Transportation Research Part C 101 (2019) 233–253

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

CB-Planner: A bus line planning framework for customized bus systems

Yan Lyu[a,*], Chi-Yin Chow[b], Victor C.S. Lee[b], Joseph K.Y. Ng[a], Yanhua Li[c], Jia Zeng[d]

[a] Computer Science Department, Hong Kong Baptist University, Hong Kong
[b] Department of Computer Science, City University of Hong Kong, Hong Kong
[c] Department of Computer Science, Worcester Polytechnic Institute, MA, USA
[d] Huawei Noah's Ark Lab, Hong Kong

ARTICLE INFO

Keywords:
Customized bus system
Bus line planning
Taxi trajectories

ABSTRACT

A customized bus (CB) system is an emerging public transportation that aims to provide direct and efficient transit services for groups of commuters with similar travel demands. Existing CB systems aggregate similar travel demands and plan bus lines manually, which is inefficient and costly. In this paper, we propose a CB line planning framework called CB-Planner, which is applicable to multiple travel data sources. A mathematical programming formulation is proposed to simultaneously optimize bus stop locations, bus routes, timetables and passengers' probabilities of choosing CB. We then developed a heuristic solution framework that includes a grid-density based clustering method for discovering potential travel demands efficiently, a bus stop deployment algorithm to minimize the number of stops and walking distance, and dynamic programming based routing and timetabling algorithms for maximizing estimated profit. We conduct an experiment on a small-scale network to verify the performance gap between the optimal solution and our proposed heuristic solution. A case study is then conducted on one-month taxi trajectory data in Nanjing, China. The study demonstrates that CB lines generated by our CB-Planner can achieve higher profit compared with baseline methods, and they also provide efficient transit services with short walk distances and small departure time adjustments. The moderate increase in travel time is paid off by the significant savings in travel fare.

1. Introduction

In recent years, a new innovative public transport mode, called Customized Bus (CB) systems (Liu and Ceder, 2015), has been springing up across China. With the advantages of congestion alleviation, environmental friendliness as well as better travel experience, CB systems enjoy high popularity in more and more major cities in China since it was first launched in Qingdao in 2013 (Liu and Ceder, 2015). Nowadays, more than 30 cities in China are operating CB services (Liu and Ceder, 2015; Ministry of transport, 2014).

CB systems aim to serve groups of passengers with similar travel demands with direct and efficient transit services. This requires a CB system to be able to discover groups of similar travel demands from a set of various demands. The bus lines should have very few intermediate stops between its origin area and destination area and the timetables should be adaptive to the demands of target

* Corresponding author.
E-mail addresses: yanlyu@comp.hkbu.edu.hk (Y. Lyu), chiychow@cityu.edu.hk (C.-Y. Chow), csvlee@cityu.edu.hk (V.C.S. Lee), jng@comp.hkbu.edu.hk (J.K.Y. Ng), yli15@wpi.edu (Y. Li), zeng.jia@huawei.com (J. Zeng).

https://doi.org/10.1016/j.trc.2019.02.006
Received 28 May 2017; Received in revised form 5 February 2019; Accepted 8 February 2019
Available online 23 February 2019
0968-090X/

passengers. These features make CB systems become a more affordable and equally efficient transit choice between large communities and central business districts, compared with alternatives such as taxis and ride-sharing.

Existing CB systems manually plan CB lines by aggregating travel data collected from on-line surveys (Liu and Ceder, 2015), which is tedious, inefficient and costly. Compared to the conventional transit network design, designing bus lines for CB systems faces some new challenges: (i) Discovering the travel demand patterns of their interest efficiently. CB systems should allocate their transit resources to the massive travel demands with nearby origins and destinations and similar departure times in order to be profitable. (ii) A well designed CB line should consider the trade-off between two conflicting goals: providing efficient transit services vs. maximizing ride-sharing to earn more profit.

In this paper, we investigate how to systematically plan bus lines for CB systems and propose a new bus line planning framework **CB-Planner** that discovers groups of target passengers with similar travel demands, deploys bus stops for the discovered travel demands and plans bus lines that can maximize the estimated daily profit. It is worth to mention that CB-Planner is not limited to satisfy the actual travel demands submitted by CB passengers; it can also mine potential travel demands from other real travel data sources, such as taxi GPS trajectories and public transport transaction records which provide real travel demands citywide. It thereby has two advantages. (i) CB-Planner can be used at the initial stage of building a CB system in a new city, i.e., setting up initial CB lines for passengers to reserve based on the discovered travel patterns. (ii) By utilizing these readily available travel data sets regularly, it is possible for CB-Planner to detect any significant change in travel patterns, thus allowing any necessary adjustment to existing CB services. Any discrepancy observed after full operation of the planned CB lines can easily be fed back to the system for tuning the models for service adjustment and future planning. The main contributions of this study are as follows.

- A mathematical programming formulation is proposed to simultaneously optimize bus stop locations, bus routes, timetables and passengers' probabilities of choosing CB buses.
- We developed a heuristic solution framework that includes a grid-density based clustering method for discovering potential travel demands efficiently, a bus stop deployment algorithm to minimize number of stops and walking distance, and dynamic programming based routing and timetabling algorithms for maximizing estimated profit.
- We conducted a numerical experiment on a small-scale network to verify the optimal gap between the optimal solution and our proposed heuristic solution. CB-Planner was then evaluated in a realistic situation using one-month taxi trajectory data in Nanjing, China. The results show that our framework can generate CB lines with higher profit, compared with baseline methods, and they can provide efficient transit services in terms of short walk distances, small departure time adjustment, low travel fare and short detour time.

The rest of this paper is organized as follows. Section 2 highlights the background of CB and related work. Section 3 proposes the mathematical formulation of CB line planning. Section 4 provides the heuristic solution framework with details. Section 5 presents the numerical experiment and the case study for evaluating the performance of CB-Planner. Finally, we conclude this paper and discuss future work in Section 6.

2. Background and related work

In this section, we briefly introduce CB systems and present related work.

2.1. Background of CB systems

A CB system aims to provide demand-oriented, express, and efficient transit services. It has been vigorously promoted by governments (Ministry of transport, 2014; Chinadaily, 2014; Xinhuanet, 2013), due to the advantages of congestion alleviation, environmental friendliness as well as better user experience. Liu and Ceder (2015) first presented a detailed and comprehensive analysis on CB systems in 2015. It elaborates the background of CB systems and analyzes their operation-planning processes, including on-line demand collection, route network and timetable development. CB systems are significantly different from traditional bus transports (Liu and Ceder, 2015) in terms of operating procedures, service features and bus lines as discussed below:

Operating procedures. Existing CB systems usually have the following operating procedures: (1) Passengers submit their requests including their origins, destinations and departure times via on-line platforms such as web sites and smart phone apps. (2) Each submitted request is matched with the existing CB lines. If there exists any CB line that satisfies a request thoroughly, the system will suggest the most suitable schedule to the passenger. Then the passenger can purchase a seat on-line in advance if there are available seats. (3) The unsatisfied requests will be collected for further improving the CB system. (4) The system periodically plans new CB lines or re-plans existing CB lines based on the recent unsatisfied requests. The updated CB lines are then published for passengers to reserve.

Service features. CB services differ from traditional bus services in two folds: First, CB services aim to serve groups of similar travel demands that appear at a certain period of time everyday (or every weekday/weekend), in order to have sustainable profit. The CB lines, including stops, routes and timetables, are fixed for a period of time (e.g., a month), and they will be updated on a monthly basis to capture demand changes. Second, most of the existing CB systems require pre-pay for booking a seat, in oder to guarantee that the profitability is less likely affected by "no-show" passengers. The travel fare is usually in proportion to the travel distance.

Bus lines. A typical CB route, as illustrated in Fig. 1, has the following features (Liu and Ceder, 2015). (1) Multiple bus stops are set up in the origin area and destination area, this would help passengers access CB services within a short walk distance. (2) No

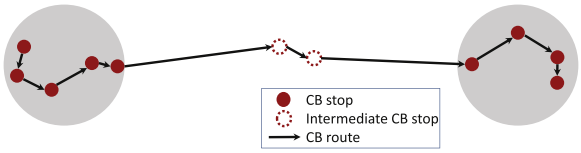


Fig. 1. Illustration of a typical customized bus route.

interchanges or transfers, this would offer passengers direct express transit services. (3) Only a few or no intermediate stops are arranged along a CB route, and thus traveling by a CB bus would be more efficient and direct than a traditional bus. Note that intermediate stops are not indispensable to a CB route, they are plotted with dash lines in Fig. 1.

2.2. Related work

2.2.1. Customized bus network design

Despite the very new concept of CB, the idea of vehicle sharing is not new. Similar public transport services, such as subscription bus (Chang and Schonfeld, 1991; Chang and Yu, 1996; Chien et al., 2001), Com-Bus (McCall, 1977) and Flexi (Bastani et al., 2011), have been proposed, and some of them have successfully operated in various major cities (Chien et al., 2001; McCall, 1977). These transit systems have the common feature that their services are demand oriented. The success of these transit systems indicates the promising future of CB systems.

Although CB systems enjoyed a great popularity in recent years, only a few studies have focused on design and optimization for CB systems. Cao and Wang (2017) investigated the passenger assignment problem for CB systems, given the fixed CB lines and travel demands. Chuanyu et al. (2017) studied CB bus dispatching problem and proposed a detouring strategy and a vehicle replacement strategy to avoid traffic congestions. Ma et al. (2017) proposed a model for stop planning and timetables for CB systems with an immune genetic algorithm. Unlike CB-Planner, neither of these studies provided a holistic solution for CB line planning.

The customized bus service design is systematically explored in Tong et al. (2017). The authors designed shuttle services that optimize stops, routes and timetables with a multi-commodity network flow-based (MCNF) model. The differences between MCNF and our CB-Planner are twofold. (1) MCNF optimizes existing CB services. It takes the actual travel requests submitted to a CB system as the input, and aims to minimize the number of unserved requests as well as the routing cost. So it is unable to set up initial CB lines at the initial stage of building a CB system in a new city when very few or no requests are submitted. In contrast, CB-Planner can leverage the readily available data sources such as taxi trips and public transit records to discover potential customers. The initial CB lines can be set up based on the discovered travel patterns. These CB lines strike the best balance between profit and attracting more passengers with high service quality. (2) MCNF utilizes the current bus stops of existing traditional bus system. However, these stops may not have good accessibility in the area where public transit systems are not well developed. CB-Planner deploys stops based on demand data, so the deployed stops could provide better accessibility for CB customers.

Another CB line planning framework is proposed in Ma et al. (2017). This study proposes an area clustering algorithm on travel demands for allocating CB resources. The CB lines are planned by pairing OD areas and selecting the OD pairs that maximize social benefits and minimize operating cost. CB-Planner differs from this study in two aspects. Firstly, CB-Planner provides more detailed solutions including deploying stops and scheduling timetables. Secondly, Ma et al. (2017) clusters origin locations and destination locations separately. It fails to capture the OD connections of each travel demand, and hence is unable to accurately discover travel patterns from the travel demand data. In contrast, CB-Planner clusters similar travel demands based on their origins, destinations and departure times. Each demand cluster will have a high chance to be served by a sequence of bus trips along a CB line. In other words, building a CB line for such a demand cluster is likely to be profitable.

2.2.2. Traditional transit network design

Transit network design (TND) (Guihaire and Hao, 2008; Ibarra-Rojas et al., 2015) determines the bus route layouts and the associated attributes such as bus frequencies, timetables and space between stops, by optimizing specific objective functions such as maximum service coverage and minimum passenger discomfort. In the following, we present a literature review on TND and highlight the difference between the existing studies and our CB-Planner.

Most of existing TND solutions optimize the route layouts based on a given infrastructure of road network. These studies either assumed that the bus stops have been well located on the road network (Nikolić and Teodorović, 2013; Cancela et al., 2015; Nayeem et al., 2014; Michaelis and Schöbel, 2009), or generated bus routes as sequences of adjacent nodes of road network, without considering stop deployment (Cipriani et al., 2012; Mauttone and Urquhart, 2009). For example, Cancela et al. (2015) searched the optimal bus routes on a given infrastructure of streets and stops. Nayeem et al. (2014) proposed genetic algorithms to plan the bus routes on an existing road network, with predefined locations of bus stops. Cipriani et al. (2012) determined the optimal transit network configuration in terms of bus routes represented by sequences of adjacent nodes and service frequencies. A few studies simultaneously optimize the locations of bus stops and the routes (Perugia et al., 2011; Szeto and Jiang, 2014). In these studies, a set of candidate stops are assumed to be known, so the stop location choice problem is easily integrated into routing decisions. For example, Perugia et al. (2011) selected the optimal stop locations during the process of route design for home-to-work transit services. Szeto and Jiang (2014) integrated stop location choice, route design and frequency setting into a bi-level programming model. *The above mentioned studies, however, fail to provide a concrete solution for stop deployment. As bus stops are the key component of*

CB systems, our CB-Planner proposes a stop deployment method that minimizes the total walk distance of all the passengers as well as the number of stops, such that passengers can access the CB stops with a short walk distance and they can have fewer intermediate stops and thereby shorter travel time.

Only a handful of research papers studied bus stop deployment for TND problems. Most of them modeled stop deployment as the stop-spacing problem, i.e., determining bus stop spacing along an existing route (Saka, 2001; Ceder et al., 2015). *Obviously, these works are not suitable for CB systems, as CB routes should have very few or no intermediate stops in order to provide direct and efficient services. In contrast, CB-Planner deploys stops at the origin and destination regions of demand clusters by balancing the stop accessibility with travel time.* A notable study (Bagloee and Ceder, 2011) provided a solution of generating candidate stop location before the process of laying down the routes and the frequency setting. *This work, however, cannot be directly applied to CB systems either. This is because CB systems need to discover potential groups of commuters with similar demands and design CB lines for them. In contrast, our CB-Planner provides an efficient grid-density based clustering algorithm to find potential demands for CB systems.*

2.2.3. Understanding transportation with travel data

Many researchers have foreseen the opportunity to obtain high-quality travel information at a low cost from various data sources, such as taxi trajectories (Zheng et al., 2010) and transit smart card transactions (Pelletier et al., 2011). Existing works have utilized taxi trajectories to discover vehicle travel patterns and improve transit services. For instance, taxi trajectories were used for planning night bus routes (Chen et al., 2014) and exploring new public transit modes (Bastani et al., 2011) by discovering hot pick-up/drop-off areas and hot traffic lines. Trajectory data flows of taxis can provide real-time traffic information and were utilized to recommend mobile users the most suitable transportation choice (Wu et al., 2012). The trajectories of electric vehicles (EV) were leveraged for deploying charging stations and charging points (Li et al., 2015). Smart card transactions record the public transit histories of individual commuters and have been utilized to discover urban commuting patterns. For instance, the origin-destination matrix was estimated from smart card data for a multi-modal public transport system (Munizaga and Palma, 2012). An efficient and effective data-mining procedure (Ma et al., 2013) was proposed to explore the travel patterns of individual commuters from transit smart card data. By integrating taxi trajectory data and public transaction records, (Liu et al., 2014) identifies and optimizes the flawed bus routes.

3. System model and problem formulation

In this section, we first present a mode-choice model for estimating the probability of a passenger choosing CB over a set of alternative transport modes, and then formulate the CB line planning problem by simultaneously optimizing bus stop locations, bus routes, timetables and the probabilities of passengers choosing CB.

3.1. Estimate probability of choosing CB

As a new transport mode, a CB system usually aims to shift commuters from existing transport modes such as taxis to itself. Hence, modeling the elasticity of travel demand is indispensable. We adopt a multinomial Logit (MNL) model (McFadden, 1973) to estimate the probability of a passenger choosing a CB bus over a set of alternative modes such as traditional bus, metro, and taxi.

Let $\mathscr{C}$ be the set of transport mode choices including CB, $\mathrm{CB} \in \mathscr{C}$. MNL model estimates the probability of a passenger n choosing CB as follows

$$p(n, \mathrm{CB}) = \frac{\exp(\mu(n, \mathrm{CB}))}{\sum_{c \in \mathscr{C}} \exp(\mu(n, c))}, \tag{1}$$

where $\mu(n, c)$ is the utility function of passenger n choosing mode c. Specifically, we incorporate the following four factors into the utility function: (i) WalkDist: walk distance; (ii) TimeAdj: departure time adjustment, i.e., time difference between the planned departure time and the actual departure time[1]; (iii) TravTime: travel time; and (iv) Fare: travel fare. The utility function of mode c is defined as

$$\mu(n, c) = \beta_{0,c} + \beta_{1,c}\ \mathrm{WalkDist} + \beta_{2,c}\ \mathrm{TimeAdj} + \beta_{3,c}\ \mathrm{TravTime} + \beta_{4,c}\ \mathrm{Fare}, \tag{2}$$

where $\beta_{1,c}$ to $\beta_{4,c}$ are coefficients of the four factors w.r.t. mode c, and $\beta_{0,c}$ is constant coefficient to capture the mean influence of variables which is not being explained by the four factors. In the practical implementation of CB-Planner, this model can be refined by considering more subjective or psychological factors, such as comfort and personal travel habit, and conducting more comprehensive surveys. CB-Planner is still applicable with a more comprehensive MNL model that considers subjective and psychological factors. However, the study on these factors is out of the scope of this paper.

It is noteworthy that the probability of choosing CB over alternative modes is determined by the service quality of CB, i.e., the four factors in the utility function. Given a certain CB bus m and a set of alternative transport modes, the probability of passenger n

[1] For a transport mode with a frequent timetable, such as metro, the time adjustment will be considered as the waiting time at a stop. For a mode with a less frequent timetable, such as CB, we assume a passenger would adjust her depart time earlier/later to catch a ride. For example, a passenger plans to depart at 8:00 am from a CB stop, but the nearest departure time of a bus is around 7:50 am. So the passenger needs to adjust her departure time 10 min earlier to catch the bus, namely, the time adjustment is 10 min.

choosing m, denoted by $p(n, m)$, can be calculated by Eq. (1) directly.

3.2. Problem formulation

Given a set of travel demands, we aim to find a set of locations to deploy bus stops, and generate a set of bus routes together with the timetable along each route. The objective is to maximize the total estimated profit of all the bus lines. Specifically, we consider the following inputs:

- a set of potential passengers $N = \{n\}$, in which, each passenger indicates a travel demand with an origin location o, a destination location d, planned departure time t_o and her potential travel experiences (i.e., walk distance, time adjustment, travel time and fare) by other transport modes;
- a set of available CB buses $M = \{m\}$.

To integrate bus stop deployment, route planning and timetable scheduling into a unified optimization framework, we adopt a space-time network $G = (V, A)$ that combines the physical transportation network with travel time information (Tong et al., 2017; Tong et al., 2015). Each vertex $(i, s) \in V$ represents both location i and time s, (here, location i is a node of the physical transportation network); each arc $(i, j, s, t) \in A$ indicates a directed path from location i departing at time s to location j at time t. Based on G, we define the decision variables to determine the bus network including locations of bus stops, bus routes and their timetables as well as passenger-to-bus assignment:

- $x^m_{i,j,s,t} \in \{0, 1\}$, where $x^m_{i,j,s,t} = 1$ if bus m travels through the arc (i, j, s, t), i.e., passes through location i at time s and travels directly to location j at time t, and $x^m_{i,j,s,t} = 0$ otherwise.
- $a^{m,n}_{i,j,s,t} \in \{0, 1\}$, where $a^{m,n}_{i,j,s,t} = 1$ if passenger n travels through the arc (i, j, s, t) by bus m and $a^{m,n}_{i,j,s,t} = 0$ otherwise.

Objective function. The objective function is to maximize the total estimated total profit of all the buses. The profit can be calculated by subtracting the operational cost from the fare revenue, i.e.,

$$\text{maximize} \sum_{m\in M}\sum_{n\in N} p(n, m)\rho \sum_{(i,j,s,t)\in A} a^{m,n}_{i,j,s,t} len(i, j) - \sum_{m\in M} \gamma \sum_{(i,j,s,t)\in A} x^m_{i,j,s,t} len(i, j) \tag{3}$$

where ρ is the ticket price per kilometer, γ is the operational cost per kilometer of a bus, $len(i, j)$ is the road network distance from i to j, and $p(n, m)$ is the probability of passenger n choosing bus m over the set of alternative transport modes.

$p(n, m)$ can be inferred from the mode choice model (Eq. (1)) and the passenger-to-bus assignment matrix $[a^{m,n}_{i,j,s,t}]$. Specifically, if passenger n is assigned to bus m, i.e., $a^{m,n}_{i,j,s,t} = 1$, the passenger's WalkDist, TimeAdj, TravTime and Fare by bus m can be calculated as follows. We introduce the following two binary variables to indicate the passenger's locations and times of getting on and getting off bus m:

- $O^{m,n}_{i,s} \in \{0, 1\}$, where $O^{m,n}_{i,s} = 1$ if passenger n gets on bus m at vertex (i, s), and $O^{m,n}_{i,s} = 0$ otherwise;
- $D^{m,n}_{i,s} \in \{0, 1\}$, where $D^{m,n}_{i,s} = 1$ if passenger n gets off bus m at vertex (i, s), and $D^{m,n}_{i,s} = 0$ otherwise.

The average walk distance of passenger n for taking bus m is

$$\text{WalkDist} = \frac{1}{2}\left(\sum_{(i,s)\in V} O^{m,n}_{i,s} len(o, i) + \sum_{(i,s)\in V} D^{m,n}_{i,s} len(i, d)\right), \tag{4}$$

where o and d are the passenger's origin and destination locations, respectively. The passenger's time adjustment for bus m is

$$\text{TimeAdj} = \left|\sum_{(i,s)\in V} O^{m,n}_{i,s} s - (t_o + t_{walk})\right|, \tag{5}$$

where t_o is the planned departure time at origin location o, and t_{walk} is the walking time. The passenger's actual travel time on bus m is the summation of travel time along each arc:

$$\text{TravTime} = \sum_{(i,j,s,t)\in A} a^{m,n}_{i,j,s,t}(t - s). \tag{6}$$

And the passenger's actual travel fare on bus m is

$$\text{Fare} = \rho \sum_{(i,j,s,t)\in A} a^{m,n}_{i,j,s,t} len(i, j). \tag{7}$$

Based on these four factor values by bus m, $p(n, m)$ can be calculated by the mode choice model (Eqs. (1) and (2)) and the factor values of the passenger's alternative transport modes.

Flow balance constraints. To make sure that each bus m can find a feasible route in G, $x^m_{i,j,s,t}$ should satisfy the flow balance constraint:

$$\sum_{(j,t)\in V} x^m_{i,j,s,t} - \sum_{(j,t)\in V} x^m_{j,i,t,s} = \begin{cases} 1, & i = b_0,\ s = T_0 \\ -1, & i = b_0,\ s = T_{max} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where b_0 is a depot of bus m, T_0 and T_{max} are the earliest departure time and the latest arrival time at depot b_0, respectively. There could be a waiting arc if the actual departure time t_m at the depot is later than T_0, i.e., $x_{b_0,b_0,T_0,t_m} = 1$. Similarly, there is also a waiting arc if the actual arrival time is earlier than T_{max}. In addition, bus m, if used, can only visit a vertex at most once, i.e.,

$$\sum_{(j,t)\in V} x^m_{i,j,s,t} \leqslant 1, \qquad \forall\ (i, s) \in V. \tag{9}$$

Passenger-to-bus assignment constraints. Passengers can only be assigned to valid buses and each passenger can take at most one bus, i.e.,

$$a^{m,n}_{i,j,s,t} \leqslant x^m_{i,j,s,t},\ \forall\ (i, j, s, t) \in A, \tag{10}$$

$$\sum_{m\in M} a^{m,n}_{i,j,s,t} \leqslant 1,\ \forall\ (i, j, s, t) \in A. \tag{11}$$

A passenger's trip on bus should also satisfy the flow balance constraint, i.e.,

$$\sum_{(j,t)\in V} a^{m,n}_{i,j,s,t} - \sum_{(j,t)\in V} a^{m,n}_{j,i,t,s} = O^{m,n}_{i,s} - D^{m,n}_{i,s}, \quad \forall\ (i, s) \in V. \tag{12}$$

where $O^{m,n}_{i,s}$ and $D^{m,n}_{i,s}$ should satisfy that passenger n, if chooses bus m, can get on and get off bus at only one vertex separately, namely,

$$\sum_{(i,s)\in V} O^{m,n}_{i,s} \leqslant 1, \tag{13}$$

$$\sum_{(i,s)\in V} D^{m,n}_{i,s} \leqslant 1, \tag{14}$$

$$\sum_{(i,s)\in V} O^{m,n}_{i,s} = \sum_{(i,s)\in V} D^{m,n}_{i,s}. \tag{15}$$

Capacity constraints. The total number of served passengers cannot exceed the capacity of bus m, i.e.,

$$\sum_{n\in N} a^{m,n}_{i,j,s,t} \leqslant \phi x^m_{i,j,s,t}, \quad \forall\ (i, j, s, t) \in A, \tag{16}$$

where ϕ is capacity of bus m.

In summary, we aim to maximize the estimated total profit (Eq. (3)) under the constraints (8)–(16). Compared with existing transit network design models, our proposed formulation has the following innovative features. First, we use a multinomial Logit model to estimate the probability of a passenger choosing CB over a set of alternative transport mode choices. This allows CB-Planner to consider potential demands from other transport modes such as taxis and traditional buses. Second, we integrate the Logit model into the optimization formulation to capture the interaction between passengers' transport mode choices and the design of bus lines. Namely, the design of CB lines affect passengers' choices, which in turn affect the estimated profit. This formulation thereby achieves a trade-off between two conflicting goals: providing efficient transit services vs. maximizing ride-sharing to earn more profit. Third, this formulation jointly models the high-level bus stop deployment, routing and timetabling problems and the low-level passenger-to-bus assignment problem.

4. Solution algorithms

The above proposed mixed-integer non-linear problem formulation is NP-hard and cannot be solved in polynomial time. Moreover, for the real travel data sources, such as taxi trajectories and public transport transaction records, the computational load could be huge because these real data sets usually contains billions of travel records citywide. Therefore, we develop a heuristic sequential solution to achieve the objective phase by phase. Fig. 2 depicts the CB-Planner framework with three phases:

Phase 1 (Travel demand clustering): In this phase, we first discover travel patterns by clustering travel demands with nearby origin and destination locations and similar departure times. The discovered demand clusters are the potential customers for CB systems.

Phase 2 (CB stop deployment): In each demand cluster, CB stops are deployed at the origin area and destination area with a heuristic stop deployment algorithm (CBDeploying), by minimizing number of stops and the total walk distance. This actually maximizes the probability of passengers taking CB buses in terms of walk distance and travel time (the fewer stops, the shorter the travel time).

Phase 3 (CB line planning): For each cluster with deployed CB stops, a CB line that achieves the maximum estimated profit is generated using a routing algorithm (CBRouting) and a timetabling algorithm (CBTimetabling). Finally, the CB lines that overlap in route segments and working hours are merged with a merging scheme (CBMerging), to further maximize the total estimated profit of all CB lines.

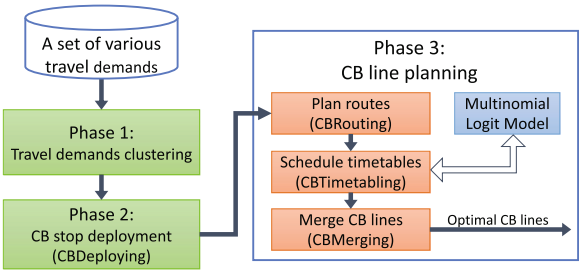


Fig. 2. Framework of CB-Planner.

4.1. Travel demands clustering

CB buses aim to serve groups of passengers who have similar travel demands. Such travel patterns can be discovered by clustering travel demands with nearby origin and destination locations and similar departure times. And the travel demands within a cluster indicate a group of potential customers of the CB system.

We present a travel demand with a five-dimensional vector, ($stLat$, $stLng$, $edLat$, $edLng$, $stTime$), which denotes the origin latitude, origin longitude, destination latitude, destination longitude, and departure timestamp, respectively. In order to cluster demands with the five dimensions, we employ a grid-density based clustering method. The computation complexity of grid-density based clustering methods has been proved to be $O(n)$ (Aggarwal and Reddy, 2013), where n is the number of grids in the data space. Since the number of grids is significantly smaller than the number of data points in most applications, grid-density based clustering methods are usually more efficient than other clustering approaches (Aggarwal and Reddy, 2013). They are also flexible in dealing with multidimensional data sets (Agrawal et al., 1998), and there is no need to measure the distance between data objects. A travel demand has spatial features (e.g., $stLat$, $stLng$), and the temporal feature (i.e., $stTime$), it is difficult to measure the distance between the demands spatially and temporally. Therefore, the grid-density based clustering methods are suitable for clustering travel demands. We implement the grid-density based clustering method with the following steps:

Space partitioning. We focus on the five-dimensional space **stLat** × **stLng** × **edLat** × **edLng** × **stTime**, where the spatial dimensions (**stLat**, **stLng**) and (**edLat**, **edLng**) are bounded by the boundaries of the city, and the temporal dimension **stTime** is bounded by the time of a day, i.e., 0:00 am to 12:00 pm. This five-dimensional space is then partitioned into non-overlapping units. All units are obtained by partitioning every dimension into intervals of equal length, i.e., each unit is the intersection of one interval from each dimension. Specifically, let u_i denote the u_i-th interval in i-th dimension, $1 \leqslant i \leqslant 5$. A unit, denoted as U, is presented as $U = (u_1, u_2, ..., u_5)$.

Map travel demands into units. We map a travel demand ($stLat$, $stLng$, $edLat$, $edLng$, $stTime$) into a unit if each attribute value of this demand falls into the interval of the corresponding dimension of the unit. After mapping all the demands into units, the *density* of a unit can be calculated as the average daily number of travel demands. We call a unit *dense* if the *density* of this unit is larger than a *density threshold*, which is an input parameter.

Merge all dense units into clusters. A demand cluster can be discovered among a set of dense units that are adjacent with each other. Two units, denoted by $U_1 = (u_1^1, u_2^1, ..., u_5^1)$ and $U_2 = (u_1^2, u_2^2, ..., u_5^2)$, are *adjacent* if there exists k-th dimension, $1 \leqslant k \leqslant 5$, such that 1) $|u_k^1 - u_k^2| = 1$, and 2) $u_i^1 = u_i^2, \ i \neq k, 1 \leqslant i \leqslant 5$.

To find sets of dense units that are adjacent with each other, we first form a graph, in which each node represents a dense unit, and there is an edge between two nodes if the two units are *adjacent*. Then sets of adjacent dense units can be extracted as the connected components from the graph by a depth-first or breadth-first traversal. Travel demands in each set of adjacent dense units form a cluster.

4.2. CB stop deployment

In this section, we deploy CB stops for the potential passengers from each demand cluster. Although not all the potential passengers from a cluster will take CB buses, the origin and destination locations of a demand cluster can indicate the most popular locations of these passengers would like to start and end their trips. Therefore, we deploy stops to cover these origin locations and destination locations for each demand cluster.

In order to attract more passengers choosing CB services, the CB stops should have very good accessibility, i.e., the walk distance between an origin/destination location to the nearest stop should be as short as possible. This thereby requires more stops to be deployed. However, too many stops will increase the travel time, which will in turn degrade the service quality and hence reduce the probability of passengers taking CB buses. Therefore, how to balance the walk distance with the number of CB stops is the key issue in CB stop deployment.

To address this issue, we first employ two parameters, service coverage radius of a stop, denoted as *CovRad*, and service coverage percentage, denoted as *CovPct*. The stop deployment problem is formulated as follows: We aim to find the minimum number of stops and their optimal locations, such that these stops can cover at least *CovPct* percentage of passengers within the radius of *CovRad*, and the total walk distance of all the passengers to these stops is minimized.

Specifically, given a set of origin/destination locations, denoted as $\{pt_i\}$, we aim to deploy a set of CB stops, denoted as $B = \{b_j\}$, so as to the number of stops, $|B|$, is minimized, under the constraints of:

- *Coverage constraint*: At least *CovPct* percentage of $\{pt_i\}$ whose distances to their nearest stops in B cannot exceed *CovRad*.
- *Distance constraint*: The total walk distance of $\{pt_i\}$ to their nearest stops in B is minimized.

This problem is a typical facility location problem, which is NP-hard and cannot be solved optimally in polynomial time (Farahani and Hekmatfar, 2009). Many heuristic methods have been proposed based on the clustering models, such as k-medoids (Jain and Vazirani, 2001; Park and Jun, 2009). The classical k-medoids is used to assign data points into different groups by minimizing the distance between data points (i.e., locations) labeled to be in a group and a point designated as the center of that group (i.e., bus stops). However, the number of groups k needs to be determined in advance, and the initial locations of the k group centers are picked randomly, which greatly affects the performance of this method. To address these two issues, namely NP-hardness and indeterminate value of k, we propose a heuristic method, called CBDeploying, that adapts the k-medoids by incrementally deploying k CB stops from $k = 1$ until the k stops satisfy the coverage and distance constraints. The initial locations of the k stops are determined by the optimal locations of the $k - 1$ stops from the previous step.

For $k = 1$, the optimal location of this stop is the location that achieves the minimum total walk distance to all other locations in $\{pt_i\}$. Then we calculate the coverage percentage of this stop. As long as the coverage percentage is less than *CovPct*, more stops are needed in order to meet the coverage constraint.

When deploying k stops for $k > 1$, the locations of the k stops should satisfy the *distance constraint*, i.e., the total distance of all locations in $\{pt_i\}$ to their nearest stops is minimized. To initialize the locations of the k stops, we first utilize the optimal locations of the $k - 1$ stops from the previous step as the $k - 1$ initial locations. For the k-th initial location, we first find the set of locations that are not covered by the previous $k - 1$ stops, i.e., the distances from these locations to their nearest stops are larger than *CovRad*. The centroid of these locations is set as the k-th initial location. The rationale behind this initialization process is that the $k - 1$ stops deployed at the previous step are already well distributed and the primary purpose of the additional stop is to increase the coverage percentage (Likas et al., 2003). To obtain the optimal locations of the k stops, we repeatedly shift the k stops to the centroid of groups of locations that are assigned to the stop, until the total distance does not decrease.

As depicted in Algorithm 1, CBDeploying first deploys one CB stop at the location which achieves the minimum total walk distance to all locations in $\{pt_i\}$, (Lines 5–6), and then checks whether the stop satisfies the coverage constraint (Line 7). If the coverage constraint cannot get satisfied, the algorithm incrementally deploys one more stop at the uncovered location which achieves the minimum total walk distance to all the uncovered locations (i.e., *unCovPts*). Then it adjusts locations of the k deployed stops using k-medoids to minimize the total walk distance from all the locations to their nearest stops (Lines 8–13). The algorithm terminates until the k stops satisfy the coverage and distance constraints. Note that our algorithm calls the k-medoids function $k - 1$ times for adjusting the locations of k stops iteratively. The complexity of each iteration in k-medoids (Lines 17–19) is $O(nk)$ (Park and Jun, 2009), where n is the number of origin/destination locations in a demand cluster.

Algorithm 1. CB stop deployment (CBDeploying)

Input: (1) A set of origin/destination locations of a cluster $\{pt_i\}$, (2) service coverage radius of CB stops *CovRad*, (3) coverage percentage of CB stops *CovPct*, (4) road network.

Output: A set of CB stops $B = \{b_j\}$.

1: Calculate the pairwise road network distance of $\{pt_i\}$
2: Set $B = \emptyset$, and the number of stops $k = 1$
3: Find the location pt^* that has the minimum total distance to other locations
4: Set the first CB stop as $b_1 = pt^*$, $B \leftarrow b_1$
5: Let *pct* be the percentage of locations that their walk distances to their nearest stops in B are no larger than *CovRad*
6: Let *unCovPts* be the set of locations that their walk distances to their nearest stops in B are larger than *CovRad*
7: **while** $pct < CovPct$ **do**
8: Increase k by 1
9: Find the location pt^* that has the minimum total walk distance to locations in *unCovPts*
10: Set $b_k = pt^*$, $B \leftarrow b_k$
11: Adjust locations of k stops B with K-MEDOIDS (B, $\{pt_i\}$)
12: Renew *unCovPts* and *pct*
13: **endwhile**
14: Get the set of CB stops $B = \{b_j\}$, $1 \leqslant j \leqslant k$
15: **function** K-MEDOIDS (B, $\{pt_i\}$)
16: **repeat**
17: Assign each location to its nearest stop in B
18: For each stop b_j, $b_j \in B$, $1 \leqslant j \leqslant k$, reset its location at the location that has the minimum total distance to the other locations that assigned to b_j
19: Calculate the sum of distances from all the locations to their nearest stops
20: **until** The sum of distances does not decrease
21: **return** B
22: **end function**

4.3. CB line planning

In CB systems, a CB line will be set for a certain travel pattern (i.e., a demand cluster or several demand clusters that overlap in route segments and departure times). So the CB line planning problem can be simplified by planning a CB line for each cluster first and then maximizing the total profit by merging the lines that overlap in route segments and working hours. In this section, we first propose a routing algorithm to plan a CB route for a demand cluster and a timetabling algorithm to schedule the timetable for a CB route. Then a CB merging scheme is proposed to merge CB lines to further maximize the total estimated profit of all the CB lines.

4.3.1. Routing

To reduce the operational cost per bus trip, a CB route should be able to satisfy a travel demand cluster with the shortest travel distance. As we have deployed multiple CB stops in the origin region and destination region of the demand cluster separately, the CB routing problem can be formulated as a traveling salesman problem (TSP) (Malandraki and Dial, 1996; Mingozzi et al., 1997), i.e., find the shortest route that visits each stop in origin region and destination region. The key issue is that, unlike the classical TSP, the pick-up stops should be visited before their corresponding drop-off stops along the route. To address this issue, we develop a dynamic programming algorithm, called CBRouting, that generates the shortest route to transit passengers from their pick-up stops to drop-off stops for a demand cluster.

Let B be the set of deployed CB stops from a cluster. We employ a dummy stop b_0, that has the following properties: 1) no passengers come from or to this stop, and 2) the distance between b_0 and any other stops is 0. We set b_0 as the origin stop for all routes. For a subset of stops $S \subseteq \{b_0, B\}$ and $b_0,\ \ b_j \in S$, let (S, b_j) be the set of possible paths that start at b_0, visit each node in S exactly once, and finish in b_j. (S, b_j) can only be valid if it satisfies the precedent constraint, i.e.,

- Precedent constraint: if b_j is a drop-off stop, then all its corresponding pick-up stops should be in S; if b_j is a pick-up stop, then any of its drop-off stops cannot be in S.

For a valid set (S, b_j), let $R(S, b_j)$ be the shortest route, and $Len(S, b_j)$ be the length of $R(S, b_j)$. We obtain the shortest route $R(S, b_j)$ by picking the best second-to-last stop b_i, such that the route length from b_0 to b_i, (i.e., $Len(S - \{b_j\}, b_i)$), plus the length of the final route segment $len(b_i, b_j)$, is minimized:

$$Len(S, b_j) = \min_{b_i \in S: i \neq j,} Len(S - \{b_j\}, b_i) + len(b_i, b_j). \tag{17}$$

We denote b_i^* as the best second-to-last stop, then

$$R(S, b_j) = (R(S - \{b_j\}, b_i^*), b_j). \tag{18}$$

Note that $(S - \{b_j\}, b_i)$ should satisfy the precedent constraint, we prune the invalid sets for each iteration of Eqs. (17) and (18). By iteratively searching the shortest route for each subset $S \subseteq B$ and checking the precedent constraint, the CB route $R(B)$ can be generated after the set B is searched, i.e.,

$$R(B) = R(B, b_j^*),\ b_j^* = \text{argmin}_{b_j \in B} Len(B, b_j). \tag{19}$$

As depicted in Algorithm 2, our CBRouting starts searching the subset $S \subseteq B$ from the size of $|S| = 1$ to $|B|$ (Line 2). For each subset, it first checks the validity of (S, b_j), $b_j \in S$ by the precedent constraint (Line 6). If (S, b_j) satisfies the precedent constraint, the shortest route $R(S, b_j)$ and its length $Len(S, b_j)$ are obtained by Eqs. (17) and (18) (Lines 7–9). The algorithm terminates until B is searched and the CB route $R(B)$ can be obtained by Eq. (19) (Line 13). Since our algorithm searches every subset of B, there are at most $2^n \cdot n$ possible states (S, b_j), and each one takes linear time to get the shortest route $R(S, b_j)$, $n = |B|$. So the total running time is $O(n^2 2^n)$. Note that in reality, the number of stops deployed for a trajectory cluster is usually not very large. (As shown in Fig. 8, the case study show that the number of deployed CB stops for each cluster is smaller than 6, i.e., $n < 6$.) Therefore, the CB route can be obtained in a relatively short time.

Algorithm 2. CB routing (CBRouting)

Input: (1) A set of CB stops of a cluster $B = \{b_i\}$, $(1 \leqslant i \leqslant |B|)$, (2) $len(b_i, b_j)$, $(b_i, b_j \in B)$.

Output: Optimal CB route $R(B)$.

```
1: Set the dummy stop b_0 and let Len({b_0}, b_0) = 0
2: for s = 1 to |B| do
3:     for all subsets S of size s,  S ⊆ B do
4:         S = {S, b_0}, Len(S, b_0) = ∞
5:         for b_j ∈ S do
6:             Check the validity of (S, b_j) by precedent constraint
7:             if (S, b_j) is valid then
8:                 Obtain the shortest route length Len(S, b_j) and the route R(S, b_j) by Eqs. (17) and (18)
9:             end if
10:        end for
11:    end for
12: end for
13: R(B) = R(B, b_j^*),  b_j^* = argmin_{b_j ∈ B} Len(B, b_j)
```

4.3.2. Timetabling

For each demand cluster and its CB route, we schedule an optimal timetable $\mathbf{t}^*$, i.e., the sequence of start time of each trip along the route in a day, with a dynamic programming algorithm, called CBTimetabling, in order to maximize the total estimated profit of all the trips.

Let t_f and t_l be the departure time of the first and last travel demand in this cluster in a day respectively. Then we only need to search the timetable within the time interval of $[t_f, t_l]$.

Suppose from time t_f to t, $t_f < t \leqslant t_l$, we have set a certain number of trips between t_f to t, and t is the start time of the last trip. Let $\mathbf{t}(t)$ be the optimal timetable whose last trip starts at t, and $totalP(t)$ be the maximum total profit of the trips with $\mathbf{t}(t)$. We denote $profit(t)$ as the profit of the last trip, and $preP(t)$ be the profit earned before the last trip, i.e., $totalP(t) = preP(t) + profit(t)$. $preP(t)$ can be obtained by picking the best start time of the second-to-last trip $t - \tau$, such that the total profit of the trips before $t - \tau$, plus the profit of the last two trips, i.e., $profit(t - \tau)$ and $profit(t)$, is maximized:

$$preP(t) = \max_{1 \leqslant \tau \leqslant t - t_f} (preP(t - \tau) + profit(t - \tau) + profit(t)) - profit(t). \tag{20}$$

where $profit(t)$ can be calculated by counting the passengers that are not served by the previous buses and computing their probability of choosing the trip that starts at t. Let m denote the bus serving the trip at time t, $N(t)$ denote the set of passengers that are not served by the previous buses before t. The profit of bus m is

$$profit(t) = \underset{n \in N(t)}{m} \text{Fare}(n) \cdot p(n, m) - \gamma \cdot Len(R), \tag{21}$$

where ActFare(n) is the actual travel fare of passenger n along the route R, and $Len(R)$ is the length of route R. $p(n, m)$ is the probability of passenger n choosing m, and it can be calculated using Eqs. (1) and (2). We assume that passenger n chooses the closest stops to get on and get off the bus. Note that the number of passengers on board could exceed the bus capacity, not all the passengers in $N(t)$ will be considered into $profit(t)$. Instead, at each origin stop, the passengers are sorted according to their time adjustment from the shortest to the longest. Those passengers who ranked behind will be excluded if the bus becomes fully occupied at some stops. In the case that $profit(t)$ could be below 0, i.e., a trip starts at t could lose money, we consider this trip as a dummy trip with profit of 0.

We denote $t - \tau^*$ as the start time of the best second-to-last trip, then $\mathbf{t}(t)$ can be obtained by iteratively recording the profit-making trips, i.e.,

$$\mathbf{t}(t) = \begin{cases} (\mathbf{t}(t - \tau^*), t) & \text{if } profit(t) > 0, \\ \mathbf{t}(t - \tau^*) & \text{otherwise.} \end{cases} \tag{22}$$

By searching t from t_f to t_l with Eqs. (20) and (22), the optimal timetable is recorded in $\mathbf{t}(t_l)$, and the maximum profit would be $preP(t_l) + profit(t_l)$.

As depicted in Algorithm 3, CBTimetabling first calculates the profit of the trip that starts at t_f, and initializes the total profit of previous trips $preP(t_f)$ as 0 (Lines 1–3). It then incrementally obtains the optimal timetable $\mathbf{t}(t)$ from $t = t_f + 1$ to t_l, by searching the best second-to-last trip such that the total profit from t_f to t is maximized (Lines 4–8). The algorithm terminates until $\mathbf{t}(t_l)$ is searched (Line 9). Since our algorithm traverses the optimal timetables from $t = t_f$ to t_l and each timetable takes linear time, the time complexity is $O(n^2)$, where n is the number of time slots from t_f to t_l.

Algorithm 3. CB timetabling (CBTimetabling)

Input: A travel demand cluster with its CB route
Output: Optimal timetable $\mathbf{t}^*$, and the maximum total estimated profit $totalP^*$.
1: Calculate $profit(t_f)$ with timetable $\mathbf{t} = t_f$ by Eq. (21)
2: If $profit(t_f) > 0$, set $\mathbf{t}(t_f) = t_f$, otherwise, set $\mathbf{t}(t_f) = \varnothing$
3: Set $preP(t_f) = 0$
4: **for** $t = t_f + 1$ to t_l **do**
5: Calculate $preP(t)$ by Eq. (20)
6: Let τ^* be the parameter that achieves the maximum in Eq. (20), and get the previous timetable $\mathbf{t}(t - \tau^*)$
7: Obtain $\mathbf{t}(t)$ by Eq. (22)
8: **end for**
9: $\mathbf{t}^* = \mathbf{t}(t_l)$, and $totalP^* = preP(t_l) + profit(t_l)$

4.3.3. CB line merging

Given the set of demand clusters, we have planned a CB line that achieves the maximum estimated profit for each cluster. Despite the fact that CB lines from different clusters differ from origin locations, destination locations and departure times, some of them may overlap with each other in terms of route segments, and working hours, (i.e., the time interval between the start times of the first and the last trips). For example, in Fig. 3, the lines 1, 2 and 3 overlap in some route segments, as well as the working hours, and the same holds for lines 3 and 4. If we merge the overlapped CB lines into one line, this new line could be more profitable than the total profit of each single line, with the fact that one bus ride along the new line can serve more passengers. However, not all overlapped CB lines can be merged into a more profitable line. This is because the merging may lead to more intermediate stops or longer detour distance for the merged line, and thus a longer travel time, resulting in a loss of passengers. In order to further maximize the total profit of CB

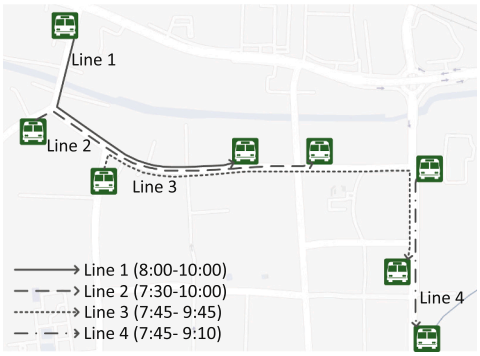


Fig. 3. Example of overlapped CB lines.

lines, we explore which CB lines could be merged together, and how to achieve the maximum total estimated profit by merging CB lines.

Algorithm 4. CB line merging (CBMerging)

Input: A set of CB lines $\mathscr{L} = \{L_i\}$ from each cluster
Output: A set of CB lines $\mathscr{L}^* = \{L_k\}$
1: $\mathscr{L}^* \leftarrow \mathscr{L}$
2: **repeat**
3: For each pair of CB lines in $\mathscr{L}^*$, select the pairs that satisfy the merging criteria
4: For each of the selected pairs, merge the two CB lines by re-planning a route and a timetable with Algorithms 2 and 3 separately, and get the estimated profit growth of the merged CB line
5: Select the merged line, denoted as L_{ij}^*, that achieves the maximum profit growth; let (L_i^*, L_j^*) be the pair of lines that form L_{ij}^*
6: Remove (L_i^*, L_j^*) from $\mathscr{L}^*$, and $\mathscr{L}^* \leftarrow L_{ij}^*$
7: **until** The total profit stops increasing by merging

Merging criteria. Two CB lines need to be merged if they (1) share some route segments, and (2) overlap in working hours.

Merging scheme. It is challenging to find the optimal merging scheme that achieves the maximum total profit. The main reasons are: (1) The profit of a CB line is determined by how well the CB route and timetable satisfy the travel demands of passengers. Whether a new merged CB line can be more profitable is unknown before its route and timetable being planned. (2) For a set of n CB lines in which any two lines satisfy the merging criteria, the number of possible combinations for merging can be counted by Bell numbers $B(n)$ (Rota, 1964), whose upper bound is $(\frac{0.792n}{\ln(n+1)})^n$ (Berend and Tassa, 2010). Calculating the profit for all the possibilities is not efficient. To this end, we propose a greedy merging scheme, called CBMerging, by iteratively merging the pair of CB lines that achieves the maximum profit growth.

As depicted in Algorithm 4, given a set of CB lines, we first merge each pair of CB lines that satisfies the merging criteria, and calculate the profit of each merged line with Algorithms 2 and 3. Then the pair that achieves the maximum profit growth is replaced by the corresponding merged CB line. We repeat the previous steps until the total profit of all the CB lines stops increasing by merging. For a set of n CB lines in which any two lines satisfy the merging criteria, in the worst case, our merging scheme greedily checks the profit for $O(n^2)$ possible merging combinations, which is much smaller than the exhaustive search. Our case study (Section 5.2.5) also shows that our merging scheme greatly improves the total profit.

5. Experiment

This section first provides a numerical experiment on a small-scale network to study the optimality gap of the proposed heuristic solution. A case study is then conducted on a city-scale network using a real taxi trajectory dataset as the travel demand data source.

5.1. Numerical experiment on Sioux Falls network

We test the proposed model and algorithms using a simplified Sioux Falls network[2] (Tong et al., 2017; LeBlanc, 1988). As shown in Fig. 4, the network consists of 24 nodes which can be regarded as candidate stop locations and 38 bidirectional transportation links. Each link is labeled with the travel time in minutes. We randomly generate travel demands from the north part of the network to the south. Each demand is associated a randomly generated departure time within a 20-min time window. We also assume that each passenger has another mode choice, taxi, which provides the shortest travel time in the network. The taxi fare is in proportion to the travel time. Table 1 lists a group of 20 travel demands, together with their travel time and fare if they travel by taxi. These passengers' origin nodes and destination nodes are marked in blue and yellow in Fig. 4 separately. Given choices of a certain CB bus

[2] https://github.com/bstabler/TransportationNetworks/tree/master/SiouxFalls.

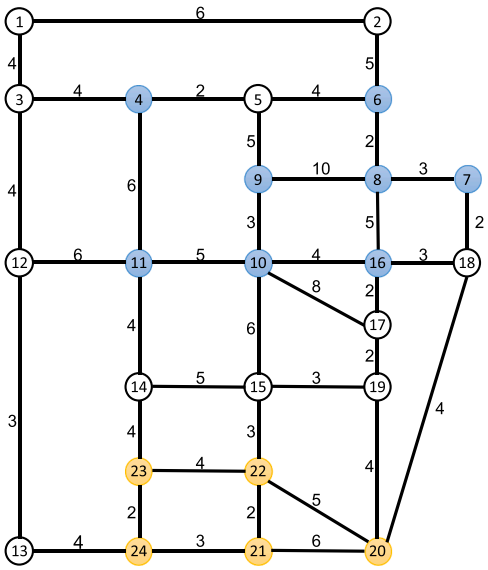


Fig. 4. Sioux Falls network.

Table 1
Passenger demands.

PSGR ID	Origin Node	Dest. Node	Dep. Time	TravTime (taxi)	Fare (taxi)
1	11	24	3	10	20
2	9	20	11	14	28
3	16	24	3	15	30
4	16	21	6	12	24
5	16	23	10	14	28
6	7	22	12	11	22
7	4	21	8	17	34
8	4	23	12	14	28
9	6	20	7	11	22
10	4	22	12	18	36
11	9	21	8	14	28
12	16	22	13	10	20
13	16	22	12	10	20
14	8	20	6	9	18
15	10	20	10	11	22
16	11	22	13	12	24
17	11	22	15	12	24
18	10	20	14	11	22
19	10	20	9	11	22
20	10	20	10	11	22

and a taxi, the probability of each passenger choosing the CB bus can be estimated by MNL model (Eqs. (1) and (2)). Here we use the set of parameters obtained by a stated-preference survey, which will be detailed in Section 5.2.2. We assume there are at most two buses, i.e., $|M| = 2$. Each bus has the capacity of 10 passengers and the operational cost is 2.5 RMB/km. The ticket price is 3 RMB/km.

To obtain the optimal solution of the proposed model, we adopt a brute force method to enumerate all possible solutions. Given the above network and travel demands, there are 121,735 possible routes starting from one of the origin nodes and ending at one of those destination nodes. These routes further generate a number of 121, 735×20 of vertex sequences, i.e., (*location*, *time*) sequences, because each route can have 20 possible timetables in the time window of 20 min. For each vertex sequence, we calculate the probability of each passenger taking this vertex sequence and further compute the profit. For each pairwise combination of vertex sequences, where two buses are used, we assign each passenger to the sequence that can bring higher expected ticket revenue and then compute the total profit. This method can find the optimal assignment in a shorter time comparing to the method of enumerating all the possible assignments of 20 passengers to the two sequences. The two optimal routes with timetables, i.e., two (location, time) sequences, passengers who have larger than 10% probability of choosing each CB route, and the maximum profit are listed in Table 2(a). We also implement our proposed heuristic algorithms to this network and the 20 travel demands. Firstly, two groups of passengers with similar demands are discovered by our clustering method, i.e., passengers (2, 15, 18, 19, 20) and passengers (8, 10, 16, 17). Then we apply the stop deployment, routing, and timetabling methods on each group and obtain two bus lines. Finally, we

Table 2
Comparison of optimal and heuristic bus lines.

Bus ID	Route $\{(i, t)\}$	Passenger ID (Prob. Choosing CB)	Profit
	(a) Optimal Bus Lines		
1	(9,7) → (10,9) → (16,13) → (18,16) → (20,20)	2 (54.6%), 15 (68.8%), 18 (55.8%), 19 (65.7%), 20 (68.8%)	49.00
2	(4,9) → (11,17) → (14,21) → (23,26) →(22,31) → (21,33)	7 (20.7%), 8 (58.1%), 10 (56.5%), 16 (62.2%), 17 (68.5%)	42.91
	(b) Heuristic Bus Lines		
1	(9,7) → (10,9) → (16,13) → (18,16) → (20,20)	2 (54.6%), 15 (68.8%), 18 (55.8%), 19 (65.7%), 20 (68.8%)	49.00
2	(4,10) → (11,18) → (14,22) → (23,27) → (22,32)	8 (67.8%), 10 (66.4%), 16 (51.9%), 17 (58.8%)	39.82

check whether the two bus lines can be merged. The heuristic results are listed in Table 2(b). From Table 2(a) and (b), we can observe that the bus lines generated by our heuristic algorithms are similar to the optimal bus lines: our heuristic bus line fail to serve passenger 7 and hence has an optimality gap of 3.36%.

We further randomly generate 20 travel demands 20 times to verify the optimality loss and computational efficiency improvement of our heuristic solution. Table 3 lists the profits and CPU times of the optimal solution and the heuristic solution. The mean optimality gap between the optimal solution and our heuristic methods is 17.55% and the standard deviation is 0.0986. The heuristic solution is 8285 times faster than the optimal solutions on average. It is worth to mention that although the brute force method for searching the optimal results takes around 160 s in this experiment, applying this brute force method to a city-scale network with millions of nodes cannot obtain the optimal result within a reasonable time. Take a network with v nodes as an instance, suppose there are $|P|$ possible paths between any two nodes. Given a time window T and a budget of $|M|$ buses, there are $|P||T|$ possible vertex (i.e., (location, time)) sequences, the searching space for the optimal combination of vertex sequences is $\sum_{m=1}^{|M|}\binom{|P\,||\,T|}{m}$, which could be huge because the number of simple paths $|P|$ in a graph can be very large, e.g. $O(v!)$ in the complete graph of order v.

5.2. Case study on the city-scale network

Taxi trajectory data is a reliable data source for estimating travel demands for CB systems, as it provides comprehensive and detailed travel information of taxi passengers and reveals the actual taxi demands citywide. A part of taxi passengers choose taxis because they prefer direct, speedy and cozy transit services, while CB systems can also provide such comparable services with the extra advantage of lower fare. Hence, these passengers have high potential to be CB customers. Note that the demands of CB systems could also come from other sources such as passengers using public transit services. In practice, our CB-Planner can be implemented with proper travel demand estimation from multiple data sources (Liu et al., 2014). This case study on the taxi trajectory data set is to demonstrate the effectiveness of our CB-Planner, i.e., CB-Planner can plan profitable bus lines given an input of city-wide travel demands, and these CB lines can provide efficient transit services with lower fare.

In this case study, each passenger trajectory is regarded as a travel demand, and each demand can be served by one of the two modes, i.e., CB or taxi. Namely, the pick-up point, drop-off point and pick-up timestamp of a trajectory are regarded as the origin, destination and departure time of the travel demand, respectively. The travel time and fare of the trajectory are taken as TravTime and Fare in utility function (Eq. (2)) of taxi mode. In the following, we first describe the data set, experimental settings and baseline algorithms, and present experimental results.

5.2.1. Taxi trajectory data set

The taxi trajectory data set was collected from Nanjing, China from June 1st to 29th, 2010. There are 7476 taxis with total 540,577,652 GPS records. The average frequency of a GPS record of a taxi is about 32 s. Each record contains the taxi ID, timestamp, latitude, longitude, and working status of the taxi. We extract passenger trajectories from each taxi trajectory, by detecting the pick-

Table 3
Optimality gaps of 20 groups of travel demand samples.

Rand. ID	Opt. Profit (RMB)	Opt. Time (sec)	Heuri. Profit (RMB)	Heuri. Time (sec)	Gap	Rand. ID	Opt. Profit (RMB)	Opt. Time (sec)	Heuri. Profit (RMB)	Heuri. Time (sec)	Gap
1	108.84	158.2	97.90	0.018	10.05%	11	165.63	153.6	108.33	0.018	34.6%
2	134.54	158.4	105.90	0.038	21.29%	12	105.55	161.8	105.55	0.010	0%
3	116.94	160.2	80.0	0.012	31.59%	13	126.60	158.2	92.01	0.030	27.32%
4	154.71	158.6	136.52	0.028	11.75%	14	178.91	148.0	154.67	0.027	13.55%
5	121.33	153.6	95.49	0.018	21.29%	15	122.31	158.8	107.20	0.026	12.35%
6	135.79	168.0	135.79	0.015	0%	16	170.93	148.5	145.29	0.025	15%
7	160.94	152.3	132.51	0.028	17.67%	17	130.08	162.3	102.87	0.033	20.92%
8	122.28	168.1	95.63	0.019	21.79%	18	118.48	159.0	81.70	0.020	31.04%
9	137.71	152.3	109.99	0.012	20.12%	19	188.93	157.0	172.48	0.027	8.71%
10	129.00	154.3	96.2	0.014	25.43%	20	100.26	155.8	93.76	0.011	6.48%
Optimality Gap Mean:					17.55%	Optimality Gap SD:					0.0986

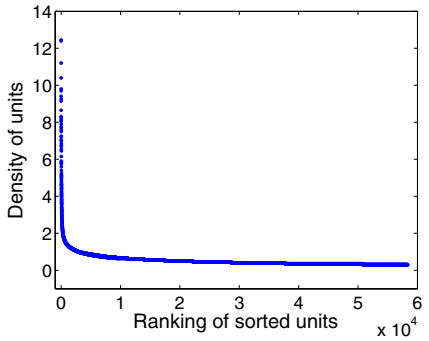


Fig. 5. Sorted density of the top 10% units.

up activities and drop-off activities. After removing those noise trajectories with less than three minutes, we extract 5,456,051 passenger trajectories. Due to the different travel patterns in weekdays and weekends & holidays, we split the one-month data into two subsets, namely, trajectories in the 20 weekdays and trajectories in the 9 weekends and holidays, and conduct experiments on the two data sets, separately.

5.2.2. Experiment settings

Parameter setting for travel demand clustering. In partitioning the travel demand space into units for clustering, the unit size should be properly set. The clustering quality will be deteriorated with too large size or too small size (Nagesh et al., 2001). We partition each spatial dimension of travel demand space, (i.e., **stLat**, **stLng**, **edLat** and **edLng**), into intervals of 500 meters, based on the service radius of CB stops, which is also set as 500 meters (Daniels and Mulley, 2013). The temporal dimension, (i.e., **stTime**), is partitioned into intervals of 30 min.

To explore density threshold, we first calculate the density for each unit, i.e., the average daily number of travel demands in each unit. Then, all the units are sorted by their density. Fig. 5 plots the density of top 10% units for the weekday data set. We observe most of the units have the density varies in a small range between 0 and 1.6, and the top units have the density that varies from 1.6 to 12.4. Therefore, we define a unit as the dense unit when its density is above 1.6, i.e., the density threshold is set to 1.6. We also get the same density threshold for the weekend & holiday data set.

Parameter estimation for the Multinomial Logit model. We employed the Multinomial Logit (MNL) model to estimate the probability of a passenger choosing CB over a set of alternative transport modes as described in Section 3.1. In this case study, as we are estimating the probabilities of taxi passengers shifting to CB buses, this MNL model is simplified into two mode choices, i.e., CB and taxi.

We conduct a small-scale Stated-Preference (SP) survey (Shiftan et al., 2006), which has been widely used for assessing the potential demands for a new transportation service. It presents hypothetical situations to the respondents, who are then asked to choose the preferred alternative (i.e., CB or taxis) based on the given attributes, without necessarily experiencing them in real situations (Hanley et al., 2001). In this survey, we consider the following six attributes, i.e., travel time by taxi ($\text{TravTime}_{\text{taxi}}$), travel cost by taxi ($\text{Fare}_{\text{taxi}}$), travel time by CB ($\text{TravTime}_{\text{CB}}$), travel cost by CB ($\text{Fare}_{\text{CB}}$), walk distance to the nearest CB stop ($\text{WalkDist}_{\text{CB}}$), and time adjustment to catch a CB bus ($\text{TimeAdj}_{\text{CB}}$). Hypothetical attribute values are generated based on the real taxi trips from the trajectory data. For example, the travel time by taxi are set from 5 min to 50 min because more than 90% of taxi trips are within this range in the real data set. The travel times by CB are generated accordingly by setting different ratios comparing to taxi travel time. The combinations of the values of each attribute are generated for SP choice scenarios using fractional factorial designs (Box et al., 2005). Table 4 illustrates one of the SP choice scenarios. There are a total of 56 respondents, and each respondent was required to choose preferred travel mode (CB or taxis) for 30 SP scenarios. And thus we collected 1680 observations in total.

We implement the maximum likelihood estimation on the survey data, and present the results in Table 5. ASC_{CB} and ASC_{taxi} are the two constant coefficients, i.e., $\beta_{0,c}$ in the utility function (Eq. (2)). As we set taxi as the reference, ASC_{taxi} is fixed to zero. The coefficients of $\text{WalkDist}_{\text{CB}}$ and $\text{TimeAdj}_{\text{CB}}$ are negative and significant, showing that with the shorter walk distance and time adjustment of CB, the more likely people choosing CB services. The coefficients of TravTime and Fare of both CB and Taxi are negative and significant, this is means that, given fixed $\text{TravTime}_{\text{taxi}}$ and $\text{Fare}_{\text{taxi}}$, the shorter CB travel time and fare, the higher probability of choosing CB, similarly, given fixed $\text{TravTime}_{\text{CB}}$ and Fare_{CB}, reducing taxi travel time and fare will increase the probability of choosing taxi.

Other parameter settings. The default values of the rest parameters are listed in Table 6. In addition, we focus on demand clusters in which the average daily number of travel demands is more than 40.

5.2.3. Baseline algorithms

We develop baseline algorithms to evaluate the performance of CB-Planner. The main reasons are: (1) CB-Planner is the first framework that designs bus lines for CB systems. (2) Existing traditional bus route planning methods cannot be used for comparison either, due to the different transit service objectives. To demonstrate the effectiveness of CB-Planner, we propose the following baselines.

Table 4
An example of an SP choice scenario.

Mode	WalkDist	TimeAdj	TravTime	Fare	Preferred mode
Taxi	–	–	35 min	40 RMB	
CB	100 m	5 min	45.5 min	24 RMB	

Table 5
Results of mode choice model estimation.

Attributes	ASC_{CB}	ASC_{taxi}	$WalkDist_{CB}$	$TimeAdj_{CB}$	$TravTime_{CB}$	$Fare_{CB}$	$TravTime_{taxi}$	$Fare_{taxi}$
Coefficients (β)	0.969	0	−0.007	−0.140	−0.241	−0.196	−0.327	−0.145
std_{err}	0.140	–	0.000	0.007	0.008	0.007	0.011	0.006
p-values	0.000	–	0.000	0.000	0.000	0.000	0.000	0.000

Baseline for CB stop deployment. We compare our CB stop deployment (CBDeploying) with the baseline method k-medoids. The classical k-medoids is unable to find the minimum number of stops k because it takes value of k as an input. Hence in this baseline, we iteratively implement k-medoids to deploy k stops from $k = 1$ to a number that *CovPct* percentage of origin/destination locations from a cluster are covered by the k stops within walk distance of *CovRad*. Different from our CBDeploying, this baseline initiates the locations of the k stops randomly at each iteration.

Baselines for CB line planning. Given the deployed CB stops, we have the following two baselines to plan CB lines.

(1) One cluster one route(1C1R): This baseline does not employ our CBMerging, and only plans a CB line with our CBRouting and CBTimetabling for each demand cluster.
(2) Fixed Time Interval (FTI): This baseline algorithm adopts a timetable with a equal headway for a route. We set the headway as 30 min, i.e., the time interval between every two successive bus trips along the route is 30 min. With this timetable, the departure time adjustment of a passenger would be no longer than 15 min.

5.2.4. Evaluation on CB stop deployment

Effectiveness of clustering. Tables 7 and 8 summarize the 9 clusters in weekdays and 16 clusters in weekends & holidays discovered by CB-Planner. These clusters reveal the similar travel demands of taxi passengers, i.e., close origin locations, destination locations and departure times, which motivates us to build CB lines for these passengers. For example, in weekdays (Table 7), Cluster 1 appears from 7:00 to 20:00 each day, during which, about 197.8 passengers travel from the origin area within 0.5 km × 1 km, to the destination area within 0.5 km × 0.5 km. The average time gap between two successive passengers is about 2.9 min, which means every 2.9 min, there is a passenger traveling from the origin area to the destination area.

The number of demand clusters (i.e., travel patterns) discovered in weekends & holidays (Table 8) is more than that in weekdays (Table 7). This reveals the actual travel pattern in a city: in weekdays, commuters share a few travel patterns, such as commuting between residential areas and working areas, while they have more diverse travel patterns in weekends & holidays, such as traveling between various entertainment or business districts.

Fig. 6 shows the standard deviation of the number of travel demands in each cluster in different days of the two data sets. We observe that the number of demands fluctuates around the average within a small range, demonstrating the repeatability of these clusters and indicating that building CB lines for these clusters could have sustainable profit.

Fig. 7 shows the distribution of departure time gap between each two successive demands using boxplot (Frigge et al., 1989) for each cluster. The band inside each box is the median value of departure time gap, and each box shows the interquartile range (IQR) of the gaps, i.e., the bottom and top of the box are the 25th and 75th percentiles. The small interquartile ranges in Figs. 7(a) and (b) demonstrate the high temporal densities of demand clusters, i.e., the departure time gaps between most of two successive passengers are within 10 min.

Effectiveness of stop deployment. Fig. 8 shows the number of CB stops deployed by our CBDeploying and the baseline k-

Table 6
Default parameter setting.

Parameter	Value
Coverage radius of a CB stop *CovRad*	500 m
Coverage percentage of CB stops *CovPct*	95%
Capacity of a CB bus ϕ	40
Ticket price for each passenger ρ	3 RMB/km
Operational cost of a bus γ	9 RMB/km
Dwell time at a CB stop	1 min

Table 7
Clusters in weekdays.

Cluster ID	Duration	Avg. no. traj.	Avg. time gap	Origin area (km × km)	Destination area (km × km)
1	07:00–20:00	197.8	2.9	0.5 × 1	0.5 × 0.5
2	07:30–09:30	41.2	2.5	1.5 × 1	0.5 × 0.5
3	08:30–20:30	189.2	3.3	0.5 × 0.5	0.5 × 0.5
4	09:00–17:00	65.3	6.9	0.5 × 0.5	0.5 × 0.5
5	09:30–17:30	157.0	3.2	1 × 1.5	0.5 × 0.5
6	09:30–17:30	93.0	5.1	0.5 × 1	0.5 × 0.5
7	10:30–17:00	87.7	3.4	1.5 × 1.5	0.5 × 0.5
8	11:30–16:30	44.8	6.7	0.5 × 0.5	0.5 × 0.5
9	21:00–24:00	101.5	1.7	2 × 1.5	0.5 × 1

medoids with $CovPct = 95\%$ and $CovRad = 500$, (i.e., 95% passengers in each cluster can access to the nearest CB stops within 500 m). Fig. 9 plots the walk distance distribution of passengers to the nearest CB stops deployed by CB-Planner. We observe that (1) our CBDeploying deploys a smaller number of CB stops than k-medoids, and (2) more than 75% passengers only need to walk to the nearest stops within 300 m for both weekdays and weekends & holidays data sets. Actually, the average walk distance is 108 m and 130 m in both data sets, respectively. Both Figs. 8 and 9 demonstrate our CB-Planner can deploy a smaller number of CB stops than k-medoids, and these stops can be accessed by the most passengers with short walk distances.

5.2.5. Evaluation on CB line planning

Profit evaluation. We measure the effectiveness of the CB line planning of CB-Planner using the estimated profit of planned CB lines. We compare the CB-Planner with the baseline 1C1R and FTI to study the effectiveness of CBMerging and CBTimetabling in our CB-Planner framework, respectively. As presented in Fig. 10(a) and (b), CB-Planner and FTI generate 5 CB lines from the 9 trajectory clusters discovered in weekdays, and 11 CB lines from the 16 clusters in weekends & holidays, with the merge scheme CBMerging, while 1C1R generates one CB line from each demand cluster without the merging scheme.

Table 8
Clusters in weekends and holidays.

Cluster ID	Duration	Avg. no. traj.	Avg. time gap	Origin area (km × km)	Destination area (km × km)
1	06:30–18:30	231.7	2.7	0.5 × 1	0.5 × 0.5
2	07:30–17:30	69.7	4.6	0.5 × 0.5	0.5 × 0.5
3	08:30–17:30	158.8	3.3	1 × 1.5	0.5 × 0.5
4	08:30–21:00	205.2	3.1	1 × 0.5	0.5 × 0.5
5	09:30–16:30	69.9	4.8	0.5 × 0.5	0.5 × 0.5
6	09:30–18:00	118.1	4.1	0.5 × 1	0.5 × 0.5
7	10:00–17:00	100.7	3.5	1.5 × 1.5	0.5 × 0.5
8	10:30–17:30	88.3	3.7	1.5 × 2	0.5 × 0.5
9	12:00–17:30	58.8	4.6	1 × 0.5	0.5 × 0.5
10	12:00–18:30	131.4	2.7	1 × 2	1 × 1
11	13:30–17:00	40.4	4.5	1 × 1	0.5 × 0.5
12	13:30–17:30	48.9	3.7	1 × 1.5	1 × 1
13	19:00–22:00	66.6	2.4	1 × 1.5	1 × 1
14	20:30–24:00	97.0	2.0	1 × 1.5	0.5 × 1
15	21:00–23:30	41.9	3.8	1 × 1.5	0.5 × 0.5
16	01:00–04:00	49.1	3.8	0.5 × 0.5	1 × 2

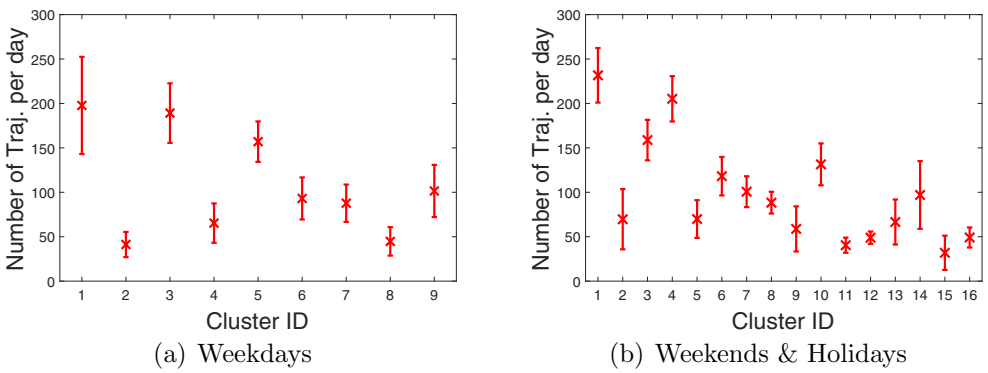


Fig. 6. Number of trajectories per day in each cluster.

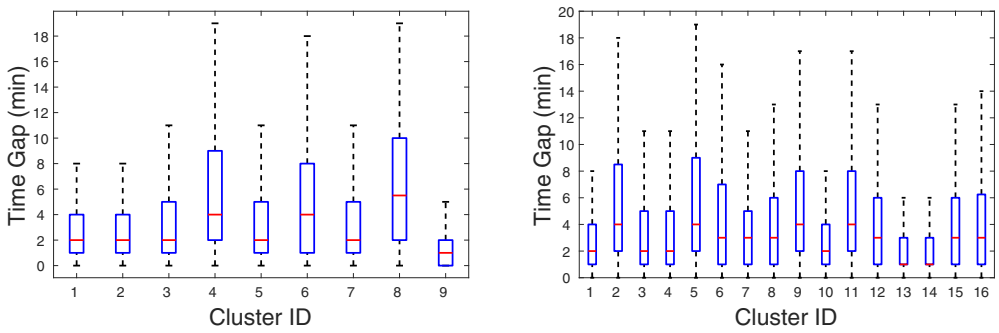


Fig. 7. Departure time gap between two successive travel demands in each cluster. (A boxplot showing the minimum, 25th percentile, median, 75th percentile, and maximum value of time gap.)

To demonstrate the effectiveness of CBMerging, we compare the profit of the merged CB lines (Lines 4 and 5 in Fig. 10(a) and Lines 10 and 11 in Fig. 10(b)) with the total profit of the corresponding unmerged CB lines (as presented with the stacked bars of 1C1R). We can observe that the merged CB lines generated by CB-Planner achieve higher profit than the total profit from the unmerged lines planned by 1C1R. Note that after merging CB lines with CBMerging, some passengers would spend more travel time along a merged CB line, due to more intermediate stops, so the probabilities for them to take the CB line will be lower. However, when the benefit earned by ride-sharing from the rest passengers are greater than the loss of passengers, the merged CB line can still be more profitable than the unmerged ones. This explains why the merged CB lines earn more profit than the lines planned by 1C1R.

To demonstrate the effectiveness of CBTimetabling, we compare CB lines generated by CB-Planner with the lines generated by FTI. We can observe that each CB line generated by CB-Planner earns more profit than the line generated by FTI. This is because FTI assigns a timetable with equal headways for each CB line, while our CB-Planner generates the optimal timetable for each CB line by maximizing its profit. In addition, the optimal timetables generated by CBTimetabling in weekdays and weekends & holidays are presented in Tables 9 and 10, respectively. We observe that the timetable of each CB line are unevenly distributed due to the profit maximization.

Travel experience validation. To validate the travel experience of the CB lines generated by CB-Planner, we first compare travel time and fare by CB buses with the travel time and fare by taxis, respectively, and then display the time adjustment of passengers to the optimal timetable of each CB line. Note that we have estimated the probability of a passenger choosing CB buses by Eq. (1). Without loss of generality, we validate the travel experience of the passengers who have more than 50% probability taking a CB bus.

Fig. 11 shows the increase in travel time and saving in fare of CB buses when comparing with those by taxis. We observe that the percentage of travel fare saving is larger than the percentage of extra travel time for each CB line in both the weekdays and weekends & holidays data sets. For example, in Fig. 11(a), passengers who choose CB line 2 would spend 17% longer travel time than by taxis, but spend 45% less money. This demonstrates that even though those passengers would sacrifice in travel time when choosing CB buses, they still benefit more in travel fare.

Fig. 12 plots the departure time adjustment of passengers to the optimal timetable of each CB line for both the weekdays and weekends & holidays data sets. We observe that more than half passengers only need to adjust their departure time within 10 min, and more than 75% passengers only need to adjust their departure time within 15 min for both data sets. The average departure time adjustments in both data sets are 8.0 min and 8.3 min, respectively.

Impact of ticket price setting. The ticket price of CB bus affects the number of passengers choosing CB buses and thus affects the profit. To study the impact of price setting, we compare the total numbers of passengers taking CB buses and the total profit under

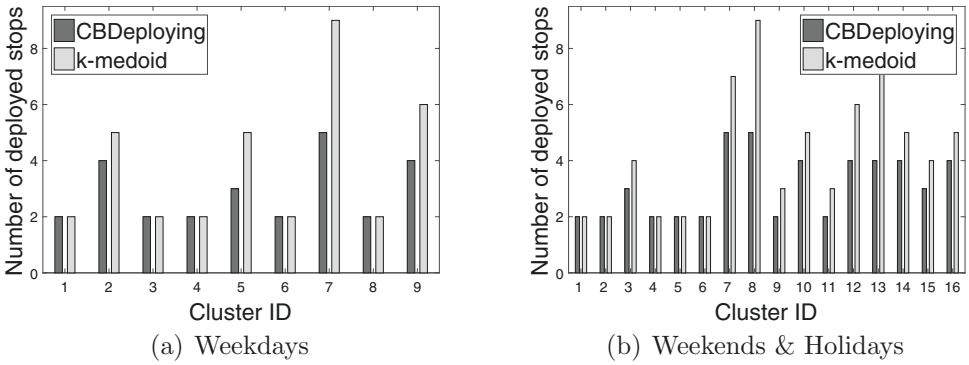
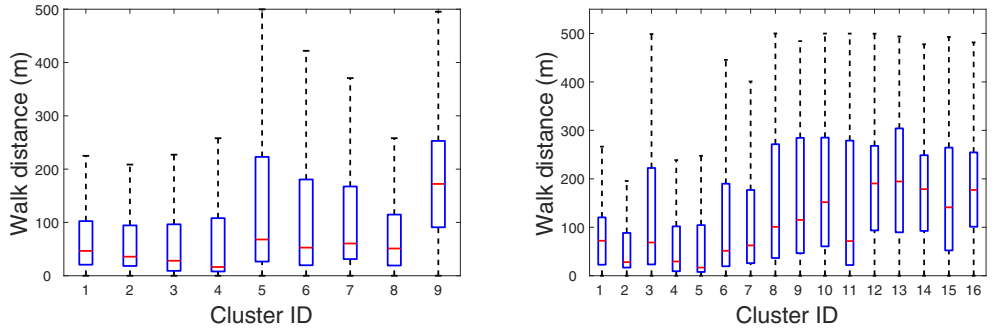
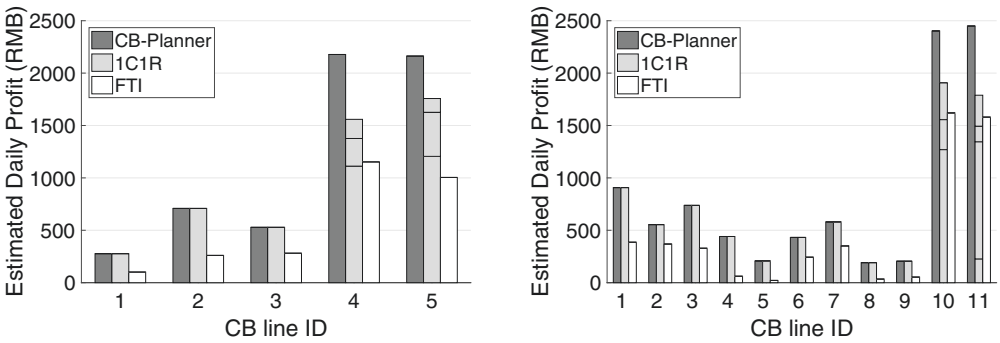


Fig. 8. Number of CB stops deployed in each cluster.



(a) Weekdays (b) Weekends & Holidays

Fig. 9. Walk distances to the nearest CB stops in each cluster. (A boxplot showing the minimum, 25th percentile, median, 75th percentile, and maximum value of walk distance.)



(a) Weekdays (b) Weekends & holidays

Fig. 10. Estimated profit of planned CB lines.

Table 9
Timetable in weekdays.

Line 1	Line 2	Line 3	Line 4		Line 5	
08:05	11:25	21:50	08:50	14:00	08:15	14:05
08:50	12:30	22:35	09:35	14:40	09:20	14:50
09:30	13:30	23:15	10:15	15:25	10:10	15:30
–	14:25	24:00	10:55	16:05	11:00	16:15
–	15:15	–	11:40	16:50	11:50	17:05
–	16:35	–	12:25	17:45	12:35	17:55
–	–	–	13:10	–	13:20	–

Table 10
Timetable in weekends & holidays.

Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10		Line 11	
10:50	13:55	12:40	13:10	14:05	20:00	21:15	21:50	01:05	07:05	13:55	08:55	15:35
12:00	15:15	14:05	14:10	15:30	20:50	22:05	22:55	02:10	07:55	14:40	09:50	16:30
13:00	17:00	15:20	15:00	16:25	21:50	23:00	23:55	03:00	08:45	15:25	10:35	17:25
14:15	–	16:20	15:55	–	–	24:00	–	03:50	09:30	16:25	11:20	18:20
15:05	–	17:20	17:00	–	–	–	–	–	10:25	17:10	12:10	19:15
16:00	–	18:25	–	–	–	–	–	–	11:15	18:00	12:55	20:40
17:00	–	–	–	–	–	–	–	–	12:10	19:05	13:50	–
–	–	–	–	–	–	–	–	–	13:05	–	14:40	–

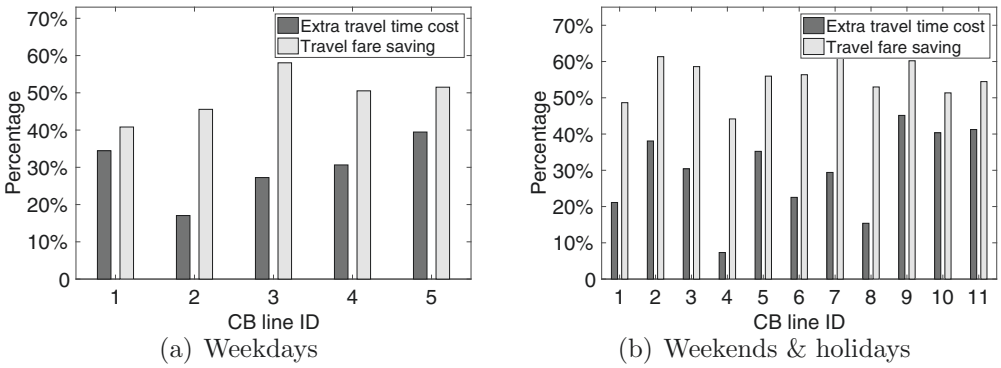


Fig. 11. Percentage of extra travel time and travel fare saving of CB buses compared with taxi.

different ticket prices.

Fig. 13(a) plots total numbers of passengers taking CB buses under different ticket prices. Generally, the number of CB passengers decreases as the price increases. However, when the price is set to 1 RMB/km, the passengers are fewer than those when the price is 2 RMB/km. This is because when the price is too low, building CB lines for some demand clusters cannot make any profit, even though these demands may have high probabilities choosing CB services due to the low price. We can observe that the number of passengers taking CB buses is maximized when the price is 2 RMB/km.

Fig. 13(b) plots total profit under different ticket prices. When the price increases from 1 to 3 RMB/km, the profit first increases as a higher price leads to more revenue. The profit then decreases when the price exceeds 3 RMB/km. This is because a higher price also causes the loss of passengers, resulting in less profit. As we can observe, the profit is maximized when the price is set to 3 RMB/km.

6. Conclusion and future work

In this paper, we proposed CB-Planner: a holistic framework which aims to strategically plan bus lines for customized bus (CB) systems. To plan bus lines that can balance service quality with profit, we proposed a mathematical programming formulation to simultaneously optimize bus stop locations, bus routes, timetables and passengers' probabilities of choosing CB. We then developed a heuristic solution framework that includes a grid-density based clustering method for discovering potential travel demands efficiently, a bus stop deployment algorithm to minimize the number of stops and walking distance, and dynamic programming based routing and timetabling algorithms for maximizing estimated profit. We conducted an experiment on a small-scale network to verify the performance gap between the optimal solution and the heuristic solution, and the results show that the mean of the gap is 17.55% with s.d. of 0.0986. CB-Planner was then evaluated in a realistic situation using one-month taxi trajectory data in Nanjing, China. The results show that our framework can generate CB lines with higher profit, compared with baseline methods. And these CB lines can provide efficient transit services with short walk distances and small departure time adjustments. The moderate increase in travel time is paid off by the significant savings in travel fare.

In future, we have two research directions: (1) Study how to develop efficient solutions to jointly optimize the CB stop locations, CB routes and their timetables, so as to capture the interactions and feedbacks between stop deployment, routing, timetabling and passengers' choices on CB buses. (2) Study on the subjective factors that affect passengers' choice on CB services, such as comfort and personal travel habit, and incorporate them for better demand estimation for CB systems.

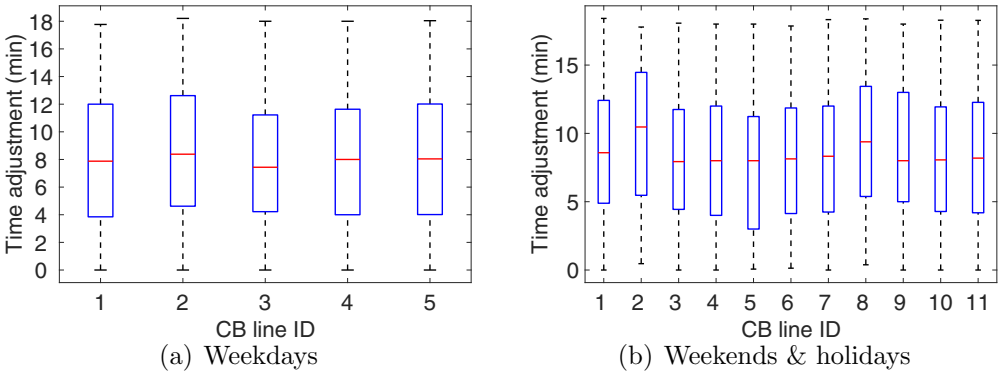
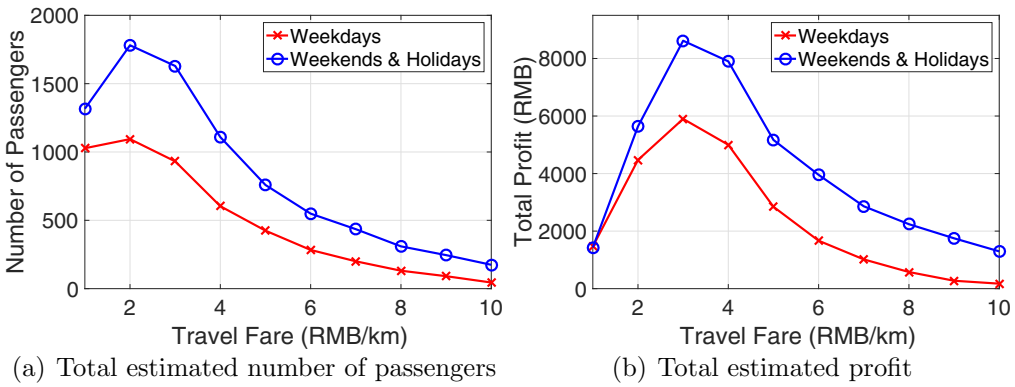


Fig. 12. Departure time adjustment to the optimal timetable in each line. (A boxplot showing the minimum, 25th percentile, median, 75th percentile, and maximum value of departure time adjustment.)



(a) Total estimated number of passengers (b) Total estimated profit

Fig. 13. Impact of different prices on number of passengers and profit.

Acknowledgements

Chi-Yin Chow was supported in part by the City University of Hong Kong under Grant 7004890. Yanhua Li was supported in part by Nissan Science Foundation grants CNS-1657350 and CMMI-1831140, and a research grant from DiDi Chuxing Inc.

References

Aggarwal, C.C., Reddy, C.K., 2013. Data Clustering: Algorithms and Applications. CRC Press.
Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P., 1998. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of ACM SIGMOD.
Bagloee, S.A., Ceder, A.A., 2011. Transit-network design methodology for actual-size road networks. Transport. Res. Part B: Methodol. 45 (10), 1787–1804.
Bastani, F., Huang, Y., Xie, X., Powell, J.W., 2011. A greener transportation mode: flexible routes discovery from gps trajectory data. In: Proceedings of ACM SIGSPATIAL.
Berend, D., Tassa, T., 2010. Improved bounds on bell numbers and on moments of sums of random variables. Probab. Math. Stat. 30 (2), 185–205.
Box, G.E., Hunter, J.S., Hunter, W.G., 2005. Statistics for Experimenters: Design, Innovation, and Discovery. Wiley-Interscience, New York.
Cancela, H., Mauttone, A., Urquhart, M.E., 2015. Mathematical programming formulations for transit network design. Transport. Res. Part B: Methodol. 77, 17–37.
Cao, Yang, Wang, Jian, 2017. An optimization method of passenger assignment for customized bus. Math. Problems Eng. 2017 Hindawi.
Ceder, A.A., Butcher, M., Wang, L., 2015. Optimization of bus stop placement for routes on uneven topography. Transport. Res. Part B: Methodol. 74, 40–61.
Chang, S.K., Schonfeld, P.M., 1991. Optimization models for comparing conventional and subscription bus feeder services. Transport. Sci. 25 (4), 281–298.
Chang, S.K., Yu, W.-J., 1996. Comparison of subsidized fixed-and flexible-route bus systems. Transport. Res. Rec.: J. Transport. Res. Board 1557 (1), 15–20.
Chen, C., Zhang, D., Li, N., Zhou, Z.H., 2014. B-planner: Planning bidirectional night bus routes using large-scale taxi gps traces. IEEE Trans. Intell. Transport. Syst. 15 (4), 1451–1465.
Chien, S.I., Spasovic, L.N., Elefsiniotis, S.S., Chhonkar, R.S., 2001. Evaluation of feeder bus systems with probabilistic time-varying demands and nonadditive time costs. Transport. Res. Rec.: J. Transport. Res. Board 1760 (1), 47–55.
Chinadaily, 2014. Zhangjiagang Opens Customized Tour Bus Lines. http://www.chinadaily.com.cn/m/jiangsu/zhangjiagang/2014-04/01/content_17397887.htm Accessed: 2015-06.
Chuanyu, Z., Wei, G., Jie, X., Shixiong, J., 2017. A study on dynamic dispatching strategy of customized bus. In: 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE). IEEE, pp. 751–755.
Cipriani, E., Gori, S., Petrelli, M., 2012. Transit network design: a procedure and an application to a large urban area. Transport. Res. Part C: Emerg. Technol. 20 (1), 3–14.
Daniels, R., Mulley, C., 2013. Explaining walking distance to public transport: the dominance of public transport supply. J. Transport Land Use 6 (2), 5–20.
Farahani, R.Z., Hekmatfar, M., 2009. Facility Location: Concepts, Models, Algorithms and Case Studies. Springer.
Frigge, M., Hoaglin, D.C., Iglewicz, B., 1989. Some implementations of the boxplot. Am. Stat. 43 (1), 50–54.
Guihaire, V., Hao, J.-K., 2008. Transit network design and scheduling: a global review. Transport. Res. Part A: Policy Pract. 42 (10), 1251–1273.
Hanley, N., Mourato, S., Wright, R.E., 2001. Choice modelling approaches: a superior alternative for environmental valuatioin? J. Econ. Surveys 15 (3), 435–462.
Ibarra-Rojas, O., Delgado, F., Giesen, R., Muñoz, J., 2015. Planning, operation, and control of bus transport systems: a literature review. Transport. Res. Part B: Methodol. 77, 38–75.
Jain, K., Vazirani, V.V., 2001. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. J. ACM (JACM) 48 (2), 274–296.
LeBlanc, L.J., 1988. Transit system network design. Transport. Res. Part B: Methodol. 22 (5), 383–390.
Li, Y., Luo, J., Chow, C.-Y., Chan, K.-L., Ding, Y., Zhang, F., 2015. Growing the charging station network for electric vehicles with trajectory data analytics. In: Proceedings of ICDE.
Likas, A., Vlassis, N., Verbeek, J.J., 2003. The global k-means clustering algorithm. Pattern Recogn. 36 (2), 451–461.
Liu, T., Ceder, A.A., 2015. Analysis of a new public transport service concept: customized bus in china. Transport Policy 39 (0), 63–76.
Liu, Y., Liu, C., Yuan, N.J., Duan, L., Fu, Y., Xiong, H., Xu, S., Wu, J., 2014. Exploiting heterogeneous human mobility patterns for intelligent bus routing. In: Proceedings of IEEE ICDM. pp. 360–369.
Ma, X., Wu, Y.-J., Wang, Y., Chen, F., Liu, J., 2013. Mining smart card data for transit riders' travel patterns. Transport. Res. Part C: Emerg. Technol. 36, 1–12.
Ma, J., Zhao, Y., Yang, Y., Liu, T., Guan, W., Wang, J., Song, C., 2017. A model for the stop planning and timetables of customized buses. PloS One 12 (1), e0168762.
Ma, J., Yang, Y., Guan, W., Wang, F., Liu, T., Tu, W., Song, C., 2017. Large-scale demand driven design of a customized bus network: a methodological framework and beijing case study. J. Adv. Transport.
Malandraki, C., Dial, R.B., 1996. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. Euro. J. Oper. Res. 90 (1), 45–55.
Mauttone, A., Urquhart, M.E., 2009. A route set construction algorithm for the transit network design problem. Comput. Oper. Res. 36 (8), 2440–2449.
McCall, C., 1977. Com-bus: A Southern California Subscription Bus Service. Tech. Rep. DOT-TSC-UMTA-77-13 Final Rpt., CACI, Incorporated, Transportation Systems Center and Urban Mass Transportation Administration.
McFadden, D., 1973. Conditional logit analysis of qualitative choice behaviour. In: Zarembka, P. (Ed.), Frontiers in Econometrics. Academic Press, New York, NY, USA,

pp. 105–142.
Michaelis, M., Schöbel, A., 2009. Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented heuristic. Public Transport 1 (3), 211.
Mingozzi, A., Bianco, L., Ricciardelli, S., 1997. Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints. Oper. Res. 45 (3), 365–377.
Ministry of transport, 2014. Ministry of transport of the people's republic of China. http://www.moc.gov.cn/zhuantizhuanlan/gonglujiaotong/gongjiaods/jingyanjl/201403/t20140324_1595555.html Accessed: 2015-06.
Munizaga, M.A., Palma, C., 2012. Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from santiago, chile. Transport. Res. Part C: Emerg. Technol. 24, 9–18.
Nagesh, H.S., Goil, S., Choudhary, A.N., 2001. Adaptive grids for clustering massive data sets. In: Proceedings of SDM.
Nayeem, M.A., Rahman, M.K., Rahman, M.S., 2014. Transit network design by genetic algorithm with elitism. Transport. Res. Part C: Emerg. Technol. 46, 30–45.
Nikolić, M., Teodorović, D., 2013. Transit network design by bee colony optimization. Exp. Syst. Appl. 40 (15), 5945–5955.
Park, H.-S., Jun, C.-H., 2009. A simple and fast algorithm for k-medoids clustering. Exp. Syst. Appl. 36 (2), 3336–3341.
Pelletier, M.-P., Trépanier, M., Morency, C., 2011. Smart card data use in public transit: a literature review. Transport. Res. Part C: Emerg. Technol. 19 (4), 557–568.
Perugia, A., Moccia, L., Cordeau, J.-F., Laporte, G., 2011. Designing a home-to-work bus service in a metropolitan area. Transport. Res. Part B: Methodol. 45 (10), 1710–1726.
Rota, G.-C., 1964. The number of partitions of a set. Am. Math. Monthly 71 (5), 498–504.
Saka, A.A., 2001. Model for determining optimum bus-stop spacingin urban areas. J. Transport. Eng. 127 (3), 195–199.
Shiftan, Y., Vary, D., Geyer, D., 2006. Demand for park shuttle services – a stated-preference approach. J. Transport Geogr. 14 (1), 52–59.
Szeto, W., Jiang, Y., 2014. Transit route and frequency design: bi-level modeling and hybrid artificial bee colony algorithm approach. Transport. Res. Part B: Methodol. 67, 235–263.
Tong, L., Zhou, X., Miller, H.J., 2015. Transportation network design for maximizing space–time accessibility. Transport. Res. Part B: Methodol. 81, 555–576.
Tong, L.C., Zhou, L., Liu, J., Zhou, X., 2017. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. Transport. Res. Part C: Emerg. Technol. 85, 451–475.
W. Wu, W.S. Ng, S. Krishnaswamy, A. Sinha, To taxi or not to taxi? Enabling personalised and real-time transportation decisions for mobile users. In: Proceedings of MDM, 2012.
Xinhuanet, 2013. Customized Bus Service Offered for City Dwellers in Beijing. http://news.xinhuanet.com/english/photo/2013-09/09/c_132705811_2.htm Accessed: 2015-06.
Zheng, Y., Chen, Y., Li, Q., Xie, X., Ma, W.-Y., 2010. Understanding transportation modes based on gps data for web applications. ACM Trans. Web 4 (1), 1:1–1:36.