Reducing Inefficiencies in Taxi Systems

Chenguang Zhu[1] and Balaji Prabhakar[2]

***Abstract*— Taxi systems are perfect examples of supply-demand systems in which taxi vehicles and drivers constitute the supply side, while passengers hailing taxis are the demand side. However, various inefficiencies can be embedded within such a large-scale system, e.g. an excessive number of taxi vehicles, a shortage of taxi supplies after an event and long idle times with no passengers in taxis. These systemic inefficiencies are often overlooked in previous literature, which focuses on taxi dispatching mechanisms to satisfy short-term demand. In this paper, we address these inefficiencies and propose a novel model for the trip assignment problem based on network flow. Compared with existing methods, our model is much more scalable. This model is capable of assigning hundreds of thousands of trips to taxis over a long time interval, e.g. a shift of 12 hours. Furthermore, the trip assignment given by this model can effectively minimize the total number of required taxis while reducing incurred idle time. Experiments show that in our model, the number of required taxis to finish all observed trips in New York City is only 72% of the size of the current taxi fleet, while the average idle time incurred per taxi drops by 32%.**

I. INTRODUCTION

Taxi systems are important components of the urban transportation system because they complement public transport for their flexibility in routes, destinations and supply. A typical taxi system comprises thousands of taxi vehicles and hundreds of thousands of passengers each day. This presents a perfect example of a supply-demand system with available taxi cabs as the supply side and waiting passengers as the demand side.

In this system, each taxi driver can be viewed as an autonomous agent: the driver makes decisions to search for passengers according to his knowledge about the temporal and spatial distribution of demand. However, if we examine the actions of taxi drivers from the driver's perspective, these actions are usually local and myopic. Without much knowledge about the current supply-demand situation outside his neighborhood, it is very easy for a driver to make a suboptimal decision in the long run. For example, suppose a New York City taxi driver decides to go to Grand Central Station for his next potential passenger as the demand there is usually high. The passenger he picks up is heading to the Brooklyn District. The driver, unaware that a large wave of visitors will be taking taxis from Times Square to nearby hotels within the next 20 minutes, will take the trip and go to Brooklyn, possibly spending a lot of time looking for the next passenger there due to Brooklyn's low density of taxi rides. In fact, it would be better for another nearby taxi which would soon change shift in Brooklyn to take this trip. We therefore argue that the decisions made by individual drivers may be locally optimal, but from a systemic perspective, there is still plenty of room for improvement.

Collectively, suboptimal individual decisions lead to global inefficiencies within the taxi systems, including an excess or shortage of taxi service supplies and long idle time with taxis having no passengers inside. For example, although up to 60% of the daily traffic flow in certain areas in Hong Kong are generated by taxis, many of them are empty trips [16]. Empty trips result in low system utilization and exacerbate road congestion. Thus, reducing inefficiencies in taxi systems is a pressing and essential task facing urban transportation regulators and governments.

To solve these issues, previous studies have been focusing on mechanisms to assign trips to taxis [21], [18], [19], [9]. Existing works modeled the taxi system as an Autonomous Mobility on Demand (AMoD) system [11], [5]. In an AMoD system, the goal is to design smarter trip assignment plans in a real-time fashion based on recently emerging trip requests and current locations of available taxis. A typical feature of an AMoD system is its temporal and spatial locality. For example, [11] built local queues at each potential drop-off and pickup location to model taxi availability and passenger arrivals. [5] framed the taxi system as a multi-agent system, proposing a real-time model to re-schedule taxi service before order acceptance confirmation to accomplish the most recent trip requests. Although an AMoD system can usually be immediately deployed and evaluated in real scenarios, these locally optimal assignments do not necessarily lead to globally optimal efficiency. By investigating this issue over the complete time frame, we should be able to excavate recurring patterns in supply and demand as well as a systematic way to balance them.

To achieve this goal, we need a bird's-eye view of the whole taxi system. Fortunately, thanks to the rising popularity of sensing technology, many cities have equipped their taxis with GPS devices to record geo-location and trip information, thereby enabling us to design a trip assignment mechanism that boosts systemic efficiency. This mechanism also allows the system to operate with fewer taxis and each taxi achieves a higher utilization rate with less idle time wasted on the road. Specifically, we want to reassign all observed trips to taxis during a time period (e.g. a shift of 12 hours), following the exact start/end time and location requirements. Our objectives are hereby two-fold:

1) Reassign trips to minimize the number of taxis to accomplish all passengers' requests;

[1]Chenguang Zhu is researcher in Microsoft AI+Research, One Microsoft Way Redmond, WA 98052, USA zcg.cs60@gmail.com

[2]Balaji Prabhakar is Professor in Departments of Electrical Engineering and Computer Science of Stanford University, 350 Serra Mall, Stanford, CA, 94305, USA balaji@stanford.edu

2) Reassign trips to taxis to minimize total idle time when using the minimum number of taxis.

The first objective is important for taxi commissions and government because they are eager to obtain an appropriate estimate of the required taxi fleet size for effective entry regulation and congestion control, while the second objective is directly correlated with the operational efficiency of taxi systems and also the earnings for drivers. It is important to note that we only minimize total idle time when the number of taxis used is minimum. Otherwise, a trivial solution where each trip is assigned to a unique taxi will have zero idle time.

To achieve the above objectives, we design a trip assignment model based on network flow. This model captures both temporal and spatial properties of taxi movement and passenger trips. Compared with previous approaches, our model is much more scalable, capable of assigning hundreds of thousands of taxi trips daily in big metropolises. We evaluate our model on New York City taxi trip datasets. One of the findings is that our model can accomplish all observed taxi trips in New York City with only 72% of the current yellow taxi fleet. Furthermore, the average idle time per taxi drops by 32%.

The rest of this paper is organized as follows. Section II frames the trip assignment problem as a combinatorial optimization problem and defines related notations. Section III examines the previous literature. We develop our model in Section IV and evaluate the model in Section V. Section VI offers concluding remarks.

II. PRELIMINARIES

We denote the set of taxi trips to be assigned as $\mathcal{T} = \{a_1, a_2, ..., a_m\}$. Each trip a_i is represented by a tuple: $a_i = (ts_i, te_i, ls_i, le_i)$, where ts_i and te_i are the start/end time of the trip, while ls_i and le_i are the start/end location of the trip. Suppose N taxis will serve these trips. An assignment of taxi trips is a mapping function $\mathcal{F} : \mathcal{T} \to [N]$. Under such a mapping function $\mathcal{F}$, each taxi i is assigned a sequence of taxi trips in time order: $a_{i_1}, a_{i_2}, ..., a_{i_{k_i}}$. A valid mapping function $\mathcal{F}$ should satisfy the following two conditions:

1) $te_{i_j} < ts_{i_{j+1}}, \forall 1 \leq i \leq N, 1 \leq j < k_i$
2) $dist(le_{i_j}, ls_{i_{j+1}}) \leq v_{max} \cdot (ts_{i_{j+1}} - te_{i_j})$

where $dist(A, B)$ is the routing distance from location A to B and v_{max} is the maximum driving speed. We set v_{max} to 25 miles per hour (40.23 km per hour) in the experiment.

The first condition regulates that two adjacent trips made by the same taxi cannot overlap temporally, while the second condition makes sure that it is feasible to reach the start location of the next trip after finishing the current trip.

Our goals are two-fold. First, we want to minimize the number of required taxis N to validly assign all trips. Second, we aim to minimize the total idle time incurred: $T_{idle} = \sum_{i=1}^{N} \sum_{j=1}^{k_i - 1} (ts_{i_{j+1}} - te_{i_j})$.

III. PREVIOUS LITERATURE

Previous literature focused on taxi dispatching strategy for efficient trip assignment: [15] aimed to reduce waiting time for passengers and boost trip success rate; [13] designed a route-recommendation mechanism to maximize driver's profits; [12] assigned trips to guarantee fairness within a group of competing drivers; [22] designed methods to optimize passengers' waiting time for taxis.

The goal of these models is to devise a more efficient real-time scheduling system. However, a more systematic approach of the trip assignment task requires optimization over the complete time frame. Such approaches can help determine an appropriate taxi fleet size to serve all passengers and reduce inefficiencies in the taxi system such as the total time taxis spend in idling status.

To this end, a network-based model was introduced in [17] to determine system performance measures at equilibrium such as vacant taxi movements and taxi utilization. The authors also computed the minimum taxi fleet size to ensure the existence of a stationary equilibrium state. In [2], the taxi system was framed as a multi-agent system and a model was proposed to enhance the utilization rate of taxi systems. [8] modeled routing behaviors of vacant taxicabs to explore more efficient passenger-finding strategies.

To the best of the authors' knowledge, the work most similar to ours comes from [20], in which a path cover model was proposed to reassign trips to taxis. We hereby present a brief introduction. Based on the notions defined in the previous section, there are m taxi trips and each trip a_i is represented by a tuple (ts_i, te_i, ls_i, le_i). These trips are then described by a graph $G = (V, E)$, where each node in V represents a trip, and an edge $i \to j$ exists if and only if trip a_i and trip a_j can be finished sequentially by a taxi. [20] proved that G is directed and acyclic. Furthermore, the nodes corresponding to a taxi's trips form a directed path in G.

Subsequently, minimizing the number of required taxis to finish all trips is equivalent to finding the minimum number of non-intersecting paths to cover every node in G. This problem can be solved by maximum matching by constructing a bipartite graph of $2m$ nodes, with details in [20].

Nevertheless, this approach has a major drawback: the scalability. In the constructed graph, the number of nodes $|V|$ is linear with the number of trips m; the number of edges $|E|$ is quadratic with m. However, the number of taxi trips in a city can be quite large. For example, in New York City, taxis make more than 400,000 trips on an average weekday, but a typical maximum matching algorithm on bipartite graphs, e.g. HopCroft-Karp [7], has a time complexity of $O(\sqrt{|V|}|E|)$. Consequently, we cannot solve the model within reasonable amount of time.

To tackle the model's limited scalability, [20] worked around the problem by assigning trips in short time intervals of 10 minutes. This approach greatly reduces the significance of the work. The reason is that trip assignment in the previous time interval cannot be automatically merged with the assignment in the next time interval, as many taxis are still taking trips. Moreover, the model can never obtain a real assignment of trips to each taxi over a day, nor can we calculate the total amount of idle time under the assignment.

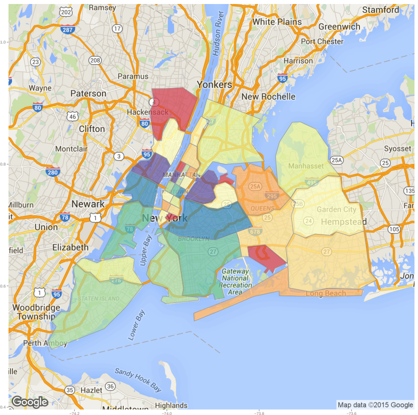


Fig. 1: Region partition of New York City.

In the next section, we will introduce a new model to solve the scalability issue which enables us to deliver assignment plans for hundreds of thousands of taxi trips within a minute.

IV. OUR MODEL

A. Discretization

The first step in our model is to discretize both trip time and trip locations. Time over a day is sliced into 1-minute bins. We then round down the start time and round up the end time of each trip to the boundaries of time bins. For trip locations, we partition a city into regions. The granularity of the discretization should be determined based on a balance between solution quality and computational efficiency. In the experiments, we partition New York City into 36 regions according to district boundaries and main roads (Fig. 1). In Section V, we will discuss in detail the impact of these discretization hyperparameters on the model.

After discretization, each trip is represented by (ts_i^d, te_i^d, ls_i^d, le_i^d). As multiple trips may share the same discretized tuple representation, the whole dataset is described as $\{(ts^d$, te^d, ls^d, le^d, $count)\}$, where $count$ is the number of trips that start from region ls^d in time bin ts^d and end at region le^d in time bin te^d.

B. Network flow model

To depict the movement of taxis in spatial-temporal dimensions, we build a network flow model, based on the concept of Time Expanded Network [3]. The basic idea is that the movement of a taxi is characterized by a unit flow in the network, capturing both available and occupied statuses.

The flow network is denoted by $G = (V, E)$. The nodes and edges are defined as follows.

Nodes. Nodes in G correspond to discretized time and regions: node $p_{t,l} \in V$ stands for discretized time t and region l. A flow passing through $p_{t,l}$ indicates that a taxi is in region l at time t.

Edges. Each directed edge $(x, y) \in E$ carries an upper-limit $u_{x,y}$ and lower-limit $l_{x,y}$ for flow value to represent the allowable number of passing taxis. Additionally, a cost $c_{x,y}$ is associated with the edge (x, y) to represent the incurred idle time[1] in minutes. In total, there are three types of edges:

1) To depict the action of taxis taking trips, edge $p_{ts^d,ls^d} \rightarrow p_{te^d,le^d}$ corresponds to the data entry (ts^d, te^d, ls^d, le^d, $count$), with $l_{x,y} = u_{x,y} = count$ and $c_{x,y} = 0$. In other words, exactly $count$ units of flow travel through this edge, with no idle time incurred.
2) Idle taxis can stay in the same region. We construct edge $p_{t,l} \rightarrow p_{t+1,l}$ for each discrete time t and region l, with $u_{x,y} = \infty$, $l_{x,y} = 0$, $c_{x,y} = 1$. It indicates that a taxi staying in region l from t to $t+1$ incurs an idle time of 1 minute.
3) Idle taxis can move from region l to another region $l' \neq l$. We construct edge $p_{t,l} \rightarrow p_{t+t_{l\rightarrow l'},l'}$ for each (t, l, l'), with $u_{x,y} = \infty$, $l_{x,y} = 0$, $c_{x,y} = t_{l\rightarrow l'}$. It indicates that each taxi moving from region l to l' incurs an idle time of $t_{l\rightarrow l'}$ minutes[2].

Finally, to complete the construction, we add a source node S and a sink node T. We add edges $S \rightarrow p_{t,l}$ and $p_{t,l} \rightarrow T$ for all t, l where a trip starts/ends in region l at time t, with $u_{x,y} = \infty$, $l_{x,y} = 0$, $c_{x,y} = 0$. It follows that a unit flow from S to T represents the movement and trip sequence of one taxi. It is worth noting that we can associate idle time to edges pertinent to S/T and add edges to corresponding nodes to incorporate initial locations and times where taxis would start to work.

An example network flow model is shown in Fig. 2. One trip starts at 15:01 in region 1 and ends at 15:05 in region 2. Two trips start at 15:00 in region 2 and end at 15:05 in region 1.

With the flow network, we aim to find the minimum feasible flow plans to minimize the number of taxis required to finish all observed trips.

Feasible flow. A feasible flow plan $f : E \rightarrow \mathbb{R}^+$ assigns $f_{x,y} \geq 0$ to each edge $(x, y) \in E$ such that the following two conditions are satisfied:

1) $l_{x,y} \leq f_{x,y} \leq u_{x,y}$ (*capacity constraint*)
2) $\sum_{x:(x,y)\in E} f_{x,y} = \sum_{x:(y,x)\in E} f_{y,x}, \forall x \in V, x \neq S, T$ (*conservation of flows*)

The size of flow plan f is defined as: $|f| = \sum_{(S,i)\in E} f_{S,i}$, which is also equal to $\sum_{(i,T)\in E} f_{i,T}$. In our network, a feasible flow plan of size N corresponds to a valid assignment of all trips to N taxis.

Minimum feasible flow (FF). To determine the minimum feasible flow size, we convert the problem into a decision problem. To judge whether a flow of size N exists for G, we define G_N as G plus an edge $T \rightarrow S$ with both cost and lower-limit as 0. The upper-limit of this edge is N. Note that G_N has no source or sink. It is obvious that a flow of size N exists for G if and only if a flow of size N exists for G_N.

[1]The network flow model can work with any non-negative bounded edge cost such as idle distance and vehicle emissions.

[2]One trick in reducing the number of edges is that $p_{t,l} \rightarrow p_{t+t_{l\rightarrow l'},l'}$ is constructed only when some trip starts at time $t + t_{l\rightarrow l'}$ in region l'. The reason is that a taxi can hold movement to region l' until the last minute: it starts the next trip as soon as it moves to region l'.

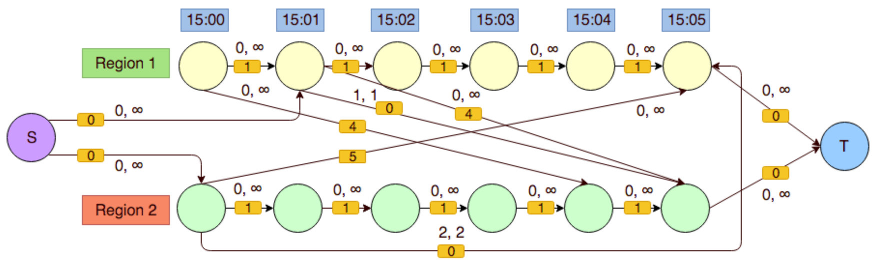


Fig. 2: An example network flow model for a two-region-six-minute taxi system. On each edge, the two numbers separated by comma are the lower-limit and upper-limit of flows. Costs are shown in yellow boxes.

A push-relabel algorithm [6] can solve the problem for G_N in time $O(|V|^2|E|)$.

Furthermore, if a feasible flow of size N exists for G, another feasible flow of size $N+1$ exists as well. We thus employ binary search to determine the minimum feasible flow size, i.e. the minimum number of required taxis, N^*. As N^* cannot exceed the number of trips, the overall time complexity is $O(log(\text{number of trips}) \cdot |V|^2|E|)$.

Minimum cost minimum feasible flow (MCFF). After determining the minimum number of required taxis N^* to finish all trips, we proceed to compute the minimum cost. Here, the goal is to find a feasible flow plan of size N^* such that the total cost C is minimized:

$$C(\{f_{x,y}\}) = \sum_{(x,y)\in E} c_{x,y} \cdot f_{x,y} \tag{1}$$

An existing algorithm [10] can solve the problem in polynomial time $O(|E|log|V|(|E| + |V|log|V|))$. The resulting cost is the minimum total idle time of these N^* taxis under optimal assignment.

A note on graph size. The time complexity of the above flow algorithms is correlated with the network size. The number of nodes $|V|$ is the product of the number of regions and the number of time bins, which is a constant. The number of edges, $|E|$, is:

- Edges of type (i): $\min([\text{number of trips}], |V|^2)$
- Edges of type (ii): $|V|$
- Edges of type (iii): $|V| \times [\text{number of regions}]$
- Edges from S or to T: $2|V|$

Total number of edges:

$$O(\min\{[\text{number of trips}], |V|^2\} + |V| \times [\text{number of regions}]\}) \tag{2}$$

As $|V|$ is a constant, $|E|$ is upper-bounded by a constant. In practice, $|E|$ is much less than that in the model in [20]. For example, for a 12-hour shift worth of data, our model contains 51K nodes and 624K edges, while the path cover model in [20], if implemented, would have 430K nodes and 23 billion edges.

After obtaining a feasible flow of size N^*, the trips can be assigned to taxis by repetitively applying breadth first search (BFS) to generate unit flows from node S to node T. Each unit flow corresponds to the working path of one taxi. In the end, we assign trips $a_{i_1}, a_{i_2}, ..., a_{i_{k_i}}$ to taxi i.

However, since we discretized the map into regions, it may be infeasible for a taxi to go to the next trip in time. For example, the drop-off location of trip a_{i_j} and the pickup location of trip $a_{i_{j+1}}$ could be in the same partitioned region, but the actual distance is so far that a taxi cannot move between these two locations in required time—we have assumed that all trips must be finished according to its time schedule.

To solve this problem, we add feasibility checking into our algorithm. We first set a speed limit of 25 mph for a taxi to move between locations. If the next trip is infeasible to reach from the current trip for a taxi, the next trip is removed from the assignment. Consequently, each taxi will only execute a feasible trip sequence. We then set those removed trips as new input into the network flow model and assign more taxis to finish these trips. The process is repeated until all trips are assigned to taxis. The details of this iterative algorithm are described in Algo. 1. This algorithm is guaranteed to end, since in each round, at least the first trip assigned to each taxi is accepted. In experiments, we observed that the iterative algorithm ended within 6 rounds for all models.

V. EXPERIMENT

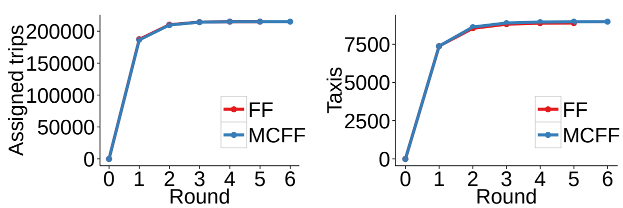
We evaluated the network flow model on New York City taxi trip dataset [4]. This dataset contains information about all trips made by yellow taxis in New York City in 2013. Each trip is described by an entry including hashed driver's license number, hashed medallion number, pickup / drop-off location and time, fares, tips and distances.

We first took taxi trips from a normal day in 2013, May 15, which was a Wednesday. As taxis work in 12-hour shifts in New York City, we focused on trips starting from 4AM to 4PM. In data-cleaning phase, we filtered out trips with duration shorter than 1 minute or longer than 1 hour. We ended up with 214,805 trips made by 12,366 taxis.

We built the network as mentioned in the previous section, with 51,842 nodes and 624,067 edges. We ran Algo. 1 to obtain feasible trip assignments. To estimate distance between locations in feasibility check, we obtained the information from data: the trip records contain distance driven for each trip. Based on this information, we built a distance table for location pairs in discretized grids of 0.003×0.003 in latitude and longitude. It turns out that 99.8% of all distance queries

Algorithm 1: Iterative algorithm for trip assignment

Set all trips as unassigned.
$Round \leftarrow 0$
while *any unassigned trip exists* **do**
  $Round \leftarrow Round + 1$
  For unassigned trips, construct network $G = (V, E)$.
  Run minimum feasible flow / minimum cost feasible flow algorithm on G.
  Use BFS to assign temporally ordered trips $a_{i_1}, a_{i_2}, ..., a_{i_{k_i}}$ to each taxi i.
  foreach i **do**
    $prev \leftarrow 1$
    Trip a_{i_1} is assigned to taxi i.
    for j `from` 2 `to` k_i **do**
      if $a_{i_{prev}}$ *and* a_{i_j} *can be finished sequentially* **then**
        Trip a_{i_j} is assigned to taxi i.
        $prev \leftarrow j$
      end
    end
  end
end



(a) Number of assigned trips **(b)** Number of taxis

Fig. 3: Number of assigned trips and taxis in Algo. 1 in each round, for FF and MCFF.

could be accomplished in this table. For the rest few requests, we used Euclidean distance.

The experiments were run on a single machine with 2.3 GHz Intel Core i7 and a memory of 16 GB 1600 MHz DDR3. The algorithm FF, which minimizes the number of required taxis, ran for 28.334s, while the algorithm MCFF, which also minimizes the total idle time, ran for 259.753s.

As shown in Fig. 3, FF ended within 5 iterative rounds and MCFF ended within 6 rounds. About 90% of all trip assignments were completed in the first round for both models. This shows that the iterative algorithm is very effective and efficient in practice.

Average statistics. As shown in Table I, the assignments given by FF and MCFF only require 8,887 and 8,972 taxis respectively to finish all observed trips (the numbers of taxis are a little different because of the iterative assignment mechanism in Algo. 1), about 28% fewer than the number of taxis in reality. With fewer taxis, the efficiency is also higher: the average idle time per taxi for the 12-hour shift drops from 4.1 hours in real data to 3.4 hours for FF and further down to 2.8 hours for MCFF, with a reduction of

TABLE I: Statistics about taxi assignments from real data, minimum feasible flow model (FF) and minimum cost feasible flow model (MCFF). The best statistics in each category are in bold.

Statistics	Real data	FF	MCFF
Number of taxis	12,366	**8,887**	8,972
Number of trips per taxi	17.4	**24.2**	23.9
Idle time per taxi (hours)	4.1	3.4	**2.8**
Earnings per taxi ($)	252.6	**351.5**	348.1
Profit per taxi ($)	124.4	**219.0**	216.1

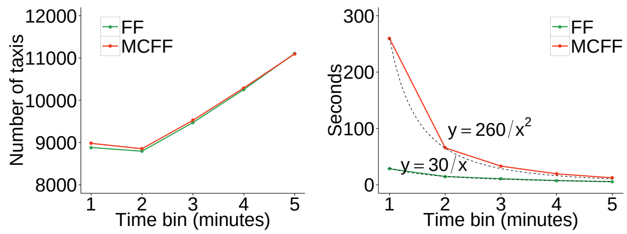
17.1% and 31.7% respectively. Although the total effective working time, i.e. the time taking passengers, is the same for all three assignments, the results indicate that under more efficient assignments, the system can waste less time in idling status with fewer taxis.

This higher efficiency also brings more income for drivers: the earnings per taxi in FF increases by 39.2%, from $252.6 up to $351.5; the earnings per taxi in MCFF increases by 37.8% to $348.1. By taking cost into consideration, we also calculated the profit of each taxi. The cost of operating a taxi comes from two main sources: gas consumption and rent. The average gas price was $3.602 per gallon and the taxi fuel economy was 29 miles per gallon in 2013 [1], while the price of renting a taxi for a 12-hour shift was $120 in 2013 [14]. By considering these costs, both FF and MCFF nearly increased the average profit per taxi by $100.

As we employed 1 minute as the length of time bin and 36 regions for partition of New York City in the experiment, we now investigate the impact of these discretization hyper-parameters on the performance of our model in terms of both effectiveness and efficiency.

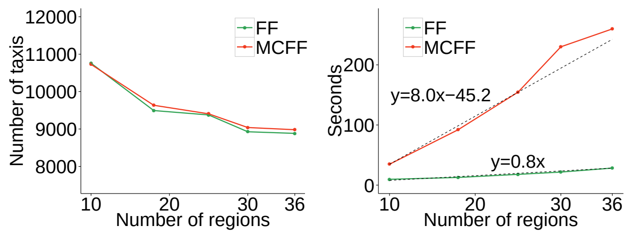
Time parameter. We experimented with time bin length from 1 minute to 5 minutes. As more than half of the New York City taxi trip records have trip start / end time right at the minute boundary—probably due to the precision of logging devices—we did not experiment with time bins shorter than 1 minute. Fig. 4 shows the number of taxis required to finish all trips and the running time for both FF and MCFF for different time bin lengths. Fig. 4(a) indicates that the number of required taxis of our models is minimal when the time bin has a length of 1 or 2 minutes. When the length of time bin further increases, the model requires more taxis. The reason is that with coarser time partition, trips close in time may be deemed infeasible. For example, if the time bin has a length of 5 minutes, then two trips 1 minute apart might not be assigned in our model because the rounded-up ending time of the first trip may be after the rounded-down start time of the second trip.

On the other hand, the time complexity of the models is related with the length of time bins because the number of nodes in the flow network grows linearly with the number of time bins. Fig. 4(b) indicates that both models run much faster with longer time bins. We fitted curves and found out that FF's running time inversely correlates with the length of time bin, while MCFF's running time inverse-squarely correlates with the length of time bin. Therefore,



(a) Number of taxis **(b)** Running time

Fig. 4: (a) The number of taxis and (b) running time of FF and MCFF models with different time bin length for discretization. The number of partitioned regions is fixed to be 36. The dotted line are the fitted curves: inverse for FF and inverse-square for MCFF.



(a) Number of taxis **(b)** Running time

Fig. 5: (a) The number of taxis and (b) running time of FF and MCFF models with different number of partitioned regions. The length of time bin is fixed to be 1 minute. The dotted line are the fitted curves: linear for both FF and MCFF.

it is important to seek a balance between computation time and the quality of optimization.

Location parameter. We experimented with different partitions of New York City: 36, 30, 25, 18 and 10 regions. The result is presented in Fig. 5. Fig. 5(a) indicates that our models can generate better trip assignment with finer geographic granularity, because a finer partition allows more accurate estimation of movement time between regions. However, the models are not very sensitive with the number of partitioned regions: with only 18 regions the number of taxis is 9,493, a 6.9% increase from the case of 36 regions.

Similar to time parameters, the number of nodes in the flow network grows linearly with the number of partitioned regions, hence affecting the running time. Fig. 5(b) shows that the running time is approximately linear with the number of partitioned regions in location discretization.

VI. CONCLUSION

In this paper, we aim to reduce inefficiencies embedded within taxi systems. We develop a novel model to assign trips to taxis over a long time period, e.g. a shift of 12 hours. The model is based on network flows and it can minimize the number of taxis required to finish all observed trips while reducing the incurred idle time. Compared with previous work, our model is much more scalable and can generate detailed assignment plans for hundreds of thousands of trips. Experiments on New York City taxi data validate the effectiveness of our model: (i) only 72% of the current yellow taxi fleet are required to finish all trips, (ii) the average idle time per taxi drops by 32%. For future work, we plan to combine our model with taxi dispatching strategies to achieve more effective dispatching mechanisms.

REFERENCES

[1] M. Bloomberg and D. Yassky. 2014 taxicab fact book. `http://www.nyc.gov/html/tlc/downloads/pdf/2014_taxicab_fact_book.pdf`.

[2] S.-F. Cheng and T. D. Nguyen. Taxisim: A multiagent simulation platform for evaluating taxi fleet operations. In *Proceedings of the 2011 International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 14–21.

[3] T. G. Crainic and G. Laporte. Planning models for freight transportation. In *European journal of operational research*, number 3, pages 409–438. Elsevier, 1997.

[4] B. Donovan and D. B. Work. New york city taxi trip data (2010-2013). `http://dx.doi.org/10.13012/J8PN93H8`, 2014.

[5] A. Glaschenko, A. Ivaschenko, G. Rzevski, and P. Skobelev. Multi-agent real time scheduling system for taxi companies. In *AAMAS'09*.

[6] A. Goldberg and R. Tarjan. A new approach to the maximum-flow problem. In *Journal of the ACM (JACM)*, number 4, pages 921–940. ACM, 1988.

[7] J. Hopcroft and R. Karp. An n^5/2 algorithm for maximum matchings in bipartite graphs. In *SIAM Journal on computing*, number 4, pages 225–231, 1973.

[8] X. Hu, S. Gao, Y.-C. Chiu, and D.-Y. Lin. Modeling routing behavior for vacant taxicabs in urban traffic networks. In *Transportation Research Record: Journal of the Transportation Research Board*, number 2284, pages 81–88, 2012.

[9] S. Ma, Y. Zheng, and O. Wolfson. T-share: A large-scale dynamic taxi ridesharing service. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 410–421. IEEE, 2013.

[10] J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. In *Operations research*, number 2, pages 338–350. INFORMS, 1993.

[11] M. Pavone. Autonomous mobility-on-demand systems for future urban mobility. In *Autonomes Fahren*, pages 399–416. Springer, 2015.

[12] S. Qian, J. Cao, F. L. Mouë, I. Sahel, and M. Li. Scram: A sharing considered route assignment mechanism for fair taxi route recommendations. In *KDD'15*, pages 955–964.

[13] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong. A cost-effective recommender system for taxi drivers. In *KDD'14*, pages 45–54.

[14] F. Salmon. New yorkers love uber. but is uber good for new york? `http://fusion.net/story/175479/uber-new-york-taxis/`.

[15] W. Shen and C. Lopes. Managing autonomous mobility on demand systems for better passenger experience. In *arXiv preprint arXiv:1507.02563*, 2015.

[16] H. Yang, Y. W. Lau, S. C. Wong, and H. K. Lo. A macroscopic taxi model for passenger demand, taxi utilization and level of services. In *Transportation*, number 3, pages 317–340. Springer, 2000.

[17] H. Yang and S. Wong. A network model of urban taxi services. In *Transportation Research Part B: Methodological*, number 4, pages 235–246, 1998.

[18] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun. Where to find my next passenger. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 109–118. ACM, 2011.

[19] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2390–2403, 2013.

[20] X. Zhan, X. Qian, and S. V. Ukkusuri. Measuring the efficiency of urban taxi service system. In *UrbComp'14*. ACM, 2014.

[21] D. Zhang, L. Sun, B. Li, C. Chen, G. Pan, S. Li, and Z. Wu. Understanding taxi service strategies from taxi gps traces. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):123–135, 2015.

[22] X. Zheng, X. Liang, and K. Xu. Where to wait for a taxi? In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 149–156. ACM, 2012.