Cluster-First, Route-Second Heuristic for EV Scheduling in On-Demand Public Transit

Thilina Perera, Alok Prakash, Deshya Wijesundera, Thambipillai Srikanthan
School of Computer Sciene and Engineering
Nanyang Technological University
Singapore, 639798
pere0004@e.ntu.edu.sg, {alok, deshya.w, astsrikan}@ntu.edu.sg

Chathura Nagoda Gamage
Dept. of Computer Engineering
University of Peradeniya
Sri Lanka, 20400
chathuratng@eng.pdn.ac.lk

***Abstract*—On-demand transit has significantly changed the landscape of personal transportation. Even traditional public transit is being overhauled by employing similar strategies, leading to the introduction of new services such as on-demand public transit (ODPT). ODPT links a geographical area using a fleet of vehicles that operate with flexible routes and timetables as opposed to its' fixed route and timetable counterparts. Further, strict regulations on reducing the carbon footprint has enforced transport operators to rely on electric vehicle (EV) fleets in public transit. However, in addition to the requirement to compute routes and schedules in real-time for ODPT services, EVs also impose constraints due to reduced driving ranges. This necessitates highly responsive real-time algorithms to cater for the significantly larger number of computations. To this end, we propose a hybrid methodology, which exploits parallel computing techniques using a clustering algorithm to decompose a large problem into smaller sub-problems, which are subsequently solved using a genetic algorithm. The result obtained from this step is used as an initial solution in a global optimization stage to further improve the quality of results. Experiments using the actual road network show that the proposed method not only improves speed of computation but also the quality of results compared to the state-of-the-art.**

***Index Terms*—genetic algorithm; shortest path; intelligent transportation systems; electric vehicles; vehicle mile traveled; hybrid systems; modelling**

I. Introduction

On-demand transit has drastically changed over the past five years in comparison to its earlier variants, which focused on para-transit services for the elderly and disabled. Nowadays, passengers and drivers are connected through mobile platforms, which enables real-time on-demand transit services such as UberX [1]. Moreover, technology has enabled services such as UberPool that allow multiple (1-2) passengers to share a ride in a vehicle without violating estimated arrival time at destination of co-passengers. Further, it has resulted in novel on-demand public transit (ODPT) services where traditional fixed route public transit is replaced with on-demand buses.

Typically, in ODPT, buses respond in real-time to the point-to-point transit requests of passengers by providing pick-up and drop-off services at designated bus stop locations. ODPT is provided in selected geographical areas, typically, during off peak hours [2]. In general, these services set a constraint on maximum travel time of a passenger request to ensure that the user-experience of the service is maintained [3]. Within these constraints, the operator strives to minimize the vehicle miles traveled (VMT) to ensure the profitability of the service resulting in a real-time combinatorial optimization problem.

However, unlike the private transit services that allow limited sharing of rides, ODPT services allow a large number of passengers to share a single ride. Also, a significantly higher number of passengers request for the service. Thus, it is necessary to perform a large number of computations in order to optimally allocate vehicles to passengers and to schedule pick-up of the allocated passengers in a vehicle. Therefore, almost all ODPT services impose a deadline before the start of the service to compensate for the significantly high runtime of the scheduling algorithms. The deadline can be a few hours before start of service for a guaranteed seat or a few minutes which does not guarantee a seat [4]. Therefore, responsiveness of the algorithm is of significant importance in ODPT.

Additionally, the growth in the adoption of alternative fuel vehicles [5] such as electric vehicles (EV) for public transit fleets have enforced further constraints on the operator. Due to the high infrastructure cost required for fast or rapid charging of EVs, operators are forced to tolerate the large downtimes and limited driving range. Thus, in addition to managing the ODPT service, a fleet of EV imposes an additional constraint of limited driving range. Hence, EV-based ODPT services have been preferred for high frequency but relatively short distance first/last mile transit [6], deployed in resort islands [7], universities [8], industrial estates [3] etc.

However, as shown in Section II, existing works on scheduling EV for ODPT is limited to low capacity sharing of rides or low number of passenger ride requests, which can potentially be used only in rural areas or off-peak hours. Thus, the responsiveness of these algorithms for high capacity deployment has not been validated. Hence, we identify a need to develop a methodology that provides near-optimal schedules for EV-based ODPT in near real-time. To this end, the contribution of this work is a hybrid methodology, which exploits parallel computing techniques using shortest path based clustering to decompose a large problem into smaller sub-problems, which are subsequently solved using a genetic algorithm for each cluster. The result obtained from this step is used as an initial solution in a global optimization to further improve the quality of results. Experimental results for an actual road network show that the proposed hybrid method not only improves speed of computation but also quality of results.

II. Literature

The concept of on-demand transit originates from the vehicle routing problem (VRP) proposed by Dantzig and Ramser [9]. The VRP has been modified by introducing numerous real-life constraints resulting in a large number of variants. In this domain, the on-demand transit problem was proposed by Psaraftis [10]. Initially, the problem was limited to a single vehicle and solutions focused on exact methodologies such as dynamic programming, integer linear programming etc. Later, variants such as multiple vehicles and depots, time-windows, loading and unloading criteria were introduced. Consequently, solutions differed from exact approaches to heuristic formulations such as tabu search, simulated annealing, genetic algorithms etc.

In the large body of research, we note several works that explicitly focus on ODPT. The works by Uchimura et al. [11], Tsubouchi et al. [12] and Uehara et al [13] propose ODPT services operated by small buses and/or vans, which provides door-to-door service similar to a shared public taxi. These services connect passengers to the inter-community express bus service at a transit hub. However, in all the works the proposed algorithms have been validated only for small test instances. Also, the service requires advance reservations to compensate for the significantly high algorithm runtime. The additional constraints of limited driving ranges have also not been considered in the works, which will further increase the algorithm runtime.

Wang [14] presents a study on a last-mile transit service with batch demands that result from passengers arriving at a mass transit node such as a train station at the same time. The author proposes algorithms to schedule passengers to a fleet of multiple types of delivery vehicles that transport passengers to their end destination with the objective of minimizing passenger travel time (waiting time and riding time). However, the arrival time of batch demands is known in advance and hence real-time route and schedule computation is not a requirement. Raghunathan et al. [15] advances the study in [14] by proposing an integrated last-mile transit problem. Here, authors consider multiple time-windows for arrival of passengers on the mass transit node. Authors propose a clustering heuristic and a subsequent integer linear programming (ILP) formulation to optimally schedule the passengers such that the total travel time is minimized. The work requires ride requests to be received in advance to compensate for the high runtime of the ILP. Further, authors do not consider the additional constraints imposed by using EV.

Scheltes and de Almeida Correia [16] study a variant of the problem in [14] by replacing conventional vehicles with EV. However, authors only analyze a personalized EV service and hence the complexities arising as a result of multiple passengers sharing an EV is not considered. Zhu et al [17] propose a dynamic path planning strategy based on a greedy algorithm for a peer-to-peer ride-sharing service. In the experiments, authors use characteristics of a Tesla model S car with five seats and supercharger facilities. Thus, the algorithm needs to be further verified with EV characteristics used for public transit. Perera et al. [18] propose a genetic algorithm to model an on-demand first mile transit service provided by EV. The work proposes to minimize the total passenger travel time within the driving range constraint of the EV fleet. Even though, results show that the proposed algorithm can produce schedules significantly quicker, it does not exploit parallel computing techniques as that proposed in this work.

III. Proposed on-demand public transit service

The ODPT service proposed in this work is based on that in [18]. Here, we summarize the operational details of the service. The ODPT service operates in a given geographical area known as a zone. We assume that passengers dispersed in a zone request for immediate rides using a mobile phone application by indicating the pick-up bus stop. Also, we assume that all passengers travel to a common destination such as a train/metro station or a shopping mall in the zone. The fleet of EV are assumed to be dispersed in the zone. EV have the same seating capacity, driving range and are empty when the algorithm is executed. We assume that the proposed algorithm is executed periodically to allocate passengers to the EV and schedule their pick-up. Here, all ride requests received during successive executions of the algorithm are considered as the input demand to the algorithm. Next, based on the outcome of the algorithm, each EV visits the allocated bus stops in the sequence specified by the algorithm to pick-up passengers and finally arrive at the common destination. We assume that EV can only visit the charging stations when it reaches the destination. Thus, when performing computations each EV needs sufficient driving range to reach the destination after picking-up each passenger.

However, the excessive travel time of passengers compared to a private vehicle ride to the destination is a drawback of high capacity ride-sharing systems such as the ODPT in this work. Here, travel time is equivalent to the addition of waiting time for an EV and the riding time on the EV to the destination. To mitigate this drawback, we impose a maximum travel time constraint for each passenger request based on the travel time to the destination from the origin under prevailing traffic conditions. Thus, each request is constrained by a multiplicative factor (for example 4x) of the travel time to the destination using a private vehicle ride, which is imposed as the maximum travel time constraint. Therefore, each rider is guaranteed that the EV reaches the destination before the maximum travel time constraint is reached. This multiplicative factor is known as the quality of service (QoS) of the system. This additional constraint will aid to uphold user-experience of the ODPT service as well as increase its predictability. The objective of the study is to derive a set of schedules in real-time to minimize the total VMT of the fleet of EV. This will ensure that operator profits are maintained while providing the ODPT service by reducing the number of charging instances that is required for the EV.

IV. Methodology

In this section, we present the proposed methodology for generating routes and schedules for the fleet of EV that relies on a cluster-first, route-second approach shown in Fig. 1. The approach of clustering a large problem into smaller sub-problems that can be solved independently using parallel computing techniques can significantly improve speed of computation of the algorithm. In contrast, badly selected clusters degrade the solution quality as EV have to detour a long distance to pick-up passengers instead of following the path to the destination. In the proposed algorithm, the clustering phase is a one-time offline step. In this phase, bus stops are divided into clusters. Therefore, this step is only required if a new bus stop or a road is added to the network. The routing phase is executed periodically when passengers are scheduled to the EV. Here, EV are first allocated to the clusters based on the real demand. Also, as the objective of the study is to minimize the VMT, it is equally important that most of the EV are fully utilized. This will reduce the number of EV required to serve passengers and in turn reduce the VMT. Therefore, only the minimum number of EV are allocated to a cluster. Thus, after clustering bus stops and allocating EV, we use a state-of-the-art metaheuristic proposed in Perera et al. [18] to solve each smaller sub-problem to near optimality. The genetic algorithm (GA) proposed in [18] strives to minimize the total passenger travel time. However, based on the objective of this work, the code is modified to minimize the VMT. Next, any unscheduled passengers are scheduled using the remaining EV. Finally, the results obtained from the previous steps are used as an initial solution in a global optimization to allow inter-cluster exchange of passengers to improve solution quality. Next, we describe the steps of the proposed method in detail.

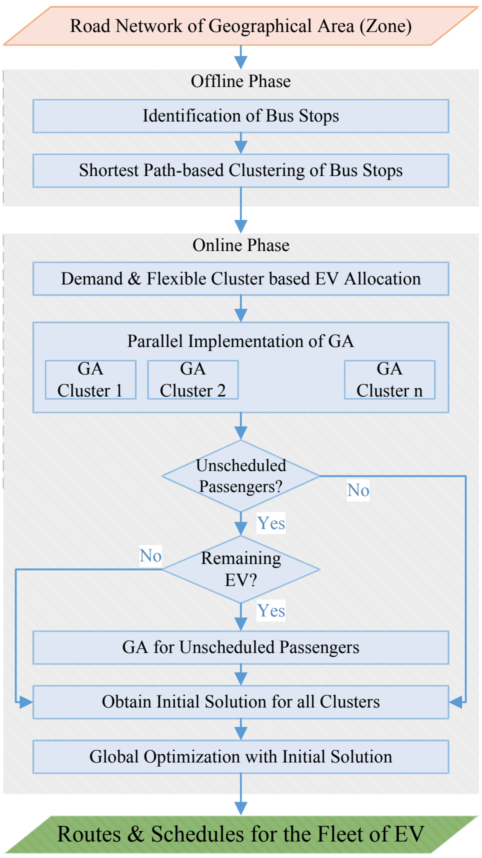


Fig. 1: Proposed Methodology

A. *Shortest Path-based Clustering*

The input to the offline phase consists of the road network of the selected geographical area. Next, bus stop locations are identified and the GPS coordinates are extracted. Thereafter, a clustering algorithm is used to decompose the problem into smaller sub-problems. In general, clusters are formed based on the shape of the zone. For example, k-means clustering finds clusters that are spherical and k-medians clustering finds clusters that look like diamonds [19]. Thus, traditional clustering techniques are not suitable for ODPT services such as the one presented in this work. Therefore, existing works use a variant of the sweep algorithm [20] to generate clusters. However, as the sweep algorithm works with polar angle, the clusters generated using the algorithm in a road network may include paths which require significant detours. Thus, we propose shortest path based clustering of bus stops. Also, since this is an offline phase, the runtime does not deter the responsiveness of the algorithm. The technique for generating clusters based on the shortest path is explained subsequently.

Initially, we obtain the shortest path data from Open Source Routing Machine [21] by providing the GPS coordinates of the bus stops and the destination. After obtaining all route data, it is converted to a tree data structure. Here, the root of the tree represents the destination, the branches represent waypoints (junctions in the road network) along the route to the destination and the leaf nodes represent the bus stops. Thus, traversing the tree from a leaf node to the root shows the route an EV requires to take in order to reach the destination. Thus, multiple bus stops which coincide on the same shortest path, with only the distance from the last waypoint to the bus stop differing, will overlap in this data structure. These stops are formed into clusters which will reduce the VMT of the EV. However, if a bus stop is located along the shortest path of multiple other bus stops, they are excluded from the tree and kept in a separate list which is allocated to the final clusters based on the demand during the online phase. These are termed as bus stops that belong to flexible clusters.

The formulation of clusters is explained using the road network shown in Fig. 2. Here, the black dots show the bus stops and the purple triangle at the bottom left shows the destination. There are 24 bus stops in the selected zone. The constructed tree of the bus stops is shown in Fig. 3. Here, the number at the leaf node represents bus stops directly connected to the waypoint. We identify 3 clusters as shown in Fig. 3 that have 5, 8 and 7 bus stops each. Cluster 1, which has 5 bus stops is shown in Fig. 2 outlined with a black ellipse. They are all connected to the same waypoint and have the same shortest path from the waypoint. In addition, we identified 4 bus stops that have been grouped into flexible clusters.

B. *EV Allocation to Clusters*

In this step, the EV are allocated to the clusters based on the real demand. The pseudo code of the algorithm used for allocation is given in Algorithm 1. Initially, in Algorithm 1, the demand is aggregated and the minimum number of EV required for each cluster is computed (line 2-3). Then, the required number of EV, which are nearest to the cluster center (center of the GPS coordinates) is allocated (line 4). This is repeated for all the clusters. At the end of the first stage of allocation, we compute the remaining seating capacity of each cluster (line 5). Then, passengers at bus stops that belong to flexible clusters are allocated such that the number of EV used is minimized (line 9). However, if the capacity is insufficient

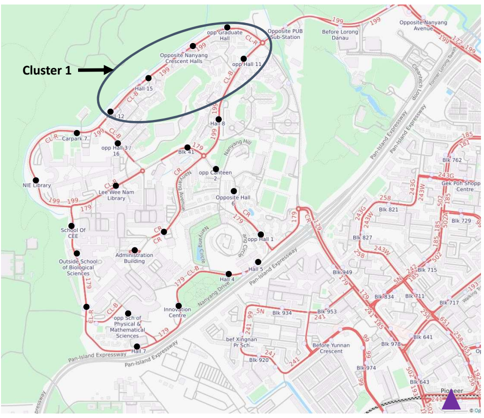


Fig. 2: Road Network of the Zone

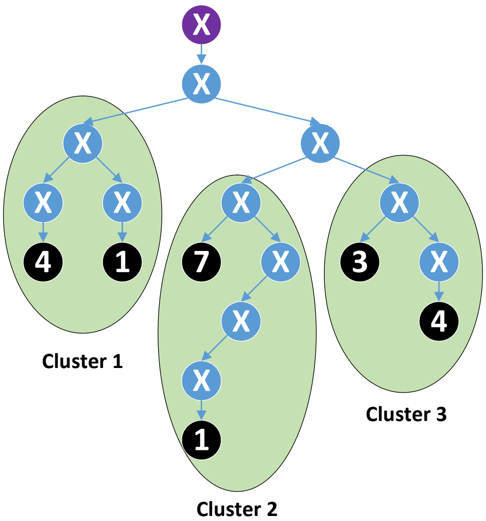


Fig. 3: Tree of the Bus Stops

new EV are allocated from the fleet (line 10). At the end of EV allocation, we use Open Source Routing Machine [21] to generate the distance and time matrices required for the GA. Here, the OSRM Route Service API is used to obtain the distance and time matrices of the fastest route by providing the GPS coordinates of the bus stop, EV locations and destination.

Algorithm 1 Pseudo Code of the EV allocation algorithm

Input: Clusters of Bus Stops, Demand at Bus Stops, Bus Stop and EV Locations
Output: Final Clusters with Passengers and EV

```
1: repeat(
2:     aggregate demand of passengers
3:     calculate the minimum number of EV required
4:     allocate EV nearest to the cluster centre
5:     compute the remaining capacity
6:     )
7: until final cluster
8:
9: allocate passengers at bus stops with flexible clusters
10: if (insufficient capacity) allocate EV from the fleet
```

C. Parallel Implementation of the GA

The next step in the method relies on the modified GA, which is executed in parallel for each cluster. As mentioned, we modify the algorithm in [18] to optimize for reducing VMT. Also, the work in [18] considers that all passengers are served by the EV. However, in this scenario, we modify the GA such that it only schedules the passengers that meet the constraints. Later, all unscheduled passengers are scheduled using the remaining EV. The output of this step consists of the schedules of the allocated EV in each cluster and the list of unscheduled passengers if any.

Main steps of the GA are explained here for clarity. GA is a multi-population algorithm. The algorithm starts with a pool of initial solutions that are generated using local search techniques and random allocation. Each solution in the solution pool, termed a chromosome consists of the schedule for the EVs. Fitness value of a chromosome represents the quality of the solution. Thus, a chromosome with a higher fitness value implies the schedule of the EVs are superior (less VMT) in comparison to a chromosome with a lower fitness value. The GA strives to minimize the VMT of a chromosome by performing genetic operations, namely mutation and crossover. However, certain chromosomes are selected from the solution pool prior to performing genetic operations. A roulette wheel based selection strategy is used for selection. After genetic operations, the suitable subset of candidates are selected for the next iteration of the algorithm. This process is repeated until the termination criteria (a predefined number of iterations) is satisfied.

D. Implement GA for Unscheduled Passengers

In this step, unscheduled passengers of all the clusters are scheduled using the GA. However, if all passengers are scheduled or if all the EV are used at the initial allocation step, the algorithm advances to generate the initial solution.

E. Generate the Initial Solution

The initial solution consists of the schedules for all EV in the given problem. It is generated by combining the solutions obtained in each cluster and the solution for unscheduled passengers if any. It is noteworthy that the initial solution obtained at this step also gives a feasible solution for the ODPT service. However, we perform another step to optimize the schedules since clustering can degrade the solution quality.

F. Global Optimization to Improve Solution Quality

In the final step of the method, the initial solution is used as an input with the large problem to the GA used in [18]. Therefore, at this step the GA can perform any inter-cluster optimizations which are not possible in the earlier step as the problem is decomposed into small sub-problems. Further, as we provide an initial solution to the large problem, runtime is not affected by performing this step. The output of this step gives the final schedules for the fleet of EV.

V. Results

In this section, we present the experimental results of the proposed method. Initially, we explain the selected geographical area followed by the experimental parameters. Then, we discuss the comparison strategy. Finally, we show the results obtained for runtime and VMT of the fleet of EV. The method proposed in Section IV is implemented in C++. The runtime is measured on a PC with 16 GB RAM, running Windows 7 Professional on an Intel Xeon E5-1650V2 CPU at 3.50 GHz.

A. Geographical Area (Zone)

As mentioned in Section I, EV-based ODPT services are used for frequent trips in areas where existing traditional public transit penetration is relatively low. Therefore, we select two zones for the experiments, which display such characteristics. In both zones, the nearest train/metro station is selected as the common destination of the ODPT service. The first zone is selected surrounding a University. The other zone is a large residential area. A University generally has a high population density and typically provides fixed route shuttle services to the nearest train/metro station. Thus, it is an ideal platform to validate the ODPT service. The residential area is selected as a proof of concept that ODPT services should not be restricted to specific zones. Table I indicates the number of bus stops, clusters and bus stops with flexible clusters in the selected zones.

TABLE I: Output of the Offline Phase

Geographical area	Bus stops	Clusters	Bus stops with flexible clusters
Zone 1	45	3	12
Zone 2	36	3	14

B. Experimental Parameters

Experimental parameters used for performance evaluation of the ODPT service is given in Table II. The parameters in the setup consists of number of passengers and EV, the capacity and driving range of EV, and QoS. The number of passengers in the experiments are chosen based on the real demand that is observed on fixed route transit services using historical data. EV characteristics are based on the parameters used in [18]. QoS parameter indicates that each passenger is guaranteed to reach the destination within 4 times the travel time to the destination using a private vehicle. For example, if the travel time to the destination from the location of the bus stop is 6 minutes, maximum travel time constraint of the passenger will be 24 minutes. Also, in the experiments, passengers and EV are randomly distributed in the zone. Thus, some bus stops contain multiple passengers while some will be empty.

TABLE II: Values of the Parameters

Parameter	Test Case Number		
	1	2	3
Number of Passengers	60	90	120
Number of EV	10	15	20
EV capacity	8		
EV driving range	30 km		
QoS	4x		

C. Comparison Strategy

In order to evaluate the benefits of the proposed method in terms of runtime and quality of results, we use 2 versions of

TABLE III: Parameters of the GA

Parameter	Method		
	Proposed	SOA V_1	SOA V_2
Population	50	50	1000
Iterations	100	100	5000

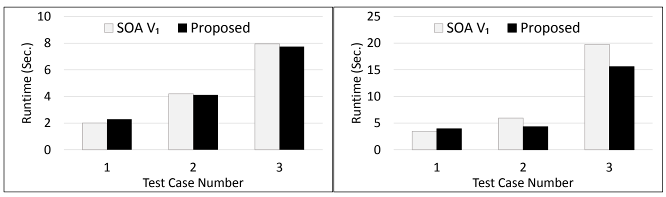
the GA proposed in [18] (henceforth referred as the state-of-the-art (SOA)). The runtime of the proposed method is compared with the SOA by using identical parameters. To this end, we implement the GA with a population size of 50 and 100 iterations (SOA V_1). The solution quality is compared with an instance of the SOA by implementing the GA with a population size of 1000 and 5000 iterations (SOA V_2). Parameter values have been tuned using the generate and test principle [22]. This provides baseline results for comparison. In addition, the result obtained in SOA V_1 is also used to show that the proposed method not only improves runtime but also quality of the solution. Table III depicts the parameters used in the 3 methods.

D. Performance Evaluation

Here, we present results of the 3 experiments in the two zones. Initially, we show the distribution of passengers among the clusters in each experiment. Next, the runtime is compared with the SOA V_1 method. Finally, we show the VMT from the solution obtained from the proposed method and compare it with SOA V_1 and SOA V_2 methods. Here, a comparison with the former method shows the improvement of the solution and the latter shows the deviation from baseline results.

TABLE IV: EV and Passenger Distribution

Zone	Test Case Number	{EV,Passenger} tuple for Cluster Number			
		1	2	3	Unscheduled
Zone 1	1	1,6	5,35	3,19	-
	2	3,21	6,46	3,23	-
	3	2,15	8,64	6,41	1,2
Zone 2	1	2,15	4,25	3,20	1,2
	2	3,22	6,43	4,25	1,8
	3	5,38	6,45	5,37	-

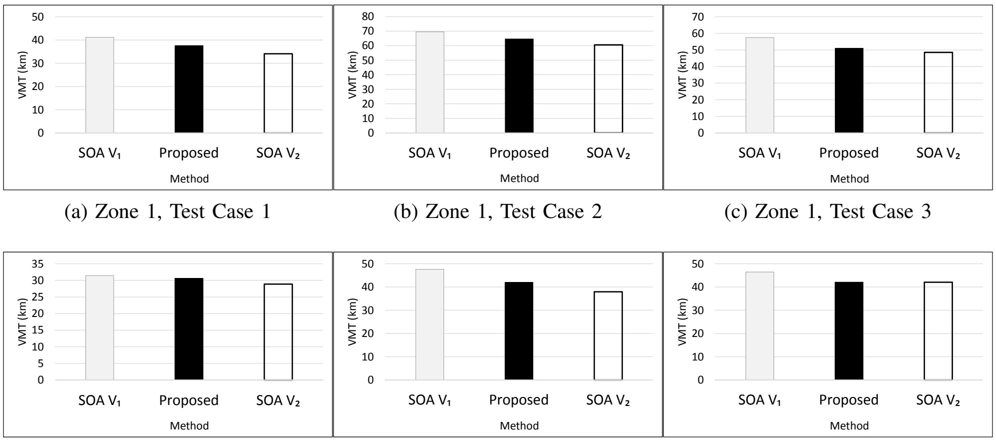


(a) Zone 1 (b) Zone 2

Fig. 4: Runtime Comparison of the Algorithm

Table IV shows the EV and passenger distribution of the experiments. Results show that cluster 2 in zone 1 is crowded compared to the others. In zone 2, passengers are uniformly distributed. Also, the number of unscheduled passengers in the initial step is higher in zone 2. This is mainly due to the large geographical area of zone 2, which increases the travel time between the bus stops that leads to failed travel time constraints.

Figure 4 shows the runtime comparison between SOA V_1 and the proposed methods. The horizontal axis in the graph shows the test case number while the vertical axis shows the



(a) Zone 1, Test Case 1 (b) Zone 1, Test Case 2 (c) Zone 1, Test Case 3

(d) Zone 2, Test Case 1 (e) Zone 2, Test Case 2 (f) Zone 2, Test Case 3

Fig. 5: VMT of the fleet of EV

runtime in seconds. Here, we observe that in both zones the runtime of the proposed algorithm is in the same order as the SOA V_1 method. In fact, in 4 test cases the runtime of the proposed method is superior. However, as we observe in Fig. 5, the proposed algorithm generates schedules which are significantly superior to SOA V_1 method without compromising the runtime. Figure 5 shows VMT of the EV fleet from the 3 methods. Here, the horizontal axis shows the 3 methods, namely SOA V_1, proposed and SOA V_2 respectively. The vertical axis shows VMT in km. We observe an average 7.6% improvement of the quality of results compared to SOA V_1 and an average deviation of 6.6% compared to the SOA V_2 (baseline).

VI. Conclusion

This work proposes a hybrid methodology for scheduling an ODPT service, which links a geographical area using a fleet of EV that responds in real-time to point-to-point transit requests of passengers. The problem incorporates finding the schedules of the EV fleet that minimizes the VMT under various constraints. In comparison to the state-of-art, we propose a method to decompose a large problem into small subproblems by using a shortest path based clustering algorithm and subsequently generate schedules for each cluster using a GA. Finally, a global optimization is performed to mitigate the impact of clustering. Experiments on two geographical areas prove that the proposed algorithm can significantly improve the quality of results without compromising on the runtime. In future, we plan to derive algorithms for moving passengers from a common origin to multiple destinations. Also, we plan to extend the work to consider city-wide deployment of ODPT services that requires to identify profitable clusters in real-time in order to implement the proposed algorithm.

Acknowledgment

This research project is partially funded by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program with the Technical University of Munich at TUMCREATE.

References

[1] Uber, "Uber Ride: Always the ride you want," https://www.uber.com/us/en/ride/, 2019.

[2] Land Transport Authority, Singapore, "On-Demand Public Transit Trial," https://www.lta.gov.sg/content/ltaweb/en/public-transport/buses/odpb-trial.html, 2018.

[3] RideCo, "RideCo improves mobility in an industrial area with on-demand shuttles, in partnership with a government agency," https://bit.ly/2WXeUNM, 2019.

[4] GrabShuttle Plus, "GrabShuttle: On-demand Shuttle Service for a Smarter Commute," https://www.grab.com/sg/shuttleplus/, 2019.

[5] M. Matthias, "Driving ban for diesel-powered vehicles in major cities: an appropriate penalty for exceeding the limit value for nitrogen dioxide?" *International Archives of Occupational and Environmental Health*, vol. 91, no. 4, pp. 373–376, 2018.

[6] Circuit, "Circuit: Ditch the car and enjoy the ride," http://www.thefreeride.com/, 2019.

[7] Z. Abdullah, "Driverless shuttle trials start on Sentosa; on-demand service for public from 2019," https://bit.ly/2Y9Ffoi, 2018.

[8] D. Etherington, "Ford and MIT project provides on-demand electric shuttles for students," https://tcrn.ch/2avLDmb, 2016.

[9] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[10] H. N. Psaraftis, "A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem," *Transp. Sci.*, vol. 14, no. 2, pp. 130–154, 1980.

[11] K. Uchimura *et al.*, "Demand responsive services in hierarchical public transportation system," *IEEE TVT*, vol. 51, no. 4, pp. 760–766, 2002.

[12] K. Tsubouchi *et al.*, "Innovative on-demand bus system in Japan," *IET Intell. Transp. Syst.*, vol. 4, no. 4, pp. 270–279, 2010.

[13] K. Uehara *et al.*, "Evaluation of a hierarchical cooperative transport system using demand responsive bus on a dynamic simulation," *IEICE Trans. Fundamentals*, vol. E99-A, no. 1, pp. 310–318, 2016.

[14] H. Wang, "Routing and Scheduling for a Last-Mile Transportation System," *Transp. Sci.*, vol. 53, no. 1, pp. 131–147, 2019.

[15] A. U. Raghunathan *et al.*, "The Integrated Last-Mile Transportation Problem (ILMTP)," in *ICAPS*, 2018.

[16] A. Scheltes and G. H. de Almeida Correia, "Exploring the use of automated vehicles as last mile connection of train trips through an agent-based simulation model: An application to Delft, Netherlands," *Int. Journal of Transp. Sci. and Tech.*, vol. 6, no. 1, pp. 28 – 41, 2017.

[17] M. Zhu *et al.*, "An Online Ride-Sharing Path-Planning Strategy for Public Vehicle Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 616–627, 2019.

[18] T. Perera *et al.*, "Hybrid genetic algorithm for an on-demand first mile transit system using electric vehicles," in *ICCS*, 2018.

[19] F. Cakir *et al.*, "Revisiting cluster first, route second for the vehicle routing problem," 2015.

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

[21] Open Source Routing Machine, "OSRM API Documentation," https://bit.ly/2XCChvM, 2019.

[22] A. E. Eiben and S. K. Smit, *Evolutionary Algorithm Parameters and Methods to Tune Them*. Springer Berlin Heidelberg, 2012, pp. 15–36.