A Scalable Heuristic Algorithm for Demand Responsive Transportation for First Mile Transit

Thilina Perera*, Alok Prakash† and Thambipillai Srikanthan†
Nanyang Technological University, Singapore 639798
Email: *pere0004@e.ntu.edu.sg, †{alok,astsrikan}@ntu.edu.sg

***Abstract*—First/last mile transit using public transport has consistently been a bottleneck for commuters due to the relatively higher time spent in these legs when compared to the overall journey. Recently, demand responsive transportation(DRT) services have been proposed for the first/last mile transit. However, in contrast to the requirements of a public transportation system, existing DRT services either match only a few passengers to a vehicle or require advance booking. Hence, in this paper we propose a DRT system, specifically for the First mile transit, by matching multiple passengers to a vehicle in real-time. We first model the problem as a convergent graph and obtain an exact solution. Next, a scalable heuristic algorithm has been proposed that not only provides near optimal solution, but also does that in real-time (≈ms) as opposed to minutes/hours taken for the exact solution.**

I. Introduction

Public transit is a crucial service for any society due to its high impact on the quality of life. A public transit journey typically consists of multiple legs and uses different modalities of transportation, such as buses and subway, that have different average speeds. While subways typically run at higher average speeds due to dedicated tracks, public buses share the road infrastructure with other vehicles and hence run at lower average speeds, especially during peak traffic hours. Buses are also frequently used to service the first/last legs of a multi-modal journey to pick passengers from a neighborhood and bring them to a major transit node, typically a subway system. This creates a bottleneck during the first and last legs of transit in a multi-modal journey due to slower average speed of buses when compared to the faster subway system. This is known as the first/last mile (FM/LM) problem.

The root causes for the FM/LM problem are fixed routes, schedules, capacity and designated stops of public transit services [1]. These restrictions and the resulting FM/LM problem contribute negatively to the user experience of public transit [2]. Thus, it is evident that, in order to enhance the quality of service (QoS) and user experience of public transit, the FM/LM problem needs to be addressed with innovative solutions. There have been numerous research and commercial efforts to address the inherent issues in FM/LM trips using public transit. A few common solutions consist of personal rapid transit solutions such as segways and e-scooters, bike sharing schemes, covered walkways, casual car-pooling schemes and demand responsive transportation (DRT) systems [3] [4] [5] [6].

DRT services are characterized by (a) vehicles that do not operate over a fixed route or on a fixed schedule except, perhaps, on a temporary basis to satisfy a special need; and (b) typically, the vehicle may be dispatched to pick-up several passengers at different pick-up points before taking them to their respective destinations and may even be interrupted en-route to these destinations to pick-up other passengers [7]. Traditionally, DRT services have been used in rural areas and areas of low passenger demand [8] or to transport elderly and disabled persons. However, with the recent technological advancements in mobile communication and GPS-based location tracking systems, real-time demand management and servicing has become a reality. Therefore, owing to its 'on-demand' feature, DRT has emerged as the preferable solution to the FM/LM problem [9] [10]. However, existing real-time DRT systems is limited to one or two ride matches [11]. In contrast, solutions focusing on multiple ride matches require advanced booking [12] to cater for a significantly high response time in these systems. However, in a public transit system, several riders ride in a vehicle and hence need to be matched to this single vehicle. Also, this information has to be communicated to the riders, along with an estimated time of arrival (ETA), preferably in a near-instantaneous time limit. This necessitates rapid and scalable solutions for a real-time DRT system before it can be effectively employed for the FM/LM trips.

Additionally, other reported DRT services such as [13] have been shown to be unsustainable over time due to their inability to match riders along the same direction at the same time [14]. Hence, in our work we specifically propose to limit the DRT services within a neighborhood and for passengers going to a common destination i.e. the nearest rapid transit node.

In this paper, we explicitly focus on the FM transit issue and propose a DRT based solution. Specifically, the proposed DRT system comprises of a homogeneous fleet of vehicles with fixed capacity (in terms of maximum number of passengers per vehicle) dispersed in a neighborhood. These vehicles respond to the demands of passengers, in real-time, by picking them from their origin and dropping them off at a pre-determined nearest rapid transit node. The passengers request the service specifying the intended pick-up time window and the origin. The backend infrastructure of the proposed DRT system logs all passenger requests as well as the real-time traffic conditions. It then computes, in **real-time**, the appropriate routes and schedules for the fleet of vehicles and communicates the relevant information to the passengers, as well as the drivers of the vehicles. It should be noted that, in this work, we discuss the algorithmic aspect of the problem and not the design of the infrastructure to facilitate such a service. Moreover, our optimization goal is to devise a set of routes to minimize the total vehicle miles traveled (VMT) by the fleet of vehicles while serving all passenger requests and adhering to constraints such as vehicle capacity and pick-up time window requested by passengers. We measure the QoS of the proposed system by the average waiting time of a passenger, defined as the time interval between the actual pick-up time and the beginning of the pick-up time window.

Formally, our problem is classified as a static convergent DRT (CDRT) problem. However, it differs from the state-of-the-art as we consider a constantly high rate of passenger requests which necessitates scalable solutions. Specifically, we consider practical scenarios such as large-scale organizations, industrial estates and universities where the population density is significantly high and the penetration of public transit is relatively low compared to an urban area. Thus, the contribution of the paper is twofold; firstly, we propose an optimal mixed integer programming (MIP) formulation for the static CDRT problem and show that the time to solve the problem grows exponentially with the number of passenger requests. Secondly, we develop a rapid and scalable greedy local optimization based heuristic algorithm which provides a set of near-optimal routes in real-time.

The rest of the paper is organized as follows. In Section II we discuss the existing state-of-the-art papers on the FM/LM and the CDRT problems while Section III presents the proposed methodology. We present the results of our study in Section IV and conclude the paper in Section V.

II. Related Work

FM/LM problem is initially drawn from telecommunications, and subsequently from supply chain management [15]. Lately, it has been studied with respect to the context of public transit [16] [17]. A study in Wake County, North Carolina reports that 72% of the population is located outside the comfortable distance of the nearest mass rapid transit node [4]. Similarly, a study in Singapore reveals that the first mile journey for a significant number of trips originating at a mass rapid transit node is beyond $1km$ [18].

Ride-sharing is a mode of transportation whereby, drivers travelling towards a single destination, pick-up and drop-off other passengers travelling towards the same destination or travelling on the same route. The ride-sharing problem is formulated by extending the vehicle routing problem (VRP) to the dial-a-ride problem (DARP). The standard objective of DARP is cost minimization. Cost is modelled in-terms of both travel time and distance [19]. Similarly, maximizing the number of passengers is also studied [20]. Both exact and heuristic solutions have been proposed to solve the static and dynamic versions of the DARP. Agatz et al. propose optimization based approaches to minimize the system wide vehicle miles in a dynamic setting [21]. Cordeau et al. present a tabu search heuristic to solve the static multi-vehicle DARP. However, the authors do not compare the results of the proposed method with the optimal results due to the inability to obtain optimal

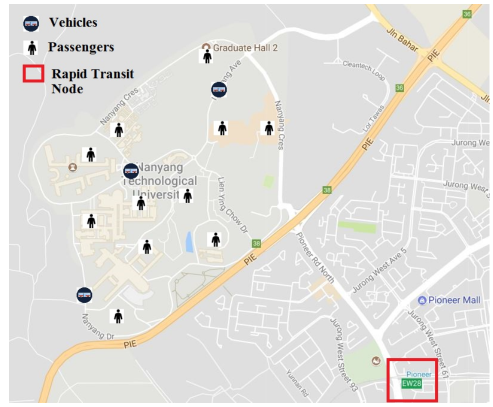


Fig. 1: Map of a potential DRT location

solutions [22]. Paquette et al. further improve the algorithm developed in [23] to combine multicriteria analysis with tabu search heuristic to solve the static DARP. However, the average runtime of the algorithm is 24 minutes.

DRT services, which is a specific case of the DARP have emerged as the preferred solution to the FM/LM problem [9] [10]. Therefore, many studies have been conducted on the implementation of DRT services. Deakin et al. present a case study of a practical ride-sharing system in San Francisco Bay Area [24]. The authors study the travel patterns and potential markets for a ride-sharing system in downtown Berkley, California and the University of California, Berkley. A similar study by MIT real-time ridesharing group, discuss the existing challenges and opportunities of ride-sharing [25]. Further, the survey in [26] provides an extensive discussion on the state-of-the-art and future directions in ride-sharing systems. [27] [13] [28] report instances of DRT services in operation. However, few DRT projects have been abandoned due to financial issues caused by the inability to match riders along the same direction at the same time [14] [29]. In order to avoid this issue, we specifically propose to limit the DRT services within a neighborhood and for passengers going to a common destination i.e. the nearest rapid transit node.

The closest work to the work presented in this paper is proposed in [30]. The authors present three algorithms to solve the multi-vehicle CDRT problem by generalizing it to n-Travelling Salesman Problems (n-TSP). The authors propose two exact methods (dynamic programming, depth first search) and a heuristic approach (genetic algorithm). However, a major drawback of their work is the requirement to book seats 4 hours prior to the ride and hence cannot be deployed for a real-time DRT system.

III. Methodology

A. Problem Statement

We study the CDRT problem where passengers within a neighborhood request the service by providing the origin and the pick-up time window as input. The requests and real-time traffic conditions are logged in the backend infrastructure of the proposed system and refreshed periodically every **2 minutes**. It then uses an optimization algorithm to schedule the shared vehicles (dispersed in the neighborhood) to pick-up each passenger from the origin and drive them to the nearest

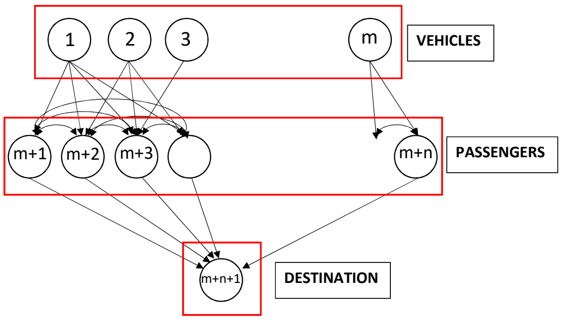


Fig. 2: Convergent Graph

rapid transit node. Figure 1 shows a potential location with passengers and vehicles dispersed in the neighborhood. We consider a realistic fixed capacity constraint for each vehicle. Additionally, we assume that either the **supply exceeds or equals the demand** for the service. Thus, ideally all passenger requests are satisfied. The optimization goal in this paper is to minimize the total VMT by all vehicles. This optimization goal is in line with the main objective of a service provider, who typically strives to minimize the vehicle miles and hence save costs.

In addition, it should also be noted that in reality, vehicles may be interrupted to pick new passengers along the way in a dynamic environment. However, since the dynamic problem can be modeled using multiple static problems interleaved in time, we focus on the static CDRT problem in this paper.

B. Problem Definition

We define our problem using a directed and acyclic convergent graph [30], G, to represent the fleet of v vehicles denoted by $\{V_1, V_2, V_3, \cdots V_v\}$, set of p passengers denoted by $\{P_1, P_2, P_3, \cdots P_p\}$, and a node to represent the common destination of the system. Hence, G consists of $(m + n + 1)$ nodes, where nodes $1, 2, 3, \cdots, m$ refer to the fleet of vehicles, nodes $m + 1, m + 2, m + 3, \cdots, m + n$ refer to the set of passengers and node $[m + n + 1]$ refers to the convergent point, where $m = v$ and $n = p$. The set of all nodes, $1, 2, 3, \cdots, m + n + 1$ is denoted as $\varPhi$. The subset of nodes $1, 2, 3, \cdots, m$, representing vehicles is denoted ν and the subset $m+1, m+2, m+3, \cdots, m+n$, representing passengers is denoted ρ. The set of edges, λ, represents all the direct connections between the fleet of vehicles, the set of passengers and the convergent point. There are no edges ending at node set ν and beginning from node $m + n + 1$. Each edge (i, j) has an associated cost, c_{ij}, and time, t_{ij}, measured in terms of distance and travel time respectively where $i, j \in \varPhi$ & $i \neq j$. Also, the travel time for each edge ending with an element of the set ρ has an additional constant service time. Figure 2 shows a sample convergent graph.

Each passenger P_i has a pick-up time window denoted, $P_{i[a,d]}$, where $a < d$ and the vehicle serving the passenger must strictly arrive at passenger P_i before $P_{i[d]}$. However, if a vehicle arrives prior to $P_{i[a]}$, it has to wait at the customer location and we assume that there is no restriction on the waiting time due to prior arrival of the vehicle. Similarly, each vehicle has a maximum capacity l. Also, it is assumed that l, a, d are positive integers and c_{ij} and t_{ij} are non-negative integers. It should be noted that we do not require the triangle inequality to be satisfied by both c_{ij} and t_{ij} which signifies the added complexity of the problem.

The two decision variables in the model x and s are defined below. The binary decision variable x_{ijk} is defined for each edge (i, j) and each vehicle V_k, where $i \neq j$, $i \neq m+n+1$ & $j \neq 1, 2, 3, \cdots, m$; denotes if vehicle V_k travels along the edge (i, j) from i to j. Similarly, the decision variable s_{ik} is defined for each node i and each vehicle V_k, where $i \neq 1, 2, 3, \cdots, m$ & $i \neq m + n + 1$; denotes the service time of passenger P_i by vehicle V_k. Specifically;

$$x_{ijk} = \begin{cases} 1, & \text{vehicle } V_k \text{ travels from node } i \text{ to node } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$s_{ik} = \begin{cases} z, & \text{vehicle } V_k \text{ services passenger } P_i,\ P_{i[a]} \leq z \leq P_{i[d]}, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The objective of our study is to devise a set of routes that minimizes the total VMT (cost), with the constraints (a) each passenger is serviced by exactly one vehicle; (b) all routes start at node set ν and end service at the destination node, i.e. node $m + n + 1$; (c) service time at each customer is within the pick-up time window; and (d) capacity of each vehicle is not exceeded.

C. Mathematical Formulation

Here, we present an optimal solution based on a MIP formulation. We used *IBM ILOG CPLEX Optimization Studio 12.7.1* [31] to find the optimal solution. CPLEX, Optimization Programming Language is used to model the problem and the in-built MIP solver is used to obtain the optimal solution. The MIP formulation is presented below.

Objective function:

$$minimize \sum_{k \in \nu} \sum_{i \in \varPhi} \sum_{j \in \varPhi} c_{ij} x_{ijk}; \quad (3)$$

Subject to:

* *Temporal Constraints*

$$P_{a[i]} \leq s_{ik} \leq P_{d[i]} \quad \forall i \in \rho, \forall k \in \nu; \quad (4)$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad \forall i, j \in \varPhi, \forall k \in \nu; \quad (5)$$

* *Spatial Constraints*

$$\sum_{i \in \rho} \sum_{j \in \varPhi} x_{ijk} \leqslant l \quad \forall k \in \nu; \quad (6)$$

* *Routing Constraints*

$$\sum_{j \in \varPhi} \sum_{k \in \nu} x_{ijk} = 1 \quad \forall i \in \rho; \quad (7)$$

$$\sum_{j \in \varPhi} x_{kjk} = 1 \quad \forall k \in \nu; \quad (8)$$

$$\sum_{i \in \varPhi} x_{ibk} - \sum_{j \in \varPhi} x_{bjk} = 0 \quad \forall k \in \nu, \forall b \in \rho; \quad (9)$$

$$\sum_{i \in \rho} x_{ijk} = 1 \quad \forall k \in \nu, j = m + n + 1; \quad (10)$$

* *Completion Constraints*

$$x_{ikk} = 0 \quad \forall i \in \rho,\ \forall k \in \nu; \tag{11}$$

$$x_{iik} = 0 \quad \forall i \in \rho,\ \forall k \in \nu; \tag{12}$$

$$x_{ijk} = 0 \quad \forall j \in \Phi,\ \forall k \in \nu,\ i = m + n + 1; \tag{13}$$

The objective of the problem (Equation 3) is to minimize the total VMT. For clarity, we have divided the constraints into four sub-categories, namely temporal, spatial, routing and completion constraints. The first temporal constraint (Equation 4) presents the timing relationship at each node. It affirms that the pick-up time window of each passenger is met by the servicing vehicle. Equation 5 models the timing relationship along an edge from the origin to the destination. It states that the service time at the destination should be higher than or equal to the addition of the service time at the origin and the travel time along the edge. However, due to the multiplicative factor this constraint is non-linear. Thus, we use the method proposed in [32] to linearize the constraint. The linearized constraint is given in Equation 14. Equation 6 deals with the spatial constraints of a vehicle by limiting the maximum passenger allocation to the capacity of the vehicle. Each passenger is allocated to only 1 vehicle by Equation 7. Next, Equation 8 ensures that the origin of a journey for each vehicle is the node representing the vehicle itself. Equation 9 affirms that after a vehicle arrives at a passenger node it has to leave for another destination. Finally, Equation 10 guarantees the destination of a journey to be the convergent point. The three completion constraints ensure that, a route does not end at a vehicle node (Equation 11), there are no loops in the routes (Equation 12) and a route does not begin at the convergent point (Equation 13). Also, it should be noted that in the standard VRP there could be multiple cycles formulated as paths, typically known as subtours, instead of one complete path from origin to destination. These are eliminated by explicitly declaring subtour elimination constraints. However, in our approach it is not required to provide this as Equation 4 prevents subtours.

$$s_{ik} + t_{ij} - s_{jk} \leq M_{ij}(1 - x_{ijk}) \quad \forall i \in (\Phi \backslash m + n + 1), \quad \forall j \in \Phi,\ \forall k \in \nu; \tag{14}$$

where;

$$M_{ij} = \begin{cases} P_{d[i]} + t_{ij} - P_{a[j]}, & \text{if } P_{d[i]} + t_{ij} - P_{a[j]} \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The solution from the above formulation provides the optimal set of routes for the fleet of vehicles. However, as shown later in Section IV, it is achieved at a high runtime, especially as the number of passenger requests grow. Hence, it is viable only when the number of passenger requests are low. Therefore, we are motivated to develop a scalable algorithm providing comparable results to the optimal, but with a low runtime. Thus, in the next subsection we present our greedy local optimization based heuristic algorithm which provides results with comparable accuracy in only several milliseconds.

D. Greedy Local Optimization Heuristic Algorithm

As mentioned, the main objective of the algorithm is to obtain a set of routes to minimize the total VMT. However, exhaustively searching the complete set of solutions to find the optimal solution is time consuming. Thus, we propose a local optimization based greedy heuristic algorithm. Not only does the proposed heuristic execute in several milliseconds but also its' accuracy is comparable to the optimal results. Algorithm 1 depicts the pseudo code of the proposed solution. It consists of two phases, namely *Initialization* and *Execution*. The following sections describe each phase in detail.

- Initialization

During the initialization phase, all passenger requests yet to be served (P) and available vehicles (V) are extracted from the backend database. Next, the cost (c_{ij}) and time (t_{ij}) variables are initialized and populated using the method proposed in [33]. We define variables for minimal distance between a vehicle and each passenger (m_{td}), set of candidate vehicles for each passenger (a_{cv}) and remaining capacity of the currently selected vehicle (m_{cv}). Also, the variables are initialized with values during this phase.

- Execution

As mentioned in Section III-A, we consider the scenario, in which the remaining capacity of the fleet of vehicles exceeds or equals the passenger requests. Therefore, ideally all the passenger requests can be serviced. Initially, the algorithm sorts the passenger requests in the ascending order of the starting time of the pick-up time window ($P_{i[a]}$). Next, for each passenger request, it iterates the fleet of vehicles, validate the constraints (capacity ($k_{rem.capacity}$) and service time (s_{ik})), identify the vehicle/s with the minimum distance to the passenger and populate the set of candidate vehicles. Thus, in each iteration of the problem the algorithm finds a local optimum solution. Once the set of candidate vehicles are populated, a second optimization step is executed. During this step, the algorithm allocates the vehicle with the maximum remaining capacity as the potential candidate vehicle. Finally, the passenger is assigned to the vehicle and remaining capacity, current location and service time are updated with new values. The algorithm, repeats the execution phase for all the passenger requests. The outcome of this process is a set of feasible routes that minimize the total VMT.

IV. Results

In this section, we present the results obtained from the proposed algorithms. *IBM ILOG CPLEX Optimization Studio 12.7.1* was used to implement the MIP formulation, presented in Section III-C, for the exact solution. The greedy heuristic proposed in Section III-D was implemented in C++. Both algorithms were executed on a PC with 8 GB RAM, running Windows 7 on an Intel Xeon E5-1650V2 CPU at 3.5 GHz.

A. Data Sets

In order to test the accuracy and scalability of the proposed solution, we developed 2 test data sets. These data sets are based on realistic data for distance and travel time obtained

Algorithm 1 Local Optimization Algorithm

Input: passenger requests (P), fleet of vehicles (V)
Output: route/schedule of the fleet of vehicles

Initialization:
$m_{td} \leftarrow \infty$
$a_{cv} = \emptyset$
$m_{cv} = 1$

Execution:
sort P in ascending order of $P_{i[a]}$
for $i \in P$ **do**
  for $k \in V$ **do**
    if $k_{rem.capacity} > 1$ **then**
      if $s_{ik} < P_{i[d]}$ **then**
        search minimal distance;
        update a_{cv};
  done
  if $|a_{cv}| > 1$ **then**
    for $u \in a_{cv}$ **do**
      search minimum utilization;
      allocate vehicle;
    done
  assign passenger;
  update vehicle parameters;
  update variables;
done

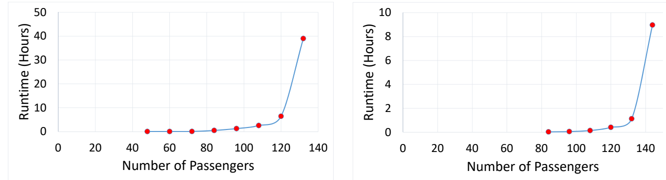
TABLE I: Test Data Parameters

Data Set	DS1	DS2
No of passengers	48 - 132	84 - 144
No of vehicles	4 -11	7 - 12
Max capacity/vehicle	12	12
Vehicle utilization (%)	100	53 - 92
No of test scenarios	8	25

from *Google Maps* [34], for the neighborhood shown in Figure 1. In both data sets, the number of passengers and the number of vehicles are varied while the maximum capacity of a vehicle is maintained as a constant. Dataset DS1 contains 8 possible test scenarios, where we vary the passengers from 48 to 132 in steps of 12. Since, this dataset assumes 100% vehicle utilization, the 8 test scenarios refer to the passenger-vehicle pairs of (48,4), (60,5) ... (132,11). Thus, dataset DS1 represents the scenario when supply equals demand. In contrast, dataset DS2, contains 25 test scenarios, where the vehicle utilization is varied within the range of 53% - 92%, representing scenarios where supply exceeds demand. Table I presents a summary of the data sets.

B. Evaluation Criteria

The evaluation of the algorithms is performed under three criteria; Criteria 1: The scalability of the optimal solution and the proposed algorithm in terms of the runtime, Criteria 2: The performance of the proposed solution in terms of the deviation of results from the optimal values and Criteria 3: The QoS of the proposed solution measured in terms of the average waiting time of a passenger.



(a) DS1 (b) DS2

Fig. 3: Runtime for Optimal Solution

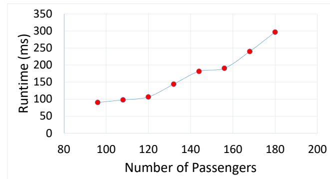


Fig. 4: Runtime for Heuristic Algorithm

- Criteria 1: Scalability

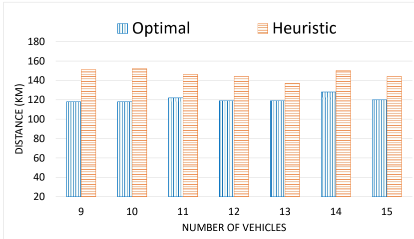
Scalability is measured by the execution time of the proposed algorithms to compute the routes and schedules for all vehicles in the fleet. The runtime is measured for both datasets. Figure 3a shows the runtime of obtaining the optimal solution for the 8 test scenarios of data set DS1. As the number of passenger requests is increased from 48 to 132, the runtime of this algorithm increases from **4 seconds** to **39 hours**. This confirms an exponential increase in the runtime with increasing number of passenger requests. Similarly, Figure 3b shows the average runtime for the 25 test scenarios of data set DS2. Even though, the runtime is reduced when the vehicle utilization constraint is relaxed, the exponential trend is still observed. Therefore, it can be inferred that in a realistic real-time DRT based public transit system, if it is expected that the system should be able to compute the routes and schedules within minutes instead of hours, the number of passenger requests that can be optimally served is limited to only a small number of passengers. On the contrary, as shown in Figure 4 the runtime of the proposed heuristic algorithm scales linearly with the number of passenger requests. The average runtime for the heuristic algorithm for the two data sets is **168 milliseconds**.

- Criteria 2: Performance

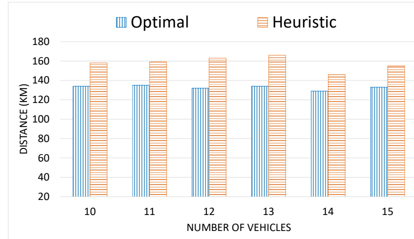
The trade-off between accuracy and runtime is a crucial factor when selecting a suitable algorithm. To this end, criteria 2 assess the accuracy of the proposed algorithm. Identical inputs (DS2) were provided to both the optimal formulation and the heuristic and the output of the two methods are compared. Figure 5 shows the results for the 25 test scenarios of DS2. For clarity, the figures are categorized by the number of passengers in each scenario. The **average deviation** of the proposed solution is **18%**. This concludes that the significantly large runtime penalty incurred by the optimal solution leads only to a marginal deficit in terms of performance.

- Criteria 3: Quality of Service

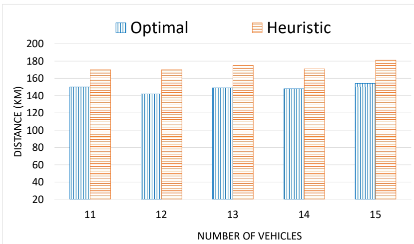
Even though, minimizing the total VMT is the prime performance metric of the algorithm used in this paper, it is also important to consider the waiting time before the ride begins from a passengers' perspective. Hence, the average waiting time is considered as the QoS factor. Equation 16



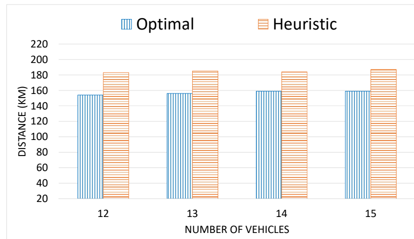
(a) 96 passengers



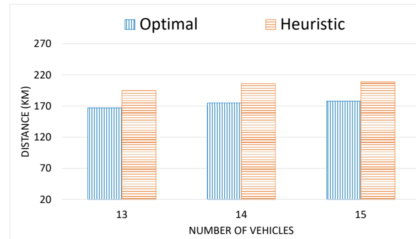
(b) 108 passengers



(c) 120 passengers



(d) 132 passengers



(e) 144 passengers

Fig. 5: Performance: Optimal vs Heuristic

measures the total waiting time (T_{wt}) for all passengers (N) and Equation 17 measures the QoS.

$$T_{wt} = \sum_{i \in \rho} \sum_{k \in \nu} s_{ik} - P_{i[a]} \quad \forall s_{ik} > 0 \tag{16}$$

$$QoS = T_{wt}/N \tag{17}$$

Similar to criteria 2, identical inputs (DS2) were provided to both the optimal formulation and the heuristic to measure the QoS. Results indicate that the average waiting time for a passenger using the optimal formulation is 6.5 min. On the contrary, the proposed heuristic algorithm provides routes with nearly no waiting time.

V. Conclusion

This paper proposes a scalable, greedy local optimization based heuristic algorithm to solve the CDRT problem with large number of passenger requests. The problem is initially formulated as a MIP model and results show that obtaining optimal results is only achievable when the number of passenger requests are relatively low. Next, a heuristic solution is presented that can achieve accurate results in few milliseconds with near perfect quality of service. In future, we will also include other goals in the proposed system such as reducing the average waiting and traveling time for all passengers. Also, we plan to explore the effectiveness of our heuristic algorithm for the last mile problem using a DRT service.

References

[1] E. Osaba et al. An asymmetric multiple traveling salesman problem with backhauls to solve a dial-a-ride problem. In *SAMI*, 2015.

[2] Neil Paulley et al. The demand for public transport: The effects of fares, quality of service, income and car ownership. *Transport Policy*, 2006.

[3] Singapore Ministry of Transport. Fact Sheet - Public Transport Improvements and Future Plans. http://bit.ly/2gZSGpO, 2015.

[4] Alex Gibson. Eliminating Public Transits First-Mile/Last-Mile Problem. http://bit.ly/1ZdbbCj, 2016.

[5] Olivia Greenway. Fixing the first and last mile problem. http://bit.ly/2tWqe9C, 2016.

[6] Susan Carpenter. The last mile to public transit is often the most challenging. http://bit.ly/2ux6JGq, 2016.

[7] David Koffman. Operational experiences with flexible transit services. *TCRP Synthesis of Transit Practice*, 2004.

[8] N. Floudas et al. Public transport optimisation based on traveller requests and network efficiency. In *RTIC*, 2014.

[9] Spencer Woodman. WELCOME TO UBERVILLE. http://bit.ly/2uwXd6e, 2016.

[10] Sand Logan et al. First and Last Mile Connections: New Mobility. http://on.nrdc.org/2utPxUd, 2016.

[11] Peter Lim. GrabShare Vs uberPOOL Which Carpooling App Is Best for Singaporeans? http://bit.ly/2tWudTx, 2016.

[12] Beeline Singapore. https://www.beeline.sg/, 2017.

[13] Final report on the kutsuplus trial: work to develop ride-pooling worth continuing. http://bit.ly/2h01gVz.

[14] Olli Sulopuisto. Why helsinki's innovative on-demand bus service failed. http://bit.ly/1VXDDqu, 2016.

[15] David A. King. What do we know about the first mile/last mile problem for transit? http://bit.ly/2e1nKVk, 2016.

[16] Desheng Zhang et al. Last-mile transit service with urban infrastructure data. *ACM Trans. Cyber-Phys. Syst.*, 2016.

[17] Z. J. Chong et al. Autonomous personal vehicle for the first- and last-mile transportation services. In *CIS*, 2011.

[18] Hasan Poonawala et al. Singapore in motion: Insights on public transport service level through farecard and mobile data analytics. In *KDD*, 2016.

[19] Alexander Kleiner et al. A mechanism for dynamic ride sharing based on parallel auctions. In *IJCAI*, 2011.

[20] Agatz Niels et al. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 2012.

[21] Niels A.H. Agatz et al. Dynamic ride-sharing: A simulation study in metro atlanta. *Transportation Research Part B: Methodological*, 2011.

[22] Jean-Franois Cordeau et al. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 2003.

[23] Sophie N. Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 2011.

[24] Deakin Elizabeth et al. Markets for dynamic ridesharing? *Transportation Research Record: Journal of the Transportation Research Board*, 2010.

[25] Mit ”real-time” rideshare research. http://bit.ly/2tGJaxT, 2009.

[26] Furuhata Masabumi et al. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 2013.

[27] Introduction of on demand bus (english). http://bit.ly/2vWVCq0.

[28] Autonomous infrastructure for cities. http://www.bridj.com/, 2017.

[29] Curt Woodward et al. Bridj, local on-demand bus service, is shutting down. http://bit.ly/2pBRTyh, 2017.

[30] R. Chevrier et al. Comparison of three algorithms for solving the convergent demand responsive transportation problem. In *ITSC*, 2006.

[31] IBM ILOG CPLEX Optimization Studio. https://ibm.co/2vXgzRC.

[32] J.-F. Cordeau and others. The vehicle routing problem. chapter VRP with Time Windows. 2001.

[33] E. S. Rigas et al. Algorithms for electric vehicle scheduling in mobility-on-demand schemes. In *ITSC*, 2015.

[34] Google Maps. http://bit.ly/1bQa31p.