot so that unrelated traffic and noises could be minimized. The location we choose to study is the Night Safari, which is one of the most popular tourist attractions in Singapore. Besides geographically isolated, the Night Safari is also shown to be an ideal location for study since it only opens from 7pm to midnight and its volume of visitors is quite stable and predictable.

In this study, we use data collected from the month of September in 2008. In this data set, there are 95,390 GPS traces and they are plotted according to their coordinates in Fig. 2. From Fig. 2 we could clearly see that despite our best effort in choosing an isolated location, the selected data set is still very noisy, and we need to properly filter it before performing any meaningful analysis. The satellite map of the area covered by all GPS traces can be seen in Fig. 3.

B. Processing Noisy GPS Traces

As introduced in Section IV, by building the filtering procedure, we attempt to achieve two goals: 1) remove noisy GPS traces, and 2) infer unobservable yet important information like queue length and queuing time.

Majority of the noises in the data set are the result of GPS errors. By noticing the fact that GPS traces are generated by vehicles traversing the road network, we could filter or correct these errors by marking GPS traces to roads. For a sequence of GPS traces, we should be able to establish a reasonable moving pattern along the underlying road network (moving along a road segment A and then jump to road segment B, which is not directly connected to A, is an example of unreasonable moving pattern). In the case where most GPS traces are connected smoothly except for a few GPS traces, we could fix the problem by *marking* these straying traces to the correct road segment. However, if a sequence of GPS traces is particularly noisy, and no reasonable topological connection could be easily established, this sequence should be considered as noise and removed completely.

A prerequisite for the above correction and filtering process is to have a well-defined road network. Unfortunately, in a lot of cases, the exact road network within the queueing area might not be well-defined. For example, by comparing Fig. 2 and 3 we can see that although most taxis follow the

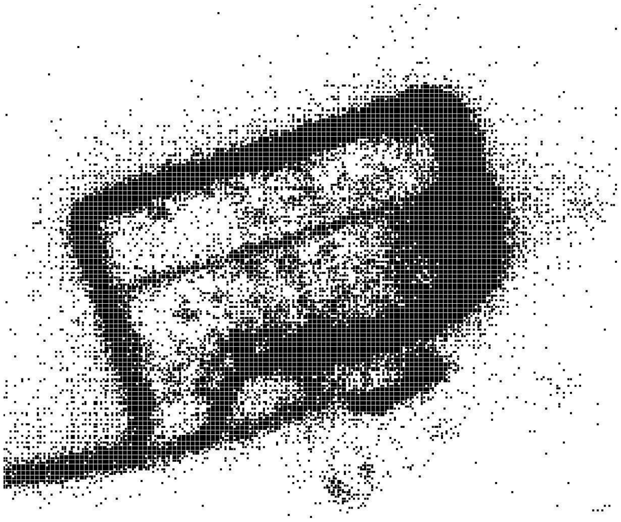


Fig. 2. The superposition of all GPS traces from September, 2008.

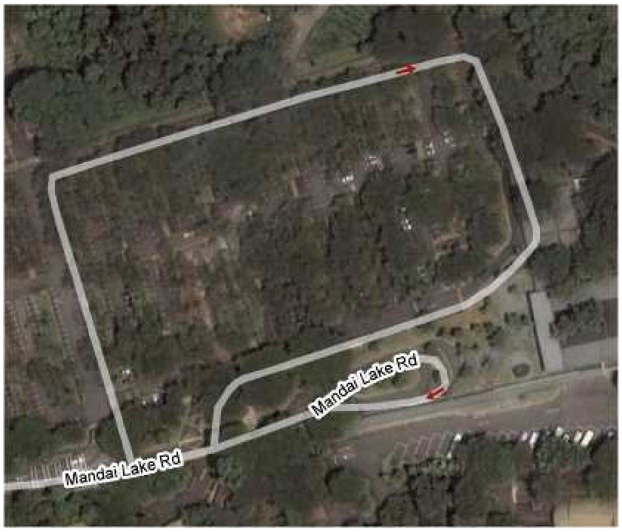


Fig. 3. The satellite map of the Night Safari area (from Google Maps).

main road illustrated in the map, there are several packs of GPS traces that do not belong to any road segment on the map. To prepare a better road network for the queueing area, we need a general procedure that can induce road network from GPS traces. In the following subsection, we would describe one such procedure that is implemented in our study.

C. Learning Road Network from GPS Traces

As stated earlier, the purpose of road network learning is to identify routes taken by queueing taxis even when these routes are not on the digital map. Despite the noisiness of GPS traces in Fig. 2, we can still easily recognize the road network underneath these traces. Based on this observation, we adopt an intuitive procedure that induces the existence of a road segment when there are enough GPS traces *near* it. A straightforward realization of this intuitive idea is the well-known k-means clustering algorithm. Bruntrup et al. [8] have proposed a similar clustering approach for map generation.

In general, the k-means clustering procedure divides points into k clusters (k is pre-determined) such that the sum of the squared distance between each point and the center of the cluster it belongs to is minimized. To put this procedure into our context, we can view these cluster centers as centers of road segments. When k is large enough, we should be able to approximate the real road network. Coming up with the optimal clustering can be quite computational intensive, and in many cases, not necessary. There are many efficient algorithms available for approximating k-means clustering; however, most of them are not very suitable for handling data with temporal orders. In our study, we adopt a heuristic that approximates the clustering reasonably well under temporal consideration:

1) Divide all points into N hypercube of equal size. For each hypercube n, count how many points belongs to it and denote that as p_n. Let the desired number of clusters be k. Set $i = 1$.
2) While $i \leq k$: pick the hypercube n with the most point, i.e., $n = \arg\max_n \{p_n\}$. Let the center of cluster i be computed as the mean of all points in hypercube n. Set $p_n = 0$, $i = i + 1$, and repeat this step.

Although the accuracy of this heuristic is not formally established, it works well empirically for our queueing study, and it also scales up nicely for very large data set.

The definition of a point in our context contains three components: longitude (x), latitude (y), and direction (θ). While x and y are GPS readings that are readily available, θ needs to be computed as follows: for all GPS traces generated by a single taxi during one visit, connect them following temporal order, θ is then measured as the angle from the connecting arc to the horizontal line in counterclockwise direction, as illustrated in Fig. 4. As noted earlier, the introduction of direction θ is the reason why we don't consider classical k-means approximation algorithms. To initialize the clustering algorithm, we divide the longitude and the latitude by using an interval of 5×10^{-5} degrees (roughly 5.6 meters). For direction, we simply divide it into four quadrants. With this division, we have $16,324$ hypercubes defined for our domain.

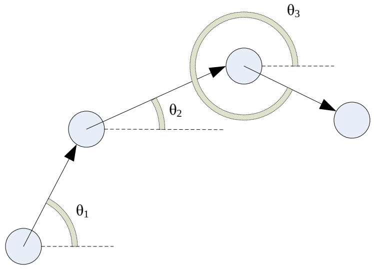


Fig. 4. The definition of θ.

In our numerical analysis, we perform two phases of k-means clustering; in the first phase, we follow the commonly suggested formula to determine k, and set $k = \lfloor\sqrt{95390/2}\rfloor = 218$. To further consolidate these centers for route generation, we perform the second-phase clustering, with $k = \lfloor\sqrt{218}\rfloor = 14$. The cluster centers identified in both phases are illustrated in Fig. 5.

To obtain the road network from the computed clusters, we would connect a cluster to its *closest* neighbor. The distance measure between two clusters i and j is defined as:

$$S_{ij} = \alpha\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \beta|\theta_i - \theta_j|.$$

By specifying a starting cluster i_0, we could find the next cluster to connect to by finding $j = \arg\min_j S_{i_0,j}$. By repeating this step iteratively, we could then identify a chain of cluster centers which can be viewed as the underlying road network that characterizes the queuing area for the taxi stand. Fig. 5(b) illustrates the connected clusters.

D. Filtering GPS Traces

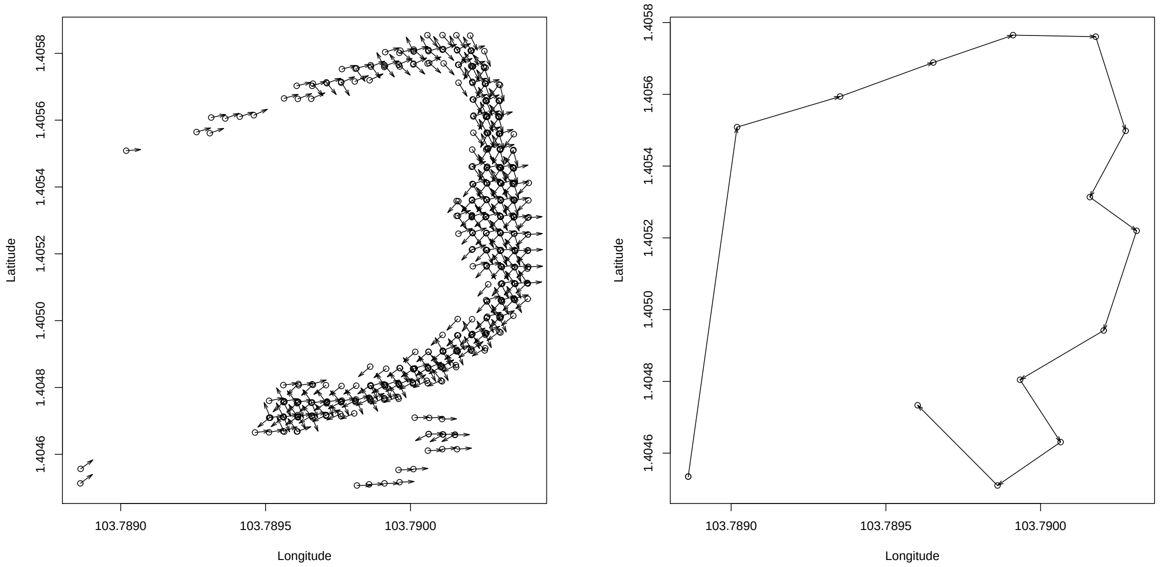
Given the road network definition (which can be learned from GPS traces or extracted from the digital map), we can then filter noisy GPS traces systematically. The idea is to match GPS traces to road segments and filter out GPS traces that are too far away from the matched road segments. More specifically, for each road segment, we would identify GPS traces that belong to it and establish a distribution of distances (from GPS traces to road segment). The filtering is done at the 90% confidence interval around the road segment.

For each taxi, by connecting valid traces belong to it, its queuing route can then be constructed. Here we would perform the final checking: the order in which road segments appear in a taxi's route should match the chain of clusters. The taxi trip should be dropped if it fails this final test.

Out of $95,390$ original GPS traces, around 85% of these traces survive the above filtering process. The remaining GPS traces are plotted in Fig. 6.

E. Identifying Queue Length and Waiting Time

Two most important pieces of information, queue length and waiting time, can also be inferred by using the learned road network. Based on the learned network topology, we define the first node as the entering point of the queue and the last node as the exiting point of the queue.



(a) 218 cluster centers obtained in the first phase.

(b) 14 cluster centers after the second phase.

Fig. 5. The two phases of the k-means clustering algorithm.

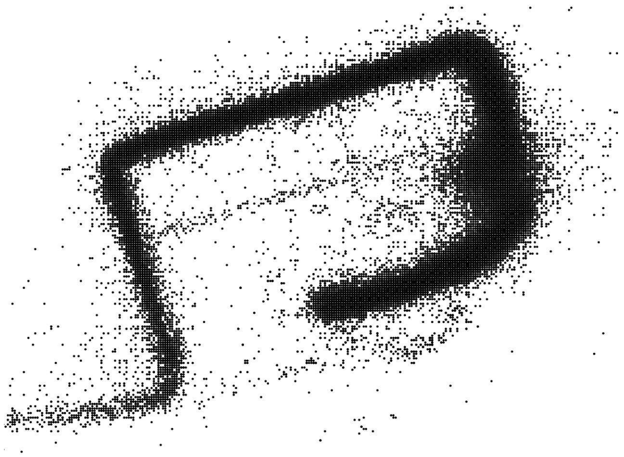


Fig. 6. The remaining GPS traces after filtering.

If we treat the GPS traces we receive from the central server as discrete events, then whenever a taxi crosses the entering point to enter the queue, the queue counter should be increased by one and the entering time will be recorded. Similarly, whenever a taxi crosses the exiting point to exit the queue, the queue counter will be decreased by one and the exiting time will be recorded as well. For this taxi, the queue length it observes as it enters the queue is the value of the queue counter when this taxi joins the queue. On the other hand, its waiting time is computed as the difference between the exiting time and the entering time.

F. Optimal Policy and Analysis

Before deriving the optimal policy for our service choice model, we need to first define $R_a(\cdot)$, $S_a(\cdot)$, $R_b(\cdot)$, and $W(\cdot)$. $R_a(t)$ and $S_a(t)$ are defined as the expected revenue and service time *per trip* out of the Night Safari at time t. $R_b(t_1, t_2)$, on the other hand, is defined as the expected revenue (in the general network) during the period $[t_1, t_2]$, and is simply computed by $R_b(t_1, t_2) = r(t_1)\ (t_2 - t_1)$, where $r(t_1)$ is expected revenue *per hour* at time t_1.

Finally, to estimate $W(L, t)$, we assume that all L taxis in the queue would be served at a fixed interval, $m(t)$. Therefore, $W(L, t) = m(t)L$, in which $m(t)$ denotes customer's inter-arrival time at time t.

With these function definitions, the optimal threshold L^* can be found by solving:

$$R_a(t) = \frac{S_a(t) + W(L^*, t)}{60}\, r(t). \qquad (2)$$

By substituting functions according to the above definitions, we have:

$$L^* = \frac{1}{m(t)} \left[\frac{60\, R_a(t)}{r(t)} - S_a(t) \right]. \qquad (3)$$

In our analysis, we define each time period to be one-hour long and compute relevant problem parameters accordingly. The summary of the analysis can be seen in Table I. To illustrate how real drivers perform against the optimal threshold policy we obtain, we also include the average queue length drivers observe at the point of entry and the average waiting time. As shown in Table I, the Night Safari taxi stand is always too crowded with taxis, particularly during the busier hours.

An interesting finding is that the average waiting time during the busiest hour (22:00-23:00) is actually the longest. This seemingly counter-intuitive result might be explained by the following reasoning:

TABLE I

OPTIMAL POLICIES FOR DIFFERENT TIME PERIODS

Time periods	19:00- 20:00	20:00- 21:00	21:00- 22:00	22:00- 23:00	23:00- 24:00
Avg. customer number (per hour)	3.3	11.8	27.6	28.2	10.8
Avg. fare from Night Safari: $R_a(t)$	14.6	15.0	14.8	15.1	17.5
Avg. service time from Night Safari: $S_a(t)$ (min)	21.5	22.3	22.1	21.2	20.2
Avg. customer inter-arrival time: $m(t)$ (min)	18.2	5.1	2.2	2.1	5.6
Avg. general revenue per hour: $r(t)$	20.1	19.2	18.9	18.8	16.1
Threshold Policy (L^*)	1.2	4.8	11.5	12.8	8.1
Avg. queue length	2.3	7.2	15.0	20.3	15.8
Avg. waiting time (min)	37.7	34.2	33.4	39.7	27.8

- Drivers are ill-informed on the queueing conditions and also other drivers' intentions.
- On knowing that 22:00-23:00 is the time period with most customers, all self-interested drivers would simultaneously decide to go to the Night Safari if they failed to reason strategically (considering other drivers' probable actions).

This analysis suggests that by releasing information regarding the taxi queues to appropriate drivers, overall queueing performance could be improved.

VI. CONCLUSION

In this paper we propose how to derive an optimal service choice model for taxi drivers serving a taxi stand. By incorporating GPS data, this model can be used in real-time and assist drivers in improving the delivery of their services (and in the process, earn more income). We demonstrate the potential of our methodology by applying our model to a real-world scenario. As expected, most drivers, when making their decisions independently, are making sub-optimal decisions.

We believe that the performance of a taxi fleet can be significantly improved if proper mechanism could be introduced so that optimal decisions computed by our model can be revealed to appropriate drivers. This area, along with the implementation and experimentation of our approach, are our major future research directions.

VII. ACKNOWLEDGMENT

The authors would like to thank C. Jason Woodard and Darshan Santani for their valuable suggestions and advices in the early stage of this work. This work was partially supported by the School of Information Systems, Singapore Management University.

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] D. G. Kendall, "Some problems in the theory of queues," *Journal of Royal Statistics Society B*, vol. 13, no. 2, 1951.

[4] M. W. Sasieni, "Double queues and impatient customers with an application to inventory theory," *Operations Research*, vol. 9, no. 6, pp. 771–781, 1961.

[5] 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.

[6] K. I. Wong, S. C. Wong, and H. Yang, "Modeling urban taxi services in congested road networks with elastic demand," *Transportation Research Part B: Methodological*, vol. 35, no. 9, pp. 819–842, 2001.

[7] K. I. Wong, S. C. Wong, H. Yang, and J. H. Wu, "Modeling urban taxi services with multiple user classes and vehicle modes," *Transportation Research Part B: Methodological*, vol. 42, no. 10, pp. 985–1007, 2008.

[8] R. Bruntrup, S. Edelkamp, S. Jabbar, and B. Scholz, "Incremental map generation with GPS traces," in *IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, 2005, pp. 574–579.