TASeT: Improving the Efficiency of Electric Taxis With Transfer-Allowed Rideshare

Yunfei Hou, *Student Member, IEEE*, Weida Zhong, Lu Su, *Member, IEEE*, Kevin Hulme, Adel W. Sadek, and Chunming Qiao, *Fellow, IEEE*

***Abstract*—We consider a promising application for electric taxis (eTaxis) in transportation cyberphysical systems. The new rideshare scheme introduced herein takes into consideration both the limited battery of eTaxis and the user requirements. In the proposed eTaxi-sharing system, a passenger may share a taxi with others to enjoy a reduced fare and can potentially transfer from one eTaxi to another before reaching her destination, as long as her total trip time is within the maximum she specifies to be tolerable. Transfers are restricted to only take place at the designated (safe and convenient) battery charging stations scattered around the city. When an eTaxi comes to a charging station to pick up/drop off a transfer passenger, the eTaxi's battery can be charged. In this paper, we address a new optimization problem called Transfer-Allowed Shared eTaxis (TASeT), whose goal is to schedule an eTaxis service and find optimal rideshare and transfer plans to maximize the system throughput in terms of the number of passengers served by the taxi service within a given time period. A mixed-integer programming (MIP) formulation is presented to solve TASeT, along with an efficient greedy heuristic. Aside from large-scale simulation, we also present a case study that utilizes real taxi traces collected from the city of Shanghai, China. Compared with the nontransferable taxi-sharing (NTT) case, our solution could improve the number of served passengers and shared travel distance by 22% and 37%, respectively, during rush hours.**

***Index Terms*—Electric vehicle, intelligent transportation systems, transfer-allowed taxi sharing, transportation cyberphysical systems, transshipment.**

I. Introduction

Electric taxis (eTaxis) are emerging on the global market. Electric vehicles exhibit significant environmental benefits over their regular fossil fuel counterparts and have, thus, received wide support from various governments around the globe. Large cities such as Paris, New York, Beijing, and Shenzhen all have pilot programs for eTaxis as a means to improve transportation system sustainability. For example, New York City plans to replace one third of its taxi fleet (about 4000 taxis) with eTaxis by 2020 [1]. Electric vehicle is also getting popular in the rideshare community. Since 2011, the Autolib' car-sharing program (Paris, France) has 1800 electric vehicles with 4000 dedicated charging points. For taxis, however, while green and clean energy is an important consideration, other essential requirements, including high availability and minimum cost for passengers, still cannot be ignored. Moreover, an electric taxi with a fully charged battery can only travel about 100 miles (compared to 300 miles for a common taxi with a full tank). Hence, eTaxis will require three times as frequent recharging (compared to refueling). When it comes to battery charging, even with a direct-current quick-charging system such as CHAdeMO or SAE Combo, it will take about 30 min to reach 80% of the battery's capacity. Clearly, battery and related charging issues have been major challenges for the large deployment of eTaxis.

Additionally, a more efficient taxi service is still in need. It becomes more and more difficult to hail a taxi in large cities, particularly during rush hours. Instead of adding more taxis to crowded urban traffic, researchers have looked into various ways to leverage communication technologies to improve taxi service. Specifically, ridesharing has been considered as a potential approach to improve taxi utilization [2], [3].

By studying existing eTaxi fleets and investigating how current taxi systems operate, we made the following observations. (Without causing confusion, taxis/eTaxis and requests/passengers will be used interchangeably in the remainder of this paper.)

First, traditional taxi sharing usually operates in a nontransferable model, which cannot fully utilize vehicles with available space. In other words, a taxi carpooling passenger is only considered to be served by one vehicle. As shown in Fig. 1, let us assume that two passengers A and B need taxi services and taxis 1 and 2 already have passengers onboard (their routes are marked in blue). With traditional nontransferable taxi sharing (NTT), only passenger A can be served by Taxi 1 since they are going to the same direction. Passenger B, however, cannot be served because taxi 1's route does not include B's destination and taxi 2's route does not include B's pickup location. In this case, we propose to allow a passenger to transfer from one taxi to another at an intermediate location such that multiple taxis can cooperatively serve more carpooling requests.

Second, charging stations and the time needed for charging the battery can be utilized to support taxi sharing and transfer. In

Manuscript received October 29, 2015; revised March 20, 2016 and June 17, 2016; accepted July 6, 2016. Date of publication July 19, 2016; date of current version December 14, 2016. This work was supported by the Transportation Informatics University Transportation Center (TransInfo). The review of this paper was coordinated by Guest Editors.

Y. Hou is with the School of Computer Science and Engineering, California State University San Bernardino, San Bernardino, CA 92407 USA (e-mail: hou@csusb.edu).

W. Zhong, L. Su, and C. Qiao are with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: weidazho@buffalo.edu; lusu@buffalo.edu; qiao@buffalo.edu).

K. Hulme is with the Center for Engineering Design and Applied Simulation, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: hulme@buffalo.edu).

A. W. Sadek is with the Department of Civil, Structural and Environmental Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: asadek@buffalo.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2016.2592983

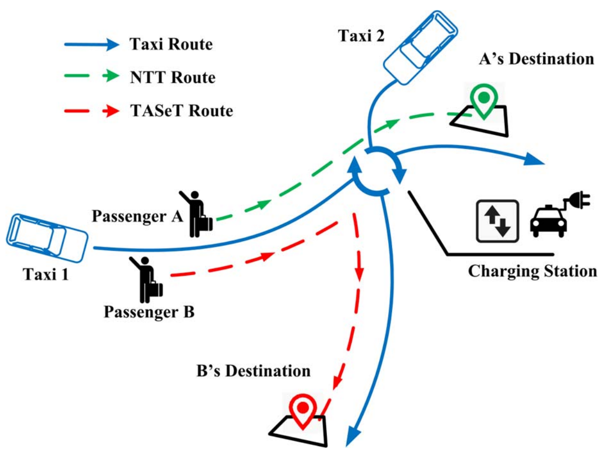


Fig. 1. Example illustrating NTT and TASeT eTaxi.

many large cities in the USA, the European Union, and Japan, the number of public charging stations is catching up with the number of gas stations [4], and we are expecting that the charging networks will keep growing in the coming years. These charging stations are usually easy to access, which makes them convenient for taxi sharing. Additionally, while waiting for taxi-sharing passengers at a charging station, an eTaxi can have its battery charged for a short time. Referring to Fig. 1 again, passenger B can be served if transfer is allowed because she can be picked up by taxi 1 at her source, then transfer to taxi 2 at a charging station (i.e., a transfer location), and finally be dropped off at her destination. Upon arriving at a charging station both taxis 1 and 2 can be charged for a while if time permits.

We propose herein a new dispatch system for eTaxis with transfer and sharing. In this system, we introduce an optimization problem that we refer to as Transfer-Allowed Shared eTaxis (TASeT). For TASeT, given a) a number of passengers; and b) a list of taxis, our objective is to maximize the number of requests served with carpooling during a given time period (e.g., rush hours) where each passenger is allowed to make *at most one transfer* between taxis per trip. Additionally, we are considering the need for recharging of an eTaxi. Passengers will transfer to other taxis at one of the charging stations that are scattered throughout the city. An eTaxi will be directed to charge its battery while waiting for a transfer passenger or when its battery level is too low to serve the next passenger.

Many existing vehicular ad hoc network (VANET) applications can be integrated into the TASeT system to further improve the performance and user experience such as traffic prediction [5], [6], passenger demand analysis [7], taxi cruising guidance [8], [9], and charging station placement [10], [11].

There are three main advantages of the proposed TASeT system. First, given the same number of taxis, many more passengers can be served during rush hours or bad weather, which could greatly improve the taxi services. Second, due to rideshare, the total travel distance of all taxis can be much reduced compared with the case where no taxi sharing is allowed. This will help reduce energy consumption and alleviate traffic congestion. Third, with an appropriate pricing scheme, not only passengers can save money but the taxi drivers (and their company) can also reduce their cost and improve their profits.

The major contributions of this paper are as follows.

1) To the best of our knowledge, no existing work has looked into transfer-allowed taxi-sharing paradigm with eTaxis.
2) As expected, there are many design issues to consider in order for the proposed eTaxi-sharing system to work reasonably well. The mathematical formulation of the problem needs to account for the transfer and ride-sharing process, and hence, it is significantly more challenging than classic routing problems. In this paper, we formulate the TASeT problem by mixed-integer programming (MIP) and prove it to be NP-hard. We introduce an effective rideshare planning strategy with practical considerations.
3) We conduct comprehensive simulations to evaluate the performance of the proposed solutions. Additionally, we extend two existing solutions to enable them to handle eTaxis and compare their performances. Our results show that TASeT can significantly improve the utilization of taxi services and reduce energy consumption. We also present a case study in the city of Shanghai by using GPS trajectories collected from taxis and provide useful insights.

The remainder of this paper is organized as follows. In Section II, we discuss the related work on taxi sharing. In Section III, we present the overview of the proposed taxi-sharing system. We formally describe the TASeT problem and formulate it with MIP in Section IV, and then, we present a greedy heuristic algorithm to efficiently solve the TASeT problem. We present other taxi-sharing models in Section V and report the results of simulations in Section VI. A case study in the city of Shanghai is presented in Section VII, and Section VIII concludes this paper.

II. Related Work

Traditional research on eTaxi dispatch system usually focuses on how to mediate/offset the negative impact of the frequently performed charging task. In a few studies, it has been proposed to extend traditional system with charging plans and considered constraints on the limited travel distance [12], [13]. In our work, instead of considering charging and serving passengers as separate issues, we propose an integrated operation model that utilizes charging time to serve carpooling requests, thus has the potential to further improve the efficiency of eTaxi services. We also attempt to harness the existing research on taxi-sharing and transshipment problem. These works are briefly reviewed subsequently before the proposed algorithm is described in Section III.

A. Taxi Sharing

Most of the existing works on taxi sharing and general carpooling looked into the problem by assuming that one request can only be served by one vehicle, i.e., transfer at intermediate locations is not allowed. Quite a few taxi-sharing systems have been proposed. In [2], a dual-side heuristic searching algorithm along with a lazy shortest path strategy was designed to solve the taxi-sharing problem. Similarly, coRide [3] system introduced a brand-and-bond algorithm for rideshare planning. Other recent efforts in designing taxi-sharing algorithms include [14]–[17]. In this paper, we consider such kind of taxi service that allows passengers to transfer at most once, and we show that this service model will be particularly beneficial for electric vehicles.

To the best of our knowledge, only three papers [18]–[20] have considered allowing transfers in carpooling. In [19], three heuristic algorithms were proposed, each of which gave a tradeoff in terms of effectiveness and computational cost, to find rideshare plan with transfers. However, time windows for passenger pickup and delivery were not considered in their model. In [20], the problem was modeled with time windows and a genetic algorithm was provided to find a rideshare plan for a single passenger. In comparison, TASeT schedules carpooling for multiple passengers rather than an itinerary for a single passenger. In our previous work, we proposed transfer-allowed carpooling (TAC) [18]. However, it cannot be directly applied to TASeT: First, in TAC, a driver provides a ride along her way from her source to the destination, whereas in TASeT, we need to consider where to dispatch a taxi to pick up one or more passengers. Second, charging issues were not studied in TAC. On the other hand, there are a few inconsequential similarities between TASeT and TAC problems. For example, the previous study on TAC showed that allowing just one transfer improves the carpooling efficiency most, whereas allowing more than one transfer does not bring any noticeable benefits. In the proposed eTaxi-sharing system, passengers will also make at most one transfer (and such a transfer time must be tolerable to passengers).

B. Pickup and Delivery Problem With Transfers

TASeT problem can be viewed as an extension of the general pickup and delivery problem with a transfer option (PDPT) and time windows. The classical problem that is most related to the current work is the transshipment problem [21], which considers the shipment of goods to an intermediate destination and then from there to yet another destination. As presented in the following section, much more attention to details is necessary for making use of the PDPT to solve the TASeT problem. The major differences between PDPT and TASeT include objective, time constraints at transfer locations, and additional constraints regarding taxi service model and electric vehicles.

Various research has been published with focus on solving practical instances of PDPT: In three seminal papers by Coltin and Veloso [19], [22], [23] and Coltin's Ph.D. thesis [24], heuristic algorithms were proposed to schedule rideshare routes with transfer. They also tested their scheduling algorithms with mobile robots. Masson *et al.* proposed an adaptive large neighborhood search algorithm for PDPT in [25]. They later improved the neighborhood-based heuristics with a constant-time method to efficiently insert request through transfer points [26]. In most of the aforementioned works, the typical objective function to be minimized is the total distance traveled by all vehicles; more complicated objective functions were considered in [27]–[29], which included both the total waiting and travel time of passengers, combined with some measure of the operation cost and were weighted differently according to practical considerations. Many authors have introduced the idea of transfer into the dial-a-ride problem (DARP), and most of them added transfer points and proposed heuristic strategies [30], [31]. In DARP, traditional research typically targets at minimizing the fleet size to satisfy all the passenger demands. In comparison, TASeT tries to serve as many passengers as possible with a given number of eTaxis. Their differences are summarized in Table I.

TABLE I
COMPARISON BETWEEN TASeT AND PDPT

	TASeT	**PDPT**
Optimization objective	Max. number of passenger request	Min. travel distance / time cost
Time constraints at transfer	Yes	N/A
Serve all requests	No	Yes
Number of transfer allowed	Once	Multiple times
Constraints on battery	Yes	N/A

Several mathematic formulations have been proposed for the PDPT problem; however, only a handful of papers have so far addressed the PDPT problem involving time constraints for transfer passengers, and they are fairly recent [23], [34], [35]. Our formulation is based on the work in [33]. A complete formulation of PDPT was proposed in [32], in which passenger transportation and used decision variables for both arcs and nodes in the underlying network were considered. They used arch variable for the vehicle flows and binary node variables to track the passengers. For each transfer node, they split the node into two to handle precedence relationships for transfer passengers. In comparison, we use flow variables for the flow of passengers and do not split transfer nodes (which avoids adding additional nodes and links between them). This results in a fewer number of decision variables and constraints in our formulation.

III. SYSTEM DESIGN

The proposed eTaxi-sharing system works according to Fig. 2: Upon receiving a taxi request, the dispatch center will first try to find a vacant taxi to pick up the passenger based on taxis' current locations. If there is no vacant taxi available at that time, the dispatch center will ask the passenger whether she would consider sharing a taxi with others and provide an estimated pickup time based on historical data. If the new passenger is willing to carpool, the dispatch system will look for a rideshare plan. After a rideshare plan is found, the dispatch center will contact the corresponding taxi and provide a reduced taxi fare to passengers (which will also increase the taxi driver's income). The taxi driver and her current passenger can decide whether to carpool with the new passenger upon receiving the new rideshare plan. After all parties agree on the shared fare, the system will reply to the new passenger. During this process, the new passenger is expected to wait for a while before the dispatch system could find a rideshare plan. To provide a better

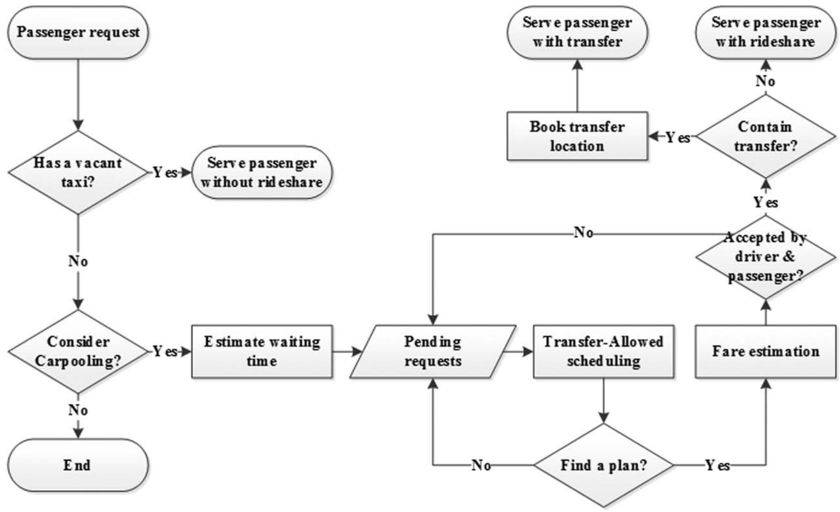


Fig. 2. Workflow to process a request.

user experience, the system should provide a real-time status update and allow passengers to cancel carpooling requests at any time.

There are four major components in the system: 1) dispatch module, which schedules taxis and provides rideshare itinerary for corresponding passengers; 2) communication module, which keeps track of taxis and connects the system with passengers through mobile device; 3) routing module, which provides taxis with routing suggestions based on real-time traffic condition; and 4) station management module, which manages charging stations for taxis and support passenger transfers.

In this paper, we focus on the dispatch module. To simplify the problem, we have the following assumptions of other modules: Communication is always reliable; routing decisions are based on the shortest travel time; and charging stations are only available for taxis and works in a first-come–first-served manner.

IV. Problem Description

Here, we define the proposed TASeT problem. We consider taxi-sharing planning by a dispatch center, which has the information about the current status of taxis offering rides (including taxis' current locations, routes, and battery status, etc.) and passengers that need to be served. As stated previously, we consider the TASeT model, in which it is possible for a passenger to transfer from one taxi to another, and charging stations also serve as transfer locations. For practical reasons, we look for rideshare plan that passengers will transfer at most once during their trips and eventually will get to their destinations within their tolerable delay time. Given a limited number of taxis, the dispatch center tries to serve as many passengers as possible in a given time period. Here, we first provide an optimal solution for the problem. We will formulate the TASeT problem and prove it to be NP-hard. To handle the problem in real-world scenarios, we also introduce a greedy heuristic algorithm at the end of this section.

A. *Mathematical Formulation of TASeT*

We formulate the TASeT problem with MIP. Conceptually, we build a directed graph with additional data on its nodes and arcs: Each node represents a location of the road network, and each arc is associated with travel time and distance between two locations. Additionally, we use an overlaid network to represent taxi routes and passenger routes.

Let $G(N, A)$ be a directed graph with node set N and arc set A. An arc from node i to node j is denoted by $ij \in A$. We use $T \subseteq N$ to denote the set of transfer nodes in G (i.e., set T represents the location of charging stations). Let K be the set of taxis, the status of each taxi $k \in K$ at the time of dispatch is defined as a three-tuple $(u_k, b_k, o(k))$ in which u_k denotes the seat capacity of taxi k, b_k represents the battery status (to simplify the battery model, the battery level is represented in terms of expected travel distance with remaining battery), and $o(k) \in O$ denotes the start location (i.e., the initial location at the time of a dispatch). O is the set of start location of all taxis, $O \subseteq N$.

Let R be the set of requests indexed by $r = 1, 2, \ldots, |R|$. Each request can be defined as a three-tuple $(p(r), d(r), q_r)$, in which the first two parameters $p(r)$ and $d(r) \in N$ are the pickup and dropoff locations of request $r \in R$, respectively, and q_r is the number of passengers associated with this request. Let $D = \{d(r)|\forall\, r \in R\}$ be the set of dropoff nodes and $P = \{p(r)|\,\forall\, r \in R\}$ be the set of pickup nodes. If two requests have a common pickup or dropoff location, the corresponding node is duplicated. In this model, each request is associated with exactly one pickup and dropoff pair. In graph $G, N = O \cup T \cup P \cup D$, each node in N is connected to all the other nodes, with the exception that each node in O is only connected to all the nodes in P. In other words, a taxi may visit multiple pickup, transfer, or dropoff locations but not any start location of other taxis.

In this formulation, taxi route and passenger route are modeled as network flows in which the $\{0, 1\}$ variables represent the binary decision *taxi* k (*or request* r) that *uses link* ij. Specifically, we use two decision variables, i.e., $x_{ij}^k = \{0, 1\} k \in K$,

$ij \in A$, and $y_{ij}^{kr} = \{0,1\} k \in K, r \in R, ij \in A$. Let $x_{ij}^k = 1$ if the taxi k uses arc ij and $x_{ij}^k = 0$ otherwise. Let $y_{ij}^{kr} = 1$ if request r is served by taxi k on the arc ij; otherwise, $y_{ij}^{kr} = 0$. We state the MIP model as follows:

$$\text{Maximize} \sum_{k\in K}\sum_{r\in R}\sum_{i:ij\in A, j\in d(r)} y_{ij}^{kr} q_r$$

Subject to

$$\sum_{j:ij\in A} x_{ij}^k \leq 1 \quad \forall k \in K \quad \forall i = o(k) \tag{1}$$

$$\sum_{j:ij\in A} x_{ij}^k - \sum_{j:ji\in A} x_{ji}^k \leq 0 \quad \forall k \in K \quad \forall i \in T \cup D \tag{2}$$

$$\sum_{j:ij\in A} x_{ij}^k - \sum_{j:ij\in A} x_{ji}^k = 0 \quad \forall k \in K \quad \forall i \in P \tag{3}$$

$$\sum_{k\in K}\sum_{j:ij\in A} y_{ij}^{kr} \leq 1 \quad \forall r \in R \quad \forall i = p(r) \tag{4}$$

$$\sum_{k\in K}\sum_{j:ij\in A} y_{ij}^{kr} - \sum_{k\in K}\sum_{j:jl\in A} y_{jl}^{kr} = 0 \quad \forall r \in R$$

$$i = p(r),\ l = d(r) \tag{5}$$

$$\sum_{k\in K}\sum_{j:ij\in A} y_{ij}^{kr} - \sum_{k\in K}\sum_{j:ji\in A} y_{ji}^{kr} \leq 0 \quad \forall r \in R$$

$$\forall i \in T \cup D \tag{6}$$

$$y_{ij}^{kr} \leq x_{ij}^k \quad \forall ij \in A \quad \forall k \in K \quad \forall r \in R \tag{7}$$

$$\sum_{r\in R} q_r y_{ij}^{kr} \leq u_k x_{ij}^k \quad \forall ij \in A \quad \forall k \in K \tag{8}$$

$$x_{ij}^k \in \{0,1\} \quad \forall ij \in A \quad \forall k \in K \tag{9}$$

$$y_{ij}^{kr} \in \{0,1\} \quad \forall ij \in A \quad \forall k \in K \quad \forall r \in R. \tag{10}$$

Unlike PDPT problem that usually focuses on minimizing operational cost or travel time, the objective of the TASeT problem is to *maximize the number of passengers that can be served by the taxi fleet*. Constraint (1) enforces that each taxi is scheduled at most once from its origin. This implies that TASeT works in a best-effort manner: If there are not too many requests, not all the taxis have to be dispatched. Constraints (2) and (3) maintain the flow conservation at transfer and dropoff locations. We use "$\leq$" instead of "$=$" because we assume that the current dispatch will be ended once a taxi drops off all of its passengers.

Similar to constraints (1)–(3) on taxi flow (which represents taxi routes), constraints (4)–(6) specify passenger flow (i.e., passenger routes). Constraint (4) enforces that each request is served at most once. Constraint (5) guarantees that passengers will reach their destinations if they are picked up by a taxi. To allow passengers to transfer from one taxi to another, constraint (6) maintains the flow conservation at the designate transfer nodes. Similar to constraint (2), by using "$\leq$," we allow the case that one taxi ends its trip at a dropoff or transfer node.

Constraints (7) and (8) link the taxi flow and request flow. Constraint (7) states that, if there is a request flow on an arc, there are some taxi flows on the same arc. Constraint (8) enforces that, on each and every arc, each taxi would not carry more passengers than its seat capacity. Additionally, we define binary decision variables with constraints (9) and (10).

Because the graph G is very likely to be cyclic, constraints (11)–(14), shown below, eliminate subtour in a taxi's route. We choose to use method provided in [33] because it provides tight bounds. Let $z_{ij}^k = 1$ if node i precedes (does not have to be immediately) node j in the route of the taxi k. Constraint (11) enforces that, if there is a taxi flow, there should be no cycle in the taxi route. Constraints (12) and (13) preserve the precedence relationship between two nodes

$$x_{ij}^k \leq z_{ij}^k \quad \forall ij \in A \quad \forall k \in K \tag{11}$$

$$z_{ij}^k + z_{ji}^k = 1 \quad \forall ij \in A \quad \forall k \in K \tag{12}$$

$$z_{ij}^k + z_{jl}^k + z_{li}^k \leq 2 \quad \forall ij \in A \quad \forall k \in K \tag{13}$$

$$z_{ij}^k \in \{0,1\} \quad \forall ij \in A \quad \forall k \in K. \tag{14}$$

To capture the time constraints in the taxi service, we define a few additional notations. For an arc $ij \in A$, let t_{ij} be the estimated time for a taxi to travel from node i to j and l_{ij} be the corresponding distance. We use a_i^k and d_i^k to denote the arrival and departure times of taxi k at node i. Then, if a taxi k chose to travel on arc ij, i.e., $x_{ij}^k = 1$, it must satisfy $a_j^k \geq d_i^k + t_{ij}$ and $d_j^k \geq a_j^k$ to handle the time sequence. Constraints (15) and (16) enforce these constraints by using the Big M method (M is a large positive constant) as follows, and constraint (17), shown below, enforces that vehicles will not run out of battery before finishing serving all of its assigned passengers:

$$d_i^k + t_{ij} - a_j^k \leq M\left(1 - x_{ij}^k\right) \quad \forall ij \in A \quad \forall k \in K \tag{15}$$

$$a_j^k \leq d_j^k \quad \forall j \in N \quad \forall k \in K \tag{16}$$

$$\sum_{i:ij\in A} l_{ij} x_{ij}^k < b^k \quad \forall k \in K. \tag{17}$$

Two time windows $[s_{p(r)}, e_{p(r)}]$ and $[s_{d(r)}, e_{d(r)}]$ are associated with a request r that has pickup node $p(r)$ and dropoff node $d(r)$. The pickup window $[s_{p(r)}, e_{p(r)}]$ defines how long a passenger is willing to wait for a taxi. We assume that, at $s_{p(r)}$, a passenger needs a taxi, and after a maximum waiting time (MAXWAIT), she will choose other means of transportation; thus, we have $e_{p(r)} = s_{p(r)} + \text{MAXWAIT}$. The dropoff window $[s_{d(r)}, e_{d(r)}]$ enforces that the total travel time of a passenger will not exceed her maximum tolerable trip time. Let TRIPTIME be the travel time for a request that is served by a single taxi without any transfer and MAXDELAY be the maximum additional trip time a passenger would accept. Start time and end time of the dropoff window can be calculated by $s_{d(r)} = s_{p(r)} +$ TRIPTIME and $e_{d(r)} = s_{d(r)} +$ MAXDELAY. For simplicity, we write constraints (18) and (19), shown below, using the start time and end time of the time windows

$$s_{p(r)} \leq a^k_{p(r)},\ d^k_{p(r)} \leq e_{p(r)} \quad \forall k \in K \quad \forall r \in R \tag{18}$$

$$s_{d(r)} \leq a^k_{d(r)},\ d^k_{d(r)} \leq e_{d(r)} \quad \forall k \in K \quad \forall r \in R. \tag{19}$$

We use a logical counter to handle transfer. Let $c^{kl}_{ir} = 1$ if the request r is transferred from taxi k to taxi l, $l \neq k$, at some transfer node $i \in T$, and $c^{kl}_{ir} = 0$ otherwise. We have the following constraints:

$$a^k_i - d^l_i \leq M\left(1 - c^{kl}_{jr}\right) \qquad \forall r \in R \qquad \forall i \in T \qquad \forall k, l \in K, \quad k \neq l \tag{20}$$

$$\sum_{j:ij \in A} y^{kr}_{ji} + \sum_{j:ji \in A} y^{lr}_{ij} \leq c^{kl}_{ir} + 1 \qquad \forall r \in R \qquad \forall i \in T \qquad \forall k, l \in K, \quad k \neq l \tag{21}$$

$$\sum_{i \in T} \sum_{l \in K, l \neq k} c^{kl}_{ir} \leq 1 \qquad \forall r \in R \qquad \forall k \in K \tag{22}$$

$$c^{kl}_{ir} \in \{0, 1\} \qquad \forall r \in R \qquad \forall i \in T \qquad \forall k, l \in K, \quad k \neq l. \tag{23}$$

Constraints (20) and (21) together enforce that, at node i, a request r could transfer from taxi k to l only if taxi k arrives before the departure of taxi l. Note that this implies that the pickup taxi can arrive at the transfer node either earlier or later than the arrival of the dropoff taxi. For each request, constraint (22) allows passengers to transfer at most once at some transfer nodes. Finally, constraint (23) defines c^{kl}_{ir} as a binary variable.

B. Computational Hardness

Theorem 1 TASeT is NP-Hard: Without elaborating on the formal proof, we show that TASeT is NP-hard as follows.

The decision version of TASeT can be stated as follows: Is there a set of rideshare plans such that the number of served requests is less than a given positive number p?

To show TASeT $\in$ NP, suppose that we are given a set of taxi rideshare plans. Clearly, we can verify in polynomial time if the number of served requests is less than p. To prove TASeT is NP-hard, we reduce classic vehicle routing problem (VRP) [34] to TASeT. VRP focuses on finding the minimum total route cost in which all requests are served and is known to be NP-complete. Additionally, given an algorithm that can efficiently find the minimum total route cost, the related decision problem of "finding the maximum number of requests that can be served" can be readily solved (by converting VRP to a decision problem). Without considering transfer and battery constraints, VRP is reducible to TASeT; thus, TASeT is NP-hard.

C. Greedy Heuristic Algorithm for TASeT

The complexity of TASeT problem comes from two aspects: 1) Decide the order for serving the passengers, and 2) find potential rideshare plans for each passenger, which may cause changes on the existing plans. Although the MIP model can find optimal solutions and have theoretical values, it is only feasible for a small number of requests at a time, which makes it unsuitable to be directly applied to the real world. Here, we propose a heuristic strategy that addresses aforementioned challenges with practical considerations and can accommodate a large number of taxi requests. Due to the limited space, we only outline the major steps of the algorithm and omit other inessential details.

Since the delay caused by detour and waiting is the major concern of taxi carpooling, it is natural for us to design rideshare plans taking into consideration the processing time and the requested service time. The proposed heuristic algorithm for TASeT tries to pick up as many passengers on time as possible and transfer them between taxis. The algorithm also strives to balance between reducing the trip delay for the passengers and increasing the battery charging time during transfer for taxi drivers. Additionally, we need to consider the availability of charging stations and allow multiple threads of the algorithm to run in parallel to improve system response speed. The algorithm works in two stages. First, the dispatch system seeks to serve passengers without carpool. In the second stage, the algorithm proceeds with carpooling schedules. Traditional taxi dispatch stops after the first stage, whereas in TASeT, we use traditional dispatch strategy to decide which passengers to be picked up first by vacant taxis. The passengers' itineraries may be changed in the second stage if they are willing to participate in carpooling and a rideshare plan can be found. Specifically, the heuristic of TASeT has the following procedure.

In line 1 of Algorithm 1, we implement traditional taxi dispatch strategy that schedules taxis based on the shortest waiting time for the passengers. The consideration here is to reduce taxis' idling time to serve more passengers as soon as possible. Please note that the DispatchVacantTaxi() function can be changed to any dispatch strategy that a taxi company may currently be using (e.g., shortest waiting time and shortest cruising distance). By making the new carpooling service compatible with the existing system, it will deliver similar user experience even if a passenger is not willing to carpool with others.

Algorithm 1 Transfer-Allowed Scheduling

procedure Dispatch (K, R)
1: *lst1* $\longleftarrow$ DispatchVacantTaxi (K, R)
2: $K' \longleftarrow$ taxis that accept carpool
3: $R' \longleftarrow$ remaining passengers that accept carpool
4: *lst2* $\longleftarrow$ GreedyTASeT(K', R')
5: **for each** $r \in R$
6: **if** r involves in $itnry1 \in$ *lst1* && in $itnry2 \in$ *lst2*
7: *lst1* $-= itnry1$
8: **return** *lst1* $\cup$ *lst2*

The heuristic of TASeT is shown in Algorithm 2. In line 1, we maintain a lookup table LT to keep track of potential taxis that can be arranged for transfer at a charging station. $LT[t][k]$ stores the earliest arrival time for taxi k to charging station t without violating the tolerable delay for k's current passengers (i.e., check if setting t as a new via point will exceed the tolerable delay for onboard passengers); otherwise, $LT[t][k] = \infty$. Statements in line 8 and line 10 also use LT to look up for potential transfer locations and relaying taxis.

Algorithm 2 Greedy Heuristic for TASeT

procedure GreedyTASeT(K, R)
1: $initialize\ LT$
2: $itinerarylist = \emptyset$
3: $R' \longleftarrow$ sort R by the number of possible pickup taxis
4: **for each** $r \in R'$
2: $templist = \emptyset$
5: **for each** k can pickup r
6: $posbitnry \longleftarrow$ FindPlanwithNoTransfer$(r.src, r.dst, k)$
7: $templist+ = posbitnry$
8: $chargestationlist \longleftarrow$ potential transfer locations for k
9: **for each** $t \in chargestationlist$
10: **for each** k' that k can transfer to at t
11: *itnryp1*$\longleftarrow$ FindPlanwithNoTransfer$(r.src, t, k)$
12: *itnryp2*$\longleftarrow$ FindPlanwithNoTransfer$(t, r.dst, k')$
13: $itnrywithchargeplan \longleftarrow$ CreateChargePlan (*itnryp1,itnryp2*)
14: $templist+ = itnerywithchargeplan$
15: sort templist by delay and transfer time
16: $selected \longleftarrow$ itinerary with longest transfer time in the top $\beta\%$ shortest delay
17: update k, k' and LT
18: $itinerarylist+ = selected$
19: **return** $itinerarylist$

The line 3 in Algorithm 2 performs request selection, which decides the order of requests to be processed. We rank the requests in descending order by the number of taxis that are possible to pick them up in time (without considering transfer or destination). The request that has the fewest eligible taxis is served first. Here, we use this approach to roughly estimate the probability that a request can be served by some taxis. The basic idea is that the less taxis that the passenger can be picked up by, the higher priority she should receive, because such a request will hardly get a chance to be served if other requests have already taken up many carpooling resources.

In line 6, the algorithm searches for nontransfer itineraries, and continues to look for one-transfer itineraries in line 11–13. Note that charging stations are considered in line 13 as transfer locations. Both routing constraints and the availability of a charging station will be evaluated. If a feasible rideshare plan can be found, we will check if the corresponding charging station is available at the time of scheduled transfer. If it is available, which means a possible taxi-sharing plan is found for the passenger. The system will reserve charging stations after the algorithm decides the itineraries for each passenger.

For one passenger, line 11 and 12 in Algorithm 2 may find multiple rideshare itineraries. To select one from all possible plans, we consider the following two criteria in Algorithm 3: a) To serve a passenger with fewer detours, we only keep the top $\beta\%$ plans with shortest delay induced by a detour (where β is a configurable parameter and is set to 20% in our simulation) for further consideration; and b) in an effort to provide a buffer time for passengers to connect to the next taxi, while also allowing taxis to charge more of its battery, we will choose the plan with longest transfer time among the remaining candidate plans selected by a). Note that for such an itinerary, the total delay of the selected plan (which includes both detour and transfer time) still needs to be within the maximum tolerable delay for all participating passengers.

Algorithm 3 Find Rideshare Plan Without Transfer

procedure FindPlanwithNoTransfer(r, k)
$n \longleftarrow$ number of existing passengers in k
$lst = \emptyset$
pkorder $\longleftarrow$ plan to pickup r after picking up n passengers
for each $dropofforder$
lst+ = CreateItinerary$(pkorder + dropofforder, k)$
return FindItineraryWithMinimumDelay(lst)

For a given request (specified by its source and destination as in Algorithm 2) and a possible pickup taxi, we calculate the feasible rideshare plan, as shown in Algorithm 3. We use two heuristics that are based on practical considerations: First, we assume that the new passenger is the last one to board a taxi so that pickup time will not change for the previous scheduled passengers (if a taxi already has n passengers on board, there will be $(n + 1)!$ possible dropoff orders). Second, among all delivery orders that satisfy time constraints, we choose the one that has minimum changes on delivery time of existing passengers.

V. Other Taxi-Sharing Models

Here, we introduce two existing taxi-sharing models. Although they are not designed for electric vehicles, we simply extend their scheduling algorithms by checking taxis' batteries before each dispatch. If the remaining battery of a taxi is not sufficient to serve the next request, it will be directed to its nearest charging station. The details of these algorithms can be found in [18]. We will compare their performance with TASeT in Section V.

A. NTT Model

We use traditional taxi-sharing scheduling strategy, i.e., without considering transfer at an intermediate location, as the baseline for this study. As stated in Section I, we refer to it as the NTT model. The NTT strategy usually focuses on finding passengers with similar start and end locations and then serves them as a group. Most existing solutions assume all passengers

TABLE II
COMPARISON OF MIP AND GREEDY HEURISTIC ON THE NUMBER OF SERVED REQUESTS AND CPU TIME

NoT	NoR	NoTrans	NoSRM	NoSRH	MIP Running Time	Heuristic Running Time
3	4	2	4	4	1”48	0”101
3	5	2	5	5	3”34	0”139
3	6	2	6	6	22”20	0”126
3	7	2	7	6	1’21”81	0”122
3	8	2	8	8	45’15”94	0”141
4	5	2	5	5	8”50	0”118
4	6	2	6	6	52”22	0”125
4	7	2	7	7	11’19”59	0”132
4	8	2	8	8	37’03”99	0”141
4	9	2	9	9	105’02”76	0”152

Note: NoT = Number of Taxis, NoR = Number of Requests, NoTrans = Number of Transfer Locations, NoSRM = Number of Served Requests with MIP, NoSRH = Number of Served Requests by Heuristic.

must be severed, which is corresponding to find a spanning tree in graph G. In our TASeT model, however, the system operates in a best effort manner.

B. Spontaneous Transfer-Allowed Taxi-Sharing Model

In [18], we introduced the idea of transfer to share private vehicles. We use this model to evaluate the effect of not having dedicated location for transfer-allowed taxi sharing. We proposed a heuristic algorithm that focused on improving passenger’s carpool experience. We modify this solution to handle eTaxi sharing and refer to it as *spontaneous transfer-allowed taxi-sharing* (STAT) strategy. Similar to TASeT, STAT allows two taxis working cooperatively to server one request. However, given that taxi routes are based on different passenger’s requests that are generally random in both time and space, the transfer locations are decided in a spontaneous manner. The intersections between two existing taxi routes are considered to be potential transfer locations. In STAT, we assume that taxis can pick up passengers at any location if time constraints permit; however, roadside parking is not allowed so that the taxi will not wait for a transfer passenger (which complies with traffic regulations in downtown area). In other words, at a transfer location, transfer passenger needs to arrive before the next relaying taxi.

VI. PERFORMANCE EVALUATION

A. Evaluation of Mathematical Formulation

For computational analysis, we implemented MIP and greedy heuristic algorithms in CPLEX and JAVA, respectively. We used the real-world request and network described in Section VII as the test case. The results are tabulated in Table II.

As we expected, the running time for MIP exponentially increases as the size of node increases. In our experiment, the MIP needs almost 2 h to compute the result for a 24-node scenario (with four nodes as taxi start location, 9×2 as request start/end, and two as transfer location). It also shows that the heuristic solution is effective to produce optimal solutions with the small test set.

TABLE III
DEFAULT VALUES OF EXPERIMENT PARAMETERS

Road network	11×11 grid
Length of a link	[500, 1500] m
Avg. travel speed on each link	[15, 45] km/h
Avg. distance between two charging station	4 km
Charging time to 80% of battery	30 minutes
Travel distance with 80% of battery	80 km
Number of eTaxis	100
eTaxi capacity	3 seats
Number of passenger requests per hour	500 requests
Passenger trip length	[5, 15] km
Maximum tolerable delay	20 minutes
Total simulation time for a given setting	10 hours

B. Evaluation on Different Taxi-Sharing Models

Here, we evaluate the performance of the proposed solutions for TASeT. For simplicity, we ignore the queuing at charging stations in this simulation and assume that charging stations are always available for eTaxis. Moreover, all taxis are electric and are fully controlled by the dispatch center. At any time during the simulation, an eTaxi is in one of the three states: a) serve passenger; b) charge battery; and c) wait for dispatch. After serving a request, an eTaxi will wait for new dispatch from the control center instead of cruising on the street looking for new passengers. These assumptions allow us to focus on the battery usage for serving passengers and related charging issues. Table III shows the default values of the parameters in our simulation.

To capture the performance, we use the following two metrics to evaluate the different service models.

Definition 1: Service Rate (SR) is defined as the average number of passengers that is served by an eTaxi over 1 h.

Definition 2: Benefit Ratio (BR) is defined as the ratio of the sum of saved travel distance by taxi sharing to the sum of travel distance when passengers are served individually. The saved travel distance, i.e., benefit, is the difference of travel distance between without and with taxi sharing to serve the same group of passengers.

Fig. 3(a) shows the performance of different strategies in a single dispatch with the default settings. It shows that allowing transfer can increase served requests by 25%–35% compared with nontransferable case. In TASeT, about half of the transfer plans contain charging plans for the relaying taxis. This is because, in the current model, we assume that charging only happens when a taxi is waiting for a transfer passenger. Fig. 3(a) also shows that carpooling in general can significantly increase the number of served requests (when compared with the case that each taxi only serves one request at a time), which is in line with previous studies on taxi sharing [2], [3].

Since taxi sharing can improve taxis’ availability, we are interested in how many taxis it would take to serve all the requests. Fig. 3(b) shows the number of taxis needed to serve 99% of the total passengers in one dispatch. Given that passengers may appear at random time and locations, additional taxis are needed to serve passengers within their maximum waiting time. Our result shows that, by carpooling, we could reduce 19% to 32% of the number of taxis to serve all requests in time, and we can achieve an additional 10%–19% reduction by allowing transfer. The difference between STAT and TASeT is

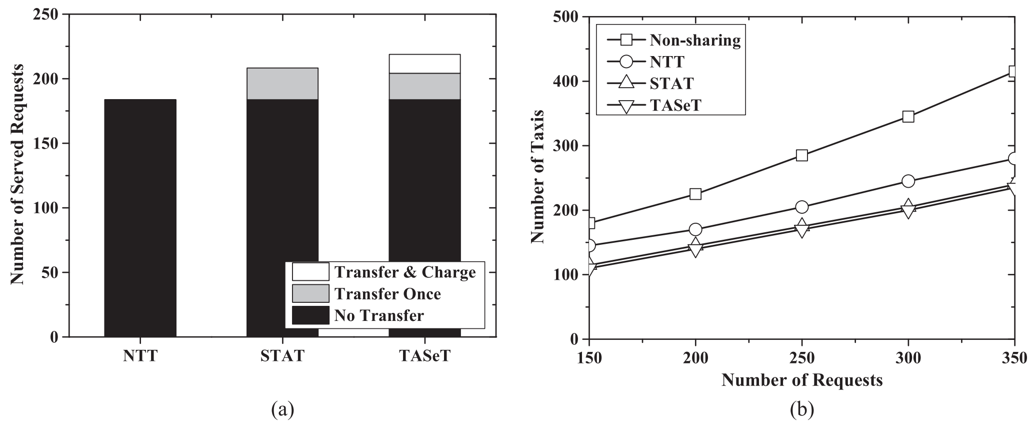


Fig. 3. Testing result. (a) Decomposition of served requests. (b) Comparison on the number of taxis required to serve all requests.

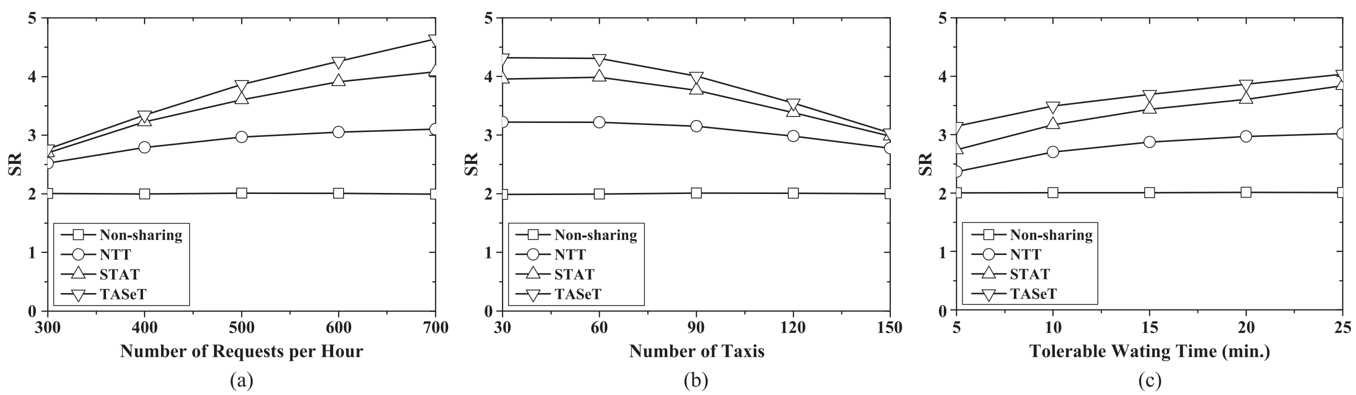


Fig. 4. (a) Performance with different numbers of requests. (b) Performance with different numbers of taxis. (c) Performance with different maximum tolerable waiting times.

not significant because of the oversupply of eTaxis. Although we choose a special setting as the example, it shows the potential for TASeT to reduce the number of eTaxis. By using less eTaxi to serve the same number of requests, this implies that TASeT is capable of improving traffic condition.

In the rest of simulations, we look at the performance during a period of time. Fig. 4(a) shows the performance under different passenger demands. Compared with nontransfer cases, the SR increases more quickly as the demand increases, which indicates that TASeT has the potential to alleviate the undersupply of taxis during rush hours. (For NTT cases, the SR remains the same as the demand increases.)

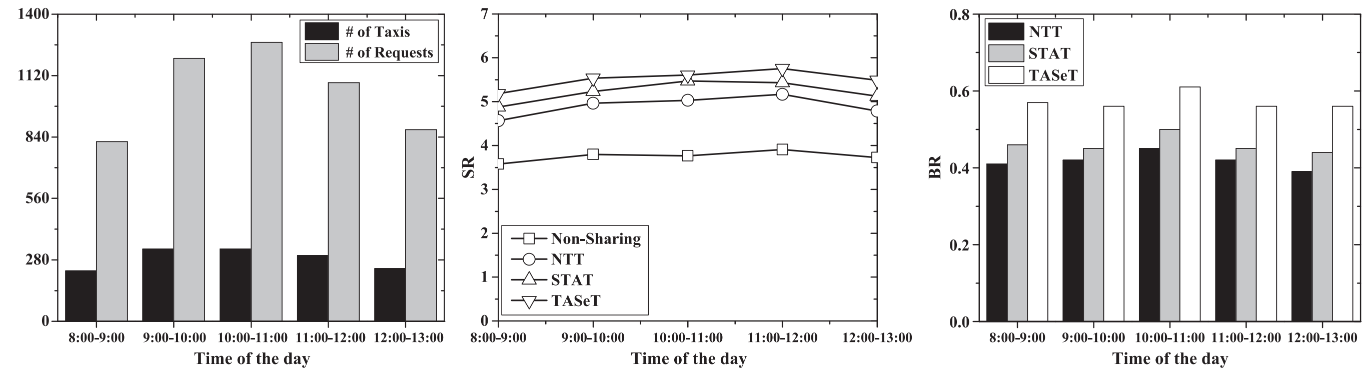
In Fig. 4(b), we evaluate the system performance with different taxi supplies. In general, TASeT outperforms others, which indicates allowing transfer and utilizing battery charging time are efficient approaches. Compared with STAT, the improvement of SR comes from two facts: 1) In TASeT, eTaxis spend less time dedicated to charging batteries. Part of their batteries is charged while waiting for taxi-sharing passengers, which saves their trips to the charging stations. Picking up passengers at a charging station also means less vacant time for eTaxis. 2) Charging stations that scattered in the city provide convenience locations for transfer, which leads to fewer detours for passengers to connect to another eTaxi. We also observed an improvement of 9%–34% when comparing TASeT with NTT in terms of SR, which indicates that, in both taxi undersupply and oversupply cases, allowing one transfer would greatly improve the carpooling opportunity.

In Fig. 4(c), we evaluate sharing-induced delay for passengers, i.e., the tolerable waiting time determines the maximum delay for passengers (compared with their travel time without taxi sharing). Not surprisingly, as passengers can tolerate a longer delay, the SR is improved. Our results show that, even with a little carpooling-induced delay, i.e., 5 min, allowing transfer can still significantly increase the performance in terms of SR.

VII. Case Study in Shanghai

Here, we evaluate the performance of different taxi-sharing models by using real-world taxi traces collected from the city of Shanghai. The taxi data set consists of second-by-second GPS trajectories and taxi meter records (which include load status, travel distance, waiting time etc.). We focus on an area of 8 km $\times$ 8 km, and the data set contains traces of about 1000 taxis on a typical Tuesday.

We build our test scenarios as follows. First, to simplify the road network, we plot the entire taxi trace data set and build an overlay road network that overlaps with many of those traces. This overlay road network may still be different from the real-world arterial road network because it contains many lower class roads that taxis have chosen to travel on. Second, to model



(a) (b) (c)

Fig. 5. (a) Number of requests and taxis retrieved from real trace at different times of a day. (b) Performance with real trace at different time of a day. (c) Percentage of benefit at different times of a day.

realistic passenger appearances, we utilize the load status information included in the taxi meter data. In particular, at a given time, if a taxi's status changed from vacant to loaded, it means that the taxi has just picked up a passenger on a specific road (which can be also identified through the longitude and latitude information in the GPS data). Since a passenger may already have waited for some time before getting picked up, in our simulation, the appearing time of that passenger is randomly selected within the time period $[t - \mathrm{MAXWAIT}, t]$. Similarly, we identify the destination of the current passenger when the taxi status changes from loaded to vacant. Third, to model charging stations, the location of gas stations in this area are mapped to the road network. Due to the large number of gas stations in this area, 20 out of the total 39 stations are selected. Finally, the number of taxis is decided based on the real-world data set. This number is updated on an hourly basis, in an effort to mimic the changes of taxis that operate in the real-world region. Taxis' initial locations are retrieved from their GPS trajectories. In addition to the modifications just noted, other parameter settings are the same as the settings presented in Section VI.

Fig. 5(a) shows the total number of passenger appearances we retrieved from the taxi traces at different times during 8 A.M.–1 P.M. along with the number of taxis operating during that hour. Fig. 5(b) compares the performance of each model at different times of the day. In general, it confirmed our previous observation that TASeT yields better performance and allowing transfer outperforms NTT in terms of higher SR (with an average improvement of 22%). The SR is generally higher than our simulation in Section V; this might be mainly because the size of the region in the case study is smaller. (We discard the trips that were going out of the region, which also leads to a shorter average trip length.)

The results in Fig. 5(c) show that transfer-allowed taxi sharing can significantly increase the BR, which reflects the advantage of TASeT in energy reduction (by reducing taxis' travel distance). In our experiment, average BRs for NTT and TASeT strategies are 0.42 and 0.57, respectively. In other words, allowing transfer can further improve the percentage of taxi-sharing distance by about 37% compared with those in traditional strategies. When looking at Fig. 5(b) and (c) together, in general, the changes in terms of BR during different times of the day are similar to those of SR. It confirms our observations that TASeT is an efficient approach under different passenger-demand/eTaxi-supply scenarios.

VIII. Conclusion

In this paper, we have introduced a new taxi-sharing paradigm called TASeT, which aims to increase the eTaxi carpooling performance by trying to fully utilize a taxi's available space. In TASeT, battery charging stations are utilized to support taxi sharing: Passengers are allowed to transfer from one eTaxi to another, and eTaxis can charge their batteries while waiting for passengers. In addition to formulating the problem with MIP and proving it to be NP-hard, we described an efficient greedy heuristic and performed large-scale evaluation. Additionally, we presented a case study in the city of Shanghai by using real taxi GPS trajectories. From this work, our major findings are as follows: 1) TASeT can significantly improve the number of passengers served by eTaxis (e.g., by 118% when compared with noncarpooling and by 35% when compared with NTT); 2) if TASeT is adopted, the number of taxis can be reduced by up to 41% while still providing service to the same number of passengers in a timely fashion; and 3) both passengers and taxi drivers can benefit from eTaxi carpooling, and allowing transfer can significantly improve the sustainability of taxi services. By allowing transfer, the percentage of shared distance can be increased by 37% when compared with NTT.

References

[1] "Take charge: A roadmap to electric New York City taxis," NYC Taxi Limousine Comm., New York, NY, USA, 2015.

[2] M. Shuo, Z. Yu, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE 29th ICDE*, 2013, pp. 410–421.

[3] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He, "coRide: Carpool service with a win–win fare model for large-scale taxicab networks," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, 2013, pp. 1–14.

[4] Recargo, Inc. PlugShare, Venice, CA, USA. [Online]. Available: http://www.plugshare.com/

[5] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.

[6] B. Zhang, K. Xing, X. Cheng, L. Huang, and R. Bie, "Traffic clustering and online traffic prediction in vehicle networks: A social influence perspective," in *Proc. IEEE INFOCOM*, 2012, pp. 495–503.

[7] Y. Ge, C. Liu, H. Xiong, and J. Chen, "A taxi business intelligence system," in *Proc. 17th ACM SIGKDD*, 2011, pp. 735–738.

[8] Y. Hou *et al.*, "Towards efficient vacant taxis Cruising Guidance," in *Proc. IEEE GLOBECOM*, 2013, pp. 54–59.

[9] D. Zhang and T. He, "pCruise: Reducing cruising miles for taxicab networks," in *Proc. IEEE 33rd RTSS*, 2012, pp. 85–94.

[10] A. Y. S. Lam, Y.-W. Leung, and X. Chu, "Electric vehicle charging station placement," in *Proc. IEEE Int. Conf. SmartGridComm*, 2013, pp. 510–515.

[11] C. Jin, J. Tang, and P. Ghosh, "Optimizing electric vehicle charging with energy storage in the electricity market," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 311–320, Mar. 2013.

[12] J.-L. Lu, M.-Y. Yeh, Y.-C. Hsu, S.-N. Yang, C. H. Gan, and M.-S. Chen, "Operating electric taxi fleets: A new dispatching strategy with charging plans," in *Proc. IEEE IEVC*, 2012, pp. 1–8.

[13] H.-J. Kim, G.-L. Park, and J. Lee, "Incorporative relocation team planning and staff member allocation in electric vehicle sharing systems," *Int. J. Multimedia Ubiquitous Eng.*, vol. 9, no. 3, pp. 249–256, 2014.

[14] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transp. Res. B, Methodological*, vol. 57, pp. 28–48, Nov. 2013.

[15] J. Jung, R. Jayakrishnan, J. Y. Park, and G. Goyang-si, "Design and modeling of real-time shared-taxi dispatch algorithms," in *Proc. Transp. Res. Board 92nd Annu. Meet.*, 2013, pp. 1–20.

[16] S.-Y. Chiou and Y.-C. Chen, "A mobile, dynamic, and privacy-preserving matching system for car and taxi pools," *Math. Problems Eng.*, vol. 2014, 2014, Art. no. 579031.

[17] C. Tian, Y. Huang, Z. Liu, F. Bastani, and R. Jin, "Noah: A dynamic ridesharing system," in *Proc. Int. Conf. Manage. Data*, 2013, pp. 985–988.

[18] Y. Hou, X. Li, and C. Qiao, "TicTac: From transfer-incapable carpooling to transfer-allowed carpooling," in *Proc. IEEE GLOBECOM*, 2012, pp. 268–273.

[19] B. Coltin and M. Veloso, "Ridesharing with passenger transfers," in *Proc. IROS*, 2014, pp. 1–6.

[20] W. Herbawi and M. Weber, "Modeling the multihop ridematching problem with time windows and solving it using genetic algorithms," in *Proc. IEEE 24th ICTAI*, Athens, Greece, 2012, pp. 89–96.

[21] S. Mitrovic-Minic and G. Laporte, "The pickup and delivery problem with time windows and transshipment," *Inf. Syst. Oper. Res.*, vol. 44, no. 3, pp. 217–227, 2006.

[22] B. Coltin and M. Veloso, "Scheduling for transfers in pickup and delivery problems with very large neighborhood search," in *Proc. AAAI*, 2014, pp. 1–7.

[23] B. Coltin and M. Veloso, "Online pickup and delivery planning with transfers for mobile robots," in *Proc. ICRA*, 2014, pp. 1–6.

[24] B. Coltin, "Multi-agent pickup and delivery planning with transfers," Ph.D. dissertation, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2014.

[25] R. Masson, F. Lehuede, and O. Peton, "An adaptive large neighborhood search for the pickup and delivery problem with transfers," *Transp. Sci.*, vol. 47, no. 3, pp. 344–355, Dec. 2012.

[26] R. Masson, F. Lehuede, and O. Peton, "Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers," *Oper. Res. Lett.*, vol. 41, no. 3, pp. 211–215, May 2013.

[27] J. S. Shang and C. K. Cuff, "Multicriteria pickup and delivery problem with transfer opportunity," *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 631–645, Sep. 1996.

[28] S. R. Thangiah, A. Fergany, and S. Awan, "Real-time split-delivery pickup and delivery time window problems with transfers," *Central Eur. J. Oper. Res.*, vol. 15, no. 4, pp. 329–349, Nov. 2007.

[29] P. Bouros, D. Sacharidis, T. Dalamagas, and T. Sellis, "Dynamic pickup and delivery with transfers," *Adv. Spatial Temporal Databases*, vol. 6849, pp. 112–129, 2011.

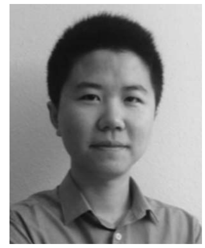
[30] I. L. Gørtz, V. Nagarajan, and R. Ravi, "Minimum makespan multi-vehicle dial-a-ride," in *Proc. ACM Trans. Algorithms-ESA*, 2009, pp. 1–29.

[31] R. Masson, F. Lehuede, and O. Peton, "The dial-a-ride problem with transfers," *Comput. Oper. Res.*, vol. 41, pp. 12–23, Jan. 2014.

[32] C. E. Cortés, M. Matamala, and C. Contardo, "The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method," *Eur. J. Oper. Res.*, vol. 200, no. 3, pp. 711–724, Feb. 2010.

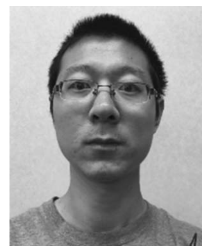
[33] A. Rais, F. Alvelos, and M. S. Carvalho, "New mixed integer-programming model for the pickup-and-delivery problem with transshipment," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 530–539, Jun. 2014.

[34] P. Toth and D. Vigo, *The Vehicle Routing Problem*. Philadelphia, PA, USA: SIAM, 2001, pp. 29–52.



Yunfei Hou (S'15) received the Ph.D. degree in computer science from the University at Buffalo, The State University of New York, Buffalo, NY, USA, in 2016.

He is currently an Assistant Professor with the School of Computer Science and Engineering, California State University, San Bernardino, CA, USA. His current research interests include applications in transportation cyber-physical systems, and data and information analysis for transportation engineering.



Weida Zhong is currently working toward the Ph.D. degree in computer science and engineering with the University at Buffalo, The State University of New York, Buffalo, NY, USA.

His research interests include vehicular ad-hoc networks and vehicular cyber-physical systems.

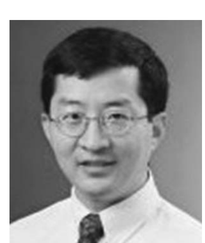
Lu Su (M'15) received the M.S. degree in statistics and the Ph.D. degree in computer science from the University of Illinois, Urbana-Champaign, IL, USA, in 2012 and 2013, respectively.

He has been with the IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is currently an Assistant Professor with the Department of Computer Science and Engineering with the University at Buffalo, The State University of New York, Buffalo, NY, USA. His research interests include cyber-physical systems, wireless and sensor networks, and mobile computing.

Kevin Hulme directs the Motion Simulation Laboratory, Center for Engineering Design and Applied Simulation, where he focuses on the custom design and development of ground vehicle simulations for applications in clinical research, education and training, and next-generation transportation studies. Recent areas of focus include standardization of simulators in teen driver safety, fidelity requirements in simulation system specification, and multi-participant civilian driving simulators.

Adel W. Sadek is currently a Professor of civil, structural, and environmental engineering with the University at Buffalo (UB), The State University of New York, Buffalo, NY, USA. He also serves as the Director of UB's Institute for Sustainable Transportation and Logistics, Director of the Transportation Informatics Tier I University Transportation Center, and Chair of UB2020's Strategic Strength in Extreme Events. Before joining UB, he was an Associate Professor with the School of Engineering, University of Vermont, Burlington, VT, USA, and held a secondary appointment in the Department of Computer Science.

Dr. Sadek received a National Science Foundation (NSF) CAREER Award and a 2011 IBM Smarter Planet Faculty Innovation Award.



Chunming Qiao (F'14) is currently the Director of the Laboratory for Advanced Network Design, Analysis, and Research (LANDER) at the University at Buffalo, The State University of New York, Buffalo, NY, USA. He has pioneered research on optical Internet and, in particular, optical burst switching.

Dr. Qiao served as the Chair of the IEEE Technical Committee on High Speed Networks (HSN) and the IEEE Subcommittee on Integrated Fiber and Wireless Technologies (FiWi), which he founded.