The Value of Prepositioning in Smartphone-Based Vanpool Services Under Stochastic Requests and Time-Dependent Travel Times

Donghui Li
Department of Industrial Engineering
Tsinghua University

Hai Jiang
Department of Industrial Engineering
Tsinghua University
Email: haijiang@tsinghua.edu.cu

Constantinos Antoniou
Department of Civil, Geo and Environmental Engineering
Technical University of Munich

Qianyan Xie
Research and Advanced Engineering
Ford Motor Company

Wei Shen
Asia Pacific Research
Ford Motor Company

Weijian Han
Research and Advanced Engineering
Ford Motor Company

1 Introduction

Smartphone-based vanpool services have become an emerging trend, which has attracted much attention in the research field [1, 2], and have been made available around the world. In smartphone-based vanpool services, passengers request vans from their smartphones for point-to-point pickups and deliveries, and vans are dynamically routed to passengers with committed pickup and delivery times. One of the fundamental issues in smartphone-based vanpooling is to schedule a fleet of vans to serve passengers efficiently, which corresponds to the classical dial-a-ride problem (DARP) and its many variants have been investigated extensively in the literature [2]. The dynamic and stochastic DARP is the most realistic one for dynamic vanpool services since it considers both dynamic requests and future stochastic information [3, 4].

One of the major limitations of existing literature is that when a vehicle is dispatched, it is limited to pickup and delivery locations of received requests. The operator would not preposition a vehicle to locations where future demand may appear. FIGURE 1 is an example to demonstrate the benefits of prepositioning. Assume that the current time is 10:00, we have one van available in this area, and the van is now empty. We have received two requests, one is from A to A', and the other one is from B to B'. Suppose there is a high probability that we will receive a request from C to C' in the next 10 minutes. Due to the limit of waiting time, we can only choose one of these requests under current traffic conditions. Existing literatures only consider sending the van to A or B, and use the stochastic information of future requests and traffic conditions to evaluate these decisions. If prepositioning is considered, we will also consider sending the van to C. This decision can lead to high profit, when the probability of receiving this request is high.

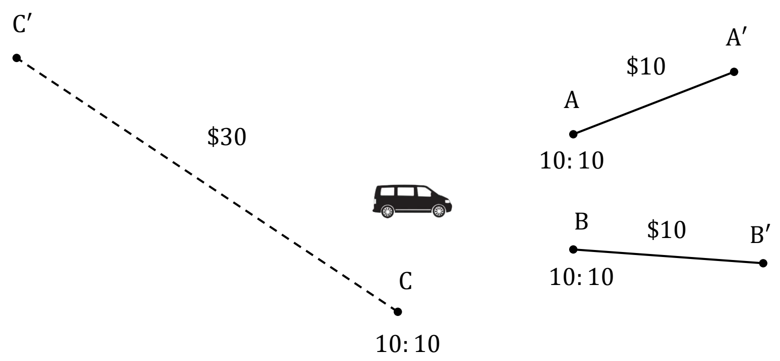

Figure 1: An example to demonstrate the benefits of prepositioning.

We develop a metaheuristic scheduling algorithm for the dynamic and stochastic DARP. The algorithm uses multiple scenarios which include future requests and traffic conditions to generate and evaluate each potential decision. Prepositioning is considered in this algorithm. We use a real dataset, which includes requests from Pandabus in Dalian and travel speed data achieved from a map service provider, to test our algorithm. The results show that incorporating stochastic requests without considering prepositioning can improve the average profit by 18.6%. And the prepositioning improves the average profit by 23.8% and reduces the average waiting time by 74.7%.

2 Problem Description

Our research problem comes from Pandabus, which operates a pilot dynamic vanpool service in Dalian, China. They use several vans to provide transportation services for passengers within a service area. They have a smartphone app for passengers to send requests. Each request has a pickup location, a delivery location, and expected pickup time. They also have a smartphone app for drivers to receive scheduled routes and locations of requests.

They need an online scheduling algorithm to decide whether to accept each request and how to route each van. We model this problem as a DARP. $R = \{r_1, r_2, \cdots\}$ is the set of requests, which is being updated during the operation to include newly received requests. Each request r_i has its pickup node r_i^p, delivery node r_i^d, status r_i^s

which can be new, rejected, waiting for pickup, picked, or delivered, route r_i^v which is the index of the van picking this request. Each request r_i has a pickup time window $[e_{r_i^p}, l_{r_i^p}]$ and a delivery time window $[e_{r_i^d}, l_{r_i^d}]$. We use the pickup time window to ensure passengers' expected pickup time and use the delivery time window to limit maximum detour. For the pickup time window, we set $e_{r_i^p}$ as passenger's expected pickup time, and $l_{r_i^p} = e_{r_i^p} + u_w$, where u_w represents the maximum allowed waiting time. For the delivery time window, we set $e_{r_i^d} = e_{r_i^p}$, and $l_{r_i^d} = l_{r_i^p} + u_d DTT(r_i^p, r_i^d)$, where u_d represents the maximum allowed detour ratio and $DTT(r_i^p, r_i^d)$ is the direct travel time between the pickup and the delivery node under average travel speed.

We use scenarios to represent the stochastic information about future requests and traffic conditions. $S^r(t) = \{s_1^r(t), s_2^r(t), \cdots\}$ is the set of request scenarios we use at time t. $S^s(t) = \{s_1^s(t), s_2^s(t), \cdots\}$ is the set of travel speed scenarios we use at time t. $S(t) = \{s_1(t), s_2(t), \cdots\}$, where $s_k(t) = \langle s_k^r(t), s_k^s(t)\rangle$, is the set of scenarios we use at time t, which combines request scenarios and travel speed scenarios. The goal of this research is to design an online scheduling algorithm to optimize the operating profit and the user experience. The operating profit is calculated as the operating cost minus the service revenue. The user experience includes the waiting time and the detour. In our implementation, the objective function is a linear combination of the cost, the revenue, the waiting, and the detour.

3 Solution Methods

To develop and test the scheduling algorithm, we need a simulation framework which provides a simulated online environment. It simulates the arrival of new requests, movement of vans and calls the scheduling procedure when needed, such as when new requests arrive or after a given time interval. The scheduling procedure uses scenario-based approaches to decide whether to accept each new request and design schedules for each van. For each given scenario, we need to solve a deterministic problem and this is done by a tabu search algorithm. In this abstract, we concentrate on the scheduling procedure. The scheduling procedure gets the following inputs: (1) the set of vans V which includes the position of each van; (2) the set of requests R which includes newly received requests and accepted requests with their status, such as, whether the request is picked or not, which van the request in on. The scheduling procedure first decides whether to accept each new request, then decides the routes of each van.

In the first step, to decide whether to accept each request, the brief idea is to compare the expected objective function value when accepting the request with the one when rejecting the request. To achieve this goal, we develop an evaluation procedure as shown in Figure 2. In this evaluation procedure, we need to input the current state of vans and requests, $state(t)$. The evaluation procedure estimates the average objective function value of current state. It loops through each scenario $s_k(t) \in S(t)$. With a given scenario $s_k(t)$, the stochastic problem becomes a deterministic problem. Tabu search is used to solve the deterministic DARP under the given state and scenario, which gives an optimal objective function value $obj_k(t)$ under each scenario. We use the average value of these objective function values under different scenarios to represent the expected objective function value of current state. With this evaluation procedure, we first mark the request as rejected and use the evaluation procedure to evaluate the average objective function value under given state, denoted as $obj_{rejected}$. Then we mark the request as accepted and insert it into a random route, run the evaluation procedure again, and get $obj_{accepted}$. If $obj_{rejected} > obj_{accepted}$ we reject the request, otherwise we accept it.

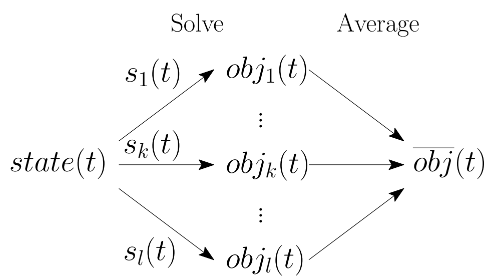

Figure 2: Illustration of the evaluation procedure.

In the second step, to decide the route of each van, we develop a scenario-based search to generate and evaluate potential decisions. The main idea is demonstrated in Figure 3. Similarly, the $state(t)$ represent the current state of the system. In each iteration of the loop, we select a scenario $s_k(t) \in S(t)$. We can use tabu search to find the optimal decision for the given state and scenario. For each scenario $s_k(t)$ we can get an optimal decision, denoted as $decision_k(t)$, which we call a candidate decision. Because $s_k(t)$ includes potential future requests, in some candidate decisions, the vans may be dispatched to future requests if this can lead to a better solution. After generating candidate decisions, we need to evaluate these decisions and choose a final decision. For each $decision_k(t)$, we first update the state according to the decision. By doing this, we get a new state $state_k(t + \Delta t)$ which represents the consequence of executing the decision. Then we use the evaluation procedure to loop through scenarios again to get an expected objective function value $\overline{obj}_k(t+\Delta t)$ of the candidate decision. Finally, we choose the candidate decision with the best expected objective function value as our final decision.

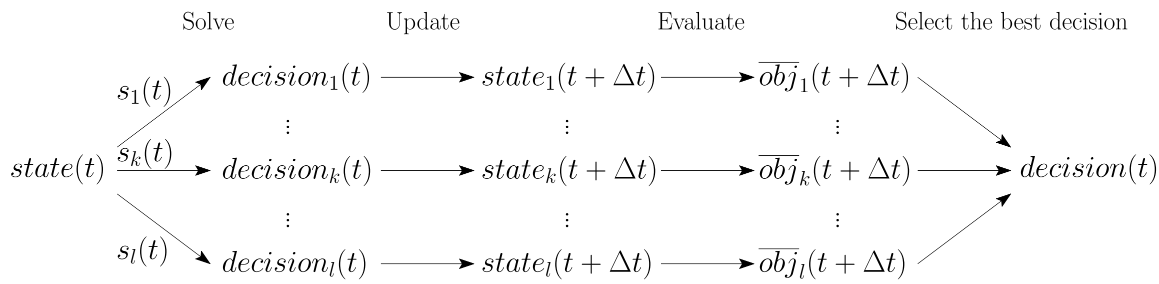

Figure 3: Illustration of the scenario-based search.

4 Numerical Experiments

Our request dataset comes from Pandabus. The dynamic vanpool service is run in a 5 km $\times$ 5 km service area which includes business and residential areas. We have historical requests in September 2016, which includes 20 work days. There are 4 vans running in this area, each van has 17 seats. The average number of daily requests is 163.5 and the average travel distance between pickup and delivery locations is 3.17 km. In the following tests. If it is not stated, we set the ticket price as 1 Chinese Yuan / km, the fuels cost as 0.5 Chinese Yuan / km, which are estimated from actual service price and cost. We set the maximum allowed waiting as 10 min and the maximum allowed detour ratio as 1.4, which are estimated according to the current operation.

We use multiple scenarios to represent the stochastic distributions of future requests and travel times. These scenarios can be generated according to forecasting models. Since this research does not focus on requests and travel time forecasting, we implement a simple method to generate scenarios. Each scenario is generated directly from the data of one historical day. For example, travel speed scenario $s_k^s(t)$ is the travel speed of one historical day. For request scenario $s_k^r(t)$, we use a parameter H to limit the horizon of future requests to guarantee the running performance. $s_k^r(t)$ contains all requests received between $(t, t + H]$ during one historical day. The algorithm is implemented with C++ and tested on a computer with i7-4790K 4.0GHz CPU and 16GB RAM.

We use the following metrics to measure the scheduling performance: (1) daily profit which is the difference between daily revenue from ticket price and daily cost of fuels, (2) average waiting time which is the average value of each passenger's waiting time between expected pickup time and actual pickup time, and (3) average detour ratio which is the average value of each passenger's detour ratio, where the detour ratio is the ratio between actual travel time and average direct travel time between the pickup and delivery locations.

Both stochastic requests and travel times are considered in this research. We set up multiple tests to measure the benefits of considering stochastic requests in our algorithm. The improvements of considering stochastic requests in this research can be divided into these two aspects: the first one is stochastic, which means that we use multiple scenarios with future requests to generate and evaluate candidate decisions; and the second one is prepositioning, which means that we consider sending the vans to locations with potential requests. We set up the following tests to inspect the benefit of these two aspects respectively: Test 1, deterministic, we set the horizon of scenarios to zero, which means we do not consider future requests; Test 2, stochastic, we set the horizon of scenarios to 15 min and only consider dispatching vans to received requests; and Test 3, prepositioning, we set the horizon of scenarios to 15 min and consider sending the vans to locations with potential requests. Stochastic time-department travel times are considered in all of these tests.

We run these tests with perfect request scenarios. Perfect request scenarios mean that the request scenarios contain the real requests from the corresponding date. For example, in the test of 2016-09-01, the scenarios are generated from the historical data of this date. We use this way to demonstrate the scheduling performance when we have very accurate forecast. From the average differences of these tests, we can find that considering future requests (without prepositioning) improves the average profit by 18.6%. And the prepositioning improves the average profit by 23.8% and reduces waiting time by 74.7%. However, we may not have perfect forecast in the practice. Thus we run tests with scenarios generated from two historical days. From the average differences, we can find that considering future requests (without prepositioning) improves the average profit by 14.0%. And the prepositioning improves the average profit by 10.1% and reduces waiting time by 55.9%.

We also do similar tests for stochastic time-dependent travel times. However, the performance of these tests does not have significant differences. These may be because the traffic conditions in the service area are good and considering stochastic time-dependent travel times does not have significant improvements under this circumstance.

5 Conclusion

In this research, we develop a scheduling algorithm for dynamic vanpool services considering both stochastic requests and stochastic time-dependent travel times using scenario-based search and tabu search. Prepositioning is considered in the algorithm. We use a real dataset which includes requests from our partner in Dalian and travel speed data achieved from a map service provider. The results show that incorporating stochastic requests can improve the solution quality significantly, especially that prepositioning can increase profit and reduce waiting time significantly.

References

[1] M Grazia Speranza. Trends in transportation and logistics. *European Journal of Operational Research*, 264(3):830–836, 2018.

[2] Sin C Ho, WY Szeto, Yong-Hong Kuo, Janny MY Leung, Matthew Petering, and Terence WH Tou. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 2018.

[3] Michael Schilde, Karl F Doerner, and Richard F Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research*, 38(12):1719–1730, 2011.

[4] Michael Schilde, Karl F Doerner, and Richard F Hartl. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research*, 238(1):18–30, 2014.

[5] Suleyman Karabuk. A nested decomposition approach for solving the paratransit vehicle scheduling problem. *Transportation Research Part B: Methodological*, 43(4):448–465, May 2009.

[6] Fabien Lehud, Renaud Masson, Sophie N. Parragh, Olivier Pton, and Fabien Tricoire. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society*, 65(7):983–1000, July 2014.

[7] Thomas Hanne, Teresa Melo, and Stefan Nickel. Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces*, 39(3):241–255, 2009.

[8] Sophie N Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19(5):912–930, 2011.

[9] Sophie N Parragh, Jorge Pinho de Sousa, and Bernardo Almada-Lobo. The dial-a-ride problem with split requests and profits. *Transportation Science*, 49(2):311–334, 2014.

[10] Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[11] Jean-François Cordeau and Gilbert Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594, 2003.

[12] Sophie N Parragh and Verena Schmid. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 40(1):490–497, 2013.

[13] Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. Dynamic programming based metaheuristics for the dial-a-ride problem. *Annals of Operations Research*, 236(2):341–358, 2016.

[14] Ying Luo and Paul Schonfeld. Online rejected-reinsertion heuristics for dynamic multivehicle dial-a-ride problem. *Transportation Research Record: Journal of the Transportation Research Board*, (2218):59–67, 2011.

[15] Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*, 24(3):343–355, 2012.