Last-Mile School Shuttle Planning With Crowdsensed Student Trajectories

Panrong Tong, Wan Du, *Member, IEEE*, Mo Li, *Member, IEEE*, Jianqiang Huang,
Wenqiang Wang, and Zheng Qin

***Abstract*— By processing a large dataset composed of daily trajectories of thousands of students in Singapore, we find that, instead of simply picking up students from their homes, an optimal school shuttle planning system needs to learn the real transportation usage and plan across all potential pickup locations for every student to generate need-satisfying routes. It is challenging, however, to perform route planning over a large number of students each having multiple potential pickup locations. We develop a graph-based data structure that embeds potential pickup locations of all students with the awareness of real-world constraints and existing public transits. Based on the graph structure, we prove that the optimal last-mile school shuttle planning problem is NP-hard and thereafter design a Tabu-based expansion algorithm to solve the problem, which strikes at a proper balance between the savings of students' commute time and the total cost of operating the shuttle buses. Extensive experiments with large-scale real-world crowdsensed trajectory data demonstrate that our last-mile school shuttles can save the traveling time for most students by over 20% and the savings can be up to 65% for 10% of the students.**

***Index Terms*— Last-mile shuttle planning, crowdsensing systems, trajectory processing, graph-based data structure.**

I. Introduction

LAST-MILE shuttles have become critical for urban transportation in modern cities, since they complete last legs of individual trips by getting people from transportation networks to their final destinations (usually where public transportation does not reach) [22]. Intuitively, providing such services to students would be both beneficial and low-cost due to their common destinations (i.e., the schools) and commuting hours. However, planning a need-satisfying last-mile school shuttle service involves two key tasks, i.e., estimating transport demands (locations where users may need the shuttles) and optimizing bus routes according to the estimated demands [2].

Existing practices of school shuttle planning heavily relies on offline surveys or empirical experience to estimate transport demands, which may not be accurate and is often inefficient. They either completely ignore the public transportation by assuming all the transport demands start from students' homes(e.g., the door-to-door school shuttles, which pick up students directly from their homes), or have simplified approximation on how students utilizing the public transportation (e.g., the metro school shuttles, which assumes most of the students take metro to one or few major stations nearest to their school and picks up them from the one or few stations). Some recent data-driven studies learn the true transport demands from personal mobility data (e.g., cellular footprints or taxi trips) [5], [6], [19], [38]. With that, they formulate the route planning problem into classic optimization problems such as vehicle routing problem (VRP) [15]. For example, Feeder [38], a most recent work, plans a last-mile shuttle route that takes commuters from a metro station to their destinations. With the cellular data of mobile users, Feeder learns the rough locations of their destinations, clusters them as bus stops, and plans routes accordingly.

However, the above data-driven approaches are limited by the granularity of their observations and thus often oversimplify the true transport demands for the following two reasons. First, previous works [5], [6], [19], [38] implicitly assume only one potential pickup location for each individual, which is often not the case in practice. Given the multiple choices of transportation (e.g., bus, metro, walk) and their combinations, potential pickup locations of a student could include the home, the entries and exits of used public transits (e.g., bus stops, metro stations) and all the road segments walked. For example, Figure 1a shows a representative home-school trip that contains 23 potential pickup locations, including home (point A), metro stops (point B and C), bus stops (point D and E) and all the road segments traversed by walk (dash lines). According to our observation from the crowdsensed trajectory data, most students have 6 to 52 potential pickup locations, as summarized in Figure 1b. Second, previous works [5], [6], [19], [38] are based on simple proximity model which is unaware of real-world constraints and existing public transits. For example, simply clustering geographically proximate transport demands and assigning one bus stop to serve all of them has been widely used in previous works, which however is not accurate because walking between geographically proximate locations may incur high cost due to road constraints (e.g., crossing a street but from a faraway pedestrian flyover). Such unawareness

Manuscript received January 22, 2019; revised June 20, 2019 and October 9, 2019; accepted November 18, 2019. This work was supported in part by Singapore MOE Tier 2 under Grant MOE2016-T2-2-023, in part by NTU CoE under Grant M4081879, and in part by Alibaba-NTU JRI under Grant AN-GC-2018-006. The Associate Editor for this article was P. Wang. *(Corresponding author: Panrong Tong.)*

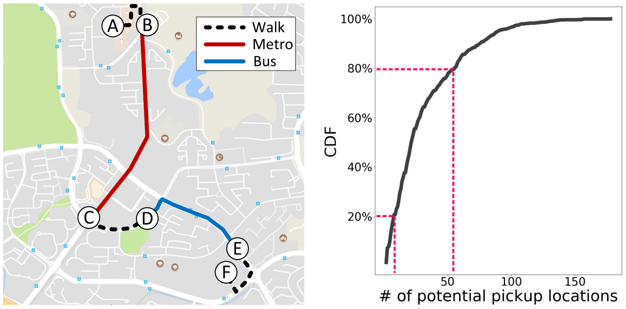
P. Tong and M. Li are with the Interdisciplinary Graduate School and the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: tong0091@e.ntu.edu.sg; limo@ntu.edu.sg).

W. Du is with the Department of Computer Science and Engineering, University of California, Merced, CA 95343, USA (e-mail: wdu3@ucmerced.edu).

J. Huang is with Alibaba DAMO Academy, Hangzhou 310030, China, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jianqiang.hjq@alibaba-inc.com).

W. Wang and Z. Qin are with the Institute of High Performance Computing, Singapore (e-mail: wangweq@ihpc.a-star.edu.sg; qinz@ihpc.a-star.edu.sg).

Digital Object Identifier 10.1109/TITS.2019.2956786



(a) A representative trip (b) Pickup locations of NSE trips

Fig. 1. Multiple pickup locations for each student.

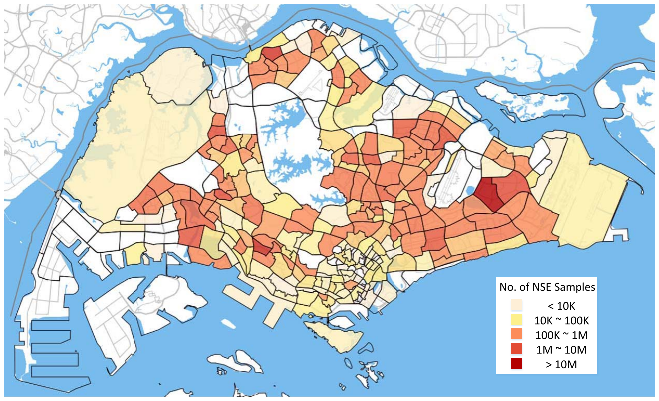


Fig. 2. NSE trajectories spatial distribution.

of constraints or conveniences leads to suboptimal bus route planning.

This paper presents last-mile school shuttle planning system that considers multiple pickup locations for each student with real-world constraints and existing public transits. We study daily trajectories crowdsensed from 2809 students through a nation-wide experiment of Singapore. The trajectories contain students' periodical locations and activity updates when they commute between home and school. The granularity and scale of our data allow us to finer profile their trips, fully examine the real transportation usage and generate more truthful demands to the shuttle service. Data samples in our trajectories are usually subject to a large location drift and may suffer from sparsity in some regions (see in Section II-B). We thus propose a trip profiling scheme to infer precise traveling paths and transportation modes, and thus extract potential pickup locations for all students.

With multiple pickup locations for every student, we gain an extra dimension of optimization and can thus derive better shuttle route plans. However, it is computational infeasible to extend existing VRP based algorithms to process such scale of demands. For 500 students and each having 20 potential pickup locations, blindly applying existing algorithms would result in the steps of selecting one possible pickup location for each student, and then running the algorithms once for each one out of 20^{500} possible combinations. Thus, this paper further proposes a novel graph-based data structure that embeds all transport demands on the road network. Such a graph based data structure aggregates similar demands from different students and provides a set of operators that facilitate route plan and update.

With the proposed data structure, we thus develop a customized Tabu expansion algorithm to find a proper subset of nodes in the graph as bus stops and lay the bus routes. The proposed solution is able to balance the commute time saved for all students and the operating cost of the shuttle buses.

We evaluate the performance of our last-mile school shuttle routes with the trajectories of 2809 students from 7 schools in Singapore. According to the evaluation results, our solution is able to save the commute time of most students by over 20%, where 10% of the students can save up to 65% while 75% of the students can save at least over 8%.

In summary, this paper makes the following contributions:

- This is the first paper, to the best of our knowledge, to demonstrate the necessity of considering multiple pickup locations of each individual for more efficient shuttle bus planning.
- This paper develops a holistic last-mile bus planning system with a set of techniques, including trip profiling from trajectory data, a novel graph-based data structure for embedding travel demands, and a graph-based Tabu expansion algorithm.
- Extensive experiments are performed on real-world crowdsensed data to evaluate the proposed system and compare with benchmark solutions.

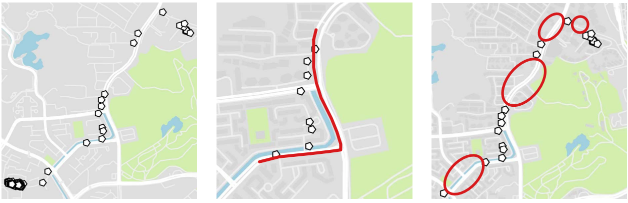
II. Motivation

We introduce the crowdsensing platform used in this study and demonstrate opportunities as well as challenges to utilize the crowdsensed data for last-mile school bus planning.

A. The National Science Experiment (NSE)

NSE [23] is a nation-wide experiment of Singapore that mobilizes the government and social forces to experiment a large-scale mobile crowdsensing system. A special designed mobile device is developed and assigned to a student during school days [32]. The device is equipped with a variety of sensors to measure the motion and environmental parameters, including three dimensional accelerations, light, temperature, noise levels, air pressure, etc. The data are sensed periodically and uploaded to the server opportunistically whenever the device connects to the wireless@SG WiFi hotspots (15,000+ free hotspots covering major public areas in Singapore [36], sponsored by SingTel [33] for free). In addition, the device also scans and sends back the signal strengths (RSSIs) from nearby WiFi hotspots and a third-party localization service from Skyhook [34] is invoked to determine the geolocations from its geo-WiFi database. The average time interval between geolocation updates is 15 seconds. Figure 2 visualizes the spatial distribution of all NSE samples (188,100,399 samples in total in one semester).

In this study, we primarily make use of the geolocations of each device which constitute a mobility trajectory of a specific student. The trajectory data have the following two



(a) A sample trajectory (b) Localization drift (c) Sparsity

Fig. 3. Imperfection of NSE data.

distinctive advantages compared with traditional survey based investigation or origin-destination based data profiling:

- **Data representativeness.** NSE trajectories consists of independently selected students from each participating school. The data group is representative to plan the last-mile school shuttles for each school.
- **Data richness.** NSE trajectories offer detailed trip profiles of individuals - the intermediate location updates during the trip as well as the time taken between those locations. With detailed understanding of transport choices of individuals, we are able to cross study the last-mile shuttle planning together with existing public transportation alternatives.

B. Challenges

Previous works [5], [6], [19], [38] are subject to inaccurate approximation on how and where the students may take a school shuttle. The crowdsensed NSE trajectory data brings opportunities to extract all potential pickup locations for each student. Such consideration factors in the students' route preferences and choices of available public transits, so the corresponding shuttle planning is based on user preference and at the same time offers higher freedom of optimization. Nevertheless, special challenges need to be carefully addressed to develop an effective and efficient solution.

Challenge 1 (Trajectory Profiling From Imperfect Trajectories): To extract potential pickup locations for each student, we need a detailed profile including precise traveling path and transportation modes on different path segments. However, NSE data are limited in localization accuracy. The geolocations of NSE data are estimated by WiFi hotspot based localization, which is not released by the third-party company [34]. The localization error is inevitable and often higher than those of GPS based approaches. Figure 3b shows how the derived locations (denoted as white dots) deviate from the real traveling path (denoted in red lines). According to our assessment, the NSE localization error ranges from a few meters to hundreds of meters with an average of 120 meters. At the same time, the locations are not evenly updated because there are inadequate number of audible WiFi hotspots in certain areas. As a result, the trajectory data have uneven location granularity, as Figure 3c suggests.

It has been known difficult to accurately map a sequence of coarse locations to a trajectory on the road map [21]. It is also difficult to accurately detect the transport mode with a trajectory of low resolution location samples [37].

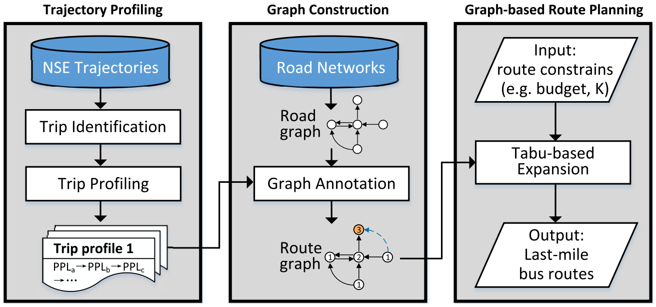


Fig. 4. System overview.

In this paper, we develop a novel approach to generate a representative travel profile for each student. It combines the NSE trajectory data and public transits information via Google Directions Service [16].

Challenge 2 (Embedding All Pickup Locations in an Efficient Data Structure): Previous works implicitly assume one specific pickup location for each student, which cannot be extended to handle the problem that each student having multiple potential pickup locations. Blindly applying existing algorithms would result in unacceptable computational cost, e.g., a problem with 500 students and each having 20 potential pickup points incurs the steps of selecting one possible pickup location for each student, and then running the algorithms once for each one out of 20^{500} possible combinations. In this paper, we propose a novel graph-based data structure that embeds all potential pickup locations of different students into the road networks. Similar transport demands of different students can thus be aggregated, and representative pickup locations can be derived to facilitate bus route optimization.

Challenge 3 (Computationally Feasible Bus Route Planning): Even with the aggregated transport demands, brute force searching for the best bus route is still infeasible, which we prove being NP-hard (in Section III-C). We extend the idea of Tabu search algorithm [14] - a metaheuristic originally designed for guiding a search to overcome local optimality in combinatorial optimization problems. The effectiveness of a tabu-based algorithm requires an application-specific design of its core components, however there is no graph-related design of tabu components. With the observation of convergent mobility pattern of students, we propose a tabu-based expansion algorithm which defines tailored tabu components under the graph structure and can efficient yield close to optimal bus routes.

III. Design

We design a holistic system for last-mile school shuttle planning to tackle the above three challenges. Figure 4 depicts the architecture of proposed system which consists of three key components, i.e., trajectory profiling, graph construction and graph-based route planning. First, trajectory profiling learns real transportation usage and extract potential pickup locations for every student, which offers an extra dimension of optimization and more need-satisfying bus routes can thus be derived. To handle extremely large search space brought by massive potential pickup locations, graph construction constructs a graph-based data structure that aggregates similar demands with the awareness of road networks; Finally, graph-based routing planning efficiently plans the routes that are able to balance the commute time saved for all students and the costs of the shuttles.

A. Trajectory Profiling

Potential pickup locations of a student include the home, the entries and exits of used public transits (e.g., bus stops, metro stations) and all the road segments walked. To extract those from NSE trajectories, we need to infer the precise traveling paths and transportation modes on different path segments.

Conventional methods solve the above problem by combining travel mode detection [37] and map matching [20], [21]. To infer the sample-wise travel modes, a classification model is normally trained from labeled historical data in terms of mobility features (e.g., distance, speed and acceleration). Many classification models can be used, such as Decision Trees, Random Forest, Bayesian Network, Support Vector Machine and Multilayer Perceptron. By feeding a target trajectory into the trained model, samples lying in the interchange of different travel modes are selected. Finally, map matching replaces the identified samples with corresponding points projected to the closest road segments.

However, according to our study, the above process performs poorly due to the two characteristics of NSE data (as suggested in Figure 3). First, most travel mode detection algorithms require consistent accuracy of inferred mobility features [37]. However, NSE data only provide coarse-grained and inconsistent mobility information. For example, due to the large localization error, the distance between two consecutive samples can be as large as several hundred meters, which results in overestimated speed. Large localization drifts also lead to inaccurate approximations for real locations. Second, due to the data sparsity problem, we cannot exploit potential pickup locations in some areas without enough samples.

We propose a two-step algorithm for trajectory profiling: 1) trip identification first detects the origin (home) and destination (school) of each student from NSE data; 2) trip profiling then infers precise traveling paths, transportation modes and thus potential pickup locations by combining the NSE trajectories and public transits information via Google Direction Service [16].

Trip Identification: A trip refers to a commute trajectory between home and school. Raw NSE trajectories are highly skewed: a small number of samples (2.1%) contains useful trip information, but the vast majority of samples are "stay" points, where the students stay for a long time (like at homes or schools). These "stay" points provide little travel information, but lead to high computational overhead and false positives when extracting potential pickup locations. Trip identification clusters nearby "stay" points as one representative point so that valid trips are consisted of informative points.

TABLE I

SUCCESS RATE OF REAL ROUTE EXTRACTION BY GOOGLE DIRECTIONS SERVICE. THE SUCCESS RATE OF THE GOOGLE PATH SET INDICATES THE PROBABILITY THAT THE SET CONTAINS THE REAL PATH TRAVERSED BY A STUDENT. THE SUCCESS RATE OF THE HEURISTIC SELECTION METHODS INDICATES THE LIKELIHOOD THAT THE METHOD OUTPUTS THE REAL PATH

Trajectory type		Walking	Public Transits	Driving	Overall
Google route set		91%	100%	89%	93%
Heuristics	Distance	0%	0%	44%	15%
	Duration	25%	0%	33%	19%
	Walking	69%	50%	0%	40%
	Transfers	69%	25%	0%	31%

Conventional algorithms normally detect "stay" points by clustering all points with either distance [39] smaller or density [11] larger than a threshold as one point. They do not perform well in processing NSE trajectories, due to 1) SENSg sensors use the received signal strengths from city-wide Wi-Fi access points to determine the device locations. The localization error is large and varies dynamically in space and time. It is hard to decide a global distance threshold as random noises could result in many false-positive identifications. 2) SENSg sensors automatically enter the sleep mode after 15-minute inactivity of movement. Some stay points may not have enough samples to be clustered by the density threshold even though the device did remain there for a long time.

Therefore, we adopt a time-weighted density clustering algorithm. Different from conventional algorithms, we use both density and time duration as the metric to identify "stay" points. Specifically, points for each student are classified as core points, reachable points or outliers based on the following criteria: 1) A point p is a core point if the total duration of nearby points (within 200 meters) exceeds an hour. Those nearby points are said to be directly reachable from point p. 2) A point q is reachable from p if there is a path $p1, ..., pn$ with $p1 = p$ and $pn = q$, where each $pi + 1$ is directly reachable from pi. 3) All points that are not reachable from any other points are outliers. Stay point clusters are formed by core points and all points that are reachable from them. We use the centroid locations to represent each cluster and identify valid trips.

2) Trip Profiling. With the home and school locations obtained from trip identification, we use the Google Directions Service [16] to generate all possible paths for each home-school pair. The service returns several well-segmented paths. Each path contains travel mode (i.e. walking, driving and taking public transits), precise intermediate locations and the estimated trip distance and duration. For each home-school pair, 9 paths can be suggested in general. The set of all suggested paths contains almost all reasonable choices, and usually includes the real path traversed by students. We manually label the real paths of 20 randomly selected students and their travel modes along the paths. The results in Table I indicate that 93% Google sets contain the real path and it works perfectly for students taking public transportation.

Next, we find the real path from the Google set. Google recommends the best route by 4 different heuristics, i.e., the route

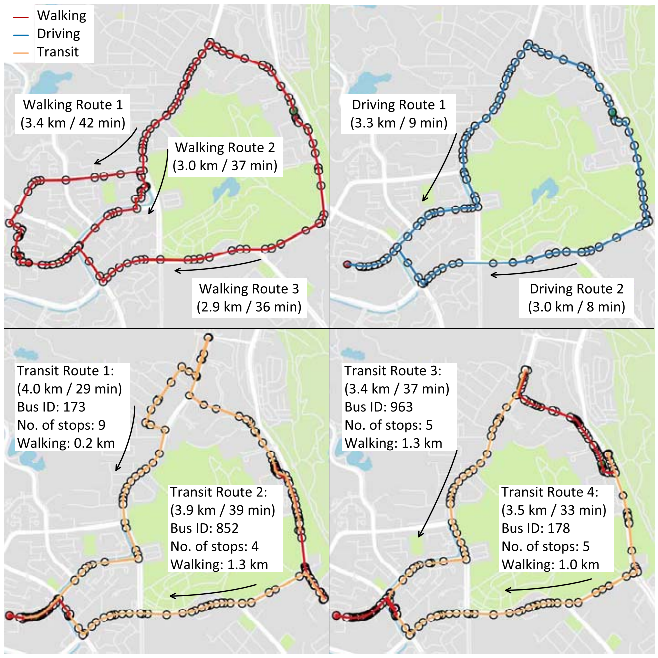

Walking
Driving
Transit
Walking Route 1 (3.4 km / 42 min)
Walking Route 2 (3.0 km / 37 min)
Walking Route 3 (2.9 km / 36 min)
Driving Route 1 (3.3 km / 9 min)
Driving Route 2 (3.0 km / 8 min)
Transit Route 1: (4.0 km / 29 min) Bus ID: 173 No. of stops: 9 Walking: 0.2 km
Transit Route 2: (3.9 km / 39 min) Bus ID: 852 No. of stops: 4 Walking: 1.3 km
Transit Route 3: (3.4 km / 37 min) Bus ID: 963 No. of stops: 5 Walking: 1.3 km
Transit Route 4: (3.5 km / 33 min) Bus ID: 178 No. of stops: 5 Walking: 1.0 km

Fig. 5. Routes suggested by Google Directions Service.

with the shortest distance, minimum duration, the shortest walking distance, or few transit transfers. As shown in Table I, none of these heuristics can match the real routes traversed by students with a reasonable accuracy.

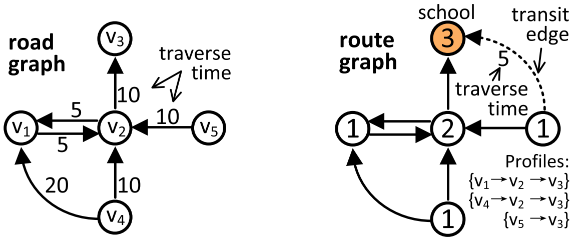
Therefore, for each NSE trajectory, we find its most similar path from the Google set by a hierarchical rule-based classifier with following features: 1) Path shape. We implement a fast approximation Dynamic Time Warping (DTW) algorithm [29] to measure the similarity between two paths. 2) Time duration. It is for situations when Google paths with different travel modes have similar shapes. 3) Distance. For short trips, the driving and walking Google paths tend to have similar shape and commute time. In that case, the walking path is of higher probability.

In this way, we infer the precise paths that are well segmented by transportation modes and thus extract all potential pickup locations for each student.

B. Graph Construction

We describe three key properties of effective last-mile bus design and how previous works fail to address them.

- **The capability of structuring multiple potential pickup locations for each student.** Previous works oversimplify students' demands and model each student as a single point of VRP. When each student has multiple pickup locations, such structure cannot select the best pickup location for each student without calculating best routes for all possible combinations (500 students each having 20 potential pickup points will result in 20^{500} combinations). Worse, changes in even one student's demands require recalculating all those combinations. Thus, a good data structure should be able to simultaneously represent all potential pickup locations, aggregate similar ones and adapt to small changes.
- **The awareness of road networks.** Previous works employ simple proximity models such as Euclidean-based or grid-based model. Those models incur inaccurate distance estimations as geographically proximate locations could be far by walk due to road constraints(e.g. highways, one-way streets). The awareness of road networks imposes those constraints on shuttle bus design.
- **The awareness of existing transits.** Previous works aim at standalone services. The ignorance of existing public transits results in high-cost or replicated services. The awareness of existing transits ensures practical shuttle bus design.



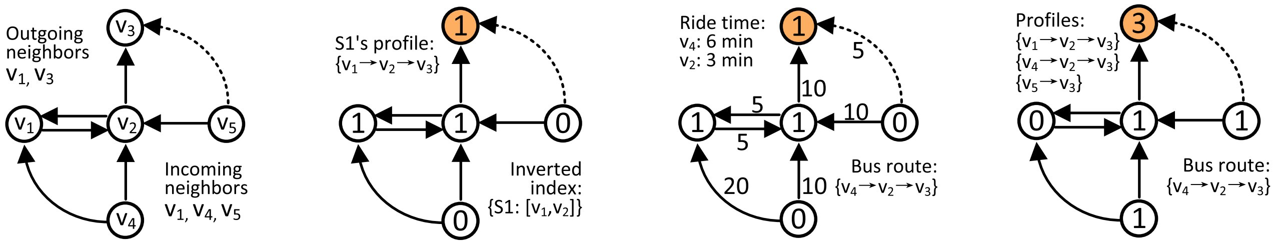
(a) Build the road graph (b) Annotate profiles

Fig. 6. The construction of the route graph.

To address above problems, we proposed a new graph-based data structure named the route graph, which can be built via the following three stages.

Stage 1 (Generate a Road Graph From the Road Networks): As shown in Figure 6a, a road graph $G_{road} = (V, E)$ is a directed graph built from road networks, where a vertex set V represents all road segments and an edge set E denotes the linkage and physical direction between road segments. We first extract information about each road segment (i.e., locations of origin and destination, name, length, category, accessibility for vehicles) from OpenStreetMap [24], which is later used to derive the linkage and walking distance between adjacent road segments. We store these information as the attributes of Structure Vertex and Edge (summarized in Table II). For each road segment $v_i \in V$, attribute "student_set" stores students that can get on our bus at v_i and is initialized as an empty list. For each edge $e_j \in E$, we set the edge type as "road" indicating physical connectivity in road networks and the "traverse_time" is estimated by the walking time from one road segment to the other. In this way, the data structure is aware of road networks.

Stage 2 (Build a Route Graph With Graph Annotation): A route graph $G_{route} = (V, E, D)$ is built by annotating a road graph G_{road} with students' demands, where three types of information are embedded - the school vertex, potential pickup locations and the usage of public transits. The school vertex is learnt from all input profiles and annotated by setting the vertex's attribute *type* to "School". Potential pickup locations of each student are sequentially annotated to corresponding vertices. For each pickup location, the student ID will be added to the attribute "student_set" of the corresponding vertex.



(a) Neighbors of v_2 (b) Insert the profile of S1 (c) Estimate the ride time (d) Allocate a graph

Fig. 7. Examples of supported operators in proposed data structure G_{route}.

TABLE II

SKELETON OF THE ROUTE GRAPH

Structure	Attributes	Description
Vertex	id type student_set shortest_time	Vertex ID Either ”road” or ”school” Associated students Shortest time to the school
Edge	id type from_v to_v traverse_time	Edge ID Either ”road” or ”transit” Origin vertex ID Destination vertex ID Time needed to traverse through
Data	inverted_index tabu_list	e.g. {Student 1: [v_1, v_2, v_3]} tabu vertices

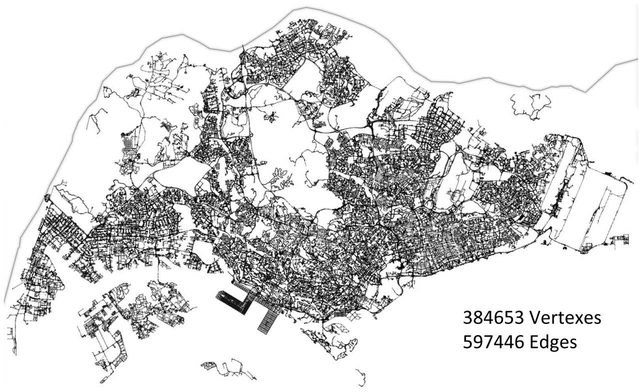


Fig. 8. Visualization of the real route graph.

During this process, if consecutive pickup locations are not connected by a direct edge, the usage of existing transits is identified (as shown in Figure 6b). A new edge will be added to G_{road}, where the *type* is “transit” and *traverse_time* is estimated from the duration between corresponding pickup locations. To facilitate an efficient search of a student on G_{route}, we build an inverted index that maps students to associated vertices. The inverted index is a dictionary organized by student ID and stored in class Data as shown in Table II. Figure 6b shows the annotation result of three student profiles.

Stage 3 (Estimate the Time to the School Vertex): After graph annotation, G_{route} is now aware of both the road networks and the existing transits. We thus estimate the shortest time from each vertex to the school vertex by Dijkstra algorithm, which is then stored in vertex attribute *shortest_time.*

Figure 8 visualizes the route graph we built for the entire city of Singapore, which consists of 384653 vertices and 597446 edges. The route graph sketches the contours of roads in Singapore.

The proposed G_{route} supports the following operators.

Neighbor() returns one-hop neighbors of a given vertex, where the users can specify to return incoming neighbors, outgoing neighbors, or both. This operator has $\mathcal{O}(1)$ complexity as we maintain adjacency lists for each vertex. Figure 7a shows both incoming and outgoing neighbors of vertex v_2.

Insert() inserts a student’s ID into a specific vertex’s “student_set” (duplicates can automatically be handled by set) and updates the inverted index accordingly. Insert() can be easily extended to process demands. Figure 7b shows the example of inserting S1’s demands by sequentially calling Insert() on its potential pickup locations. In the meantime, a new record indexed by S1 is added to the inverted index.

Remove() incurs a reversed process and is typically used for ensuring one student can only be picked up at one vertex. Same as Insert(), Remove() works directly on vertices and thus is of $\mathcal{O}(1)$ complexity.

Locate() takes a student’s ID as input and returns all the associated vertices by checking the inverted index. This operator is of $\mathcal{O}(1)$ complexity as we maintain an inverted index in the route graph.

Estimate() takes a shuttle route as input and estimates the ride time to school for stops in the bus route. The ride time is estimated by a shortest path that connects all bus stops and thus is of $\mathcal{O}(V^2)$ complexity. This operator is efficient as the number of related vertices between two stops is small. Figure 7c shows an example output of Estimate().

Allocate() estimates the best vertex for each student to get on the bus given a certain bus route. We make two reasonable assumptions, i.e., students will be willing to get on the shuttle as long as the total commute time of themselves can be reduced and they will always pick the bus stop that results in highest commute time reduction. After calling Allocate(), vertices in shuttle routes will sequentially check their “student_set”. For each student in a “student_set”, all traversed vertices will be retrieved by calling Locate(). The best vertex is the bus stop that results in the lowest commute time considering all the traversed vertices and the ride time. After that, other potential pickup locations of the student will be deleted from corresponding vertices by calling Remove(). This operator is of $\mathcal{O}(nm)$ complexity, where n is the number of bus stops and m is the size of "student_set". The operator is efficient as m and n are usually small compared with the number of vertices. Figure 7d gives an allocation on the route graph of Figure 6b: the student that origins from v_1 chooses to get on the bus at v_2; the student origins from v_4 will get on the bus at v_4; and the student that origins from v_5 sticks to his/her original path.

Score() calculates the beneficial score of an allocation. In our route planning, we aim at routes that strike a proper balance between route gains and operating costs. Therefore, we model these two objectives (i.e., saving more time for students vs. having shorter ride time) into the beneficial score ϕ. The basic mechanism behind is penalizing route gains (i.e., $time\_saving$) with operating costs (i.e., $ride\_time$). But instead of a reciprocal, we use an exponential function of $ride\_time$ to offer decision makers the flexibility to adjust their preference between the two objectives.

$$\phi(routes) = time\_saving * \alpha^{ride\_time}, \quad 0 < \alpha \leq 1 \qquad (1)$$

where α is a user-defined tuning parameter to set the preference on the gain and cost according to a specific application's sensitivity towards the operating cost. A larger α indicates that more preferences are given to the reduction of students' commute time (e.g., $\alpha = 1$ means that we do not care about the system cost), while a smaller α puts emphasis on the system cost. Given an allocation, the overall time saving can be easily calculated via the number of students who remain in bus route vertices and the bus ride time can be estimated by Estimate(). Thus, this operator is of $\mathcal{O}(n)$ complexity, where n is the number of bus stops. For the allocation depicted in Figure 7d, Score() returns $\phi(\{v_4, v_2\}) = (1*(20-6)+1*(10-3))*\alpha^6$.

A practical Issue in Graph Annotation: The Vertex Ambiguity: During the graph annotation, we aim to find a best matched vertex v for each potential pickup location p in trip profiles. The ambiguity stems from two facts: (1) pickup locations are far less in granularity compared with vertices of the road graph. (2) pickup locations returned by Google Directions Service [16] are not always close to its corresponding vertices of the road graph generated by OpenStreetMap [24].

In our study, we observed that location proximity between v and p is not enough and sometimes misleading, especially when it comes to vertex-dense regions such as intersections, junctions, and overpasses. The key observation here is that Google Directions Service [16] usually reports turning points as they define the shapes of trajectories. Such a characteristic makes the graph distance of the correct vertex sequence the shortest. Thus, we propose finding the best match of each p via anchor vertices that result the shortest path among all candidates. For each pickup location, we first extract vertices within a certain distance as its candidate set. Then the anchor vertices are selected by calculating the shortest path among the combination of candidates from different sets, where a Viterbi algorithm is applied to speed up the calculation. Finally, we complete the path between consecutive anchor vertices with corresponding shortest path and the best match of each p is the nearest vertex along the completed path.

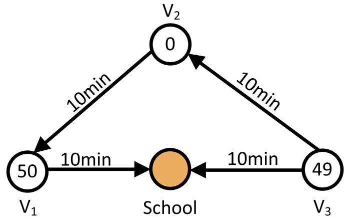


Fig. 9. The shortsightedness of greedy expansion.

C. Graph-Based Route Planning

With the proposed graph structure, we are now able to efficiently evaluate a specific bus route. However, given that enormous possible bus routes exist in city-wide route graph, brute force searching is still infeasible.

In this subsection, we first define the graph-based route planning problem and prove its NP-hardness. Then we propose a computationally feasible algorithm to ensure that we find a good route plan in limited time.

1) Problem Definition: Given the route graph $G_{route} = (V, E, D)$ with $|V| = v$, expected number of routes K, a ride time distance b, we want to find a sequence of vertices $V' \subseteq V$, which maximizes the total beneficial score ϕ and fulfills two constraints: 1) V' forms exact K routes, 2) the ride distance of each route is no more than budget b. Mathematically, we formulate this problem as an integer programming problem, where we use binary variables x_{ij} equal to 1 if and only if students in vertex j is served by service route i.

$$\begin{aligned}
\underset{N'}{\text{maximize}}\ & \phi(N') = (\sum_{i=1}^{K}\sum_{j=1}^{V} g_{ij}x_{ij}) * \alpha^{C} \\
\text{subject to}\ & \sum_{j=1}^{V} c_j x_{ij} \leq b \quad (i = 1, \ldots, K) \\
& \sum_{i=1}^{K} x_{ij} \leq 1 \quad (j = 1, \ldots, V) \\
& x_{ij} \in \{0, 1\} \quad (i = 1, \ldots, K,\ j = 1, \ldots, V)
\end{aligned}$$

where C stands for the total ride time of all routes, which equals $\sum_{i=1}^{K}(\sum_{j=1}^{V} c_j x_{ij})$, g_{ij} and c_{ij} denotes the saved time and the ride time respectively. The first constraint bounds the ride time of each route under budget b; the second constraint ensures that the students in one road segment are served by only one bus.

2) NP-Hardness: Finding K budget constrained last-mile bus routes with maximal beneficial score is NP-hard. We detail our prove in Appendix.

3) Greedy Expansion Algorithm: We propose a greedy expansion algorithm to deal with the NP-hardness and plan the bus routes on the graph. It is based on an important mobility pattern of students, i.e., they eventually converge to the school from their homes scattered in the city. We initialize a solution

Algorithm 1 Tabu-Based Expansion

```
Input: $G_{route}$, $k$
Output: $route$*
route ← Initialization(k)
while not Termination do
    neighbors ← ∅
    For $r \in$ G_route.neighbor(route) do
        if r not in G_route.tabu_list then
            neighbors ← r
        end
    End
    candidates ← Evaluate(neighbors)
    if candidates.best.score > route.score then
        route ← candidates.best
    end
end
Return $route$
Function Evaluate(neighbors)
    candidates ← ∅
    For $r \in$ neighbors do
        allocation ← G_route.allocate(r)
        c.score ← G_route.score(allocation)
        c.route ← r
        if new_score ≪ old_score then
            G_route.tabu_list ← r
            continue
        end
        candidates ← c
    End
    return candidates
```

at the school and progressively improve the solution by adding neighbor vertices as bus stops. In each iteration, the algorithm always chooses the neighbor vertex with the largest improvement on the beneficial score. If adding the neighbor vertices offers no improvement to current solution, the search terminates. This heuristic is computationally efficient, but it often results in a local optimum due to its greedy nature.

Figure 9 illustrates an example of this problem. For simplicity, we set the ride time of each edge equals 2 minutes, and tuning parameter $\alpha = 0.95$ in Equation 1. The search heuristic starts at the school and chooses v_1 as $\phi(\{v_1\}) = (50*8)*0.95^2 = 361$ is larger than $\phi(\{v_3\}) = (49*8)*0.95^2 = 354$. When the greedy search continues to expand from v_1, it terminates because expanding to its only neighbor v_2 incurs a decrease in the beneficial score ($\phi(\{v_2, v_1\}) = (50 * 8) * 0.95^4 = 319$). Such shortsightedness stops the algorithm from searching further and finding the global optimal solution by including V3 ($\phi(\{v_3, v_2, v_1\}) = (50*8+49*4)*0.95^6 = 438$).

4) Tabu-Based Expansion Algorithm: We propose a tabu-based expansion algorithm that integrates the idea of tabu search [14] to explore the solution space beyond local optimality. Generally, tabu search starts with an initial route and searches for the best solution in a defined neighborhood of current solution. It then updates current solution by the solutions in the neighborhood and repeats the process until a certain termination condition is satisfied. It allows non-improving moves, but maintains a tabu list of forbidden moves to prevent cycling back to solutions that have already been visited. To leverage tabu search to perform route planning on our graph, we develop a tabu-based expansion algorithm that includes application-specific design, including the tabu list, neighbors and termination conditions. As depicted in Algorithm 1, our algorithm has the following stages:

Initialization. The algorithm initializes a subgraph, k initial routes and an empty Tabu list. The subgraph is created with vertices that have a smaller "shortest_time" in respect to budget b. This reduces search space and ensures that all the planned routes within budget b. The initial routes start from the school.

Tabu Iteration. In each iteration, the algorithm first calculates the neighbor set of current solution. The neighbor set contains the routes that can be reached from the current route by adding an adjacent non-tabu vertex into current route or removing a non-tabu vertex from current route. If adding or removing a vertex results in a significant drop in the beneficial score, we regard this vertex as a tabu vertex and append it to the "tabu_list" of the G_{route}. Tabu vertices are not allowed to be included in the neighbor set. The aforementioned shortsightedness is overcame by allowing non-improving searches until all the neighbors are tabu vertices. In each iteration, we update the route and score when the best route formed by neighbors has a larger score.

Termination. The algorithm terminates when either of the following two criteria is met, i.e., the maximum iteration number or no performance improvement in the last 5 iterations. The best solution identified in the whole search process is returned as the final solution.

IV. Evaluation

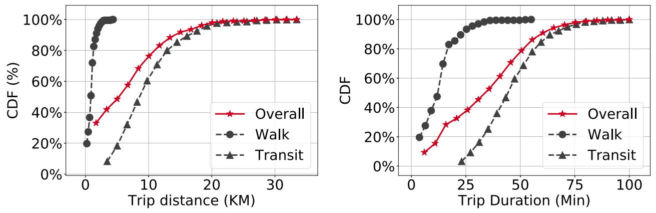
In this section, we conduct extensive experiments to evaluate our system using real-world data.

A. Experiment Settings

We use the NSE data crowdsensed from 2809 students of 7 schools within a semester (i.e., from 11/04/2017 to 31/07/2017). During this four-month experiment period, one SENSg device is assigned to each student to continuously record the student's daily trajectories for one week. A total of 11236 trajectories that traversed ∼ 80k kilometers are extracted from the collected data. For each student, we use four-day data to perform our last-mile bus planning and use the remaining one-day data to evaluate the proposed system. We extract road networks of Singapore from OpenStreetMap [24], which contains 384653 road segments and 597446 edges.

We compare our results with the following methods.

- **Door-to-Door shuttles**. It offers minimum-distance routes by solving a capacitated VRP [18] formed by students' home locations.
- **Feeder shuttles** [38]. It first performs a KMeans clustering over students' home locations with the K having the largest Bayesian information criterion, and then



(a) Trip distance distribution (b) Trip duration distribution

Fig. 10. NSE mobility statistics.

TABLE III

SUMMARY OF STUDENT COMMUTE PATTERNS

	School A	School B	School C	School D	Overall
# Students - Total	448	418	472	292	2809
# Students - Walk	358	107	60	84	1052
# Students - Transit	90	311	412	289	1757
Avg distance (KM)	2.8	6.9	9.1	7.0	6.9
Avg duration (Min)	25	35	44	36	33
Avg # pickups	**20**	**30**	**46**	**30**	**32**

offers minimum-distance routes by solving a capacitated VRP [18] formed by the centroids of obtained clusters.

- **Metro shuttles**. It offers routes that directly link the largest transportation hub and the school.

We empirically set system parameters as follows: the parameter in objective function $\alpha = 0.995$; the bus capacity is 100; the number of available buses is the minimum fleet size required in terms of the total number of students. Each route aims to serve a certain set of students with on waiting time and is traversed once by a bus. As suggested by Figure 10a, the overall trip distance of NSE students distribute almost evenly from a few kilometers to 20 kilometers. Since it will be hard and unfair to determine a service cut-off regarding trip distance, we adopt the original setting of serving all the students for Door-to-Door shuttles and Feeder shuttles.

All the algorithms are implemented and experimented with a HP Z440 workstation with 12 3.5GHz Intel Xeon CPU cores and 32GB memory.

B. Mobility Statistics of NSE Trajectories

In NSE data, after omitting students who traveled to schools by private cars, we have 37% of students who walked to school and 63% of students who commuted via public transits. We summarize their mobility statistics.

1) Trip Distance Distribution: Figure 10a summarizes cumulative distribution function of the trip distance. From the figure, it is clear that the majority of transit trips have a longer trip distance compared with walking trips, i.e., 90% of transit trips are shorter than 16.7 KM while 90% walking trips are under 1.8 KM. The overall average trip length is 6.9 KM.

2) Trip Duration Distribution: Figure 10b illustrates cumulative distribution function of the trip duration, where 90% of transit trips take less than 63 minutes while 90% of the walk trips are within 23 minutes. This is because, upon a longer commute time, students are more prone to be driven to school by their parents. The overall average trip duration is 33 minutes.

3) Trip General Patterns: We list three general patterns we found in NSE data: 1) Most of the students (85%) arrive at school during 7:00 a.m. to 7:45 a.m.. This proximity reveals the chance of finding last-mile bus routes that can benefit the majority of students and remain cost-efficient; 2) Most of the students' homes are near their schools, while a certain number of students live far from their schools. The proportion of students who live far away varies from school to school; 3) Students rarely change their choices of transportation in different days, which makes it plausible to use historical data for route planning.

C. System Performance

Table III summarizes detailed information on the schools used in our evaluation. Due to the space limit, we list 4 schools as well as an overall summary of all schools. The other schools have similar statistics. For each school, we list information including the total number of students, the number of students who walk to school, the number of students who take public transits, the average distance and duration of all trips, and the average number of pickup locations. As shown in Table III, the percentage of students who take public transits differs among schools due to the difference in public transits accessibility. Besides, some schools have students living relatively far away than other schools, resulting in longer trip distance and duration.

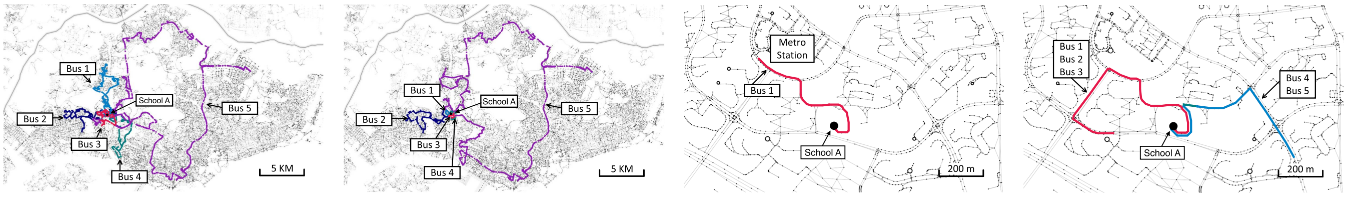
Table IV summarizes the performance of four methods in terms of fours metrics: the average reduction on students' commute time (denoted as Δt), the total ride distance of bus services (denoted as d), the percentage of students that got served (denoted as %) and the beneficial score (denoted as ϕ and defined in III-B). Figure 11 visualizes the shuttle routes planned for School A.

In Table IV and Figure 11, we can see:

- Proposed system produces routes that outperform Feeder, Metro shuttles by 6.6× and 2.9× respectively in terms of overall beneficial score. This is due to its awareness of both students' real demands and existing public transits. By understanding how the students' transport demands are addressed by current public transits and locating the stops that can help them most in a global view, proposed system yields reasonable commute time reductions with relatively low ride distances.
- Door-to-door routes have the largest ride distances in all schools due to the total ignorance of existing public transits. The long ride incurs unnecessary long ride time for students who get on the bus on the early stage, which further lowers the reduction on all students' commute time. For Feeder, although the total ride distance is reduced by performing a clustering beforehand, the students, on average, need to travel ~0.92 extra kilometers and 11.6 minutes to nearest service station due to its unawareness of road networks and existing transits. Consequently, the reduction to students' commute time is also limited.

TABLE IV
SUMMARY OF OVERALL PERFORMANCE

	School A				School B				School C				School D				Overall			
	Δt	d	%	ϕ	Δt	d	%	ϕ	Δt	d	%	ϕ	Δt	d	%	ϕ	Δt	d	%	ϕ
Door-to-door	-7.7	278.7	100%	-	-15.9	529.8	100%	-	-16.3	727.4	100%	-	-35.5	431.0	100%	-	**-12.0**	**479.2**	100%	-
Feeder	2.7	157.0	100%	1.2	1.2	375.7	100%	0.2	3.7	198.2	100%	1.4	3.6	187.7	100%	1.4	**2.9**	**209.5**	100%	1.0
Metro	1.3	0.8	21%	1.3	1.5	0.4	42%	1.5	2.6	1.5	17%	2.5	2.2	1.0	26%	2.2	**2.4**	**1.2**	24%	2.3
Our method	5.1	4.5	96%	5.0	4.0	5.0	96%	3.9	9.4	7.7	73%	9.0	7.1	3.5	85%	7.0	**6.8**	**6.1**	87%	6.6



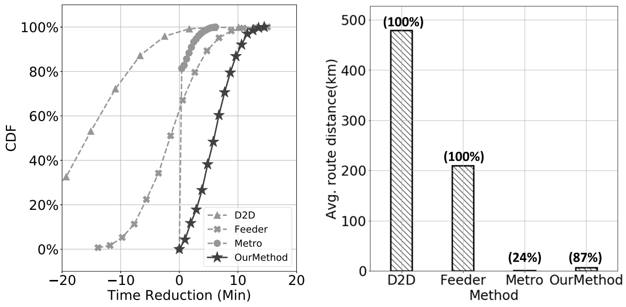
(a) Door-to-Door shuttle routes
map scale = 1:1000000
total bus routes = 278.7 km

(b) Feeder shuttle routes
map scale = 1:1000000
total bus routes = 157.0 km

(c) Metro shuttle routes
map scale = 1:25000000
total bus distances: 0.8 km

(d) Routes by our approach
map scale = 1:25000000
total bus distances: 4.5 km

Fig. 11. Visualization of last-mile bus routes planned for School A.



(a) Trip duration reduction

(b) Average routes distance

Fig. 12. Overall performance.

- Although fetching students from the largest transportation station nearby possesses a small ride distance, it can only serve a limited amount of students due to the fact that there are not much students' commute rely on one specific transportation station. The lack of understanding on students' demands limits the percentages of students who can benefit from metro shuttles (24% on average) and results in small reductions on students' commute time.
- Overall, the proposed system can benefit the majority of students (i.e., 87%) and offer a reasonable commute time reduction (i.e., 6.8 minutes).

Figure 12 shows the comparison on the overall gains and costs of routes suggested by different methods. Figure 12a illustrates the CDF of trip duration reduction of all students, where the right half of the figure summarizes the percentage of students who benefit from the system. As shown, most students receive no benefit from Door-to-Door (D2D) and Feeder shuttles due to the long ride time. Although the CDF of Metro shuttles locates at the right half of the figure, both the population of served students and the time reduction for them remain low. This is because Metro shuttles usually ride short distances. In contrast, routes suggested by our method offer a reasonable time reduction to most students. Figure 12b shows on average routes suggested by our method serve 87% of students with short route distances.

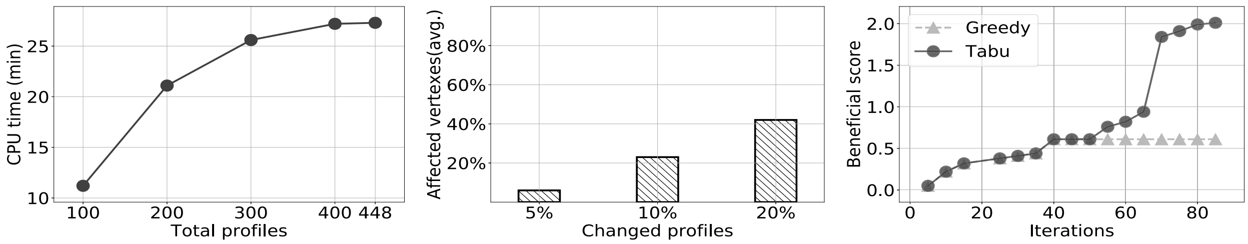
D. System Components

We further study the performance of the four key components in our proposed system separately.

1) Trip Identification: It needs to identify homes and schools by classifying trip points and stay points. We manually labeled 77797 points from 20 randomly selected students, which form 68 valid trips with 1598 trip points. We compare proposed time-weighted density clustering with both threshold and density based counterparts (introduced in Section III-A) in terms of precision and recall. We tune parameters of above algorithms based on grid searches over sets of empirical values and set a fine distance threshold as 50 meters and duration threshold as 1 hour in threshold-based algorithm, the minimum distance as 200 meters and the minimum number of points as 10 in DBSCAN, and the minimum distance as 200 meters and the minimum duration as 1 hour in proposed method.

Table V shows that proposed method yields a significant improvement on precision for identifying trip points. It is because proposed method can 1) adapt to inconsistent localization accuracy by leveraging density-domain information and 2) tolerate sudden drifts by exploring time-domain information. For identifying home and school locations, our method outperforms DBSCAN on recall because the latter cannot identify home/school clusters with insufficient sample points when a SENSg device enters its sleep mode.

2) Trip Profiling: We evaluate the proposed Google-based algorithm by the accuracy of sample-wise travel mode detection and the performance of pickup location extrac-



(a) Computational efficiency (b) Scalability on profile changes (c) Effectiveness of Tabu expansion

Fig. 13. Evaluation on the graph structure and Tabu-based expansion algorithm.

TABLE V
COMPARISON OF TRIP IDENTIFICATION ALGORITHMS

	Trip points		Home & School	
	Precision	Recall	Precision	Recall
$Threshold$	4.8%	92.4%	19.3%	100%
$DBSCAN$	72.4%	97.0%	90.1%	88.2%
$OurMethod$	90.5%	97.7%	98.5%	100%

TABLE VI
THE ACCURACY OF TRAVEL MODE DETECTION

	Walk	Transit	Drive	Overall
$DT + MM$	54%	40%	36%	44%
$OurMethod$	91%	93%	89%	91%

TABLE VII
THE PERFORMANCE OF PICKUP LOCATION EXTRACTION

	Precision	Recall	F1 score
$DT + MM$	41%	63%	65%
$OurMethod$	91%	95%	96%

tion. We compare proposed method with the state-of-the-art algorithms leveraging decision tree and map matching (DT + MM), where mobility features including sample-wise distance, duration, speed and acceleration are calculated from consecutive samples. We manually labeled 11085 walking, 10075 transit and 7735 driving samples, where 70% of samples are randomly selected for training the DT classifier while the remaining samples are used for testing.

The results in Table VI reveal that proposed method yields an overall 91% accuracy in detecting sample-wise travel modes, while decision tree classifier achieve only 44% accuracy for all travel modes. Since the decision tree classifier relies heavily on the consistency of speed and acceleration, it suffers from large localization errors and sparsity of NSE data. As a consequence of inferior performance in travel mode detection, the DT + MM algorithm extract pickup locations with only 41% precision. Table VII shows that the proposed method has a dominated performance in terms of the precision, recall and F1 score on extracting pickup locations.

3) Benefits of the Route Graph: We show the benefits of proposed graph data structure by comparing it with TSP-based route planning in terms of two aspects: computational efficiency and scalability on profile changes. We conduct a trace-driven simulation based on trips of 448 students from School A.

Figure 13a shows that graph-based route planning scales well in terms of CPU time as the number of profiles increases. We omit the CPU time of TSP-based route planning as it requires solving 20^{100} TSP instances for 100 profiles and this overhead increases exponentially as the number of profiles increases.

Figure 13b depicts that proposed graph data structure is highly scalable towards profile changes. The reason is that profile changes are reflected in the graph structure by modifying the attributes of related vertices, which is less cumbersome than updating distances with all the other points in TSP-based method.

4) Tabu Expansion: We demonstrate the effectiveness of proposed tabu expansion algorithm with a trace-driven simulation based on 448 students from School A. As shown in Figure 13c, the greedy algorithm stops searching too early due to its shortsightedness. Proposed tabu-based expansion algorithm exploits a bigger search space by allowing non-improving moves and finds a route that has 3.3× higher beneficial score than the greedy route.

V. RELATED WORK

A. Bus Stop Selection (BSS)

Given the road network consisting of home, school, bus depot and the origin-destination (OD) matrix, BSS seeks to select a set of bus stops and assign students to those stops. According to two comprehensive survey studies [10], [26], many works assume that the potential locations of bus stops are given. With that, BSS is then formulated as an assignment problem to minimize the number of bus stops or the total student walking distance. Only a few works solve BSS in conjunction with route planning by heuristics, which can be classified into the following three strategies: the location-allocation-routing strategy [12], the allocation-routing-location strategy [4] and the location-routing-allocation strategy [31]. All above studies did not take into consideration of the existing public transits and assumed that the best pickup locations for students are within walking distance from their homes. This is often not the case in practice given the multiple choices of public transportation and diversity among students' home-school trips. Our work learns potential pickup locations for individual students from their daily trips with public transits, and considers all possible pickup locations with proposed graph-based data structure to ensure a low cost of suggested routes.

1) Bus Route Generation (BRG): Given the selected bus stops and the number of students assigned to them, BRG searches for the optimal routes and is very similar to the vehicle routing problem (VRP) [18]. Due to the problems' NP-hardness, only relatively small instances can be precisely solved via optimization algorithms (e.g., dynamic programming, branch-and-bound). Therefore, in practice, classical heuristics that combine a construction heuristic (e.g., the savings algorithm [7], the sweep algorithm [13], and the Fisher and Jaikumar algorithm [13]) and an improvement method (e.g.,λ -opt) are often used to obtain a feasible solution. More recently, a significant research effort has been dedicated into metaheuristics such as genetic algorithm [1], simulated annealing [25] and Tabu search [8], which are capable of consistently producing high quality solutions at the expense of speed and simplicity. However, those methods cannot handle the problem considered in our paper where student have multiple potential pickup locations. Directly applying those methods requires constructing and solving a number of VPR instances and results in infeasible computation cost. We thus propose a graph-based data structure to reduce search space and develop a customized Tabu search algorithm upon the graph to construct proper routes.

2) Common Practices in School Bus Planning: Traditionally, planners have to design school bus routes based on costly surveys and their own expertise [17]. There are two commercial practices regarding school bus planning: door-to-door shuttles and transportation hub expresses. Both methods first require that each student provides a best pickup location (i.e. usually his/her home or a major transportation hub traversed). The bus planning problem is then formulated into an optimization problem such as TSP variants (e.g. mTSP [3]) and VRP variants (e.g. CVPR [18], DARP [8]). Different from above approaches, we learn the best pickup locations for individual students instead of assuming a homogeneous pickup strategy (either homes or transportation hubs) is best for all students. Learning from trajectories can also reveal the system-wise optima, while surveying students can only obtain knowledge of individual-wise optima.

3) Data-Driven Bus Route Planning: Today, there are several recent projects that leverage information from different data sources to facilitate bus route planning. In [27], [38], researchers learn the metro passengers' final destinations from cellphone data. The bus routes are determined by solving TSP-like optimization problems that are formulated by cluster centers of the learned destinations. In [5], [6], taxi records between two regions are used as an indicator of poor public transit coverage and bus routes are generated to bridge those regions. In [28], similar origin and destination locations of commuters are learnt by a clustering over smart card data, where bus routes are designed to link those locations. In [19], researchers build a transportation mode choice model from both taxi records and bus transactions. With this model, region pairs with low probability of passengers taking buses are identified, where new bus routes are designed by maximizing the expected utilization of public bus service. For those works, they implicitly assume that each passenger has only one origin and destination pair. They cannot be used in our scenarios, because a student usually has multiple potential pickup locations.

4) Understanding Transportation Choices From Trajectories: The closest works to us are travel mode detection algorithms. Most existing methods share a general principle. First, a classification model (e.g. Decision Trees, Random Forest, Bayesian Network, Support Vector Machine and neural network) [37] is trained from features of historical trajectories. Then, by feeding mobility features of new traces into trained model, sample-wised travel modes can be determined. Most approaches regard mobility patterns like distance, speed and IMU readings as features, while some recent works involve new feature (e.g. barometer readings [9], [30]) or new information (e.g. GIS information [35]) to improve accuracy. These approaches assume that mobility features possess a consistent error level across samples. However, this assumption is not satisfied in NSE data due to its sparsity and noisiness. Thus, directly applying those methods leads to erroneous results.

VI. Conclusion

In this work, we have investigated the problem of last-mile school shuttle planning by mining massive crowdsensed daily trajectories. The proposed system automatically learns potential pickup locations for each individual, efficiently hosts possible demand combinations by a graph data structure and heuristically searches for system-wise optimal routes. Our experiments show that proposed system is able to plan routes that are more beneficial and low-cost than existing solutions.

Appendix
Proof of the NP-Hardness

Proof: We derive our problem by reducing a generalized assignment problem. We can view each road segment $v_j \in V$ being assigned to a route r_i as a task being assigned to an agent. Each assigned task has a profit (i.e. students time saved) and a cost (i.e. the route time), while each agent has a budget (i.e. the route budget). The final decision set V' is viewed as the task-agent assignment. In this way, for any instance of the decision version of the generalized assignment problem, we can find an instance of the decision version of the problem of finding K budget constrained last-mile bus routes with maximum beneficial score by setting budget unbound, and their answers are the same. Thus, the generalized assignment problem is reducible to our problem, which completes the proof of NP-hardness. □

References

[1] B. M. Baker and M. A. Ayechew, "A genetic algorithm for the vehicle routing problem," *Comput. Oper. Res.*, vol. 30, no. 5, pp. 787–800, Apr. 2003.

[2] B. Balcik, B. M. Beamon, and K. Smilowitz, "Last mile distribution in Humanitarian relief," *J. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 51–63, 2008.

[3] T. Bektas, "The multiple traveling salesman problem: An overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, Jun. 2006.

[4] R. Bowerman, B. Hall, and P. Calamai, "A multi-objective optimization approach to urban school bus routing: Formulation and solution method," *Transp. Res. A, Policy Pract.*, vol. 29, no. 2, pp. 107–123, 1995.

[5] C. Chen, D. Zhang, N. Li, and Z.-H. Zhou, "B-Planner: Planning bidirectional night bus routes using large-scale taxi GPS traces," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1451–1465, Aug. 2014.

[6] S. P. Chuah, H. Wu, Y. Lu, L. Yu, and S. Bressan, "Bus routes design and optimization via taxi data analytics," in *Proc. ACM CIKM*, 2016, pp. 2417–2420.

[7] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, 1964.

[8] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: Models and algorithms," *Ann. Oper. Res.*, vol. 153, pp. 29–46, Sep. 2007.

[9] W. Du, P. Ton, and M. Li, "UniLoc: A unified mobile localization framework exploiting scheme diversity," in *Proc. IEEE ICDCS*, Jul. 2018, pp. 818–829.

[10] W. A. Ellegood, S. Solomon, J. North, and J. F. Campbell, "School bus routing problem: Contemporary trends and research directions," *Omega*, to be published.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM SIGKDD*, vol. 96, 1996, pp. 226–231.

[12] M. Galdi and P. Thebpanya, "Optimizing school bus stop placement in Howard County, Maryland: A GIS-based heuristic approach," in *Geospatial Research: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2016, pp. 1660–1676.

[13] B. E. Gillett and L. Miller, "A heuristic algorithm for the vehicle-dispatch problem," *Oper. Res.*, vol. 22, no. 2, pp. 340–349, 1974.

[14] F. Glover, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.

[15] B. L. Golden, S. Raghavan, and E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43. Springer Science & Business Media, 2008.

[16] Google. *Directions API*. Accessed: Jan. 16, 2018. [Online]. Available: http://Developers.google.com/maps/documentation/directions

[17] V. Guihaire and J.-K. Hao, "Transit network design and scheduling: A global review," *Transp. Res. A, Policy Pract.*, vol. 42, no. 10, pp. 1251–1273, 2008.

[18] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, no. 3, pp. 345–358, 1992.

[19] Y. Liu *et al.*, "Intelligent bus routing with heterogeneous human mobility patterns," *Knowl. Inf. Syst.*, vol. 50, no. 2, pp. 383–415, 2017.

[20] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proc. ACM SIGSPATIAL*, 2009, pp. 352–361.

[21] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. ACM SIGSPATIAL*, 2009, pp. 336–343.

[22] *NSE Project*. Accessed: Jul. 31, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Last_mile_(transportation)

[23] *NSE Project*. Accessed: Jul. 31, 2017. [Online]. Available: https://www.nse.sg

[24] *OSM*. Accessed: Jan. 23, 2018. [Online]. Available: https://www.openstreetmap.org

[25] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Ann. Oper. Res.*, vol. 41, no. 4, pp. 421–451, 1993.

[26] J. Park and B.-I. Kim, "The school bus routing problem: A review," *Eur. J. Oper. Res.*, vol. 202, no. 2, pp. 311–319, 2010.

[27] F. Pinelli, R. Nair, F. Calabrese, M. Berlingerio, G. D. Lorenzo, and M. L. Sbodio, "Data-driven transit network design from mobile phone trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1724–1733, Jun. 2016.

[28] G. Qiu, R. Song, S. He, W. Xu, and M. Jiang, "Clustering passenger trip data for the potential passenger investigation and line design of customized commuter bus," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3351–3360, Sep. 2019.

[29] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.

[30] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh, "Using mobile phone barometer for low-power transportation context detection," in *Proc. ACM SenSys*, 2014, pp. 191–205.

[31] P. Schittekat, J. Kinable, K. Sörensen, M. Sevaux, F. Spieksma, and J. Springael, "A metaheuristic for the school bus routing problem with bus stop selection," *Eur. J. Oper. Res.*, vol. 229, no. 2, pp. 518–528, 2013.

[32] *SENSg*. Accessed: Jul. 31, 2017. [Online]. Available: https://www.nse.sg/sensg/about-sensg

[33] *Singtel*. Accessed: Jul. 31, 2017. [Online]. Available: https://www.singtel.com

[34] Skyhook Wireless. Accessed: Jul. 31, 2017. [Online]. Available: https://www.skyhookwireless.com

[35] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and GIS information," in *Proc. ACM SIGSPATIAL*, 2011, pp. 54–63.

[36] *Wireless@SG*. Accessed: Jul. 31, 2017. [Online]. Available: https://www.imda.gov.sg/wireless-sg

[37] L. Wu, B. Yang, and P. Jing, "Travel mode detection based on GPS raw data collected by smartphones: A systematic review of the existing methodologies," *Information*, vol. 7, no. 4, p. 67, 2016.

[38] D. Zhang, J. Zhao, F. Zhang, R. Jiang, and T. He, "Feeder: Supporting last-mile transit with extreme-scale urban infrastructure data," in *Proc. ACM IPSN*, 2015, pp. 226–237.

[39] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. ACM WWW*, 2009, pp. 791–800.

Panrong Tong received the B.E. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2015. He is currently pursuing the Ph.D. degree with the Interdisciplinary Graduate School, Nanyang Technological University, Singapore. His research interests include smart city, data mining, distributed computing, and intelligent transportation systems.

Wan Du (S'10–A'11–M'15) received the B.E. and M.S. degrees in electrical engineering from Beihang University, China, in 2005 and 2008, respectively, and the Ph.D. degree in electronics from the University of Lyon (Cole Centrale de Lyon), France, in 2011. He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of California, Merced. His research interests include the Internet of Things, cyber-physical system, distributed networking systems, and mobile systems.

Mo Li (M'06) received the B.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2004, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2009. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include networked and distributed sensing, wireless and mobile, cyber-physical systems, smart city, and urban computing.

Jianqiang Huang is currently the Director of Alibaba DAMO Academy. His research interests focus on the visual intelligence in the city brain project of Alibaba. He received the Second Prize of the National Science and Technology Progress Award in 2010.

Wenqiang Wang received the B.Sc. (Hons.) and the Ph.D. degrees in computer science from the School of Computing, National University of Singapore in 2000 and 2006, respectively. He is currently a Scientist working in the A*STAR Institute of High Performance Computing. His research interests mainly focus on distributed data management and processing systems.

Zheng Qin received the B.Eng. degree in information engineering from Xi'an JiaoTong University in 2001 and the Ph.D. degree in electrical and computer Engineering from the National University of Singapore in 2006. He is currently a Senior Scientist and the Deputy Department Director of computing science at the A*STAR Institute of High Performance Computing. His research interests include cloud computing, large-scale data processing, and urban mobility and logistics.