Genetic Algorithm based Dynamic Scheduling of EV in a Demand Responsive Bus Service for First Mile Transit

Thilina Perera, Alok Prakash and Thambipillai Srikanthan

Abstract— **Demand responsive transit (DRT) services have significantly evolved in the past few years owing to developments in information and communication technologies. Among the many forms of DRT services, demand responsive bus (DRB) services are gaining traction as a complimentary mode to existing public transit services, especially to dynamically bridge the first/last mile connectivity. Simultaneously, the stern regulations imposed by regulators on greenhouse gas emission have enforced electric vehicles (EV) to replace conventional vehicles. However, state-of-the-art (SoA) work proposed to generate routes for EV-based DRB services are inhibited by the low number of ride matches and the excessively high computation time of the algorithms deeming them unsuitable for real-time computations. To this end, we propose a genetic algorithm for dynamic scheduling of EV in a DRB service that reacts to first mile ride requests of passengers. In addition, we also formulate an optimal mixed integer program to generate baseline results. Experiments on an actual road network show that the proposed GA generates significantly accurate results compared to the baseline in real-time. Further, we analyze the benefits of rescheduling passengers and flexible estimated time of arrival of EV to optimize the total travel time of passengers.**

I. INTRODUCTION

Digital revolution has paved the way for novel business models. Even public transportation, which has traditionally relied on fixed routes and schedules, is undergoing a tremendous transformation from entirely new ideas. For example, demand responsive transit (DRT) services with flexible routes and schedules that connect riders to drivers, through a mobile platform have become a common practice. Typically, in a DRT system riders use their smartphone to request a ride by indicating their origin, destination, intended pick-up/drop-off time and the vehicle type. The service provider strives to match the ride requests with the available vehicles and allocates them to the riders, thereby facilitating ride-sharing with co-riders. After the allocation, the estimated time of arrival (ETA) of the vehicle is communicated to the rider, while the new routes are updated in the vehicle.

The different variations of DRT services in operation [1] are broadly classified into static and dynamic DRT services based on the temporal scheduling capability. In a static DRT service, all ride requests are known beforehand. Thus, matching passengers and allocating vehicles can be processed off-line. These services do not allow real-time changes to the routes even though they are still flexible based on the received advance requests. In contrast, in a dynamic DRT service, ride requests must be scheduled in real-time. Therefore, real-time scheduling and routing algorithms must be developed to match passenger requests to available vehicles.

Dynamic DRT services have been typically used to satisfy the point-to-point transit requests of passengers. For example, services such as Uber which has been in operation since the inception in San Francisco, California in 2011 [1] typically match 2 - 3 passengers going in the same direction to a single vehicle. Lately, another variant of a dynamic DRT service [2] typically known as demand responsive buses (DRB), which match multiple passengers to a single vehicle, has gained traction as a complementary mode to enhance the user experience in public transit by extending its penetration through improved first/last mile (FM/LM) connectivity. DRB provide transit services to passengers with common origin/destination such as large-scale industrial estates with multiple co-located companies, universities and residential buildings in a neighborhood where the population density is significantly high and public transit penetration is low. Thus, in addition to the real-time computation requirement in algorithms, DRB services also need to ensure responsiveness for the relatively high number of passenger requests.

At the same time, stern regulations imposed on greenhouse gas emission by vehicles [3], has made electric vehicles (EV) a popular alternative mode of transit. Thus, works have studied the impact of EV-based point-to-point DRT services [4]. However, the limited driving range and high downtime during charging of the EV pose additional constraints on service providers, which restricts EV being considered as a mainstream mode in public transit. Hence, managing EV-based DRB services is a hot research topic.

To this end, our previous work proposes an EV-based DRB service that connects passengers in a neighborhood to the nearest rapid transit node and vice versa [5]. However, in [5], dynamic passenger requests are mapped to a series of periodic static requests. Thus, it does not allow real-time route changes, which is the main characteristic of a dynamic DRT service. Therefore, the proposed work advances the state-of-the art by incorporating dynamic passenger requests and passenger rescheduling to optimize the total travel time of passengers. Further, we consider a fleet of EV with disparate driving ranges and seating capacities. Hence, the major contributions of this work are (i) a genetic algorithm (GA) that generates near-optimal results in real-time for a dynamic DRB service (ii) an optimal mathematical formulation to model the proposed EV-based DRB service which generates baseline results for comparison and (iii) an analysis of the impact of rescheduling passengers.

*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.

Authors are with School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. pere0004@e.ntu.edu.sg, alok@ntu.edu.sg, astsrikan@ntu.edu.sg

II. Related Work

The problem of matching passengers and allocating vehicles in a DRB service stems from the vehicle routing problem (VRP). Earliest literature on the VRP considered the static case with a fleet of m homogeneous vehicles originating at a single depot, serving requests of n passengers. However, with the growth of computational capabilities and technological advancements, many variants of the VRP such as heterogeneous fleets, multiple depots, dynamic passenger requests etc. have been studied [6], [7]. Hence, in this section, we provide a brief overview of important literature on DRB services and highlight the limitations.

Uchimura et al. [8] propose a DRB service operated by small buses and/or vans, which provides door-to-door service. This service connects passengers to the inter-community express bus service. A similar DRB service is proposed by Tsubouchi et al. [9], where authors show the validity of the proposed service and the algorithms by evaluating it in three cities in Japan. However, in both instances the proposed algorithms have been validated only for small test instances. Uehara et al. [10] propose a hierarchical DRB service to improve the service quality of public transit in city area and its suburbs. However, the system requires advance reservations, thus it addresses only the static problem.

Wang [11] present a study on a last-mile transit system with batch demands that results from passengers arriving at a mass transit node such as a train station or bus stop. The author proposes algorithms to schedule passengers to a multi-vehicle fleet of delivery vehicles that transport passengers to their end destination with the objective of minimizing passenger waiting time and riding time. Raghunathan et al. [12] advances the study by Wang [11] by proposing an integrated last-mile transit problem. The authors propose a clustering heuristic and a subsequent integer linear programming (ILP) formulation to optimally schedule the passengers such that the total travel time (waiting time and riding time) is minimized. However in both works, the arrival times of batch demands and ride-requests are known in advance and hence, the problem can be generalized to a static case. Scheltes and de Almeida Correia [13] study a variant of this problem by replacing conventional vehicles with EV. However, the authors only analyze a personalized EV system and hence the complexities arising as a result of multiple passengers sharing an EV is not considered.

Archetti et al. [14] study a large scale DRB service, which satisfies point-to-point transit requests of passengers. The proposed DRB service has flexible routes and bus stops. It schedules dynamic ride requests with a lead time of 10 minutes. Authors test different scenarios using realistic origin-destination locations and demand patterns in a one-hour time horizon. Also, authors study the impact of lead time, size of vehicle fleet, number of passenger requests etc. on the travel time of ride requests. They compare the obtained results against the travel data from private vehicles and public transit using simulation studies. However, in comparison to our work, which uses EV, authors use conventional vehicles and do not reschedule passengers.

III. Operation of the DRB Service

We envision that passengers in a neighborhood traveling to a common destination such as a train station, request for the DRB service for immediate rides using a smartphone application. The fleet of EV that provide the DRT service are assumed to be dispersed in the neighborhood either already serving passenger requests (non-empty) or in an idle state (empty) ready to be deployed. All ride requests are logged in a centralized database and served periodically. We assume that the database is updated in real-time with the available seats, remaining driving range, GPS location etc. of the fleet of EV. Next, using this data, an optimization algorithm is executed, which in real-time matches the passengers to the fleet of EV and outputs the routes. Finally, each passenger is communicated the ETA of the EV.

Following assumptions are considered in the operation of the DRB service. (1) The driving range of EV is directly measured by the vehicle miles traveled (VMT) irrespective of the loading and tropical weather conditions. (2) EV in idle state can visit the charging stations. Hence, empty EV with scheduled passengers or non-empty EV can visit the charging stations only after it reaches the destination. (3) Quality of service (QoS) for all passengers is guaranteed by a maximum travel time (waiting time for the EV + riding time). QoS is calculated as the multiplicative factor of the travel time from the origin to the destination under prevailing traffic conditions using a single occupancy vehicle (SOV). For example, a QoS value of Nx implies that each passenger will be ensured that the upper bound on the travel time will be N times the travel time compared to a SOV. (4) Objective of the optimization algorithm is to minimize the total travel time of the passengers whilst satisfying all the constraints.

Dynamic operation of the DRB service is explained below. We assume that in a practical scenario there are passengers who have been already allocated to an EV but still not picked-up due to the shorter scheduling period compared to the ETA. This is illustrated in Fig. 1 using two instances of a passenger request. Assume, that the request is logged at T_0 and the optimization algorithm is executed at T_1. Then, the passenger is scheduled to be picked-up at T_3 with an ETA ($T_3 - T_1$). However, at T_2 when the next instance of the optimization algorithm is executed, the EV, which is scheduled to pick-up the passenger has still not arrived and the remaining ETA is ($T_3 - T_2$). Thus, at T_2 the optimization algorithm will be executed not only on the requests logged between T_1 and T_2 but also on passengers who have not been picked-up. Thus, the DRB service can dynamically reschedule the passenger (with an increased ETA but with the same QoS) to optimize the total travel time.

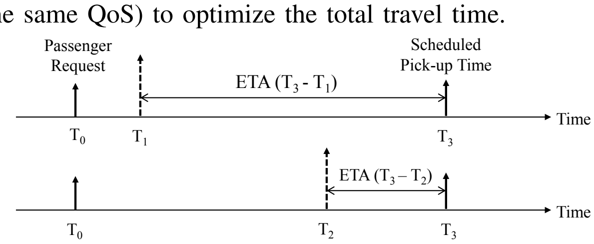


Fig. 1: Illustration of Dynamic Passenger Rescheduling

IV. METHODOLOGY

A. Problem Formulation

Assume that when the optimization algorithm is executed at T_s there are v EV denoted by Δ to serve p passenger requests denoted by Ω traveling to the destination denoted by Θ. Passenger requests (Ω) are classified as $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$, where Ω_1 denote picked-up passengers, Ω_2 denote reschedulable passengers and Ω_3 denote new passengers. Also, $p_1 =\mid \Omega_1 \mid$, $p_2 =\mid \Omega_2 \mid$ and $p_3 =\mid \Omega_3 \mid$, where $p = p_1 + p_2 + p_3$. At T_s passengers in set $\Omega_2 \cup \Omega_3$ (Π) are scheduled. This is defined using a weighted graph $G = (\Phi, E)$, where Φ defines all the nodes and E all the valid edges. G contains $v+p_2+p_3+1$ nodes (Φ), where nodes $1, 2, 3, \cdots, v$ refer to the fleet of EV (Δ), nodes $v+1, v+2, v+3, \cdots, v+p_2+p_3$ refer to the set of passengers to be scheduled (Π) and node $v + p_2 + p_3 + 1$ refers to the destination (Θ). Further, each edge $< i, j >$ has an associated weight $< t_{i,j}, d_{i,j} >$, which denotes the travel time and shortest distance between the two nodes $i \rightarrow j$ respectively. Further, all possible nodes where a valid edge $< i, j >$ can terminate is denoted Ψ ($\Phi \backslash \Delta$).

The constraints of the model consist of EV constraints, (i) the seating capacity (l_v), (ii) driving range (r_v) and (iii) maximum remaining travel time (rtt_v); and passenger constraints, (i) QoS measured by the maximum travel time (u_p) and (ii) estimated time of arrival of an EV for each passenger in Ω_2 (e_p). Even though EV does not directly have a travel time constraint, as passengers are picked-up (Ω_1), each passenger will indirectly impose a maximum remaining travel time on the EV (as a result of u_p, a travel time constraint is transferred to the EV when the passenger is picked-up). Hence, when executing the optimization algorithm, a maximum remaining travel time constraint is considered for non-empty EV. Waiting time of a passenger from Π and the travel time (waiting time + riding time) of a passenger from Ω_1 till T_s is defined by w_p and tt_p respectively. We define rescheduling ratio (R) as $p_2/(p_2 + p_3)$ and ETA flexibility window (F) as the additional time the ETA can be increased for passengers in Ω_2.

We define a two dimensional binary array (m_{pv}), which indicates the EV that has picked-up each passenger in Ω_1 and a variable (M_v), which shows the total number of passengers picked-up by each EV until T_s. Also, we define a binary variable x_{ijv}, given in Eq. 1 for each {edge, EV} tuple $\{< i, j >, v\}$, where $i \neq j$, $i \neq v + p_2 + p_3 + 1$ & $j \neq 1, 2, 3, \cdots, v$; denotes if EV v travels along the edge $< i, j >$ from i to j. The integer variable s_{iv}, given in Eq. 2 is defined for each {node, EV} pair $\{i, v\}$, denotes service time of node i by v. Table I summarizes the terminology.

$$x_{ijv} = \begin{cases} 1, & \text{EV } v \text{ travels along edge } i \rightarrow j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

$$s_{iv} = \begin{cases} f, & \text{EV } v \text{ services node } i, \ f \epsilon Z^{+}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

B. Proposed Genetic Algorithm

Block diagram of the proposed GA is given in Fig. 2. Here, we explain the modules in detail.

TABLE I: Terminology

Term	Description
Φ	set of all nodes
Δ	set of EV nodes
Ω	set of all passenger nodes
Ψ	set of termination nodes
Θ	destination node
Ω_1	set of picked-up passengers
Ω_2	set of reschedulable passengers
Ω_3	set of new passengers
Π	set of schedulable passengers ($\Omega_2 \cup \Omega_3$)
v	v^{th} EV
p	p^{th} schedulable passenger
i	node i
$< i, j >$	edge from node $i \rightarrow j$
$t_{i,j}$	travel time of edge $< i, j >$
$d_{i,j}$	shortest path distance of edge $< i, j >$
l_v	seating capacity of v^{th} EV
r_v	driving range of v^{th} EV
rtt_v	maximum remaining travel time of v^{th} EV
u_p	maximum travel time of p^{th} passenger (QoS)
e_p	ETA of an EV at p^{th} passenger from Ω_2
w_p	waiting time of a passenger from Π
tt_p	travel time of a passenger from Ω_1
R	rescheduling ratio
F	ETA flexibility window
m_{pv}	vector for picked-up passengers
M_v	total picked-up passengers in v^{th} EV
x_{ijv}	vector indicating the EV which picks up passengers in Π
s_{iv}	time v^{th} EV reaches node i (service time)

1) Encoding & Initial Population Generation: Each passenger (gene) is represented using integer encoding. A single solution (chromosome) from the solution pool gives a valid schedule for all the EV and is shown in path based representation. This classification is shown in Fig. 3 with an example of 10 passengers from set Π and 3 EV. Chromosomes are generated by modifying local search heuristics in [5]. Features such as shortest distance, travel time, maximum/minimum remaining driving range/capacity etc. are used to allocate passengers to EV. Also, random allocations are used to ensure diversity in the population.

2) Fitness Evaluation: Fitness of a chromosome is measured using Eq. 3, which minimizes the total travel time of all passengers (Ω). Here, the two terms correspond to the set of schedulable passengers (Π) and the set of picked-up passengers (Ω_1) respectively. Total travel time for schedulable passengers (Π) is equal to the summation of waiting time of each passenger (w_p) and the time the EV which picks up the passenger (x_{ijv}) reaches the destination ($s_{\Theta v}$). The total travel time for picked-up passengers (Ω_1) is equal to the summation of the travel time of a passenger (tt_p) and the time the EV which picked-up the passenger (m_{pv}) reaches the destination ($s_{\Theta v}$) after picking up other passengers.

$$\sum_{i \in \Pi} \sum_{j \in \Phi} \sum_{v \in \Delta} x_{ijv} * (s_{\Theta v} + w_i) + \sum_{p \in \Omega_1} \sum_{v \in \Delta} m_{pv} * (s_{\Theta v} + tt_p); \tag{3}$$

3) Genetic Operators: After selecting parents based on the Roulette wheel method, genetic operators are performed to improve the solution quality. Mutation helps to explore new states and avoid local optima. Crossover improves the average quality of the population. Specialized genetic operators based on the encoding scheme are used for fast convergence and maintaining diversity in the solution space [15]. In

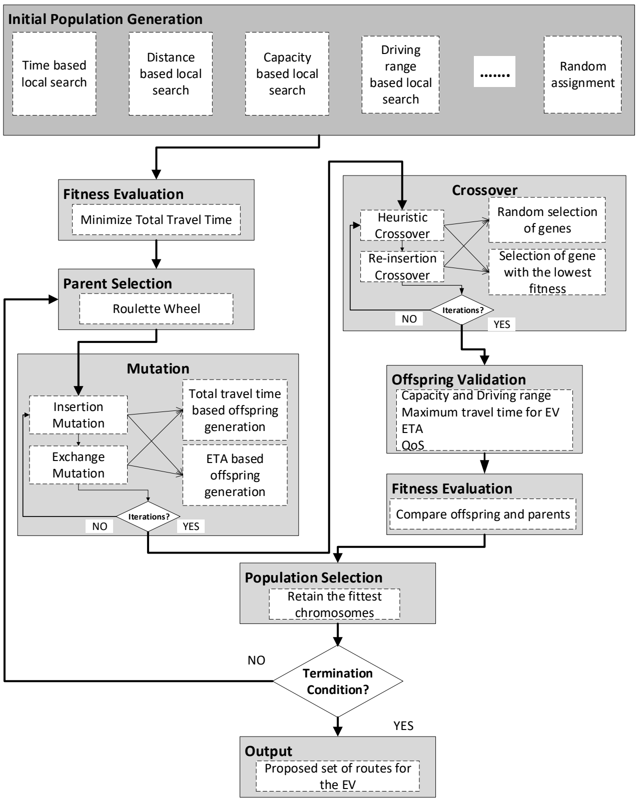


Fig. 2: Proposed Genetic Algorithm

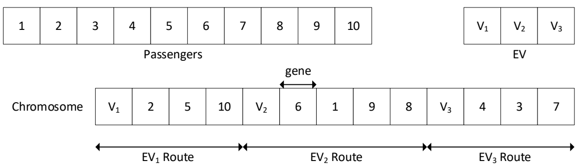


Fig. 3: Integer Encoding, Gene and Chromosome

the mutation phase, we use two operators, namely insertion mutation and exchange mutation [16]. For each operator, we generate two offspring based on reducing total travel time and also the ETA. ETA based offspring generation helps to improve the travel time of rescheduled passengers. Similarly, in the crossover phase, we use the heuristic crossover and re-insertion crossover operators [16]. In both cases, we generate offspring by random selection of genes as well as selecting the gene with the lowest fitness. In the heuristic crossover, genes from each chromosome are swapped between them while in the re-insertion crossover, the gene with the lowest fitness (passenger with highest travel time) is moved and re-inserted into the path of the other chromosome. Both operators are iterated to generate multiple offspring.

4) Offspring Validation: Offspring are validated for EV and passenger constraints given in Sect. IV-A. An offspring that violates the constraints is removed from the population.

5) Population Selection: Here, the fittest chromosomes are retained to maintain a constant population size.

6) Termination: The iterative process is terminated if consecutive iterations of the GA do not improve the solution quality. Finally, the fittest chromosome (best solution) is used as the set of routes, which minimize the total travel time.

C. Baseline Result Generation

Baseline results are generated using an optimal mixed integer program (MIP). We use the same objective as given in Eq. 3. The constraints of the model are given below.

$$\sum_{j\in\Pi} x_{ijv} = 0 \quad \forall i \in \Delta,\ \forall v \in \Delta; \quad where \quad i \neq v; \tag{4}$$

$$x_{\Theta jv} = 0 \quad \forall j \in \Phi,\ \forall v \in \Delta; \tag{5}$$

$$x_{ivv} = 0 \quad \forall i \in \Pi,\ \forall v \in \Delta; \tag{6}$$

$$x_{ijv} = 0 \quad \forall i \in \Delta,\ \forall j \in \Delta,\ \forall v \in \Delta; \tag{7}$$

$$\sum_{j\in\Psi} x_{ijv} = 1 \quad \forall i \in \Delta; \quad where \quad M_i \neq 0; \tag{8}$$

$$x_{iiv} = 0 \quad \forall i \in \Pi,\ \forall v \in \Delta; \tag{9}$$

$$\sum_{j\in\Pi} x_{vjv} \leq 1 \quad \forall v \in \Delta; \tag{10}$$

$$\sum_{j\in\Pi} x_{vjv} - \sum_{i\in\Pi} x_{i\Theta v} = 0 \quad \forall v \in \Delta; \tag{11}$$

$$\sum_{i\in\Phi} x_{ibv} - \sum_{j\in\Phi} x_{bjv} = 0 \quad \forall v \in \Delta, \forall b \in \Pi; \tag{12}$$

$$\sum_{i\in\Pi} x_{i\Theta v} \leq 1 \quad \forall v \in \Delta; \tag{13}$$

$$\sum_{j\in\Phi}\sum_{v\in\Delta} x_{ijv} = 1 \quad \forall i \in \Pi; \tag{14}$$

$$s_{iv} = 0 \quad \forall i \in \Delta,\ \forall v \in \Delta; \quad where \quad i = v; \tag{15}$$

$$x_{ijv}(s_{iv} + t_{ij} - s_{jv}) \leq 0 \quad \forall i \in (\Phi \backslash \Theta), \forall j \in \Phi, \forall v \in \Delta; \tag{16}$$

$$s_{\Theta v} \geq 0 \quad \forall v \in \Delta; \tag{17}$$

$$\sum_{i\in\Pi}\sum_{j\in\Phi} x_{ijv} \leq l_v - M_v \quad \forall v \in \Delta; \tag{18}$$

$$\sum_{i\in\Phi}\sum_{j\in\Phi} x_{ijv} * d_{ij} \leq r_v \quad \forall v \in \Delta; \tag{19}$$

$$s_{\Theta v} \leq rtt_v \quad \forall v \in \Delta; \quad where \quad M_v \neq 0; \tag{20}$$

$$s_{\Theta v} * x_{ijv} \leq u_i \quad \forall i \in \Pi,\ \forall j \in \Psi,\ \forall v \in \Delta; \tag{21}$$

$$s_{iv} \leq e_i + F \quad \forall i \in \Omega_2,\quad \forall v \in \Delta; \tag{22}$$

Valid routes always originate from EV nodes. However, a route of a particular EV originates from the node representing it. This is validated in Eq. 4. Equation 5 prevents edges originating from the destination node. Similarly, Eq. 6 and Eq. 7 prevents an edge terminating at an EV node. Equation 6 validates such edges originating from a passenger node while Eq. 7 validates the same for nodes originating from other EV nodes. Equation 8 allows EV with picked-up passengers to either travel to a passenger node (to pick-up new passengers) or to the destination (in case of new passengers not assigned). Equation 9 prevents loops at passenger nodes. Equation 10 ensures that the minimum number of EV are used to serve passengers. Equation 11 states that the sum of all EV that leave their respective origins should be equal to the sum of all EV that reach the destination. Equation 12 validates that each passenger node should have a valid incoming edge and an outgoing edge. This states that an EV arriving at a passenger node should always leave it. Equation 13 ensures for all EV that at most there is only one outgoing edge from a passenger node to the destination. Equation 14 validates that only one EV will serve a single passenger. Equation 15 initializes the service time of all EV. Equation 16 models the travel time between valid edges. However, this equation is linearized using the method proposed in Cordeau et al. [17]. Lower bound of the service time at the destination is given in Eq. 17. Equation 18 ensures that number of passengers in the EV is less than or equal to the available seating capacity. Here, when evaluating the capacity, we deduct the number of seats occupied by picked-up passengers. Equation 19 ensures that the total travel distance is less than the driving range of the EV. However, as mentioned earlier, we assume the impact of factors such as loading and air-conditioning is negligible on the driving range of the EV. Therefore, the shortest path distance between nodes (d_{ij}) is directly compared against the driving range. Equation 20 validates the maximum travel time of EV with picked-up passengers. This constraint is not applicable for EV which are empty when the optimization algorithm is executed. Equation 21 validates the QoS parameter of the system. Thus, each passenger is guaranteed that the travel time will not exceed the maximum travel time of each passenger. The final constraint in Eq. 22 validates the ETA constraint of rescheduled passengers. Here, we add a predefined ETA flexibility window to the ETA.

V. Results

This section presents the experimental results. Here, runtime is measured on a Windows 7 PC with 16GB RAM and Intel Xeon E5-1650V2 CPU at 3.50 GHz. The GA is implemented in C++. Baseline results are obtained by implementing the MIP using IBM ILOG CPLEX optimization studio 12.7.1 and the inbuilt constraint programming (cp) solver is used to solve the formulation to optimality.

1) Experimental Setup: All experiments are simulated in a locality surrounding a campus, which generally has a fixed route transit service to the nearest transit hub. Hence, this locality is ideal to test the proposed DRB service. The locality and a sample instance are shown in Fig. 4. Here, we show 6 ride requests (passenger symbol), 3 EV (bus symbol) and the destination (metro symbol). Also, we highlight different color schemes to indicate empty (red) and non-empty (black) EV and rescheduable passengers (purple) and new passengers (blue). The distance and travel time are obtained in real-time using Google Maps APIs.

2) Parameters: In each experiment, EV driving range and capacity are varied within the bounds given in Table II. QoS of the system is set at 4x and the ETA flexibility window is set to 0 minutes except in the last experiment. Other parameter values are indicated in the individual experiments. These values represent typical off-peak travel demands in the

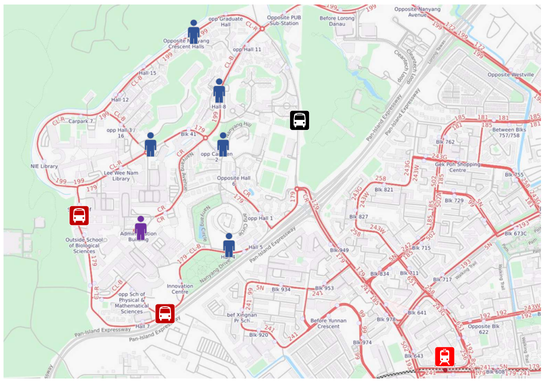


Fig. 4: Locality of the Experimental Setup

TABLE II: Parameter Values

Bound	Driving Range	Capacity	QoS	ETA Flexibility Window
Lower	25km	8	4x	0 min.
Upper	35km	12		

selected locality. Also, we distribute the passengers randomly in the vicinity. We consider a time horizon of 10 minutes and a scheduling interval of 5 minutes. Thus, the optimization algorithm is executed at instances (T_s) 5 minutes and 10 minutes. The results presented are for $T_s = 10$ instance.

3) Computation Time Variation: In this experiment, computation time is obtained by increasing the iteration count of the genetic operators and schedulable passengers ($p_2 + p_3$). The experiments are based on $p_1 = 10$, $R = 0.25$, $v = 20$. Figure 5 shows the increase in computation time with both the iteration count as well as the number of schedulable passengers. Hence, we are motivated to select a suitable iteration count for the remaining experiments.

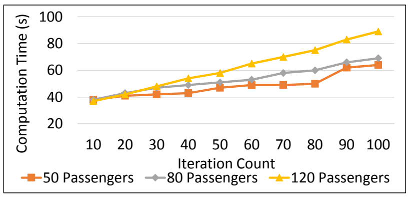
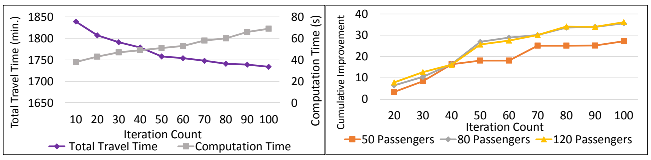


Fig. 5: Computation Time Variation

4) Selection of Iteration Count: Initially, we obtain the total travel time of the same set of experiments from the GA. Results for the case of 80 passengers is shown in Fig. 6a. The results indicate that total travel time is reduced (better solution) with the increasing iteration count. Next, we calculate the improvement of the total travel time per unit increase in computation time. Figure 6b shows the cumulative improvement of total travel time per unit increase in computation time for all the test cases. This graph indicates that after 80 iterations, the improvement for all test cases is marginal. Hence, the remaining experiments are performed with 80 iterations.

5) Accuracy of the GA: The accuracy of the GA is compared against baseline results. However, due to the complexity of the optimal formulation, baseline results are presented only for small test cases. Further, we obtain the travel time of passengers using SOV and public transit similar to Archetti et al. [14]. SOV time and public transit time indicates the lower and upper bound of the travel time



(a) Variation of Solution Quality for 80 passengers (b) Cumulative Improvement of Solution Quality

Fig. 6: Selection of Iteration Count

respectively. The results presented in Fig. 7 are based on parameter values $p_1 = 5$ and $R = 0.25$. Results indicate that the proposed GA generates schedules with high accuracy comparable to the baseline. The average deviation of results of the proposed GA is 1.13%. Further, in all cases the proposed DRB service outperforms existing public transit.

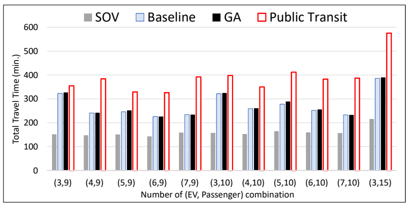
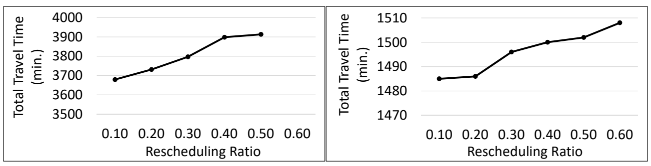


Fig. 7: Accuracy of the Proposed GA

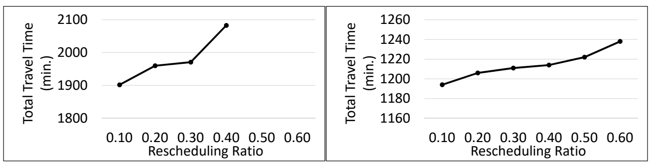
6) Impact of Reschedulable Passengers: Here, we study the impact of reschedulable passengers (p_2) on the total travel time. This experiment is run on the scenarios given in Table III. The corresponding results are shown in Fig. 8a – Fig. 8d respectively. In all graphs, the total travel time increases with R as a result of the increase in the number of reschedulable passengers (p_2). Hence, the flexibility of the GA to optimize the routes are limited due to the constraint of the ETA (e_p) of the rescheduled passengers. Further, as we observe in Fig. 8a & Fig. 8c at higher (p_2/v) values the GA does not find a feasible solution, which necessitates the use of an ETA flexibility window.

TABLE III: Experimental Scenarios

Scenario	v	p_1	$p_2 + p_3$
1	20	20	150
2	20	15	70
3	11	15	70
4	12	5	55



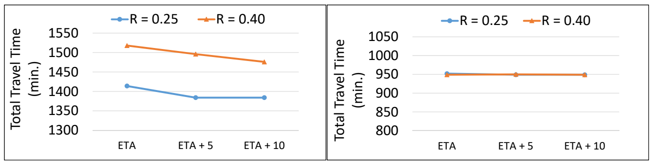
(a) Scenario 1 (b) Scenario 2



(c) Scenario 3 (d) Scenario 4

Fig. 8: Total Travel Time Variation with Rescheduling Ratio

7) Benefits of Flexible ETA: Here, we study the benefits of a flexible ETA, which allows the GA to further optimize the schedules without violating the QoS. We consider a test case with $p_1 = 5, p_2 + p_3 = 55$ with two scenarios reflecting $R = 0.25$, $R = 0.40$ and $F = 5$ min, $F = 10$ min. The results for the two scenarios with $v = 8$ and 24 are presented in Fig. 9a and Fig. 9b respectively. Figure 9a indicates that the total travel time can be reduced with a flexible ETA. Further, the benefit of rescheduling increases with higher number of rescheduled passengers. However, as shown in Fig. 9b, when the EV supply is high, benefits of rescheduling is negligible.



(a) 8 EV (b) 24 EV

Fig. 9: Benefits of Flexible ETA

VI. Conclusion

In this paper, we present a GA for dynamic scheduling of passengers in an EV-based DRB service that bridges the FM connectivity of passengers living in a neighborhood. We also present an optimal MIP to get baseline results for comparison. Further, we study the benefits of rescheduling passengers and assigning a flexible ETA. In future, we plan to vary parameters such as the number of EV, EV range and capacity to observe the changes in the solution.

References

[1] S. Shaheen *et al.*, "Shared Mobility: Current Practices and Guiding Principles," U.S. Department of Transportation, Federal Highway Administration, Tech. Rep., 2016.

[2] J. Alonso-Mora *et al.*, "On-Demand High-Capacity Ride-Sharing via Dynamic Trip-Vehicle Assignment," *PNAS*, 2017.

[3] M. Matthias, "Driving ban for diesel-powered vehicles in major cities: an appropriate penalty for exceeding the limit value for nitrogen dioxide?" *Int. Archives of Occupational and Env. Health*, 2018.

[4] M. Zhu *et al.*, "An Online Ride-Sharing Path-Planning Strategy for Public Vehicle Systems," *IEEE T-ITS*, 2019.

[5] T. Perera *et al.*, "Hybrid Genetic Algorithm for an On-Demand First Mile Transit System Using Electric Vehicles," in *ICCS*, 2018.

[6] R. M. Jorgensen *et al.*, "Solving the Dial-a-Ride Problem using Genetic Algorithms," *JORS*, 2007.

[7] J.-F. Cordeau, "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem," *ORIJ*, 2006.

[8] K. Uchimura *et al.*, "Demand Responsive Services in Hierarchical Public Transportation System," *IEEE TVT*, 2002.

[9] K. Tsubouchi *et al.*, "Innovative On-Demand Bus System in Japan," *IET Intelligent Transport Systems*, 2010.

[10] K. Uehara *et al.*, "Evaluation of a Hierarchical Cooperative Transport System Using Demand Responsive Bus on a Dynamic Simulation," *IEICE Trans. Fundamentals*, 2016.

[11] H. Wang, "Routing and Scheduling for a Last-Mile Transportation System," *Transportation Science*, 2019.

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

[13] A. Scheltes *et al.*, "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. J. of Transp. Sci. & Tech.*, 2017.

[14] C. Archetti *et al.*, "A Simulation Study of An On-Demand Transportation System," *ITOR*, 2018.

[15] R. Poli *et al.*, "Exact GP Schema Theory for Headless Chicken Crossover and Subtree Mutation," in *CEC*, 2001.

[16] H. Xu, "Comparison of Genetic Operators on a General Genetic Algorithm Package," M.Sc. diss., Shanghai Jiao Tong University, Shanghai, China, 1999.

[17] J.-F. Cordeau *et al.*, "The Vehicle Routing Problem," 2001, ch. VRP with Time Windows, pp. 157–193.