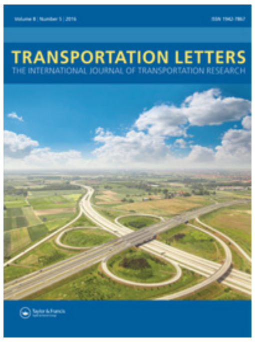


Taylor & Francis
Taylor & Francis Group

Transportation Letters

The International Journal of Transportation Research

ISSN: 1942-7867 (Print) 1942-7875 (Online) Journal homepage: https://www.tandfonline.com/loi/ytrl20

A pickup and delivery problem for ridesharing considering congestion

X. Wang, M. Dessouky & F. Ordonez

To cite this article: X. Wang, M. Dessouky & F. Ordonez (2016) A pickup and delivery problem for ridesharing considering congestion, Transportation Letters, 8:5, 259-269, DOI: 10.1179/1942787515Y.0000000023

To link to this article: https://doi.org/10.1179/1942787515Y.0000000023

Published online: 29 Feb 2016.

A pickup and delivery problem for ridesharing considering congestion

X. Wang[1], M. Dessouky[1] and F. Ordonez[2]

Traffic congestion is a significant social concern that is credited with considerable economic costs, wasted time, and associated public health risks. Efficient ridesharing solutions could help mitigate congestion. Some of the actions that government agencies have taken to encourage ridesharing include the availability of high occupancy vehicle (HOV) lanes and existing policies of discounted toll rates on HOVs. These measures encourage ridesharing by reducing costs or travel times of such trips. To study how the optimal routes change as a function of incentives for ridesharing, we modified existing pickup and delivery problems with time windows to consider changes in passenger travel time and toll cost due to vehicle load. Our computational results explore how the total route cost and time are affected by the use of HOV lanes and toll savings. In addition, our results show that it can be beneficial from a time and cost perspective to take detours to pick up additional passengers and use HOV lanes when the time saving on HOV lanes is significant.

Keywords: Ridesharing, Pickup and delivery, Time windows, Insertion, Tabu search

Introduction

Traffic congestion can be seen in some cases as evidence of social vitality. However, it is also a significant social concern that is credited with significant economic costs, wasted time, and associated public health risks. The 2012 Urban Mobility Report (Schrank, Eisele and Lomax 2012) states that, in 2011, the total cost of congestion is $121 billion in the US, and the total amount of delayed time is 5.5 billion hours with an extra usage of 2.9 billion gallons of fuel. With the expected population growth figures, for 2020, it is expected that the cost of congestion will rise to $199 billion, and the total delay is estimated to increase to 8.4 billion hours with an extra fuel consumption of 4.5 billion gallons. Moreover, the Harvard Center for Risk Analysis (HCRA) at the School of Public Health conducted a research study in 83 urban areas to evaluate the public health impacts of traffic congestion (Levy, Buonocore and Von Stackelberg 2010). These results indicate that traffic congestion led to 4000 premature deaths with a public health cost of around $31 billion in 2000.

An increased use of ridesharing has the potential to help mitigate congestion in particular, because there is a significant amount of unused capacity in vehicles on the road today. Indeed, the average vehicle occupancy rate in the US in 2009 was 1.67; this number drops to 1.13 for work commute trips (Santos *et al.* 2011). "Ridesharing is a joint-trip of more than two participants who share a vehicle and requires coordination with respect to itineraries" (Furuhata *et al.* 2013). By taking advantage of the vacant seats in most passenger vehicles, ridesharing could increase the efficiency of the transportation system, reduce traffic congestion, decrease fuel usage, and mitigate pollution (Agatz, Erera, Savelsbergh and Wang 2012). Historically, people have participated in ridesharing by posting their itinerary information on a bulletin board or a website like Craigslist so that others can find a match either manually or automatically. A good ridesharing system should provide automated matching, which means that the system should actively help drivers and riders find suitable matches (Agatz *et al.* 2012). The matching between the drivers and riders in ridesharing can be viewed as a pickup and delivery problem (PDP) with time windows. Recently, there have been a plethora of companies such as Carma (formerly known as Avego) and Sidecar that have developed technologies to help match drivers with passengers (Furuhata *et al.* 2013).

At the same time, government agencies have taken actions to encourage ridesharing. There is an increasing use of high occupancy vehicle (HOV) lanes and reduced toll rates for HOVs on many roads and bridges. For example, the New Jersey Turnpike charges a discounted toll rate to vehicles, which have three or more people, the I-15 Express Lanes in San Diego, California is free for carpooling and vanpooling, and the HOV lanes of I-110 Freeway in Los Angeles County were converted to high occupancy toll (HOT) lanes, which are also free for HOVs. If a vehicle has the required number of people, then HOV lanes can be used to reduce the travel time especially during peak hours. Therefore, ridesharing could provide cost reduction and time saving under congestion.

To study how the optimal routes change as a function of incentives for ridesharing, for example, inclusion of HOV lanes, we modified existing PDP with time windows to consider

[1] Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA, 90089-0193, USA,
[2] Industrial Engineering Department, Universidad de Chile, Republica 701, Santiago, Chile

*Corresponding author, email maged@usc.edu

Received 28 May 2014 Accepted 07 September 2015
DOI 10.1179/1942787515Y.0000000023

changes in passenger travel time and toll cost due to vehicle load. Authors assume that requests not served by the given vehicles will be serviced by an outside provider such as a taxi service. Although there are a number of studies and published methods for the PDP with time windows, to the best of our knowledge there is no previous work that considers HOV lanes and the policy of reduced toll rates on HOVs, which will save both travel time and cost by having vehicles with more passengers. That is, there may be an incentive to take detours to pick up additional passengers to qualify for HOV lanes or discounted toll rates. Each driver participating in ridesharing provides his/her start location, end location, earliest departure time, and latest arrival time. Each ride request provides their start location, end location, time windows for pickup and delivery, and the number of people who need to be served. Authors consider the static version of this problem in which all passenger requests are known in advance.

In this work, we used a heuristic algorithm to efficiently solve this specialized PDP. The heuristic uses a greedy insertion to obtain an initial solution. A Tabu procedure is applied to obtain improvements and an adjustment on the pickup time is made to reduce the ride time of each request. Authors also present a full integer programing formulation of the problem to benchmark the heuristics on small problem sizes. The rest of the paper is organized as follows. In Section 2, a literature review of the PDP is presented. Section 3 describes the problem formulation. The heuristic algorithm is proposed in Section 4. Section 5 reports the experimental results. Conclusions are presented in Section 6.

Literature review

The most closely related routing problem to ridesharing is the PDP, which is a generalization of vehicle routing problems (VRP) in which objects or people have to be transported between origins and destinations (Berbeglia, Cordeau, Gribkovskaia and Laporte 2007). Since the VRPs are proved to be NP-hard (Lenstra and Rinnooy Kan 1981), the PDPs are known to be NP-hard. The PDPs are classified into three groups: the many-to-many problem, the one-to-many-to-one problem, and the one-to-one problem (Berbeglia, Cordeau and Laporte 2010). Ridesharing can be modeled as a one-to-one PDP.

In the one-to-one PDP, each object has a pickup and a delivery location. For example, the dial-a-ride transportation service is of this type (Cordeau and Laporte, 2007). The problems can be further categorized into two groups: single vehicle and multi-vehicle problems (Cordeau, Laporte and Ropke 2008). For the single vehicle problem, dynamic programing has been used to optimally solve the problem. Psaraftis (1980) carried out one of the first studies, which worked on the immediate-request case, which was solved optimally by dynamic programing for small instances. Later, Psaraftis (1983) extended the algorithm to solve the problem with hard time windows. Desrosiers, Dumas and Soumis (1986) applied dynamic programing to solve the problem on larger instances. There are other studies on the problem, which might not solve the problem optimally but efficiently. Hosny and Mumford (2010) presented a heuristic based on an intelligent neighborhood move guided by the time window to solve the problem.

For the multi-vehicle problem, the branch-and-cut algorithm has successfully been used to optimally solve small problem instances. Lu and Dessouky (2004) developed an exact algorithm for the PDP. A branch-and-cut algorithm is developed to optimally solve the integer-programing formulation of the problem. Cordeau (2006) and Ropke, Cordeau and Laporte (2007) provided an alternative formulation and branch-and-cut algorithm to optimally solve the problem. Later, Ropke and Cordeau (2009) introduced a new branch-and-cut-and-price algorithm, which, according to the computational experiments, outperformed the branch-and-cut algorithm of Ropke *et al.* (2007).

Moreover, heuristics are developed to effectively solve large instances of the problem. Jaw, Odoni, Psaraftis and Wilson (1986) developed a heuristic procedure in which users can only specify either the pickup time or the delivery time. Potvin and Rousseau (1992) modified the insertion part of the heuristic proposed in Jaw *et al.* (1986) and added two new phases and obtained better results than that of Jaw *et al.* (1986). Diana and Dessouky (2004) presented a parallel regret insertion heuristic. Lu and Dessouky (2006) presented a new insertion-based construction heuristic, which considers the cost of reducing the time window slack due to the insertion as well as the classical incremental distance measure. Xiang, Chu and Chen (2006) proposed a fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints by applying insertions, inter-route exchanges, and secondary objective to provide diversification. Wong and Bell (2006) proposed a heuristic including parallel insertions, reinsertions, and exchanges. Some heuristics solve the problem by clustering the users first according to a proximity relation. Roy *et al.* (1983) applied the cluster-first-route-second methodology. As a modification of the methodology, Dumas, Desrosiers and Soumis (1989) created mini-clusters, combining by using column generation and applying a scheduling phase. Desrosiers *et al.* (1991) used a parallel insertion method based on the heuristic of Dumas *et al.* (1989). Later, Ioachim *et al.* (1995) applied an optimization technique in creating mini-clusters and produced better results than that of Desrosiers *et al.* (1991). Bard and Jarrah (2009) presented a three-phase procedure for clustering a large number of data points with configuration and resource constraints.

There also has been work on metaheuristics to tackle PDPs. Tabu search has been one of the most commonly used metaheuristics for solving routing problems (Cordeau and Laporte 2005). Nanry and Barnes (2000) presented a reactive Tabu search approach to solve the PDP with time windows. Li and Lim (2001) proposed a Tabu-embedded simulated annealing algorithm. Cordeau and Laporte (2003) described a Tabu search heuristic and proposed a procedure for neighborhood evaluation that adjusts the visit time of the vertices on the routes so as to minimize route duration and ride times. Moreover, there are many other metaheuristics applied to the PDP. Sombuntham and Kachitvichyanukul (2010) used particle swarm optimization for the multi-depot PDP. Parragh, Doerner and Hartl (2010) proposed a competitive variable neighborhood search-based heuristic for the static multi-vehicle dial-a-ride problem, which allows intermediate deteriorating moves. Catay (2009) and Carabetti, de Souza, Fraga and Gama (2010) applied ant colony optimization to the PDP.

In summary, the main distinction between this paper and the previous papers is that we consider load-dependent travel time and toll cost and in the experimental section, we explore the sensitivity of this dependence on the vehicle tours.

Model formulation

Authors formulate a 0–1 integer programing model for optimally solving a vehicle PDP with time windows considering time savings and discounted toll rates based on the number of people in the vehicle. It was assumed that there is one ridesharing vehicle serving n requests, and each request has a pickup and delivery location. Both the pickup and delivery have time windows. The time window of a request is always feasible, which means a direct route can always be used to satisfy the request. The requests that are not served by the vehicle will be served by an outside provider, which we refer to as taxi service. For simplicity in the description below, we assume a single vehicle. The formulation for multiple vehicles is the standard generalization of this model.

Authors follow the formulation of Lu and Dessouky (2004) as the basis for our formulation. The problem can be defined on a directed graph G=(N, A). Let N be the node set, N= {1, 2…2n+2}, and we use index $i \in (N$ to denote node i. Let H be the request set, H= {1, 2…n}, where $h \in H$ corresponds to the h-th request

$$i = 1, ..., n \quad \text{request } h'\text{s pickup locationwhen } i = h$$

$$i = n+1, n+2, ..., 2n \text{ request } h'\text{s deliverylocationwhen } i = h + n$$

$$i = 2n + 1 \text{ driver's end location}$$

$$i = 2n + 2 \text{ driver's end location}$$

Let N_P denote the set of pickup nodes. N_P= {1, 2…n}. Let N_D denote the set of delivery nodes. N_D= {n+1, n+2…2n}.

Let A be the arc set. The time and cost associated to each arc $(i, j) \in A$ depends on the number of people in the vehicle.

Parameters:

R_h pickup demand (number of passengers) of request h, $h \in H$

$$G_i = \begin{cases} R_h & i \in N_P,\ h = i \\ -R_h & i \in N_D,\ h = i - n \\ 0 & \text{otherwise} \end{cases}$$

Ca, vehicle capacity

$S_{i,n+i}$, travel time from node i to n+i using a taxi

E_i, the earliest time passenger can be picked up or delivered at node i

L_i, the latest time passenger can be picked up or delivered at node i

O, the number of people in the vehicle at the driver's start location

Y_{ij}, minimum travel time from node i to j

T_{ijk}, travel time from node i to j if there are k people in the vehicle

C_{ijk}, toll cost from node i to j if there are k people in the vehicle

D_{ijk}, travel distance from node i to j if there are k people in the vehicle

Variables

$$x_{ij} \begin{cases} 1 & \text{if the vehicle travels from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

$$u_{ij} \begin{cases} 1 & \text{if node } i \text{ is visited byataxi} \\ 0 & \text{otherwise} \end{cases}$$

$$b_{ij} \begin{cases} 1 & \text{if node } i \text{ is before node } j \text{ in thetour ofthe vehicle} \\ 0 & \text{otherwise} \end{cases}$$

v_i, the time at which a passenger is picked up or delivered at node i

z_i, the number of people in the vehicle after serving node i

t_{ij}, actual time from node i to j

c_{ij}, actual toll cost from node i to j

d_{ij}, actual distance from node i to j

The parameters, β, γ, μ, and λ, represent the weights for the different objective components, total travel time, distance, toll fee, and whether the request was serviced by taxi, respectively. The values of the T_{ijk}, D_{ijk}, and C_{ijk} parameters are data for the problem and can be determined by the travel time, cost, and distance observed on the optimal route sending k passengers from i to j using parameters β, μ, and γ in the objective function. Note that the weight λ could be indexed in the pickup node to represent the actual taxi cost for that trip. In this work, however, we consider a uniform, large, weight λ to prioritize servicing as many requests as possible by the ridesharing vehicles instead of the use of a taxi-like service.

Problem formulation:

Minimize

$$\sum_{i \in N_p} \beta * \left(v_n + \mathrm{i} - v_i\right) + \sum_{(i,j) \in A} \left(\gamma * d_{ij} + \mu * C_{ij}\right) + \sum_{(i,n+i) \in A} \lambda * u_i$$

Subject to

$$\sum\nolimits_{i \in N} x_{ij} + u_i = 1 \; j \in N \backslash \{2n+2\} \tag{1}$$

$$\sum\nolimits_{i \in N} x_{ij} + u_j = 1 \; j \in N \backslash \{2n+1\} \tag{2}$$

$$u_i = u_n + \mathrm{i} \; \mathrm{i} \in N_P \tag{3}$$

$$b_{ki} \le b_{kj} + \left(1 - x_{ij}\right) \; (i,j) \in A \backslash \{2n+2, 2n+1\} \; k \in N \backslash \{i\} \tag{4}$$

$$b_{ki} \le b_{ki} + \left(1 - x_{ij}\right) \; (i,j) \in A \backslash \{2n+2, 2n+1\} \; k \in N \backslash \{i\} \tag{5}$$

$$b_{ki} + u_i \le 1 \; i \in N \backslash \{2n+1, 2n+2\} \; k \in N \backslash \{i\} \tag{6}$$

$$b_{ik} + u_i \le 1 \; i \in N \backslash \{2n+1, 2n+2\} \; k \in N \backslash \{i\} \tag{7}$$

$$b_{i,n+i} + u_i = 1 \; i \in N_P \tag{8}$$

$$x_{ij} \le b_{ij} \; (i,j) \in A \tag{9}$$

$$b_{ii} = 0 \; i \in N \tag{10}$$

$$b_{i,n+i,i} 0 \;\; i \in N_P \tag{11}$$

$$b_{2n+2n+i} 0 \;\; i \in N \tag{12}$$

$$b_{2n+2,i}0 \quad i \in N \tag{13}$$

$$z_i + (G_i + 0) * (G_i + u_i) + \sum_{m \in N} (b_{mi} * G_m) \quad i \in N \tag{14}$$

$$z_i \leq \text{Ca } i \in N \tag{15}$$

$$v_i + S_{i,n+i} \leq v_n + \mathrm{i} + (1 - u_i) * M \ i \in N_P \tag{16}$$

$$v_i + t_{ij} \leq v_j + (1 - x_{ij}) * M \ i \in N \ j \in N \tag{17}$$

$$E_i \leq v_j \leq L_i \ i \in N \tag{18}$$

$$t_{ij} \geq T_{ijk} - |z_i - k| * M - (1 - x_{ij}) * M \ i \in N \ j \in N \, k = 1, 2 \ldots \text{Ca} \tag{19}$$

$$c_{ij} \geq C_{ijk} - |z_i - k| * M - (1 - x_{ij}) * M \ i \in N \ j \in N \, k = 1, 2 \ldots \text{Ca} \tag{20}$$

$$d_{ij} \geq D_{ijk} - |z_i - k| * M - (1 - x_{ij}) * M \ i \in N \ j \in N \, k = 1, 2 \ldots \text{Ca} \tag{21}$$

$$x_{ij}\{0, 1\} \ (i,j) \in A \tag{22}$$

$$u_i\{0, 1\} \ \in N_P \tag{23}$$

$$b_{ij}\{0, 1\} \ (i,j) \in A \tag{24}$$

$$z_i \in Z \, i \in A \tag{25}$$

$$v_i \geq 0 \ i \in A \tag{26}$$

$$t_{ij} \geq 0 \ (i,j) \ i \in A \tag{27}$$

$$c_{ij} \geq 0 \ (i,j) \ i \in A \tag{28}$$

$$d_{ij} \geq 0 \ (i,j) \ i \in A \tag{29}$$

The objective is to minimize the total passenger ride time, total travel distance of the vehicle, the total toll fee, and the total taxi service cost; if there are some requests that cannot be fulfilled by the ridesharing vehicle then a taxi will be used to satisfy these requests. In the objective function, the first term is the total passenger ride time with a weight of β. The value (v_{n+1}(v_i) is the ride time of passenger *i*. The second term is the total cost of travel distance and toll fee. The summation of d_{ij} on all edges is the total travel distance for the vehicle with a weight of γ. The summation of c_{ij} on all edges is the total toll fee with a weight of μ. The third term is the total taxi cost with a weight of λ since the passengers who cannot be served by the vehicles will be served by a taxi

Constraints (1) and (2) ensure that each location is visited only once either by the vehicle or the taxi. Constraint (3) ensures that if the pickup location of the request is visited by a taxi, the delivery location of the request would also be visited by a taxi. Constraints (4)–(13) and (24) characterize variable b_{ik}, which equals 1 if node *i* is before node *k* in the tour and 0 otherwise. In particular, constraints (4) and (5) ensure that all nodes *k* that are before node *i* are also before node *j*, and vice versa if a vehicle travels from *i* to *j* (x_{ij} = 1). Constraints (6) and (7) remove node *i* from routes if it is visited by a taxi. That is, node *i* is not before or after any other node *k* in this case. Constraint (8) ensures that every pickup node *i* is either visited by a taxi or before its delivery node *n*+*i* on a tour. Constraint (9) forces *i* to be before *j* on a tour if a vehicle travels from *i* to *j*. Constraints (10)–(13) define some basic relations: no node is before itself on a tour, no drop off node is before its pick up node, no node is before the starting node, and finally the ending node is not before any node. The parameter b_{ik} is also used in the same way in the paper of Lu and Dessouky (2004).

Constraint (14) sets z_i to the number of people in the vehicle after serving node *i*. ($_{m \in N}$(b_{mi}*G_m) is the number of people picked up and have not been dropped off before node *i*. G_i is the number of people that need to be either picked up or dropped off at node *i*. The drop-offs have a negative value while the pickups have a positive value. *O* is the number of people in the vehicle at the driver's start location. Constraint (15) is to ensure that the capacity constraint of the vehicle is not violated.

Constraint (16) ensures the consistency of the time variables if the node is visited by a taxi. *M* is a big number to ensure that the inequality will always hold when u_i = 0. Actually, we could get rid of this constraint by performing a post-processing phase. Because of the minimization of the total passenger ride time term in the objective function, without constraint (16), for the taxi-served passenger (u_i = 1, $u_{n(i}$ = 1), the value of v_i and $v_{n(i}$ will be L_i and $E_{n(i}$, respectively. Constraint (17) ensures the consistency of the time variables if the node is visited by the vehicle. *M* is a big number to make sure that the inequality will always hold when x_{ij} = 0. Constraint (18) is the time window constraint for each node.

When x_{ij} = 0, constraints (19)–(21) will always hold. When x_{ij} = 1, constraint (19) corresponds to the following set of constraints

$$t_{ij} \geq T_{ij1} - |z_i - 1| * M$$
$$t_{ij} \geq T_{ij2} - |z_i - 2| * M$$
$$t_{ij} \geq T_{ij3} - |z_i - 3| * M$$

If z_i is not equal to *k*, then t_{ij} will be larger than a very small number. If z_i equals to *k*, t_{ij} will have a tighter constraint, which is $t_{ij} \geq T_{ijk}$. Since we want to minimize $v_{n(i}$(v_i, then it would make $t_{ij} = T_{ijk}$. Constraints (20) and (21) are similar to constraint (19) and give the relations $c_{ij} \geq C_{ijk}$ and $d_{ij} \geq D_{ijk}$ if z_i is equal to *k* and x_{ij} is equal to 1. The minimization will make $c_{ij} = C_{ijk}$ and $d_{ij} = D_{ijk}$. However, constraints (19)–(21) are non-linear. Authors use a standard method to linearize it. For example, constraint (19) is transformed to the following two inequalities where *P* is a big number and h_{ik} is binary

$$t_{ij} \geq T_{ijk} - (z_i - k) * M - (1 - x_{ij}) * M - P * h_{ik}(i,j) \in A \ \ k = 1, 2 \ldots \text{Ca}$$
$$t_{ij} \geq T_{ijk} + (z_i - k)^* M - (1 - x_{ij})^* M - P^* (1 - h_{ik})(i,j) \in A \ \ k = 1, 2 \ldots \text{Ca}$$

Constraints (22)–(24) are the binary constraints for the variables x_{ij}, u_i, and b_{ij}, respectively. Constraints (25) set z_i to an integer value. Constraints (26)–(29) are the non-negativity constraints for the variables v_i, t_{ij}, c_{ij}, and d_{ij}, respectively.

To help strengthen the computational capability of the model, we now describe some valid equality and inequalities as follows. These constraints are certainly redundant to the model. However, their effectiveness will be shown in the experimental results section

$$v_{\mathrm{n+i}} + v_{\mathrm{i}} \geq Y_{\mathrm{ij}} \quad i \in N_{\mathrm{P}} \tag{30}$$

$$b_{\mathrm{k,i}} + b_{\mathrm{k,i+n}} \geq x_{\mathrm{i,k}} + x_{\mathrm{k,i}} \; k \in N \backslash \{i, i+n\} \; i \in N_{\mathrm{P}} \tag{31}$$

$$b_{\mathrm{i,k}} + b_{\mathrm{i,n+k}} \geq x_{\mathrm{i+n,k}} + x_{\mathrm{k,i+n}} \quad k \in N \backslash \{i, i+n\} \; i \in N_{\mathrm{P}} \tag{32}$$

$$\sum\nolimits_{\mathrm{k \in N}} \left(b_{\mathrm{k,i}} * G_{\mathrm{k}}\right) = 0 \; i \in \{2n+1, 2n+2\} \tag{33}$$

Constraint (30) is the minimum travel time constraint since the actual travel time from node *i* to *j* will always be larger than the minimum travel time from node *i* to *j*. Since the computing time is limited, this constraint could help provide a better lower bound. Constraints (31) and (32) are the adjacent prior constraints. Constraint (33) is the vehicle return constraint that no passenger will be in the vehicle when the vehicle returns to the depot. Constraints (31)–(33) are described and shown to be valid in Lu and Dessouky (2004).

Heuristic

As previously discussed, the PDP with time windows is a NP-hard problem and optimization algorithms are only able to solve small size problems. Heuristics are necessary to solve larger size problems. In this section, we present heuristics for our PDP with time windows considering congestion. The first is an insertion heuristic, which will be used in the construction of the initial routes. The second part is the Adjust Pickup Time Algorithm, which is used to reduce the passenger ride time by adjusting the pickup time of the passengers. That is, because of the time windows, the vehicle might have to wait at some passenger's pickup location while having other passengers waiting in the vehicle. Since the passengers were picked up as soon as possible, the Adjust Pickup Time Algorithm will postpone these passengers' pickup time so that their ride time will be reduced. Third, Tabu search is applied to improve the routing results where we will repetitively attempt to insert the rejected requests to the routes again and make any adjustments within and between vehicles. Since there is a randomization in the search, Tabu search is run five times to obtain the best solution.

Insertion heuristic

Insertion techniques are widely used for VRP in obtaining initial solutions since they are efficient, easy to implement, and produce good results. Campbell and Savelsbergh (2004) provided a comprehensive review on insertion heuristics for VRP with complicating constraints. Our heuristic uses insertion as a basic technique to solve the PDP with time windows considering congestion.

Given a set of vehicles and a set of requests, the requests are inserted into the existing vehicle routes one at a time. Each request is inserted into an existing route of the vehicle, which has the lowest objective cost to serve the request among all the vehicles while the time window constraint and the capacity constraint are not violated. The objective cost is obtained after the Adjust Pickup Time Algorithm is used to improve the routing service quality by reducing the passenger ride time, which is described in Adjust pickup time algorithm. Our objective cost here to insert one request includes the increased distance, the increased passenger ride time, and the increased toll cost of the route caused by the insertion of the request. A request that cannot be served by any vehicle will be rejected and will be assumed to be served by a separate taxi service. The requests are inserted based on the width of their time windows in an ascending order since we try to serve all the requests by the vehicles. Inserting the requests with the least amount slack in their time windows first leaves the requests with the most flexibility at the end of the insertion process.

Adjust pickup time algorithm

Authors adjust the pickup time to improve the service quality, which refers to passenger ride time here. Normally, the passengers are picked up at their earliest available time whenever the vehicle has arrived. Because of the time windows, a vehicle might have to wait at some passenger's pickup location while having other passengers waiting in the vehicle so that the ride time of these passengers is increased. However, these passengers do not necessarily have to wait. That is, they can be picked up later instead of as soon as possible. Cordeau and Laporte (2003) proposed a procedure, which postpones the passenger's pickup time with the objective to minimize the violation of the ride time constraints in which the ride time of the passengers might be increased though the violation is minimized. Berbeglia, Cordeau and Laporte (2012) presented an algorithm called lazy scheduling algorithm, which is the dynamic version of the algorithm proposed in Cordeau and Laporte (2003) for the static dial-a-ride problem with the same goal to minimize the ride time violation without increasing the time window violation of any node in the route. In both algorithms, while minimizing the ride time violation, the passenger ride time could be increased. Here, we show a mechanism that adjusts the pickup time with the objective to minimize the ride time of the passengers in which the ride time of the passengers will not be increased by the adjustments.

After the insertion heuristic, we obtain a scheduled route with each passenger being picked up as soon as possible. The Adjust Pickup Time Algorithm is applied with the objective of reducing the passenger ride time by delaying the passenger pickup time without changing the routing sequence. For each node *i*, if there are no passengers in the vehicle, we set the delay time equal to F_{i}, which is the maximum amount of time we can increase without violating the time window constraints of all the nodes in the route as defined by Savelsbergh (1992). And since there are no passengers on the ride, no passenger ride time will be affected because of this delay. If there are passengers in the vehicle after serving node *i*, we set the delay time to be the minimum of F_{i} and $\mathrm{WT}_{\mathrm{ij}}$ for all *j* since the ride time of the passengers picked up before node *i* and delivered after node *i* might be increased if the vehicle delayed its departure time at node *i* for the amount of F_{i}. $\mathrm{WT}_{\mathrm{ij}}$ is the least amount of time that passenger *j* needs to wait given the scheduled route and the time windows of the nodes from the node right after node *i* to the delivery node of *j*, no matter how this amount of time is distributed. Then, we update the departure time of node *i* by including the delay time. If node *i* is a pickup node, we set the pickup time to the newly updated departure time to decrease the passenger's ride time. The pickup/delivery time of node *i* will not be changed if it is a delivery node. Since the departure time of node *i* is changed, the pickup/delivery time and departure time of all the nodes succeeding it need to be updated.

Tabu search

An insertion heuristic is used to construct the initial routes. A Tabu search algorithm is developed to improve the solution. The Tabu search algorithm applied in this work considers both between routes exchanges and within route exchanges. The neighborhoods of the between routes Tabu search are obtained from both the PD-Shift operator and PD-Exchange operator (Li and Lim 2001). The PD-Shift operator moves a pickup and delivery pair (PD-pair) from one randomly chosen route to another randomly chosen route. The PD-Exchange operator randomly picks two routes and randomly picks one PD-pair from each route and then swaps the two pairs. The neighborhoods of the within route Tabu search are obtained from the PD-Rearrange operator (Li and Lim 2001), which rearranges PD-pairs to the best position in the same route and the 2-opt operator, which randomly swaps two nodes and nodes in between (pickup or delivery) within the same route. Infeasible neighbors are forbidden with regard to the pairing (pickup before delivery), time window, and capacity constraints.

Authors first do the between routes Tabu search. At each iteration, α_{max} PD-Shift neighbors and β_{max} PD-Exchange neighbors are generated. The Tabu search then moves to the best neighbor that is not in the Tabu list. A temporary worse move is allowed to escape from the local optimal. This part of the heuristic also attempts to insert into the routes the requests that had been previously rejected. After B_NoImp iterations without improvements, random sequences of the routes chosen are generated with the intention of escaping from a local minimum. Feasibility of the routes is required. The between routes Tabu search is repeated until there is no improvement in $I_{B(max}$ iterations.

After between routes Tabu search, we do within route Tabu search for each route. At each iteration, there are γ_{max} neighbors generated from the PD-Rearrange operator and δ_{max} neighbors generated from the 2-opt operator. Same as between routes Tabu search, in each iteration, we move to the best neighbor that is not in the Tabu list allowing temporary worse move as acceptable. Different from between routes Tabu search, we use a reactive Tabu tenure size (Brandao 2004) to relate the tenure size with the route size. After W_NoImp iterations without improvements, the route is re-routed to escape from a local minimum. The within route Tabu search terminates after $I_{W(max}$ iterations with no improvement.

Experimental results

In this section, we run simulations by applying the schemes proposed in the above heuristics. Authors run several computational experiments with the objective to: (1) compare the solutions from the heuristics with the optimal solutions obtained from CPLEX using our proposed IP model; (2) compare the cost per request for the various heuristics (our proposed congestion-based heuristic versus heuristics that only use distance as a criteria in the objective) for different values on the number of vehicles and time window sizes; and (3) perform sensitivity analysis on time savings on HOV lanes to understand under what conditions it is best to take detours to make use of these incentives for ridesharing. Unless otherwise noted, all the simulations are run on the same map constructed as follows.

The experiments are run on a map constructed that considers the existence of HOV lanes and toll roads. The map is a 16*10 grid with 160 nodes, which are used as the start and end locations of the drivers and the pickup and delivery locations of the requests and 294 edges, which connect the 160 nodes as a grid. All the edges are undirected. Authors set the length of each edge to be 10km. Fifty out of the 294 edges are randomly chosen to be toll roads that charge nine dollars with no time savings to travel on. The toll rate information was derived from the Highway Performance Monitoring System of the Federal Highway Administration (Office of Highway Policy Information 2011). Authors used fees, which represented the average of the toll rate data for California. The other 244 edges are freeways, which do not charge toll fees. Authors randomly set 147 out of the remaining 244 edges to contain HOV lanes. One hundred seventeen out of the 147 are HOV lanes for two or more. The other 30 out of the 147 edges contain HOV lanes for three or more people. Authors set the time on each edge to be 10min for the general purpose lanes, 7min for HOV lanes for two or more people, and 6min for HOV lanes for three or more people. The time-saving information is gained from the HOV Performance Program Evaluation Report by Los Angeles County Metropolitan Transportation Authority (2002). By comparing the travel speed in peak hour for HOV lanes and general purpose lanes, HOV lanes are on average 36.5% faster than the general purpose lanes for the routes studied. Authors also assume that vehicles that qualify for the HOV lanes do not pay fees on the toll roads.

There are two types of inputs, which are vehicles and requests. Unless otherwise noted, we assume that all vehicles start at time 1s, end at the end of the day, and the initial number of people in the vehicle (driver) is 1. The origin and the destination are randomly selected from the above grid for both the vehicles and requests. The request time for each request is assumed to be time 0, which is right before the start time of the vehicle. That is, all the requests are known before the vehicle starts. Authors set the parameter TW, the acceptable time period that the request can be served, to be a multiple of the time needed if the passenger travels alone and directly from the origin to the destination, i.e. TW = α*direct ride time. The earliest pickup time is set in a way so that the earliest pickup time plus the time window fits within a day. If it is not specifically mentioned, the number of people to pick up for a particular request is set to 1. See Tables 1 and 2 for the detailed information.

First, we run simulations to test the effectiveness of the valid equality and inequalities in our IP model. Since it is computationally difficult to find optimal solutions for large problem instances, we run simulations on problems with one vehicle and 5, 6, 7, 8, and 9 requests with different α values of 1.5, 2, 2.5, and 3 to find the optimal solution in a reasonable amount of computation time. Here, we set the capacity of the vehicle to be 4. For this set of experiments, our objective is to serve as many requests as possible by the vehicle and to minimize the travel distance, passenger ride time, and toll fee with their coefficients $\beta=\gamma=\mu=1$. Thus, we set the weight λ to a very large number. Authors generate five random test instances for each scenario. Table 3 shows the average running time results. The experiments are implemented on a 2-Processor-Quad-Core Xeon Workstation (2.66GHz, 16GB RAM, 1.5TB hard drive) using CPLEX 9.0.

In Table 3, the results under the column of no cuts show the average running time needed to solve the five instances for each case without the valid equality and inequalities, which are constraints (30)–(33) in the model formulation. The results under the column of with cuts show the average running time needed to solve the five instances for each case with the valid equality and inequalities. The results under the column of improvement are the average improvement of the five instances for each case. In general, as shown in Table 3, those valid Equality and Inequalities are effective, and the improvement is enhanced for larger size instances. Note that as the time window size increases, it takes more computational time to find the optimal solution since the feasible region is larger.

Next, we compare the performance of our heuristics by comparing the routing results of the heuristics with that of the IP model. For the parameters in Tabu search, since our route size is not large, we set $\alpha_{\max}$, $\beta_{\max}$, $\gamma_{\max}$, and $\delta_{\max}$ to be the number of all possible neighbors. B_NoImp and W_NoImp are set to equal to 5. $I_{\mathrm{W(max}}$ is set to equal to 150. $I_{\mathrm{B(max}}$ is set to be the number of possible combinations of two routes out of the total number of routes. Table 4 shows the results. The average ratio shown in the table is the objective value of the optimal solution divided by the output objective value of the heuristic. Number of optimal found indicates among how many out of the five tests we ran for each scenario, the heuristic found the optimal solution. Some results in the table show that the average ratio equals to 1 while the number of optimal found does not equal to 5. The reason is that the optimal value and the output of the heuristic have a very small difference in these cases. Since the results round to two decimals, the ratio equals to one. From Table 4, we can see that the heuristic works well for smaller size problems. In cases where the heuristic did not find the optimal result, the heuristic failed to find a solution that served as many requests as the optimal result. The running times of the IP model are shown in Table 3 while the running time of the heuristic, for example, with $\alpha=3$ and number of requests larger then 6, is 5s on average.

Table 1. Vehicle data generation

Origin	Uniformly random in 16*10 grid
Destination	Uniformly random in 16*10 grid
Start time	00:01
End time	23:59
Initial number of people in the vehicle	1

Table 2. Request data generation

Origin	Uniformly random in 16*10 grid
Destination	Uniformly random in 16*10 grid
Request time	00:00
Earliest pickup time	Uniformly random [00:01, 23:59-TW]
Latest pickup time	Earliest pickup time+TW
Earliest delivery time	Earliest pickup time
Latest delivery time	Latest pickup time
Number of people to pick up for each request	1

Authors next compare the performances of the heuristics with the objective of minimizing distance only and the objective considering congestion, which is minimizing the travel distance, passenger ride time, and toll fee with their coefficients equal to 1. The parameters used in the Tabu search are the same as the previous simulation. Authors do simulations on 100 requests, different number of vehicles: 10, 15, and 20 and different α values of 1.5, 2, 2.5, and 3. Here, the number of people to pick up per request is a uniformly discrete random variable between 1 and 3 to test the impact of the generated routes when there are more people in the vehicle. Since the time windows of the requests we used in our simulation imposes a relatively stricter constraint than the capacity of the vehicle, we now assume that all vehicles have no capacity constraint unless otherwise noted.

In the following figures, we show the cost per request from the simulation results with the above assumptions and data inputs. All 100 requests are served by the provided vehicles without using any taxi for each instance. The results shown in the figures are averages of 10 instances. Authors simulate each instance with different heuristics: insertion heuristic with objective of minimizing distance and with Tabu Search (Distance-Tabu) and insertion heuristic with objective of minimizing congestion function (see objective function in Model formulation) and with Tabu Search (Congestion-Tabu). The Adjust Pickup Time Algorithm is applied in each scenario. Authors compare the results based on the cost per request served. Note that the cost per request is the sum of the total distance, passenger ride time, and toll fee divided by the total number of requests. The total distance refers to the total vehicle travel distance minus the distance from the origin to destination of the vehicle itself. This comparison allows us to show the benefit of explicitly accounting for the congestion in the heuristics.

Figure 1 summarizes the cost per request with different time windows and the number of vehicles using the Congestion-Tabu heuristic. Figure 2 shows the differences in cost per request, which is the cost per request under Distance-Tabu scenario minus that under Congestion-Tabu scenario.

From Fig. 1, we compare the results of different number of vehicles and time windows with the scenario of Congestion-Tabu. Authors can see that the cost per request is decreasing with a larger time window and more vehicles. That is, since all 100 requests are served, with a larger time window, there are more possible feasible routes to choose from which reduces the cost per request. Likewise, when there are more vehicles, there are more alternatives for ridesharing options, reducing the cost.

From Fig. 2, it shows that the difference in cost per request between the two heuristics, Distance-Tabu and Congestion-Tabu, is increasing with larger time windows. That is, the benefit of the congestion-based heuristics (Congestion-Tabu) increases with larger time windows. This benefit explicitly accounts for the fact that the travel time and toll fee can decrease with additional pickups.

Table 3. Performance of valid equality and inequalities

α	Number of requests made	No cuts (s)	With cuts (s)	Improvement (%)
1.5	5	2.71	0.49	50.49
1.5	6	1.87	0.80	48.42
1.5	7	5.78	1.30	62.84
1.5	8	85.38	7.12	81.36
1.5	9	74.15	7.87	79.88
2	5	1.00	0.53	38.63
2	6	1.39	0.77	43.71
2	7	131.13	11.60	53.21
2	8	1234.09	47.93	57.91
2	9	1577.76	175.22	64.84
2.5	5	1.43	0.90	40.63
2.5	6	21.49	3.78	55.81
2.5	7	53.62	63.69	33.64
2.5	8	7333.21	709.17	82.48
2.5	9	12340.98	286.44	93.71
3	5	2.69	2.75	18.47
3	6	7.66	5.52	42.24
3	7	10368.11	2219.90	68.94
3	8	820.64	104.41	66.50
3	9	342.05	74.08	73.50

Table 4. Performance of the proposed heuristic

α	Number of requests made	Average ratio	Number of optimal found
1.5	5	1.00	5
1.5	6	1.00	5
1.5	7	0.91	3
1.5	8	1.00	3
1.5	9	0.95	3
2	5	1.00	4
2	6	1.00	5
2	7	1.00	4
2	8	1.00	2
2	9	0.92	0
2.5	5	1.00	5
2.5	6	1.00	2
2.5	7	1.00	3
2.5	8	1.00	2
2.5	9	0.90	1
3	5	1.00	5
3	6	1.00	4
3	7	1.00	3
3	8	1.00	3
3	9	1.00	3

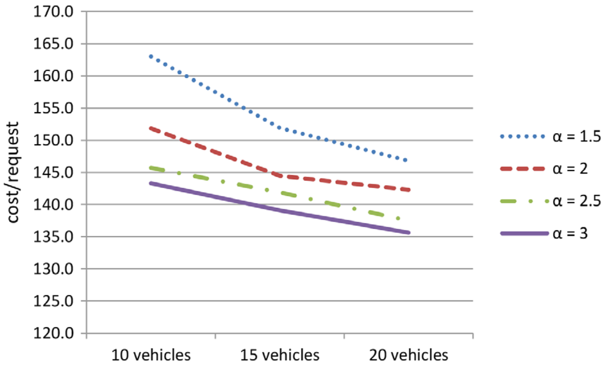
Our next set of results consists of a sensitivity analysis for the time savings on HOV lanes. Authors applied Congestion-Tabu heuristic and used the distance ratio and the ride time ratio to indicate how the ridesharing participants react to different amounts of time savings on HOV lanes: simulations at intervals of 10% reduction in travel time, starting with a 10% reduction. Here, though it is also a component of our objective function, we did not consider the toll cost since the toll cost is dominated by the ride time and the distance. The calculations of the distance ratio and the ride time ratio are shown below. $\mathrm{RD}_{\mathrm{i,n(i}}$ is the actual distance traveled from node i to node $n(i$ (the actual distance request i traveled from its pickup location to its drop off location in the vehicle). $D_{\mathrm{i,n(i,1}}$ is the direct distance if request i traveled alone. n is the total number of requests. The distance ratio is the average actual distance divided by the drive-alone distance for each request. The higher the distance ratio, the more detours the request takes. The quantity $v_{\mathrm{n(i}}(v_{\mathrm{i}}$ indicates the actual ride time that request i spends in the vehicle from its pickup location to its drop off location. $T_{\mathrm{i,n(i,1}}$ is the travel time if request i traveled alone. The ride time ratio is the average actual ride time divided by the drive-alone ride time for each request. The lower the ride time ratio, the more ride time is saved by participating in ridesharing

$$\text{distance ratio}=\frac{1}{n}\sum_{\mathrm{i}\in N_{\mathrm{P}}}\frac{\mathrm{RD}_{\mathrm{i,n+i}}}{D_{\mathrm{i,n+i,1}}}$$

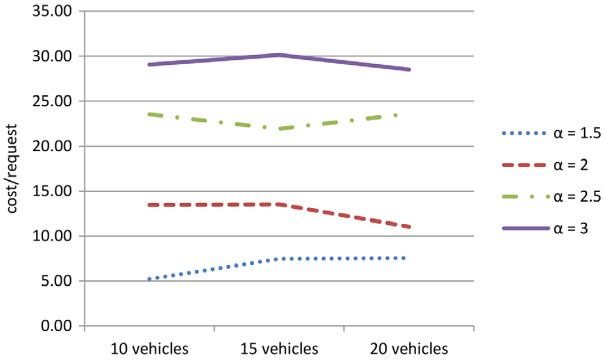
$$\text{ride time ratio}=\frac{1}{n}\sum_{\mathrm{i}\in N_{\mathrm{P}}}\frac{v_{\mathrm{n+i}}-v_{i}}{T_{\mathrm{i,n+i,1}}}$$

The heuristic settings here are the same as the previous simulation settings. However, we change the map by setting the 147 HOV lanes to be all HOV2 lanes, all HOV3 lanes, all HOV4 lanes, or NO HOV lanes according to the different tests. NO HOV lanes means there is no time savings for more people in the vehicle. There is a total of 294 edges so the other 147 edges do not contain any HOV lanes in all test instances. Therefore, the result of NO HOV lanes is indifferent with the different timing savings on HOV lanes. Authors do simulations on 100 requests, 15 vehicles, and an α value of 2. The results shown in the figures are averages of 10 instances. Figures 3 and 4 below demonstrate the sensitivity of the distance ratio and ride time ratio to the time savings on HOV lanes for the time savings, respectively.

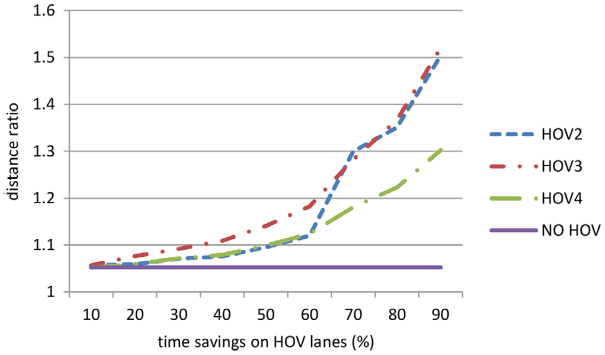
These results show that when there is no time savings for having HOV, in the NO HOV scenario, there still is some ridesharing occurring as evidenced by having both a distance ratio and ride time ratio slightly bigger than one. In the scenarios with HOV lanes, as HOV lanes become more attractive (there is a larger time savings by traveling on them) then rides become longer in length (as seen by the increase in distance ratio in Fig. 3), but travel time becomes shorter (as seen by the decrease in ride time ratio in Fig. 4), which indicate that



1 Cost/request for different α's using congestion-Tabu



2 Difference in cost/request between distance-Tabu and congestion-Tabu



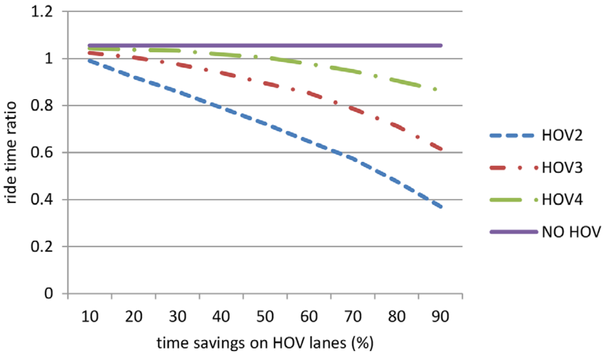
3 Distance ratio sensitivity to time savings on HOV lanes

passengers are more encouraged to take detours (participate in ridesharing) to save ride time as time savings on HOV lanes increase. Furthermore, we observe that the ride time ratio is always larger when the edges (lanes) require more people to use the HOV lanes. That is, HOV2 has the smallest ride time ratio and its difference between the other scenarios grows as the savings in time for use of HOV lanes increase. The reason being that is it is more difficult to use HOV3 lanes, even more so HOV4 lanes, to save ride time since the vehicle has to take a detour to have extra pickups and the ride time of the passenger who is already in the vehicle is increased. The behavior of the ride time ratio among the various HOV scenarios is intuitive. However, if we look at the distance ratio as shown in Fig. 3, the behavior among the various HOV scenarios is not intuitive, as it is not monotonic and the distance ratio for HOV2 jumps suddenly. To clarify this issue, we plot the total cost per request and the total distance the vehicles traveled in Figs. 5 and 6 below.

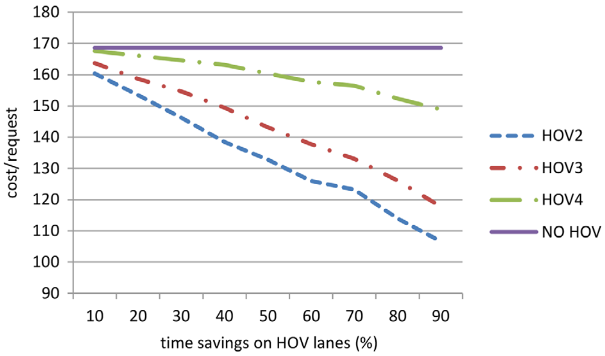
Figure 5 shows that, with larger time savings on HOV lanes, the total cost per request is consistently decreasing since the total passenger ride time is decreasing consistently as evidenced by the ride time ratio in Fig. 4. HOV2 has the lowest total cost per request as it is the easiest to be qualified to use the HOV lanes to save ride time. However, the total distance shown in Fig. 6 initially decreases but increases roughly at the same time savings point when the distance ratios for the HOV2 scenario jumps to the HOV3 scenario, which is when the time savings on HOV lanes is 60%. Authors first comment that Fig. 6 shows the total distance traveled by the vehicles and the behavior among the various HOV scenarios is, in general, in the reverse order for that of the distance ratio (from Fig. 3) since this is from the perspective of the vehicle instead of the passenger. Thus, the NO HOV scenario has roughly the highest total distance traveled by the vehicles but the lowest distance ratio.

To further analyze the behavior among the various HOV scenarios in total distance, we observe that there are two incentives for ridesharing: (1) to reduce total route distance, and (2) to save ride time. The former corresponds to the reason that there is ridesharing in NO HOV, to create ridesharing so that the vehicle can choose a more direct route to reduce the total route distance. The latter incentive is that the ridesharing vehicle can use HOV lanes to reduce the passenger ride time. When there is no time savings on HOV lanes, the incentive for ridesharing is simply to reduce the total vehicle distance. Both incentives play a role as the time savings in HOV lanes increases, but the first type of incentive tends to dominate when the time savings are less than 60%. As the time savings on HOV lanes increases, there are more feasible routes to choose and the vehicles can afford to deviate less than in the NO HOV situation. This is the reason the total distance traveled by the vehicle decreases while the distance ratio increases within this interval. When the time savings on HOV lanes is larger than 60%, both the total distance and the distance ratio go up while the ride time ratio and total cost per request go down. When the reduction in travel time is more attractive, vehicles begin to deviate more to make use of ridesharing for the second reason (to reduce travel time). That is, the decrease in ride time is more significant than the increase in total distance, which means that the passengers take detours to save ride time.

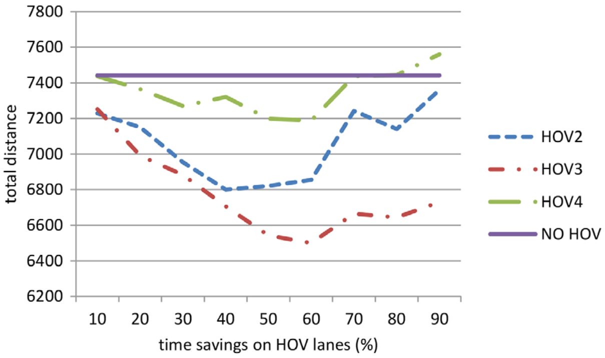
Authors can now further analyze the distance ratio shown in Fig. 3. Authors want to first comment that the result of the HOV4 scenario is always the one that is the closest to the NO HOV scenario since it is the most difficult scenario to be qualified to use HOV lanes, so it does the least amount of ridesharing. Following this argument, the counterintuitive behavior in Fig. 3 is due to the low distance ratio of HOV2. For small time savings in HOV lanes, when the first type of incentive dominates, both HOV2 and HOV3 do about the same amount of detours as evidenced by the similar curves in the total distance plot. But while vehicles in both HOV2 and HOV3 deviate a similar amount from direct routes, the HOV2 scenario



4 Ride time ratio sensitivity to time savings on HOV lanes



5 Cost/request sensitivity to time savings on HOV lanes



6 Total distance sensitivity to time savings on HOV lanes

does this without increasing the distance traveled to each passenger as much, leading to a lower distance ratio. In HOV3, the passengers in the vehicle while detouring to pick up the last passenger to qualify for use of the HOV3 lane have a large distance ratio. When the second type of incentive dominates (time savings greater than 60%), the distance ratios for the HOV2 and the HOV3 scenarios are about the same. However, the HOV2 scenario has a more significant increase in total distance than the HOV3 scenario, which indicates that the passengers in the HOV2 scenario are more involved in ridesharing to save ride time, suggesting that more ridesharing is occurring in the HOV2 scenario. The ridesharing in the HOV3 scenario requires a significant increase in distance of the vehicles. Thus, HOV2 and HOV3 have the same average distance ratio when the time savings are greater than 60% but we would expect that the variance of the distance ratio would be much higher for the HOV3 scenario when the time savings are great. To verify this issue, we computed the variance of the distance ratio. Initially, both scenarios have the same variance (0.02). The difference goes up with the increasing time savings on HOV lanes. When the time savings on HOV lanes is 90%, the HOV3 scenario has a higher variance (0.50) than the HOV2 scenario (0.40), which indicates that the HOV3 scenario has fewer passengers who participate in ridesharing to capture the HOV lanes while the passengers who participate in ridesharing to capture the HOV lanes to save ride time have much more detours since the distance ratios of both scenarios are the same. This is consistent with the observation in Fig. 6 that the passengers are more involved in ridesharing in the HOV2 scenario when the second type of incentive dominates.

Conclusion

In this paper, we modified existing PDPs with time windows to consider the passenger travel time under congestion and load dependent toll cost to study how the optimal routes change if a cost reduction and time savings are available for ridesharing. A 0–1 integer programing model is formulated to solve the problem optimally. Heuristics are developed to efficiently solve the problem. The Adjust Pickup Time Algorithm is proposed to reduce the total cost and the customer ride time.

Authors first tested how the heuristics work by comparing the routing results of the heuristics with that of the IP model. The results indicate that our heuristic performs comparably to the optimal solution for small size problems. Then, we run simulations to test different inputs and different heuristics with different objective. The results show that, as a participant in ridesharing becomes more flexible in time, the less one should pay for his/her trip. For more vehicles, there are more options in identifying the ridesharing options. Also, our results show that there is significant benefit to considering the toll cost savings and time savings with additional pickups. After that, we performed a set of computational experiments to explore how ridesharing is affected by the different time savings on HOV lanes. Authors evaluated the sensitivity to HOVs using two different measures: a distance ratio and a ride time ratio. From the results, we see that, when time savings on HOV lanes get more significant, the distance ratio will increase while the ride time ratio will decrease. This indicates that, under the policies promoting ridesharing, passengers may need to take a detour to share a ride with others to save the total route distance or to capture faster paths to save their ride time. The amount of detour of the passengers can be controlled by adjusting the time windows of the passengers or a strict time constraint can be imposed. Moreover, if it is too difficult to be qualified to use HOV lanes (e.g. HOV4 lanes), there are less intentions to take detours to share a ride. Therefore, policymakers should be aware of and further explore the ridesharing participants' reactions to those policies when designing the policies to promote ridesharing.

In this paper, we consider the load-dependent travel time and toll cost, which are more complicated than the standard PDP. It requires substantial amount of computational effort to find optimal solutions for large problem instances and hence heuristics were developed to solve the large problem instances. Thus, future work can investigate the development of tighter lower bounds in order to be able to benchmark the developed heuristics against the optimal solution for the large problem instances. Another model assumption is that the change in travel time, toll, and distance for different vehicle loads remains constant regardless of the number of vehicles, the effect of supply-demand dynamics on this PDP is also a topic for future research. Furthermore, we consider a static model in this paper. Future research could consider dynamic customer requests and include uncertainty in travel times and demand to the problem.

References

Agatz, N. A. H., Erera, A., Savelsbergh, M. W. P. and Wang, X. 2012. Optimization for dynamic ride-sharing: a review, European Journal of Operational Research, 233, 295–303.

Bard, J. F. and Jarrah, A. I. 2009. Large-scale constrained clustering for rationalizing pickup and delivery operations, Transportation Research Part B: Methodological, 43, 542–561.

Berbeglia, G., Cordeau, J. F. and Laporte, G. 2010. Dynamic pickup and delivery problems, European Journal of Operational Research, 202, (1), 8–15.

Berbeglia, G., Cordeau, J. F. and Laporte, G. 2012. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem, INFORMS Journal on Computing, 24, (3), 343–355.

Berbeglia, G., Cordeau, J. F., Gribkovskaia, I. and Laporte, G. 2007. Static pickup and delivery problems: a classification scheme and survey, Top, 15, 1–31.

Brandao, J. 2004. A tabu search algorithm for the open vehicle routing problem, European Journal of Operational Research, 157, 552–564.

Campbell, A. M. and Savelsbergh, M. 2004. Efficient insertion heuristics for vehicle routing and scheduling problems, Transportation Science, 38, 369–378.

Carabetti, E. G., de Souza, S. R., Fraga, M. C. P. and Gama, P. H. A. 2010. An application of the ant colony system metaheuristic to the vehicle routing problem with pickup and delivery and time windows, in 2010 Eleventh Brazilian Symposium on Neural Networks, 176–181; Sao Paulo, IEEE Computer Society.

Catay, B. 2009. Ant colony optimization and its application to the vehicle routing problem with pickups and deliveries, in Natural intelligence for scheduling, planning and packing problems, (eds. Chiong R. and Dhakal S.., 219–244; Berlin, Springer.

Cordeau, J. F. 2006. A branch-and-cut algorithm for the dial-a-ride problem, Operations Research, 54, 573–586.

Cordeau, J. F. and Laporte, G. 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem, Transportation Research Part B: Methodological, 37, 579–594.

Cordeau, J. F. and Laporte, G. 2005. Tabu search heuristic for the vehicle routing problem, in Metaheuristic optimization via memory and evolution: tabu search and scatter search, (eds. Rego C. and Alidaee B.., 145–163; Boston, Kluwer.

Cordeau, J. F. and Laporte, G. 2007. The dial-a-ride problemml: models and algorithms, Annals of Operations Research, 153, 29.

Cordeau, J. F., Laporte, G. and Ropke, S. 2008. Recent models and algorithms for one-to-one pickup and delivery problems, in The Vehicle Routing Problemml: Latest Advances and New Challenges, Vol. Vol. 43, 327–357; Berlin, Springer.

Desrosiers, J., Dumas, Y. and Soumis, F. 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows, American Journal of Mathematical and Management Sciences, 6, 301–325.

Desrosiers, J., Dumas, Y., Soumis, F., Taillefer, S. and Villeneuve, D. 1991. An algorithm for mini-clustering in handicapped transport, in Technical Report G-91-02. GERAD, HEC, Montréal.

Diana, M. and Dessouky, M. M. 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows, Transportation Research Part B: Methodological, 38, 539–557.

Dumas, Y., Desrosiers, J. and Soumis, F. 1989. Large scale multi-vehicle dial-a-ride problems, in Research Report G-89-30. GERAD, HEC, Montréal.

Furuhata, M., Dessouky, M. M., Ordóñez, F., Brunet, M., Wang, X. and Koenig, S. 2013. Ridesharing: the state-of-the-art and future directions, Transportation Research Part B: Methodological, 57, 28–46.

Hosny, M. I. and Mumford, C. L. 2010. The single vehicle pickup and delivery problem with time windows: intelligent operators for heuristic and metaheuristic algorithms, Journal of Heuristics, 16, 417–439.

Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M. M. and Villeneuve, D. 1995. A request clustering algorithm for door-to-door handicapped transportation, Transportation Science, 29, 63–78.

Jaw, J. J., Odoni, A. R., Psaraftis, H. N. and Wilson, N. H. M. 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, Transportation Research Part B: Methodological, 20, 243–257.

Lenstra, J. K. and Rinnooy Kan, A. H. G. 1981. Complexity of vehicle routing and scheduling problems, Networks, 11, 221–227.

Levy, J. I., Buonocore, J. J. and Von Stackelberg, K. 2010. Evaluation of the public health impacts of traffic congestion: a health risk assessment, Environmental Health, 9, 65–76.

Li, H. and Lim, A. 2001. A metaheuristic for the pickup and delivery problem with time windows, in Proceedings 13th IEEE ICTAI 2001, 160–167; Los Alamitos, CA, IEEE Computer Society.

Los Angeles County Metropolitan Transportation Authority. 2002. HOV performance program evaluation report. Metro. Available at: <http://www.metro.net/projects_studies/hov/images/hov_final_evaluation_report.pdf>. [Accessed May 21, 2014].

Lu, Q. and Dessouky, M. M. 2004. An exact algorithm for the multiple vehicle pickup and delivery problem, Transportation Science, 38, 503–514.

Lu, Q. and Dessouky, M. M. 2006. A new insertion-based construction heuristic for solving the pickup and delivery problem with hard time windows, European Journal of Operational Research, 175, 672–687.

Nanry, W. P. and Barnes, J. W. 2000. Solving the pickup and delivery problem with time windows using reactive tabu search, Transportation Research Part B: Methodological, 34, 107–121.

Office of Highway Policy Information. 2011. Toll facilities in the United States, federal highway administration. Available at: <http://www.fhwa.dot.gov/policyinformation/tollpage/cover.cfm#toc>. [Accessed April 6, 2013].

Parragh, S. N., Doerner, K. F. and Hartl, R. F. 2010. Variable neighborhood search for the dial-a-ride problem, Computers and Operations Research, 37, 1129–1138.

Potvin, J. and Rousseau, J. 1992. Constraint-directed search for the advanced request dial-a-ride problem with service quality constraints, in Computer science and operations research: new developments in their interfaces, (eds O. Balci, R. Sharda and S. A. Zenios), 457–474; Oxford, Pergamon Press.

Psaraftis, H. N. 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem, Transportation Science, 14, 130–154.

Psaraftis, H. N. 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows, Transportation Science, 17, 351–357.

Ropke, S. and Cordeau, J. F. 2009. Branch and cut and price for the pickup and delivery problem with time windows, Transportation Science, 43, 267–286.

Ropke, S., Cordeau, J. F. and Laporte, G. 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows, Networks, 49, 258–272.

Roy, S., Chapleau, L., Ferland, J. A., Lapalme, G. and Rousseau, J. M. 1983. The construction of routes and schedules for the transportation of the handicapped, in Technical Report CRT-308. Universite de Montreal, Montreal.

Santos, A., McGuckin, N., Nakamoto, H. Y., Gray, D. and Liss, S. 2011. Summary of travel trends: 2009 national household travel survey, in Technical Report. US Department of Transportation Federal Highway Administration, Washington, DC.

Savelsbergh, M. W. P. 1992. The vehicle routing problem with time windows: minimizing route duration, ORSA Journal on Computing, 4, 146–154.

Schrank, D., Eisele, B. and Lomax, T. 2012. TTI's 2012 urban mobility report. Texas Transportation Institute, Texas A&M University. Available at: <http://mobility.tamu.edu/ums/>. [Accessed May 21, 2014].

Sombuntham, P. and Kachitvichyanukul, V. 2010. A particle swarm optimization algorithm for multi-depot vehicle routing problem with pickup and delivery requests, in International Multiconference of Engineers and Computer Scientists 2010. Hong Kong.

Wong, K. I. and Bell, M. G. H. 2006. Solution of the dial-a-ride problem with multi-dimensional capacity constraints, International Transaction in Operational Research, 13, 195–208.

Xiang, Z., Chu, C. and Chen, H. 2006. A fast heuristic for solving a large-scale dial-a-ride problem under complex constraints, European Journal of Operational Research, 174, 1117–1139.