First-mile ridesharing using autonomous shuttle service and IoT cloud platform

Shyam Sundar Rampalli, Pranjal Vyas, Anuj Abraham and Justin Dauwels

Abstract

A wide range of challenges to the passengers on safety, security, comfort, and reduced waiting times are the main aspects to be considered in first/last-mile services. With the advent of autonomous vehicles (AVs), first/last-mile transits can be more safe, reliable, and sustainable in the transportation sector. Developing a system, that handles these constraints and is equipped with state-of-the-art support, is always a challenging one. Moreover, it is noticed that the passengers are bound to opt for other transit such as the private modes over public modes of transportation.

This chapter proposes a technique to solve first-mile ridesharing problems with AV as a shuttle service and uses an internet-of-things (IoT) platform. First-mile (FM) transit refers to transfer the passengers from their pick-up locations to the nearest public transport hub. The first-mile transit is solved by dividing the problem into two sub-problems. Firstly, the pick-up locations are associated with the neighbouring vehicles available. Here, we propose two techniques based on mixed-integer linear programming (MILP) and K-means clustering, in which the passenger requests and the vehicles' data are collected in an IoT cloud. The passengers are allowed to submit travel requests through a human-machine interface (HMI), such as cell phones or automatic booking kiosks within a period. At the backend, an IoT cloud computes appropriate vehicles to serve the passengers at their respective pick-up locations based on the nearest location. Thus, the pick-up locations are clustered for different vehicles in the service network on the IoT cloud.

Secondly, routing AV to the public-transport hub by minimizing the total distance travelled is performed. Here, we present a traveling salesmen problem (TSP) based solution to route each AV more efficiently. The final route for the vehicles to follow is computed in the IoT cloud and then it is communicated to the respective vehicle. The routes of the vehicles can be changed dynamically while serving the current passengers based on the condition that the new request does not change the current route drastically. The new incoming requests are dynamically allocated to the nearest vehicle from the IoT cloud and a detour from the current route is suggested if the cost of including the new pick-up location is accountable.

This chapter focuses on each of the two techniques proposed for associating pick-up locations is used with the traveling salesmen-based routing technique that showcases the overall solution to the first-mile ridesharing problem. Various numerical experiments are conducted for the proposed solution methodology using different combinations of the number of pick-up locations and vehicles. Our results show that the usage of the MILP approach proved to be more efficient for smaller fleet sizes. Although both methods performed equally for larger fleets too, the total vehicle miles travelled (VMT) for the entire fleet is lesser for the MILP approach when compared to the K-means approach.

Furthermore, the proposed techniques ensure more request services until the total vehicle capacity gets exhausted. Therefore, this approach encourages the passengers to use public transport for their first-mile transit with AV shuttles.

Index Terms

Vehicle routing, mixed-integer linear programming, K-means, ridesharing, first-mile transit, fleet management, Internet-of-Things, human-machine interface.

I. Introduction

Machine interaction and IoT have made presence in almost every industry and transportation is no more exception. IoT used for transportation can help in tracking vehicle and passengers, connect infrastructures and plan better rides for people. Ridesharing is one common transportation mode which aims in reducing congestion and hence is the need for today's urban traffic scenarios. Table I shows the recent work carried out on the ridesharing problem over several years.

In general, there are two types of ridesharing: static and dynamic. In static ridesharing, both the vehicle and passengers are known in advance, and once the ridesharing strategy is computed in advance, no further change will be made. Whereas, in dynamic ridesharing, each trip arrives online and a vehicle is assigned for an arrived trip without the knowledge of trips in the future. The most common characteristics of the ridesharing problem considered are the independent vehicles, automatic matching and cost-sharing between the vehicle and the passengers. Other factors include carpooling, dynamic assignment in real-time based on internet-enabled mobile devices. To achieve this, many mathematical optimization models are developed, to minimize either the total travel distance or time and costs of vehicles' trips or the passengers' trips and the total number of vehicles required. Other studies were also seen to maximize the number of served travel requests.

TABLE I

RECENT WORKS ON RIDESHARING PROBLEM

Ridesharing problems	**Methodology**	**Remarks**
Dial-a-ride problem [1]	A set of designated vehicles is given who can provide shared service for ride requests from the passengers	It is formulated as a single objective function
Car pooling Problem (static ridesharing) [2]	Both an exact and heuristic method is used	It is based on two integer programming (IP) formulations
Dynamic ridesharing problem [3]	A generational genetic algorithm to solve such an IP formulation	The objective function contains four optimization goals at the same time (multi-objective function)
Dynamic ridesharing problem with rolling horizon [4]	Optimization-based approaches in practical environments	Developed simulation environment based on travel demand model data
Large scale real-time ridesharing [5]	Developed a kinetic tree algorithm that is capable of better scheduling dynamic requests and adjusting routes	Model is based on large scale taxi dataset

In this chapter, we focus on first-mile ridesharing problem. First-mile (FM) problem [6] in urban transport systems refers to the connection between the passengers' homes and public transport hubs such as metro stations or bus interchange. This first-mile gap compels passengers to choose a private mode of transport over public transport systems [7]. The problem lies in providing reliable public transport with an option that is profitable to the operator and reliable to the passenger at the same time. An efficient fleet that works on-demand and served by shuttles of limited capacity rather than the traditional loop lines served by buses which have high capacity and higher operating cost might be economically viable and efficient. The problem lies where there is low or unreliable demand. In such cases, loop lines that are often served by buses are not envisioned as the best way to operate [8].

Autonomous vehicles supposedly can make more intelligent routing decisions in a real-time traffic scenario along with efficiency in fleet management, which could be an even more efficient and reliable choice. This chapter defines a first-mile transportation problem that is similar to a capacitated vehicle routing problem with multiple depots [9]. Demand from certain stops which are described in detail below will be fed to transport hubs or metro stations by Autonomous vehicles based requests made in a time window. Passengers who use this service are required to reach these designated stops to use the service.

Extensive research has been done in the field of operations research related to vehicle routing and related problems. Some of these approaches can be also be used for first and last mile transport systems. In [6], the authors proposed one such application of mixed-integer linear programming (MILP) to solve the first-mile problem using autonomous vehicles. A similar MILP model is implemented in [10] to solve the last-mile problem which included scheduling. Another relevant use case is the demand responsive transport (DRT) problem [11][12]. The research in [13] discusses a greedy local optimization heuristic algorithm to solve a typical first-mile DRT problem optimizing the vehicle miles travelled. A constrained DRT problem using the traditional MILP formulation used a two-stage approach where a heuristic is implemented first followed by MILP formulation for vehicle routing.

This chapter presents a two-level solution approach for first mile transit service. The significant contributions are listed below:

- A two-level heuristic approach and a MILP model are proposed for clustering passengers based on demand, and spatial information.
- The vehicle routing has been performed using Tabu search.
- A comparative study has been carried out by combining either of these two approaches.
- Simulation result has been presented for real-data in a Singapore based scenario.

The rest of the chapter is organized as follows: Section II describes the first-mile ridesharing problem and various assumptions considered. Section III presents the MILP approach for associating the passenger's request with the nearest vehicle. Section IV proposes a two-level heuristic approach for clustering the pick-up locations based on demand and spatial information. Section V describes vehicle routing methodology for the travelling salesman problem, which uses the results of the MILP approach and two-level heuristic approach for completing the vehicle routing. Section VI presents results with real data from an urban area in Singapore. Section VII discusses the conclusion and future work.

II. Problem description

First-mile ridesharing problem is a variant of the vehicle routing problem (VRP) that involves dispatching vehicles for transporting the passengers from their respective locations to the nearest public transport hubs. The objective here is to minimize the overall transportation cost by reducing the total vehicle miles travelled (VMT) by the entire fleet. The solution will provide the passengers reaching their destination in a shorter time compared to traditional loop lines with a fixed route. The passengers who wish to use this service need to book a ride in advance. All passengers' requests are aggregated within a time span which is then served in the next time span. The parameter time span can be decided by the operator based on certain factors such as fleet-size availability, demand and operational area.

Let $\mathbb{N}$ be the set of n pick-up locations, $\mathbb{M}$ be the set of m vehicles and $\mathbb{D}$ be the set of drop-off locations. Let $\mathbb{G}$ be the union of $\mathbb{N}$, $\mathbb{M}$ and $\mathbb{D}$ with total elements be $p = n + m + d$. In FM problem, $\mathbb{D}$ has a single element (i.e. $d = 1$) which is the public transport hub. Each pick-up location have passenger request (demand at that pick-up location) which has to be served by a single or multiple vehicles. Let the total requests be r. Each vehicle has a fixed capacity to accommodate q passengers. The total request r can be less or more than total vehicle capacity mq. If the total requests exceed the full vehicle capacity, then the requests are served until the total fleet capacity is exhausted. The remaining requests shall be considered in the next iteration(i.e. time span). The objective here is to minimize travel cost and serve every passenger until the vehicle capacity exhausts. Let $\mathbb{S} = s_1, s_2, \cdots$ be the feasible solution space. Each solution $s_i \in \mathbb{S}$ needs to satisfy basic constraints:

- Every route of a vehicle ends at a public transport hub node.
- The load of the vehicle cannot exceed the total capacity at any time.

The next section presents an MILP model to associate pick-up locations with there nearest available vehicle.

III. MILP based clustering: Approach I

This section describes a technique for associating pick-up locations with the nearest available vehicles. This problem can be formulated as multi knapsack problem using MILP. The vehicles can be assumed as analogous to different knapsacks and pick-up locations as the items to be placed in these knapsacks. The request at each pick-up location can be considered as item weights and the distances of pick-up locations with each vehicle as values. On the contrary, the objective here will be to minimize the total cost in the knapsacks (i.e. sum of distances) rather than maximizing it, as pick-up locations have to be associated with the nearest available vehicle. Each node (i.e. pick-up location) is envisioned to be shared by more than one vehicle. Hence, the items are considered as individual passengers rather than pick-up locations. The minimization of the objective function will result in a zero value as none of the items (i.e. passengers) will be allocated to the knapsacks (i.e. vehicles). Therefore, an extra constraint of limiting a minimum number of passengers for the vehicles is imposed. For this problem, it is the vehicle capacity itself.

Let E and Y be a matrix of size $m \times r$. Each row and column of the matrix represent vehicles and passengers, respectively. The elements e_{kl} of matrix E are the distances between passengers and vehicles and matrix Y represents the decision variables y_{kl} where $k \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, r\}$. The decision variables are defined as follows:

$$y_{kl} = \begin{cases} 1, \text{if } k^{th} \text{ vehicle picks up } l^{th} \text{ passenger in the route} \\ 0, \text{otherwise } . \end{cases}$$

The objective function is defined in Eq. (1) as the sum of distances between passengers and vehicles such that the passenger gets allocated to the nearest available vehicle. The objective function and constraints are defined as follows:

$$\text{minimize} \sum_{k=1}^{m} \sum_{l=1}^{r} e_{kl} y_{kl} \tag{1}$$

subject to

$$\textstyle\sum_{k=1}^{m} y_{kl} \quad \begin{cases} = 1, \text{if } r \leq mq \\ \leq 1, \text{if } r > mq \end{cases} \quad \forall l \in \{1, \ldots, r\} \tag{2}$$

$$\textstyle\sum_{k,l=1}^{m,r} y_{kl} \quad = \begin{cases} r, \text{if } r \leq mq \\ mq, \text{if } r > mq \end{cases} \tag{3}$$

$$\textstyle\sum_{l=1}^{r} y_{kl} \quad \leq m \qquad \forall k \in \{1, \ldots, m\} \tag{4}$$

Constraints can be formulated for two conditions: (a) When the total requests are less than the whole vehicle capacities, each passenger must get allocated to a vehicle while some of the vehicle seats may remain vacant and all the requests shall be fulfilled; (b) On the contrary, when the total requests are more than full vehicle capacities, the passengers are allocated until the vehicle capacity exhausts. Therefore, the remaining requests shall be considered for the next round and the total request fulfilled is equal to the total vehicle capacity. This constraint is illustrated by Eqs. 2 and 3. Eq. 4 refers to the constraint where a total number of passengers allocated to a vehicle will always be less than the vehicle capacity. The next section presents an alternative two-level heuristic approach for associating vehicles to the respective pick-up locations.

IV. K-MEANS BASED CLUSTERING: APPROACH II

This section presents the K-means algorithm-based technique for pick-up location allocation. K-means is one of the popular unsupervised clustering algorithms used for clustering spatial and non-spatial data sets. The goal of the K-means clustering algorithm is to group the data points into non-overlapping subgroups. The inter-cluster data points are kept as similar as possible while we have tried to keep the clusters as different as possible. The similarity between the data points is usually determined using distance-based measurements such as Euclidean-based distance or correlation-based distance. The decision of similarity measure is application-specific. Following are the steps for the K-means algorithm:
Step 1: Specify the number of clusters K.
Step 2: Initialize the centroids by randomly selecting K points from the dataset.
Step 3: Assign the data points to the closest centroid.
Step 4: Recalculate the centroids for each clusters by computing the average of all the data points.
Step 5: Keep iterating till there is no change in the centroids i.e assignment of the data points to the clusters is not changing.

The above K-means clustering algorithm can be used for associating pick-up locations to the nearest vehicles. The vehicle locations can be considered analogous to centroids and pick-up locations as the data points. The proposed technique uses a K-means clustering algorithm for assigning the pick-up locations to vehicles based on distance information. These pick-up locations are again clustered based on the number of requests as the passengers assigned to vehicles should not exceed the vehicle capacity.

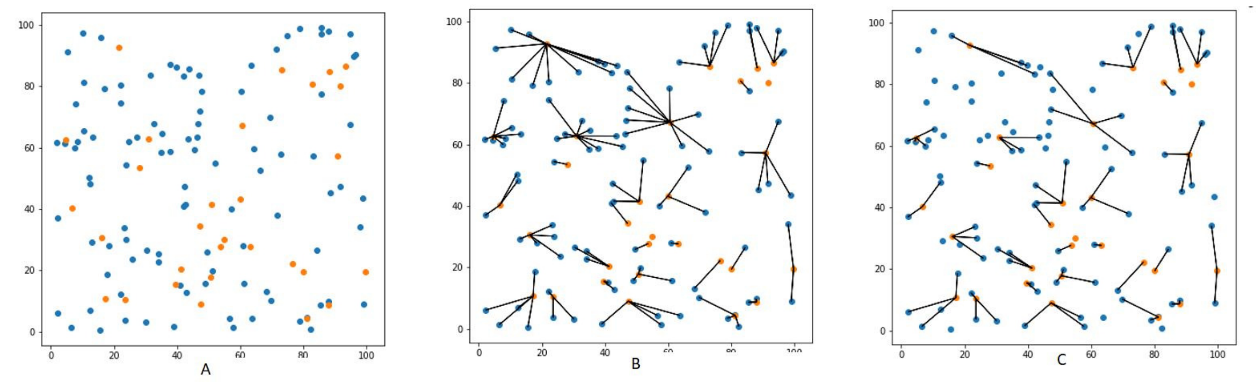
The proposed technique uses Algorithm 1 that computes the set of pick-locations to be visited for each vehicle. Let L and V be the list of pick-up locations and vehicles from sets $\mathbb{N}$ and $\mathbb{M}$, respectively. Let W be the final array of lists of pick-up locations allotted to each vehicle. Let T be a list which contains the information about vehicle associated with the respective pick-up location. This list is obtained after applying K-means algorithm on L as data-points and V as centroids.

Algorithm 1 Computation of pick-up location assignment

```
function ASSIGN_PICKUP_LOCATION(L,V)
    while sizeof(L) ≠ 0 || sizeof(V) ≠ 0 do
        T ← K_means(L, V)
        for h = 1 to sizeof(L) do
            if req(L(h)) > 0 then
                if load(T(h)) > req(L(h)) then
                    W ← {T(h), L(h)}
                    L ← remove(L(h))
                else
                    if load(T(h)) > 0 then
                        req(L(h)) ← req(L(h)) − load(T(h))
                        W ← {T(h), L(h)}
                    end if
                    V ← remove(T(h))
                end if
            else
            end if
        end for
    end while
     return W
end function
```

Algorithm 1 returns a list W which gives the information of pick-up locations associated with their respective vehicles. The algorithm stops when either the vehicle capacity exhausts or all the requests are fulfilled. List T consists of information about the particular vehicle assigned to which pick-up location. If the load of the assigned vehicle ($load(T(h))$ for the pick-up location is more than the request ($req(L(h))$) at that location, then the vehicle and pick-up location pair ($\{T(h), L(h)\}$) is added in W. Also, pick-up location $L(h)$ is removed from L. Otherwise, the remaining request which is not served is updated by reducing it by the load of the assigned vehicle and adding the vehicle and pick-up location pair ($\{T(h), L(h)\}$) in W. Since the total capacity of the vehicle is exhausted, that vehicle ($T(h)$) is removed from V. Fig. 1 illustrates a single iteration example of this algorithm where a sample dataset is considered with blue and orange points representing pick-up locations and vehicles respectively. Fig. 1(b) shows that result after the application of the K-means algorithm where pick-up locations are associated with the nearest vehicle. Since the vehicle has limited capacity, some of the pick-up locations are filtered out to be considered for the next iteration, as shown in Fig. 1(c).

The following section discusses the use of travelling salesmen problem (TSP) for individual vehicle



(a) Sample dataset with vehicles and pick-up locations

(b) Output of K-means algorithm

(c) Output after considering vehicle capacity

Fig. 1. Sample iteration for Approach II

routing by using the results from Approach I and II.

V. Vehicle Routing Methodology

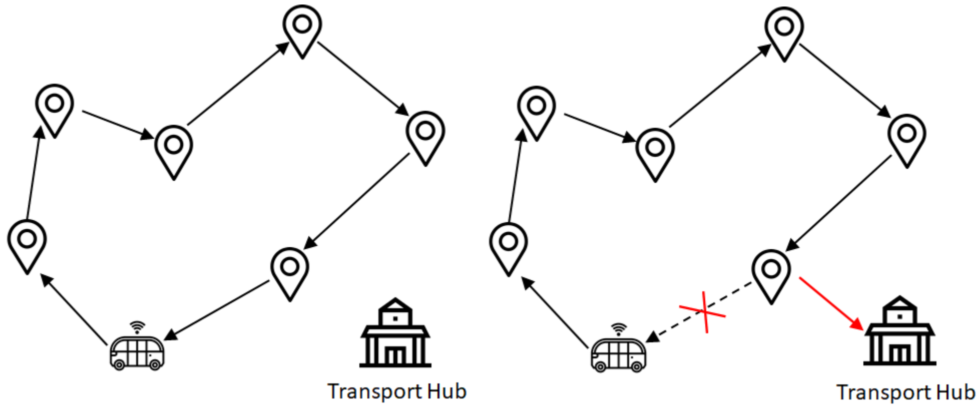
In this section, we describe a technique for vehicle routing for first-mile transit. These techniques require a list of pick-up location associated with each vehicle which is obtained from Section III and IV. Each vehicle has to visit a specified number of pick-up locations in a way that minimizes the distance travelled by the vehicle. A vehicle routing algorithm is implemented for each vehicle individually for its associated pick-up locations. Next subsection describes an algorithm that uses TSP for vehicle routing.

A. Vehicle routing using TSP

This algorithm works on applying TSP for vehicle routing. Conventional TSP [14] is used when the origin and destination of the vehicle are the same. First-mile transit service starts with the vehicle as an origin and public transport hub as a destination. The service is analogous to the TSP with different origin and destination. Given a start location (i.e. vehicle's current location) and the pickup locations the vehicles have to visit, and a route can easily be determined using TSP. This will result in a closed route that ends at the vehicle location, as shown in Fig. 2(a). After obtaining an efficient closed vehicle route using TSP, the route connection between the last pickup location and the vehicle location is removed. A new connection is created between the last pickup location and the public transport hub, as illustrated in Fig. 2(b). This results in an open-ended route with origin as vehicle location and destination as a public transport hub. The algorithm is implemented using the Tabu search meta-heuristic [15] approach. Tabu search is a meta-heuristic procedure that explores a set of solutions by repeatedly moving within a given problem space [16]. In the next section, a block diagram of the proposed workflow is presented.

B. Block diagram description

The block diagram of the proposed techniques is illustrated in Fig. 3 which shows the workflow for solving the FM problem. The overall FM problem is divided into two parts. In the first part of the problem, an optimization problem is solved for assigning pickup locations to the nearest available vehicle. The second part of the problem is responsible for applying vehicle routing for each vehicle to visit its associated pickup locations and find an optimal route from the vehicle location to the public transport hub. Section VI discusses the simulation results and comparison between various combination of the mentioned approaches.



(a) TSP with same origin and destination (b) TSP with different origin and destination

Fig. 2. Vehicle routing example using TSP

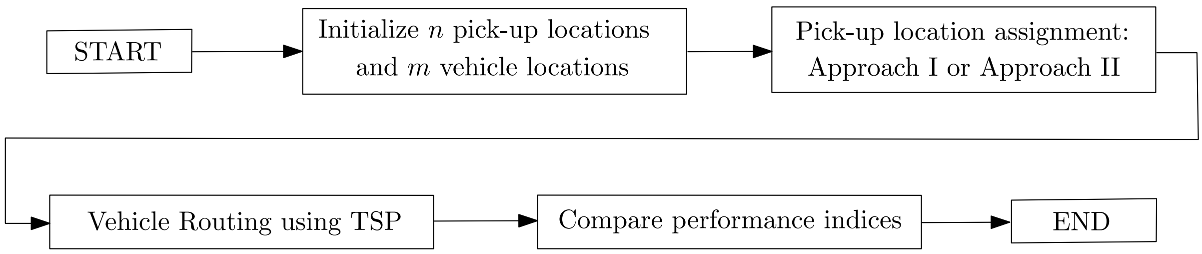


Fig. 3. Block diagram for FM problem

VI. Data description and Results

The proposed techniques for solving first-mile ridesharing problem are implemented in Python using the modules Sci-kit learn [17] and Google OR-Tools [18]. Simulations have been carried on a PC running Intel core i7-7500, 2.7 GHz processor with 24 GB RAM. An area of interest has been considered in Singapore as a motivation to analyze the results with real data.

A. *Area of Interest (AOI)*

Ang Mo Kio (AMK) is a planning area and a residential town located in the North-East of Singapore. It has two Mass Rapid Transit (MRT) train stations (Ang Mo Kio MRT and Yio Chu Kang MRT) and is further classified into twelve subzones. In this chapter, seven sub-zones that are comparatively near to Ang Mo Kio MRT are selected for the analysis as shown in Fig. 4.

The AV shuttles are assumed to have parking locations in the AOI within the traditional parking locations where they can share space with other vehicles. AV shuttles are assumed to be parked at electronic surface parking areas in the AOI. According to the data obtained from Singapore's open data portal [19] there are 62 parking locations in the AOI. These shuttles will pick-up passengers from pick-up locations, which are either existing bus stops or selected taxi pick-up/drop-off points and drop them at MRT. The next section discusses results obtained from implementing the proposed approaches.

B. *Simulation Results and Discussions*

In this section, we compare the results of the proposed techniques. The proposed techniques combined with vehicle routing techniques are (a) Approach I with TSP and (b) Approach II with TSP. Four performance indices have been considered for the comparison. These performance indices are described as follows:

- **Total vehicle miles travelled:** This is the summation of the total distance travelled by all the vehicles. The lower value of total VMT signifies lesser transportation costs.
- **Percentage of requests fulfilled:** This parameter informs about the percentage of requests served. If the total request is less than the total vehicle capacity, then the percentage of requests fulfilled will be 100% and vice versa.
- **Computation time:** It indicates the computation time required for the technique to calculate all the routes for the vehicles efficiently.
- **Percentage of fleet utilized:** The fleet utilization is also an important parameter to evaluate the performance of proposed techniques. When the total request is less than the total vehicle capacity, it can be easily comprehended that lesser the fleet utilization, more efficient is the technique.

The simulation is carried for three fleet sizes of vehicles $m = \{10, 20, 30\}$. The number of pick-up locations are varied as $n = \{30, 60, 90\}$. Maximum vehicle capacity is considered as $q = 12$ and maximum request at a pickup location is considered as 5. Vehicle locations and pickup locations are generated randomly from the AOI. Table II, III and IV tabulate the simulation results for fleet size 10,20 and 30, respectively. From the tables, it is evident that the total vehicle miles travelled are less for Approach I with TSP than Approach II with TSP. There is not much difference in the computation time of the proposed approaches. In both of the approaches, if the total request is less than the total vehicle capacity, then 100% demand is fulfilled and less than 100% when the total requests are more than the total vehicle capacity. The fleet size utilization is maximum when the total request is more than total vehicle capacity.

TABLE II

PERFORMANCE INDICATORS FOR $m = 10$ AND NUMBER OF PICK-UP LOCATIONS $(n = 30, 60, 90)$

	$m = 10$, $n = 30$		$m = 10$, $n = 60$		$m = 10$, $n = 90$	
	Approach II + TSP	Approach I + TSP	Approach II + TSP	Approach I + TSP	Approach II + TSP	Approach I + TSP
Total VMT	41223	28000	52370	35862	98135	34023
% Demand fulfilled	100	100	91	79	82	54
time-taken (s)	7.13	7.11	9.07	10.45	17	10.6
% Fleet utilized	80	80	90	100	90	100

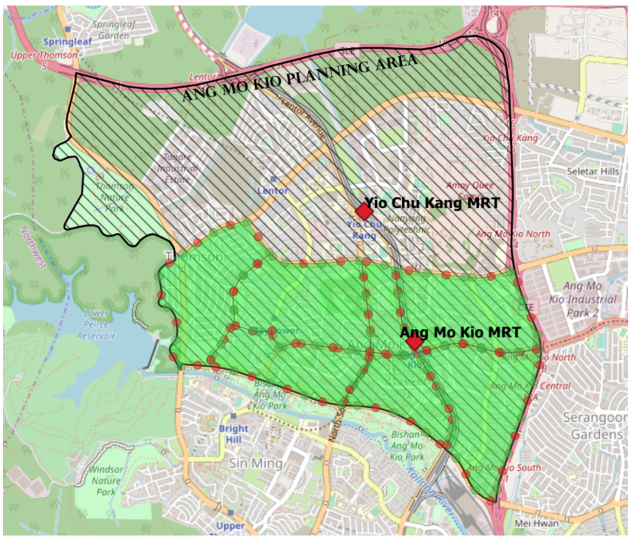


Fig. 4. AMK planning area with AOI indicated in green

TABLE III

PERFORMANCE INDICATORS FOR $m = 20$ AND NUMBER OF PICK-UP LOCATIONS $(n = 30, 60, 90)$

	$m = 20$, $n = 30$		$m = 20$, $n = 60$		$m = 20$, $n = 90$	
	Approach II + TSP	Approach I + TSP	Approach II + TSP	Approach I + TSP	Approach II + TSP	Approach I + TSP
Total VMT	52270	30102	76457	54834	98990	73872
% Demand fulfilled	100	100	100	100	100	100
time-taken (s)	7.70	6.87	13.48	14.42	17	20.5
% Fleet utilized	65	48	86	77	94	98

TABLE IV

PERFORMANCE INDICATORS FOR $m = 30$ AND NUMBER OF PICK-UP LOCATIONS $(n = 30, 60, 90)$

	$m = 30$, $n = 30$		$m = 30$, $n = 60$		$m = 30$, $n = 90$	
	Approach II + TSP	Approach I+ TSP	Approach II + TSP	Approach I + TSP	Approach II + TSP	Approach I + TSP
Total VMT	59792	31230	88828	54900	113528	75345
% Demand ful-filled	100	100	100	100	100	100
time-taken (s)	6.99	7.39	14.94	14.41	21.16	21.88
% Fleet utilized	54	34	74	54	87	73

VII. CONCLUSION AND FUTURE WORK

Vehicle routing problems, especially ridesharing problems have turned into a popular topic due to the advancements in mobility and the transportation sector. This chapter outlines distance-based optimization techniques to solve the first-mile transit problem. The FM transit problem is divided into two parts. Firstly, the pick-up locations are associated with the nearest available vehicle using Approach I and II as mentioned, then, the individual vehicle routing is performed using TSP. The FM problem is solved using a combination of pick-up location association and followed by vehicle routing as proposed in this chapter. The results suggest that the total distance is minimum for Approach I with TSP, which results in a feasible solution. The proposed techniques ensure that maximum requests are served until the total vehicle capacity gets exhausted. Therefore, this approach encourages the usage of public transport for first-mile transit using AV shuttles using IoT cloud.

Autonomous vehicles and vehicle routing have far-reaching applications with a lot of scope for improvement. Some potential future research directions include: (1) an attempt to include more constraints such as battery management of AVs, time windows for pick-up and drop-off, and fleet optimization of AVs for the first-mile use case; (2) an extension of search algorithm to emergency dial-a-ride (DAR) systems, such as medical, fire, and rescue services; (3) extend prediction techniques to large scale problems, in order to tackle the routing problems with traffic congestion and investigate the influence of passenger comfort under congestion; (4) investigation of the latest vehicular technologies, such as electric, hybrid in the context of demand-responsive transportation with heterogeneous demand, fleet, dynamic traffic conditions; (5) to enhance the computational speed-up of prediction algorithms for the vehicle routing problems; (6) Sensitivity analysis on length of operating horizon and other key parameters such as penalty cost, buffer time, and fixed cost.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the contributions of Ms. Priyanka R. Mehta of Energy Research Institute, Nanyang Technological University, Singapore for her timely support. This material is based on research/work supported by National Research Foundation under Virtual Singapore Program Award No. NRF2017VSG-AT3DCM001-018. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Research Foundation.

REFERENCES

[1] F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of operations research*, vol. 153, no. 1, pp. 29–46, 2007.

[2] V. M. R. Baldacci and A. Mingozzi, "An exact method for the car pooling problem based on lagrangian column generation," *Operations research*, vol. 52, no. 3, pp. 422–439, 2004.

[3] W. Herbawi and M. Weber, "The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.

[4] M. S. N. Agatz, A. Erera and X. Wang, "Dynamic ride-sharing: A simulation study in metro atlanta," *Transportation Research Part B: Methodological*, vol. 45, no. 9, pp. 1450–1464, 2011.

[5] R. J. Y. Huang, F. Bastani and X. S. Wang, "Large scale real-time ridesharing with service guarantee on road networks," in *In Proceedings of the VLDB Endowment*. Citeseerx, 2014, p. 2017–2028.

[6] S. Chen, H. Wang, and Q. Meng, "Solving the first-mile ridesharing problem using autonomous vehicles," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 1, pp. 45–60, 2020.

[7] T. Perera, A. Prakash, C. N. Gamage, and T. Srikanthan, "Hybrid genetic algorithm for an on-demand first mile transit system using electric vehicles," in *International Conference on Computational Science*. Springer, 2018, pp. 98–113.

[8] S. Chandra and L. Quadrifoglio, "A model for estimating the optimal cycle length of demand responsive feeder transit services," *Transportation Research Part B: Methodological*, vol. 51, pp. 1–16, 2013.

[9] J. R. Montoya-Torres, J. L. Franco, S. N. Isaza, H. F. Jiménez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Computers & Industrial Engineering*, vol. 79, pp. 115 – 129, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036083521400360X

[10] H. Wang, "Routing and scheduling for a last-mile transportation system," *Transportation Science*, vol. 53, no. 1, pp. 131–147, 2019.

[11] R. Chevrier, P. Canalda, P. Chatonnay, and D. Josselin, "Comparison of three algorithms for solving the convergent demand responsive transportation problem," in *2006 proceedings of IEEE Intelligent Transportation Systems Conference*, Sep. 2006, pp. 1096–1101.

[12] J. Brake, J. D. Nelson, and S. Wright, "Demand responsive transport: towards the emergence of a new market segment," *Journal of Transport Geography*, vol. 12, no. 4, pp. 323–337, 2004.

[13] T. Perera, A. Prakash, and T. Srikanthan, "A scalable heuristic algorithm for demand responsive transportation for first mile transit," in *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2017, pp. 000 157–000 162.

[14] I. Brezina Jr and Z. Čičková, "Solving the travelling salesman problem using the ant colony optimization," *Management Information Systems*, vol. 6, no. 4, pp. 10–14, 2011.

[15] S. Basu, "Tabu search implementation on traveling salesman problem and its variations: a literature survey," 2012.

[16] F. Glover and E. Taillard, "A user's guide to tabu search," *Annals of operations research*, vol. 41, no. 1, pp. 1–28, 1993.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[18] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: https://developers.google.com/optimization/

[19] "Singapore government's open data portal," https://data.gov.sg/, accessed: 2020-02-20.