Noname manuscript No.
(will be inserted by the editor)

The Flexible and Real-Time Commute Trip Sharing Problems

Mohd. Hafiz Hasan · **Pascal Van Hentenryck**

Received: date / Accepted: date

Abstract The Commute Trip Sharing Problem (CTSP) was introduced to remove parking pressure on cities, as well as corporate and university campuses. Its goal is to reduce the number of vehicles being used for daily commuting activities. Given a set of inbound and outbound requests, which consists of origin and destination pairs and their departure and return times, the CTSP assigns riders and a driver, as well as inbound and outbound routes, to each vehicle in order to satisfy time-window, capacity, and ride-duration constraints. The CTSP guarantees a ride back for each rider, which is a critical aspect of such a ride-sharing system. This paper generalizes the CTSP to account for uncertainties about the return trip. Each rider is assumed to have a return time specified by a distribution (learned from historical data) and, each day, a percentage of riders will want to preprone or postpone their return trip to accommodate some schedule changes. The paper proposes two generalizations of the CTSP: the Flexible CTSP (FCTSP) and the Real-Time CTSP (RT-CTSP). In the FCTSP, riders must confirm their final return times by a fixed deadline. In the RT-CTSP, riders confirm their new return times in real time with some prior notice. The paper proposes a two-step approach to address the FCTSP and the RT-CTSP. The first step uses a scenario-based stochastic program to choose the drivers and the morning routes in order to maximize the robustness of the driver assignment. The second step reoptimizes the plan at the fixed deadline or in real time once the return times are confirmed. Experiments on a real-world dataset of commute trips demonstrate the effectiveness of the algorithm in generating robust plans and reveal a trade-off between vehicle reduction and plan robustness as the robust plans tend to be conservative. A method is then proposed to evaluate this trade-off using the per-unit price ratio of vehicle increase to uncovered riders.

Keywords Robust planning · Ride sharing · Scenario sampling · Vehicle routing problem

Mohd. Hafiz Hasan
University of Michigan, Ann Arbor, Michigan 48109, USA
E-mail: hasanm@umich.edu

Pascal Van Hentenryck
Georgia Institute of Technology, Atlanta, Georgia 30332, USA
E-mail: pascal.vanhentenryck@isye.gatech.edu

1 Introduction

The Commute Trip Sharing Problem (CTSP) was introduced in [7] in an effort to reduce peak-hour traffic congestion and parking utilization for urban areas. It seeks an optimal routing plan that maximizes ride sharing for a set of commute trips. Each commuter makes two trip requests per day, one to the workplace and another back home. Each request has specific pickup and drop-off locations which must be visited in order, time windows describing allowable service times at each location, and a ride-duration limit. The routes must serve these trips exactly once according to their specifications while ensuring the capacity of the vehicles used is not exceeded. In addition to this, the vehicle drivers for any day are selected from the set of commuters; therefore, the set of drivers selected for the trips to the workplace must be identical to that for the return trips.

The CTSP is therefore a Vehicle Routing Problem (VRP) with time-window, capacity, pairing, precedence, ride-duration, and driver constraints. It is a generalization of the VRP with Time Windows (VRPTW) which is well known to be NP-hard [13]. It was solved in [7] using a three-stage approach which first clusters commuters according to their residential locations, searches exhaustively for all feasible ride-sharing routes within each cluster, and then solves a mixed-integer program (MIP) to optimize the selection of routes. Hasan et al. [8] then introduced a branch-and-price algorithm to solve the problem that uses column generation to search for feasible routes on demand.

This work generalizes the CTSP by considering a setting in which trip schedules are known in advance, however there are uncertainties associated with the return-trip schedules as they occur later in the day. More specifically, it introduces: (1) the Flexible CTSP (FCTSP) whereby commuters are required to confirm their return times by a fixed deadline and (2) the Real-Time CTSP (RT-CTSP) whereby commuters confirm their return times in real time with some advance notice. Regardless of the problem variant, the routing plan must commit the day's drivers before the return times are confirmed. The challenge is therefore to ensure that the plan is robust, i.e., to ensure that the drivers can still cover the return trips despite the uncertainties in their schedules.

This paper explores a scenario-sampling approach to handle these uncertainties. The method was first used in [1] for the VRP with stochastic customers and then by [15] to tackle temporal uncertainty in the VRPTW. It assumes knowledge of probability distributions for every commuter describing the likelihood of their return-trip times which can be sampled to obtain potential scenarios. This work incorporates the method into a multi-stage framework, whereby the first stage optimizes a plan for several scenarios while subsequent stages reoptimize the return-trip plan when additional information becomes available, i.e., just after the deadline for the FCTSP and at a regular frequency for the RT-CTSP which will reoptimize batches of trips with confirmed return times.

The approach is evaluated on a real-world dataset of commute trips from the city of Ann Arbor, Michigan. Its results show that the method produces plans that become more robust as the number of sampled scenarios increases. Unfortunately, the increase in robustness comes at a price of having additional vehicles. A method is therefore proposed to evaluate the trade-off between plan robustness and vehicle reduction.

The rest of this paper is organized as follows. Section 2 first introduces the terminologies and assumptions used throughout the paper. Section 3 then reviews the mathematical formulation of the CTSP and a column-generation algorithm by [8] for solving it. Section 4 then describes in detail the problem settings for the FCTSP and RT-CTSP and the optimization algorithm. Section 5 reports the experimental setup and results of the computational experiments. Finally, Section 6 provides some concluding thoughts.

$$
\begin{aligned}
\min \quad & T_{d_{D_r}} - T_{o_{D_r}} && (1)\\
\text{s.t.} \quad & a_{o_c} \leq T_{o_c} \leq b_{o_c} \qquad \forall c \in \mathscr{C}_r && (2)\\
& T_{d_c} \leq b_{d_c} \qquad \forall c \in \mathscr{C}_r && (3)\\
& T_{pred(o_c)} + \zeta_{pred(o_c)} + \tau_{(pred(o_c),o_c)} \leq T_{o_c} \qquad \forall c \in \mathscr{C}_r \setminus \{D_r\} && (4)\\
& T_{pred(d_c)} + \zeta_{pred(d_c)} + \tau_{(pred(d_c),d_c)} = T_{d_c} \qquad \forall c \in \mathscr{C}_r && (5)\\
& T_{d_c} - (T_{o_c} + s_{o_c}) \leq L_c \qquad \forall c \in \mathscr{C}_r && (6)
\end{aligned}
$$

Fig. 1 The Model for the Route-Scheduling Problem for a Valid Route r.

2 Notation and Preliminaries

A trip $t =< o, dt, d, at >$ consists of an origin o, a departure time dt, a destination d, and an arrival time at. On any day, a commuter c makes two trips: a trip to the workplace, t_c^+, and a trip back home, t_c^-. These trips are referred to henceforth as inbound and outbound trips respectively. A route r is a sequence of origin and destination locations from a set of inbound or outbound trips whereby each origin and destination from the set is visited exactly once. For instance, a possible route for trips $t_1 =< o_1, dt_1, d_1, at_1 >$ and $t_2 =< o_2, dt_2, d_2, at_2 >$ is $r = o_2 \to o_1 \to d_1 \to d_2$. An inbound route covers only inbound trips and an outbound route covers only outbound trips. Each route r serves a set of riders $\mathscr{C}_r$ and has a driver $D_r \in \mathscr{C}_r$. The driver must be the rider residing at the start location of the route. For instance, rider 2 must be the driver of route $o_2 \to o_1 \to d_1 \to d_2$. The total number of riders in the vehicle at any point along a route cannot exceed its capacity.

Definition 1 (Valid Route) A valid route r visits o_c before d_c for every rider $c \in \mathscr{C}_r$, starts at o_{D_r} and ends at d_{D_r}, and respects the vehicle capacity.

It is assumed that commuters sharing rides are willing to tolerate some inconvenience in terms of deviations to their trips' desired departure and arrival times as well as in terms of extensions to the ride durations of their individual trips. Therefore, a time window $[a_i, b_i]$ is constructed around the desired times and is associated with each pickup or drop-off location i, where a_i and b_i denote the earliest and latest times at which service may begin at i respectively, and a duration limit L_c is associated with each rider c to denote her maximum ride duration. Let T_i denote the time at which service begins at location i, ζ_i be the service duration at i, $pred(i)$ denote the location on a route visited just before i, and $\tau_{(i,j)}$ be the estimated travel time for the shortest path between locations i and j.

Definition 2 (Feasible Route) A feasible route r is a valid one with pickup and drop-off times $T_i \in [a_i, b_i]$ for each location $i \in r$ that ensures the ride duration of each rider $c \in \mathscr{C}_r$ does not exceed L_c.

Determining if a valid route r is feasible amounts to solving the route-scheduling linear program (LP) shown in Figure 1. Its objective is to minimize the route's total duration. Constraints (2) and (3) are time-window constraints for pickup and drop-off locations respectively, while constraints (4) and (5) describe compatibility requirements between pickup/drop-off times and travel times between consecutive locations along the route. Finally, constraints (6) specify the ride-duration limit for each rider. Note that constraints (4) allow waiting at pickup locations.

$$\min \sum_{r \in \Omega^+ \cup \Omega^-} X_r \tag{7}$$

$$\text{s.t.} \sum_{r \in \Omega^+ : c \in \mathscr{C}_r} X_r = 1 \qquad \forall c \in \mathscr{C} \tag{8}$$

$$\sum_{r \in \Omega^- : c \in \mathscr{C}_r} X_r = 1 \qquad \forall c \in \mathscr{C} \tag{9}$$

$$\sum_{r \in \Omega^+ : D_r = c} X_r - \sum_{\hat{r} \in \Omega^- : D_{\hat{r}} = c} X_{\hat{r}} = 0 \qquad \forall c \in \mathscr{C} \tag{10}$$

$$X_r \in \{0,1\} \qquad \forall r \in \Omega^+ \cup \Omega^- \tag{11}$$

Fig. 2 The Model for the CTSP.

Similar to [10,3,2], this work assumes on any day, each commuter c would specify a desired arrival time at the destination of her inbound trip, at_c^+, and a desired departure time at the origin of her outbound trip, dt_c^-. It also assumes that each would tolerate a shift of $\pm\Delta$ to the desired times. Therefore, time windows of $[a_{d_c}, b_{d_c}] = [at_c^+ - \Delta, at_c^+ + \Delta]$ and $[a_{o_c}, b_{o_c}] = [dt_c^- - \Delta, dt_c^- + \Delta]$ are associated with the destinations of inbound trips and origins of outbound trips respectively. Consequently, time windows of $[a_{o_c}, b_{o_c}] = [a_{d_c} - \zeta_{o_c} - L_c, b_{d_c} - \zeta_{o_c} - \tau_{(o_c,d_c)}]$ and $[a_{d_c}, b_{d_c}] = [a_{o_c} + \zeta_{o_c} + \tau_{(o_c,d_c)}, b_{o_c} + \zeta_{o_c} + L_c]$ are assigned to the origins of inbound trips and destinations of outbound trips respectively. Similar to [9], each rider c is assumed to be willing to tolerate an $R\%$ extension to her direct-ride duration $\tau_{(o_c,d_c)}$; therefore $L_c = (1+R) \cdot \tau_{(o_c,d_c)}$. Finally, this work assumes utilization of a homogeneous fleet of vehicles with capacity K to serve all rides, and that all travel times and distances satisfy the triangle inequality.

3 The Column-Generation Algorithm for the CTSP

The CTSP formulation uses Ω^+ and Ω^- to denote the set of all feasible inbound and outbound routes to serve a set of commuters $\mathscr{C}$. Binary variable X_r indicates whether route $r \in \Omega^+ \cup \Omega^-$ is used in the optimal plan. The problem is defined in Figure 2. Objective function (7) minimizes the number of routes used. Constraints (8) and (9) enforce coverage of each rider's inbound and outbound trips by exactly one route each, while constraints (10) ensure the sets of drivers for inbound and outbound routes are identical.

The problem has been successfully solved using a column-generation algorithm by [8]. The algorithm leverages the shadow prices of the constraints of a restricted master problem (RMP)—the linear relaxation of the original problem defined on a subset $\Omega^{+\prime} \cup \Omega^{-\prime}$ of all feasible routes—to calculate the reduced costs of routes. A pricing subproblem (PSP) is executed alternately with the RMP to search for new routes with negative reduced costs which are then added to $\Omega^{+\prime} \cup \Omega^{-\prime}$. The RMP and PSP are solved repeatedly until the PSP is unable to find any new route with negative reduced cost, at which point the optimal objective value of the RMP converges to the optimal value z^* of the linear relaxation of the original problem. An integer solution is then obtained by solving the RMP as a MIP, and its quality is evaluated by calculating an optimality gap which uses z^* as the primal lower bound.

The PSP considers each rider $c \in \mathscr{C}$ as the driver of an inbound route r_c^+ and an outbound route r_c^-, searches for such routes with minimum reduced costs, and adds them to $\Omega^{+\prime} \cup \Omega^{-\prime}$ should the costs be negative. The routes are found by first constructing a pair of graphs $\mathscr{G}_c^+ = (\mathscr{N}_c^+, \mathscr{A}_c^+)$ and $\mathscr{G}_c^- = (\mathscr{N}_c^-, \mathscr{A}_c^-)$ for each driver c whose nodes $\mathscr{N}_c^+$

and $\mathcal{N}_c^-$ represent the locations (the origins and destinations) of all inbound and outbound trips respectively and whose edges $\mathcal{A}_c^+$ and $\mathcal{A}_c^-$ represent location pairs that satisfy a priori feasibility constraints. Edge costs are calculated using the dual optimal solution of the RMP so that the total cost of any path from o_c to d_c is equivalent to the path's reduced cost. A wait-time relaxation algorithm based on the label-setting, dynamic-programming algorithm by Desrochers [4] is then used to search for the minimum-cost feasible route from o_c to d_c from each graph. The algorithm searches for a preliminary feasible route with relaxed wait times and then verifies its feasibility with the inclusion of wait times by solving the route-scheduling LP of (1)–(6). Infeasible routes are added to a set of forbidden paths whose members are prevented from being discovered in subsequent runs, and the algorithm is executed repeatedly until a feasible solution is found.

4 Robust Planning for the FCTSP and the RT-CTSP

In a practical setting for the CTSP, on any day, each commuter would make her trip request in the morning, specifying the desired arrival time at her workplace and the expected departure time for her return trip. The ride-sharing platform would then generate an optimal routing plan based on these times which consists of the day's designated drivers, the passengers they need to cover in their inbound and outbound routes, and the corresponding routes they need to take.

This work takes a step further by considering a setting in which the desired arrival times are certain as they occur in the morning, soon after the requests are made. However, the departure times might change due to unforeseen events occurring later in the day. The FCTSP considers a setting whereby the departure times are confirmed by a fixed deadline, whereas the RT-CTSP considers a more dynamic setting whereby the departure times are confirmed in real time with some advance notice. Regardless of the problem setting, an inbound routing plan which commits drivers for the day needs to be generated before the departure times can be confirmed. Therefore, *the driver assignment needs to be robust to ensure that they can cover as many return trips as possible, as the uncovered trips will need to be served by an external, more expensive resource.*

Most works on robust planning for transportation scheduling focus on addressing demand uncertainty. For instance, Serra et al. [14] proposed a two-stage stochastic programming formulation for handling uncertain passengers in the integrated last-mile transportation problem [11, 12], a problem which considers a fleet of shared vehicles working in concert with a mass transit service to provide last-mile passenger transportation. To our knowledge, Srour et al. [15] were the first tackle service-time uncertainty in the VRPTW. They utilized scenario sampling, a method which assumes the differences between confirmed and forecast service times of every customer are random variables whose distributions can be gleaned from historical data. A sampled scenario is obtained by drawing presumed service times from every customer distribution, and the stochastic information from several sampled scenarios is leveraged by deriving a routing plan for each scenario and selecting the plan that most resembles every other plan using a consensus metric.

This work incorporates scenario sampling into a two-stage approach to produce robust plans for the FCTSP and the RT-CTSP. The first stage optimizes selection of drivers and generation of inbound routes by solving a model that optimizes route selection for a single inbound scenario and a set of sampled outbound scenarios simultaneously. The second stage reoptimizes the outbound routing plan once departure times have been confirmed (FCTSP)

$$\min \sum_{r\in\Omega^+} X_r + \sum_{s\in\mathscr{S}} \sum_{r\in\Omega_s^-} X_r \tag{12}$$

$$\text{s.t.} \sum_{r\in\Omega^+ : c\in\mathscr{C}_r} X_r = 1 \qquad \forall c\in\mathscr{C} \tag{13}$$

$$\sum_{r\in\Omega_s^- : c\in\mathscr{C}_r} X_r = 1 \qquad \forall s\in\mathscr{S}, \forall c\in\mathscr{C} \tag{14}$$

$$\sum_{r\in\Omega^+ : D_r=c} X_r - \sum_{\hat{r}\in\Omega_s^- : D_{\hat{r}}=c} X_{\hat{r}} = 0 \qquad \forall s\in\mathscr{S}, \forall c\in\mathscr{C} \tag{15}$$

$$X_r \in \{0,1\} \qquad \forall s\in\mathscr{S}, \forall r\in\Omega^+\cup\Omega_s^- \tag{16}$$

Fig. 3 The First-Stage Model for Optimizing Selection of Drivers and Inbound Routes.

or reoptimizes the outbound plan in a rolling-horizon approach (RT-CTSP) as departure times are progressively confirmed over time.

4.1 Stage 1: Optimizing Selection of Drivers and Inbound Routes

In the first stage, a model which selects drivers for the day together with their inbound routes is optimized. A single inbound scenario is derived from the commuters' desired arrival times, while a set of outbound scenarios is generated either by using the expected departure times to obtain a scenario where the commuters don't change their return schedules or by sampling the departure-time distributions to obtain one where the commuters do. Let $\mathscr{S}$ denote the set of outbound scenarios and Ω_s^- denote the set of all feasible outbound routes for scenario s. The model is defined in terms of a binary variable X_r which indicates whether a route r is selected for the optimal plan. It is defined in Figure 3. Objective function (12) minimizes the total number of selected routes. Constraints (13) enforce coverage of each commuter's inbound trip by exactly one route, while constraints (14) do the same for each commuter's outbound trip in each scenario $s\in\mathscr{S}$. Finally, constraints (15) ensure the set of drivers selected for the inbound trips is identical to that for each outbound scenario $s\in\mathscr{S}$.

A column-generation algorithm similar to that used in [8] is utilized to obtain a high-quality solution for the model. An RMP is first introduced as the linear relaxation of the model defined on a subset of all feasible routes, $\{\Omega^{+\prime}\cup\Omega_s^{-\prime} : s\in\mathscr{S}\}$. Let π_c^+, $\pi_{c,s}^-$, and $\sigma_{c,s}$ denote the optimal duals of constraints (13), (14), and (15) of the RMP respectively. The reduced cost of an inbound route r^+ is then given by:

$$rc_{r^+} = 1 - \sum_{c\in\mathscr{C}_{r^+}} \pi_c^+ - \sum_{s\in\mathscr{S}} \sigma_{D_{r^+},s} \tag{17}$$

while that of an outbound route for scenario s, r_s^-, is given by:

$$rc_{r_s^-} = 1 - \sum_{c\in\mathscr{C}_{r_s^-}} \pi_{c,s}^- + \sigma_{D_{r_s^-},s} \tag{18}$$

A PSP is then composed to find new routes with negative reduced costs. The PSP considers each rider $c\in\mathscr{C}$ as the driver of an inbound route r_c^+ and $|\mathscr{S}|$ outbound routes $\{r_{c,s}^- : s\in\mathscr{S}\}$. For each such route, the PSP finds one with minimum reduced cost and then selects those with negative reduced costs to augment $\{\Omega^{+\prime}\cup\Omega_s^{-\prime} : s\in\mathscr{S}\}$. To find

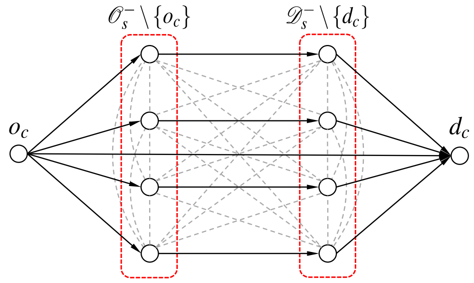

Fig. 4 Graph $\mathscr{G}_{c,s}^-$ After Edge Elimination (Each Dotted Line Represents a Pair of Bidirectional Edges).

these routes, a complete inbound graph $\mathscr{G}_c^+ = (\mathscr{N}_c^+, \mathscr{A}_c^+)$ and $|\mathscr{S}|$ complete outbound graphs $\{\mathscr{G}_{c,s}^- = (\mathscr{N}_{c,s}^-, \mathscr{A}_{c,s}^-) : s \in \mathscr{S}\}$ are first constructed for each driver c. The nodes in $\mathscr{N}_c^+$ represent all inbound origins $\mathscr{O}^+$ and destinations $\mathscr{D}^+$, whereas those in $\mathscr{N}_{c,s}^-$ represent all outbound origins $\mathscr{O}_s^-$ and destinations $\mathscr{D}_s^-$. A time window $[a_i, b_i]$ is associated with each inbound node $i \in \mathscr{O}^+ \cup \mathscr{D}^+$ based on requested trip times and similarly a time window $[a_{i,s}, b_{i,s}]$ is associated with each outbound node $i \in \mathscr{O}_s^- \cup \mathscr{D}_s^-$ based on trip times for scenario s. Edge costs are defined so that the total cost of any path from o_c to d_c is equivalent to the path's reduced cost. Let $c_{(i,j)}$ denote the cost of edge (i,j) and $\gamma^+(i)$ denote the set of outgoing edges of node i. Costs of edges $(i,j) \in \mathscr{A}_c^+$ are then given by:

$$c_{(i,j)} = \begin{cases} 1 - \pi_i^+ - \sum_{s \in \mathscr{S}} \sigma_{c,s} & \forall (i,j) \in \gamma^+(c) \\ -\pi_i^+ & \forall i \in \mathscr{O}^+ \setminus \{c\}, \forall (i,j) \in \gamma^+(i) \\ 0 & \forall i \in \mathscr{D}^+, \forall (i,j) \in \gamma^+(i) \end{cases} \tag{19}$$

while those of edges $(i,j) \in \mathscr{A}_{c,s}^-$ are given by:

$$c_{(i,j)} = \begin{cases} 1 - \pi_{i,s}^- + \sigma_{c,s} & \forall (i,j) \in \gamma^+(c) \\ -\pi_{i,s}^- & \forall i \in \mathscr{O}_s^- \setminus \{c\}, \forall (i,j) \in \gamma^+(i) \\ 0 & \forall i \in \mathscr{D}_s^-, \forall (i,j) \in \gamma^+(i) \end{cases} \tag{20}$$

A priori feasibility constraints proposed in [5,2,8] are then applied to all edges to identify and eliminate those that cannot belong to any feasible route. Figure 4 provides a sketch of $\mathscr{G}_{c,s}^-$ after edge elimination.

The wait-time relaxation algorithm from [8] is then applied to find the minimum-cost feasible route from o_c to d_c from each graph, and routes with negative costs are added to $\{\Omega^{+\prime} \cup \Omega_s^{-\prime} : s \in \mathscr{S}\}$. The RMP and PSP are solved repeatedly until the RMP converges. An upper and lower bound to z^* are maintained for this purpose. The upper bound is given by the optimal objective value of the RMP after each iteration, z_{RMP}, while the lower bound z_{LB} is calculated using the method by Farley [6], where $z_{\text{LB}} = z_{\text{RMP}}/(1 - rc^*)$ and rc^* denotes the smallest reduced cost discovered. Since constraints (15) restrict the objective value of any integer solution to be an integer multiple of $\beta = 1 + |\mathscr{S}|$, the column-generation procedure is terminated when $\beta \lceil z_{\text{RMP}}/\beta \rceil - z_{\text{LB}} < \beta$.

After convergence, an integer solution is obtained by solving the RMP as a MIP, and its quality is assessed by calculating its optimality gap. Let z_{MIP} denote the objective value of the MIP solution. The optimality gap is then given by $(z_{\text{MIP}} - z_{\text{RMP}})/z_{\text{MIP}}$ as z_{RMP} is the primal lower bound to z_{MIP}. The solution of the first stage is given by the set of selected inbound routes $\hat{\mathscr{Z}} = \{r \in \Omega^{+\prime} : X_r = 1\}$ and their corresponding drivers $\hat{\mathscr{D}} = \{D_r : r \in \hat{\mathscr{Z}}\}$.

$$\min \sum_{c\in\mathscr{C}} Y_c \tag{21}$$

$$\text{s.t.} \sum_{r\in\Omega^-_{\hat{\mathscr{D}}}:c\in\mathscr{C}_r} X_r + Y_c = 1 \qquad \forall c\in\mathscr{C} \tag{22}$$

$$X_r \in \{0,1\} \qquad \forall r\in\Omega^-_{\hat{\mathscr{D}}} \tag{23}$$

$$Y_c \in \{0,1\} \qquad \forall c\in\mathscr{C} \tag{24}$$

Fig. 5 The Second-Stage Model for the FCTSP.

4.2 The FCTSP

The FCTSP considers a second stage that reoptimizes the outbound routing plan once departure times have been confirmed by a fixed deadline. Let $\Omega^-_{\hat{\mathscr{D}}}$ be the set of all feasible outbound routes with drivers from $\hat{\mathscr{D}}$, i.e. $\Omega^-_{\hat{\mathscr{D}}} = \{r\in\Omega^- : D_r\in\hat{\mathscr{D}}\}$. Its model is defined in terms of two binary variables: X_r to indicate selection of route $r\in\Omega^-_{\hat{\mathscr{D}}}$ and Y_c to indicate if rider c cannot be covered in the outbound routing plan. The model is defined in Figure 5. The objective function minimizes the number of uncovered riders and constraints (22) ensure Y_c is set to 1 if rider c cannot be covered by any outbound route.

This model is solved using a column-generation algorithm similar to that used in the first stage. The key difference is that only outbound routes driven by riders in $\hat{\mathscr{D}}$ are generated in this model. Letting λ_c denote the optimal dual of constraints (22) of the model's RMP, the reduced cost of outbound route r is given by:

$$rc_r = -\sum_{c\in\mathscr{C}_r} \lambda_c \tag{25}$$

The PSP of this model finds routes with negative reduced costs by first constructing a graph $\hat{\mathscr{G}}^-_c = (\hat{\mathscr{N}}^-_c, \hat{\mathscr{A}}^-_c)$ for each driver $c\in\hat{\mathscr{D}}$, finding the route with minimum reduced cost from each $\hat{\mathscr{G}}^-_c$, and adding the route to $\Omega^{-\prime}_{\hat{\mathscr{D}}}$ if the cost is negative. The nodes $\hat{\mathscr{N}}^-_c$ consist of all outbound origins $\mathscr{O}^-$ and destinations $\mathscr{D}^-$, and each node $i\in\mathscr{O}^-\cup\mathscr{D}^-$ has associated with it a time window $[a_i,b_i]$ based on *confirmed outbound trip times*. Costs of edges $(i,j)\in\hat{\mathscr{A}}^-_c$ are given by:

$$c_{(i,j)} = \begin{cases} -\lambda_i & \forall i\in\mathscr{O}^- \\ 0 & \forall i\in\mathscr{D}^- \end{cases} \tag{26}$$

so that the total cost of any path from o_c to d_c is equivalent to rc_r.

The same wait-time relaxation algorithm from the first stage is used to find the minimum-cost feasible route from o_c to d_c from each $\hat{\mathscr{G}}^-_c$. The RMP and PSP are solved repeatedly until the PSP cannot find any new route with negative reduced cost, after which the RMP is solved as a MIP to obtain an integer solution.

4.3 The RT-CTSP

Consider now the RT-CTSP where departure times are continuously confirmed by riders over time. The RT-CTSP requires each rider i to confirm her actual departure time, dt_i^{actual}, in advance by at least a time interval Δ_{lead}. In other words, dt_i^{actual} is confirmed at time ct_i and

Algorithm 1 Rolling-Horizon Optimization of Outbound Routing Plan

$k \leftarrow 1$
$ot_k \leftarrow 0$, $\mathscr{Z}_{k-1} \leftarrow \emptyset$
$\mathscr{Z}_{\text{rolling-horizon}} \leftarrow \emptyset$, $\mathscr{U} \leftarrow \emptyset$
while $\mathscr{C} \neq \emptyset$ **do**
    $\mathscr{Z}_{\text{rolling-horizon}} \leftarrow \mathscr{Z}_{\text{rolling-horizon}} \cup \{r \in \mathscr{Z}_{k-1} : st_r \leq ot_k\}$
    $\mathscr{C} \leftarrow \mathscr{C} \setminus \{i \in \mathscr{C}_r : r \in \mathscr{Z}_{\text{rolling-horizon}}\}$
    $\mathscr{U} \leftarrow \mathscr{U} \cup \{i \in \mathscr{C} : b_{o_i} \leq ot_k\}$
    $\mathscr{C} \leftarrow \mathscr{C} \setminus \mathscr{U}$
    $\mathscr{C}_k \leftarrow \{i \in \mathscr{C} : ct_i \leq ot_k\}$
    $\mathscr{Z}_k \leftarrow$ Solution of FCTSP reoptimization on outbound trips of riders in $\mathscr{C}_k$
    $k \leftarrow k+1$
    $ot_k \leftarrow ot_{k-1} + \Delta_{\text{opt}}$
return $\mathscr{Z}_{\text{rolling-horizon}}$, $\mathscr{U}$

$ct_i + \Delta_{\text{lead}} \leq dt_i^{\text{actual}}$. Δ_{lead} will be referred to henceforth as the lead time, and it is assumed to be identical for every rider.

A rolling-horizon approach which executes the FCTSP optimization algorithm from Section 4.2 once every Δ_{opt}, where Δ_{opt} is a fixed time interval, is proposed for this setting. Let ot_k denote the time of the k^{th} optimization run of this approach and $\mathscr{Z}_k$ be its solution (i.e., its set of selected routes). Each optimization run includes a batch of trips whose departure times have been confirmed, i.e., the trips of riders $\{i \in \mathscr{C} : ct_i \leq ot_k\}$. At the same time, a different set of trips is excluded from the k^{th} run. These are trips whose departure-time windows have expired, i.e., trips of riders $\{i \in \mathscr{C} : b_{o_i} \leq ot_k\}$, and trips covered by routes from $\mathscr{Z}_{k-1}$ that have already departed. Let st_r denote the starting time of route r, then the latter set of excluded trips are those of riders $\{i \in \mathscr{C}_r : r \in \mathscr{Z}_{k-1}, st_r \leq ot_k\}$.

The scheme is executed until each rider $i \in \mathscr{C}$ is either served by a departed route or has her departure-time window expire without being served by any route. Riders not served by any route will have to resort to an external resource to cover their return trips. The rolling-horizon algorithm is summarized in Algorithm 1. Riders not served by any route are stored in a set of uncovered riders $\mathscr{U}$, whereas routes that have departed throughout the execution are stored in $\mathscr{Z}_{\text{rolling-horizon}}$ which represents the final solution of the approach.

5 Computational Results

The Experimental Setting The algorithms are evaluated on a dataset of real-world commute trips which is constructed from access information to 15 parking structures in downtown Ann Arbor, Michigan, collected throughout April 2017. The data is joined with the home addresses of customers to reconstruct their daily commute trips. The experiments focus on trips of commuters living within city limits which amounts to approximately 2,400 trips per weekday. To maintain tractability, the trips are partitioned into smaller problem instances by clustering the commuters based on their home locations into clusters of size $n \approx 300$ and only considering ride sharing intra-cluster.

Using historical data, a Laplace distribution is fit to the set of daily departure times of each commuter using maximum likelihood estimation and the distribution is sampled to generate outbound scenarios. For a selected day, the inbound scenario for a set of commuters $\mathscr{C}$ is obtained from their arrival times from the dataset. Departure times from the dataset, however, are treated as expected departure times, and they are used to construct an expected outbound scenario, s_{expected}. Actual departure times are simulated by first select-

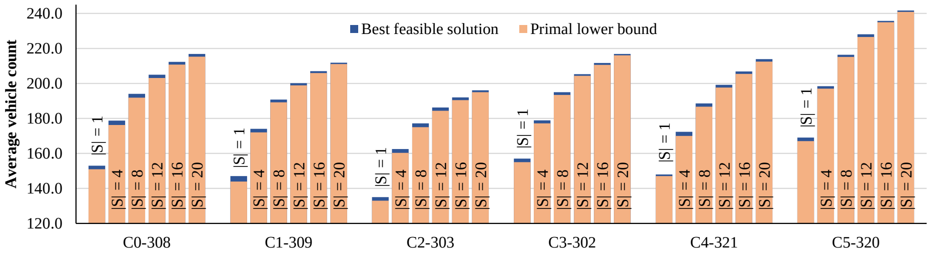

Fig. 6 Average Vehicle Counts from the First-Stage Model for Clusters C0-308, C1-309, C2-303, C3-302, C4-321, and C5-320.

ing a fraction f of the commuters uniformly at random and then sampling their departure-time distributions. The set $\mathscr{C}_f$ of selected commuters represents those who had to change their return schedules. Actual departure times of the remaining commuters in $\mathscr{C} \setminus \mathscr{C}_f$ stay at their expected values. The experiments consider scenario sets of various sizes, but every set contains s_{expected} to ensure every return trip can be covered when there are absolutely no schedule changes, as s_{expected} corresponds to the scenario where $f = 0.0$. Therefore when $|\mathscr{S}| = 1$, $\mathscr{S} = \{s_{\text{expected}}\}$. The algorithms are implemented in C++ and they invoke Gurobi 7.5.1 to solve LPs and MIPs. All experiments use $K = 4$, $\Delta = 10$ mins, and $R = 0.50$, and they are conducted on a high-performance computing cluster with 12 cores of a 2.5 GHz Intel Xeon E5-2680v3 processor and 32 GB of RAM.

The Impact of the Number of Scenarios on the Number of Vehicles The first set of experiments vary the number of scenarios in the first stage to measure the "price" of robustness, i.e., its impact on the number of vehicles. Table 1 summarizes the average results from solving the first-stage model on eight clusters and 20 instances for each $|\mathscr{S}| \in \{1, 4, 8, 12, 16, 20\}$. Its first three columns list cluster IDs, cluster sizes, and the number of sampled outbound scenarios used. The next three display the model's results in terms of the average number of columns generated, the average vehicle count, and the average optimality gap of the solution. The final three columns show average times spent on solving the RMP, the MIP, and the problem instance as a whole. Note that a 60 s time limit was placed on the MIP solver, and the vehicle count reported is from its best feasible solution. The time limit allows each problem instance to be solvable in < 15 minutes, and despite the time limit, every instance produced an optimality gap of $< 3\%$. Figure 6 summarizes the effect of increasing $|\mathscr{S}|$ for the first six clusters (the results for the remaining two are consistent). The key observation is that the vehicle count increases with $|\mathscr{S}|$, a result which can be attributed to the model having to cater to more outbound scenarios.

The FCTSP The second set of results considers the FCTSP on the same instances for $f \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$: $f = 0.0$ models a scenario whereby none of the commuters change their departure times, whereas $f = 1.0$ models one in which every commuter does. The results are summarized in Table 2. Similar to Table 1, the first three columns display cluster IDs, cluster sizes, and sampled scenario counts, while the next lists the vehicle count results of the first-stage model. The final six columns show the average number of uncovered riders resulting from the reoptimized outbound plans of 25 samples for each f value. Computation times are not shown as every instance was solved in less that 15 s. Figure 7 demonstrates how increasing the number of sampled scenarios improves the robustness of the outbound plan by showing the average number of uncovered riders for cluster C1-309

Table 1 First-Stage Optimization Results for $|\mathscr{S}| \in \{1,4,8,12,16,20\}$

Cluster ID	Cluster size	Scenario count	Average column count	Average vehicle count	Average optimality gap (%)	Average wall time (s)		
						RMP	MIP	Total
C0-308	308	1	4883.0	153.0	1.31	76.3	3.8	80.1
		4	10081.0	178.8	1.46	159.8	49.1	208.9
		8	16695.2	194.1	1.16	277.1	46.6	323.7
		12	23013.4	205.1	0.93	392.3	30.4	422.8
		16	29373.8	212.3	0.75	510.5	19.5	530.0
		20	35840.6	216.8	0.72	629.9	17.9	647.9
C1-309	309	1	5398.0	147.0	2.04	90.1	3.2	93.3
		4	11459.5	174.1	1.12	188.8	43.9	232.7
		8	18684.2	190.8	0.79	297.0	25.1	322.1
		12	25834.0	200.1	0.63	418.9	17.2	436.1
		16	32887.2	207.1	0.55	541.0	16.8	557.8
		20	40018.5	211.9	0.43	658.4	8.4	666.8
C2-303	303	1	5508.0	135.0	1.48	112.9	34.4	147.3
		4	12066.0	162.5	1.33	256.8	57.0	313.8
		8	20178.2	177.2	1.24	361.9	53.1	415.0
		12	27980.1	186.2	0.99	469.9	48.5	518.4
		16	35783.6	192.1	0.83	607.4	48.1	655.5
		20	43499.9	196.2	0.61	753.7	42.9	796.6
C3-302	302	1	4343.0	157.0	1.27	59.4	2.2	61.6
		4	9102.5	179.0	1.00	143.1	38.4	181.5
		8	15134.6	195.1	0.79	252.7	26.0	278.7
		12	21046.1	205.4	0.51	366.0	14.7	380.7
		16	26973.7	211.7	0.52	468.0	11.0	478.9
		20	32909.5	216.9	0.39	573.5	7.6	581.0
C4-321	321	1	5828.0	148.0	0.68	88.2	60.0	148.2
		4	12828.7	172.4	1.42	226.4	55.3	281.7
		8	21312.7	188.6	0.98	371.4	47.7	419.1
		12	29317.1	199.2	0.78	500.7	32.4	533.0
		16	37375.4	206.9	0.68	649.2	25.5	674.7
		20	45243.6	213.9	0.63	780.9	20.1	801.0
C5-320	320	1	4457.0	169.0	1.18	66.4	6.7	73.0
		4	9193.4	198.4	0.68	162.0	19.2	181.2
		8	15239.4	216.3	0.56	286.5	8.0	294.5
		12	21258.2	228.1	0.68	409.9	9.5	419.4
		16	27342.9	235.8	0.32	532.7	4.2	536.9
		20	33495.3	241.7	0.29	661.2	2.1	663.3
C6-300	300	1	5479.0	137.0	1.46	83.5	43.2	126.8
		4	11953.5	160.3	1.53	182.9	57.5	240.5
		8	19708.7	177.3	1.10	291.4	40.8	332.2
		12	27263.2	186.9	0.75	402.2	26.8	429.0
		16	34790.7	194.5	0.62	512.7	22.6	535.3
		20	42193.0	200.3	0.50	612.2	15.5	627.7
C7-299	299	1	3952.0	165.0	0.61	61.9	12.2	74.1
		4	8724.6	190.8	0.71	145.4	15.8	161.2
		8	14866.7	207.6	0.48	252.1	19.5	271.6
		12	20976.4	216.6	0.51	356.6	11.6	368.2
		16	27013.2	223.4	0.40	468.3	9.6	477.9
		20	33037.1	227.9	0.28	575.9	4.8	580.7

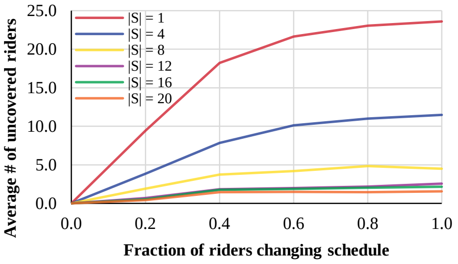

Fig. 7 Average Number of Uncovered Riders from the FCTSP for Cluster C1-309.

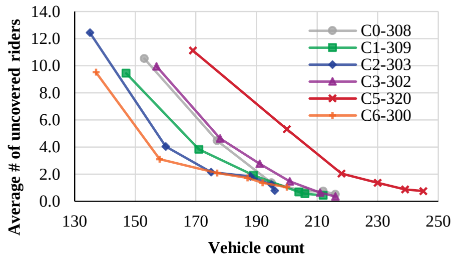

Fig. 8 Average Number of Uncovered Riders and Vehicle Count Results of the FCTSP for Several Clusters with $|\mathscr{S}| = \{1,4,8,12,16,20\}$ and $f = 0.2$.

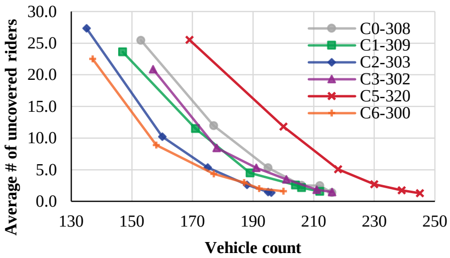

Fig. 9 Average Number of Uncovered Riders and Vehicle Count Results of the FCTSP for Several Clusters with $|\mathscr{S}| = \{1,4,8,12,16,20\}$ and $f = 1.0$.

for $f \in [0.0, 1.0]$. As expected, all riders are covered when $f = 0.0$ (as every set $\mathscr{S}$ contains s_{expected}). Moreover, the number of uncovered riders generally increases with f for any fixed $|\mathscr{S}|$, indicating the increasing challenge of accommodating progressively more changing schedules. However, when f is fixed, the average number of uncovered riders becomes smaller as $|\mathscr{S}|$ is increased, signifying an increase in plan robustness. The marginal benefits, however, diminish with increasing $|\mathscr{S}|$, and they come at the price of increases in vehicle count as shown in Figure 6. This trade-off is further illustrated in Figures 8 and 9 which plot the average number of uncovered riders for several clusters against their corresponding vehicle counts for each $|\mathscr{S}| \in \{1,4,8,12,16,20\}$ when $f = 0.2$ and $f = 1.0$ respectively. Each point in the figures represents the average number of uncovered riders and the vehicle count results for a specific $|\mathscr{S}|$ value and, for each curve, as $|\mathscr{S}|$ increases going from left to right, so do plan robustness and vehicle count, highlighting the trade-off between robustness and vehicle reduction.

The RT-CTSP The next results consider the RT-CTSP on the same instances for $\Delta_{\text{opt}} = 10$ mins, $|\mathscr{S}| \in \{1, 12\}$, and $\Delta_{\text{lead}} \in \{30, 60, 90\}$ mins. Table 3 summarizes these results. The first four columns list cluster IDs, cluster sizes, sampled scenario counts, and lead times values, while the remaining six show results of the algorithm for each $f \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ in terms of the average number of uncovered riders from 25 samples for each f value. Figure 10 then illustrates the results for cluster C1-309 and includes results of the FCTSP for additional perspective. As expected, the robustness of the rolling-horizon method is never better than that of the FCTSP. However, results of the former is only slightly worse than the latter when $|\mathscr{S}| = 12$, demonstrating the viability of the real-time approach. Its results also quickly approach those of the FCTSP as the lead time is increased.

Evaluating the Plan Robustness-Vehicle Reduction Trade-Off Figure 11 reexamines the average uncovered riders-vehicle count curve of cluster C2-303 when $f = 1.0$, which is a

Table 2 FCTSP Results for $|\mathscr{S}| \in \{1,4,8,12,16,20\}$ and $f \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

Cluster ID	Cluster size	Scenario count	Vehicle count	Average # of uncovered riders					
				$f=0.0$	$f=0.2$	$f=0.4$	$f=0.6$	$f=0.8$	$f=1.0$
C0-308	308	1	153	0.0	10.5	17.1	21.3	24.8	25.4
		4	177	0.0	4.5	8.1	10.0	10.9	12.0
		8	195	0.0	1.4	3.2	4.0	4.8	5.3
		12	206	0.0	0.7	1.8	2.1	2.8	2.5
		16	212	0.0	0.8	1.3	1.5	2.2	2.5
		20	216	0.0	0.5	1.0	1.2	1.4	1.4
C0-308	308	1	153	0.0	10.5	17.1	21.3	24.8	25.4
		4	177	0.0	4.5	8.1	10.0	10.9	12.0
		8	195	0.0	1.4	3.2	4.0	4.8	5.3
		12	206	0.0	0.7	1.8	2.1	2.8	2.5
		16	212	0.0	0.8	1.3	1.5	2.2	2.5
		20	216	0.0	0.5	1.0	1.2	1.4	1.4
C2-303	303	1	135	0.0	12.4	20.1	25.8	27.2	27.4
		4	160	0.0	4.0	7.7	9.8	10.0	10.2
		8	175	0.0	2.2	3.8	5.2	5.3	5.4
		12	188	0.0	1.8	2.2	2.2	2.3	2.6
		16	195	0.0	1.2	1.5	1.4	1.4	1.5
		20	196	0.0	0.8	1.2	1.1	1.1	1.4
C3-302	302	1	157	0.0	10.0	17.4	19.3	21.9	20.9
		4	178	0.0	4.6	8.1	7.3	8.6	8.4
		8	191	0.0	2.8	4.8	4.6	5.5	5.3
		12	201	0.0	1.5	3.5	2.8	3.9	3.5
		16	211	0.0	0.6	2.2	1.7	2.3	1.8
		20	216	0.0	0.4	1.5	1.0	1.9	1.5
C4-321	321	1	148	0.0	9.3	18.2	22.4	24.0	25.0
		4	169	0.0	5.2	9.4	11.4	11.4	12.8
		8	184	0.0	3.0	4.8	7.0	6.2	7.7
		12	195	0.0	1.4	2.6	3.8	3.6	4.6
		16	203	0.0	1.0	1.9	3.2	2.5	3.2
		20	211	0.0	0.8	1.6	2.4	2.0	2.1
C5-320	320	1	169	0.0	11.1	20.5	26.7	26.8	25.5
		4	200	0.0	5.3	10.2	12.1	11.8	11.8
		8	218	0.0	2.0	4.8	5.8	5.4	5.1
		12	230	0.0	1.4	2.8	3.3	3.1	2.7
		16	239	0.0	0.9	1.5	1.8	1.9	1.8
		20	245	0.0	0.8	1.2	1.3	1.4	1.3
C6-300	300	1	137	0.0	9.5	17.6	23.5	22.5	22.5
		4	158	0.0	3.1	6.8	9.1	9.2	8.9
		8	177	0.0	2.1	4.5	5.2	4.9	4.4
		12	187	0.0	1.7	3.2	3.9	3.9	3.0
		16	192	0.0	1.4	1.7	2.6	2.5	2.0
		20	200	0.0	1.0	1.3	1.8	2.0	1.6
C7-299	299	1	165	0.0	8.6	17.4	21.7	23.4	24.0
		4	192	0.0	3.9	7.3	8.0	8.0	9.4
		8	211	0.0	1.6	2.8	3.3	4.7	3.8
		12	218	0.0	1.8	2.8	3.0	3.7	3.7
		16	227	0.0	0.8	1.5	1.2	2.4	1.6
		20	228	0.0	0.6	1.4	1.1	2.1	1.4

Table 3 RT-CTSP Results for $\Delta_{\text{opt}} = 10$ mins, $|\mathscr{S}| \in \{1,12\}$, $\Delta_{\text{lead}} \in \{30,60,90\}$ mins, and $f \in \{0.0,0.2,0.4,0.6,0.8,1.0\}$.

Cluster ID	Cluster size	Scenario count	Lead time (mins)	Average # of uncovered riders					
				$f = 0.0$	$f = 0.2$	$f = 0.4$	$f = 0.6$	$f = 0.8$	$f = 1.0$
C0-308	308	1	30	9.0	17.8	24.2	28.7	33.0	33.2
			60	1.0	11.8	18.1	22.5	25.8	26.6
			90	0.0	11.1	17.4	21.6	25.2	25.7
		12	30	0.0	1.7	2.9	3.5	4.8	4.6
			60	0.0	0.8	1.8	2.1	2.8	2.6
			90	0.0	0.7	1.8	2.1	2.8	2.5
C1-309	309	1	30	5.0	17.0	26.4	31.7	31.5	32.6
			60	0.0	11.1	19.8	23.4	24.6	24.9
			90	0.0	10.0	18.7	22.0	23.5	24.0
		12	30	0.0	1.8	3.0	3.6	4.4	4.0
			60	0.0	0.7	1.8	2.0	2.2	2.6
			90	0.0	0.7	1.8	2.0	2.2	2.6
C2-303	303	1	30	6.0	19.6	28.2	33.7	34.9	35.6
			60	0.0	14.1	21.4	27.3	28.4	28.8
			90	0.0	13.0	20.5	26.4	27.6	27.9
		12	30	3.0	4.3	4.6	4.9	5.2	6.4
			60	0.0	2.2	2.4	2.4	2.4	2.8
			90	0.0	1.9	2.2	2.2	2.3	2.6
C3-302	302	1	30	3.0	16.4	24.7	26.5	28.5	28.1
			60	0.0	11.1	18.4	20.5	22.2	21.8
			90	0.0	10.4	17.6	19.5	22.0	21.0
		12	30	0.0	2.9	5.4	5.1	6.3	5.4
			60	0.0	1.6	3.6	3.0	4.1	3.6
			90	0.0	1.5	3.5	2.8	3.9	3.5
C4-321	321	1	30	6.0	17.3	27.0	31.5	32.7	34.1
			60	0.0	11.0	20.0	24.4	26.1	26.9
			90	0.0	10.0	18.8	23.0	24.7	25.6
		12	30	2.0	4.4	5.6	6.3	6.0	8.0
			60	0.0	1.6	2.8	3.9	3.8	5.1
			90	0.0	1.4	2.6	3.8	3.6	4.7
C5-320	320	1	30	1.0	16.3	27.5	32.1	32.7	31.6
			60	0.0	12.2	21.7	27.3	27.7	26.1
			90	0.0	11.6	20.8	26.7	27.0	25.6
		12	30	1.0	2.8	4.0	4.9	4.4	3.8
			60	0.0	1.5	2.9	3.4	3.1	2.8
			90	0.0	1.4	2.8	3.3	3.1	2.7
C6-300	300	1	30	8.0	16.8	26.3	32.4	30.9	30.6
			60	0.0	11.2	19.4	25.6	24.0	24.0
			90	0.0	10.1	18.2	23.9	22.9	23.1
		12	30	2.0	3.1	5.5	6.6	6.3	6.0
			60	0.0	1.8	3.3	4.1	4.1	3.0
			90	0.0	1.7	3.3	4.0	3.9	3.0
C7-299	299	1	30	5.0	14.6	22.9	27.0	28.3	29.6
			60	0.0	9.6	18.4	22.7	23.8	24.6
			90	0.0	8.7	17.6	21.8	23.4	24.1
		12	30	1.0	3.3	4.2	4.6	5.0	5.0
			60	0.0	2.0	3.0	3.1	3.8	3.7
			90	0.0	1.9	2.8	3.0	3.7	3.7

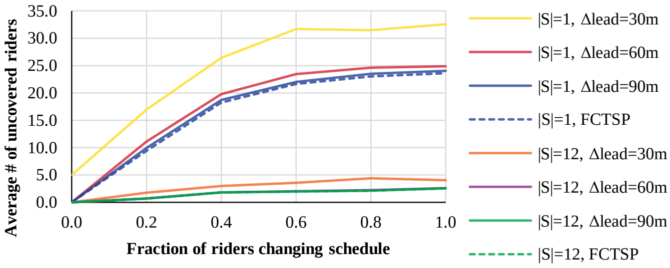

Fig. 10 Average Number of Uncovered Riders from the RT-CTSP for Cluster C1-309.

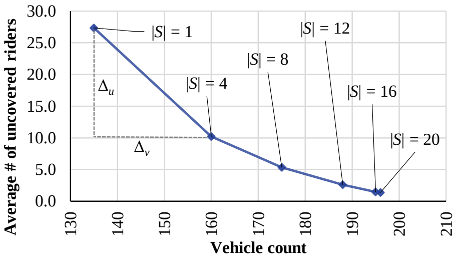

Fig. 11 Average Uncovered Riders-Vehicle Count Curve of Cluster C2-303 when $f = 1.0$.

worst-case scenario obviously. Each point on the curve represents a potential operating point for the ride-sharing platform, and the goal is to find the point with the best trade-off between plan robustness and vehicle reduction. Let Δ_u and Δ_v denote the change in the number of uncovered riders and the change in vehicle count respectively resulting from moving between two points along the curve, and let c_u and c_v be the cost per uncovered rider and the cost per unit of vehicle increase respectively. The marginal cost when moving between two points is therefore given by $\Delta_u c_u + \Delta_v c_v$. Having negative marginal cost will lead to reductions in operating cost, and therefore moving from one point on the curve to another is beneficial when $\Delta_u c_u + \Delta_v c_v < 0$. Rearranging the inequality results in $c_v/c_u < -\Delta_u/\Delta_v$ where Δ_u/Δ_v is given by the slope of the curve. Therefore, it is beneficial to move right along the curve when the c_v/c_u ratio is less then the negative of the curve's slope.

For instance, consider the leftmost curve segment in Figure 11 which has a slope of -0.69. Suppose the cost of an uncovered rider is given by the average price of using a ride-hailing service to cover a return trip, while the cost per unit of vehicle increase is well represented by the per-day cost of a parking lot (which in turn is given by its total cost amortized over its useful lifetime). Then suppose a scenario in which $c_u = \$15$ and $c_v = \$8$. The c_v/c_u ratio is 0.53 which is less than the negative of the slope. Therefore it is beneficial to move from the point with $|\mathscr{S}| = 1$ to that with $|\mathscr{S}| = 4$ as it will result in a marginal cost of $-\$57$. However, suppose an alternate scenario whereby $c_u = c_v = \$12$. The c_v/c_u ratio is now 1.0 which is larger than the negative of the slope. In this case, it is beneficial to just stay at the point where $|\mathscr{S}| = 1$ as moving to the right neighboring point will result in an increase in operating cost.

6 Conclusion

This paper proposes a two-stage algorithm for generating robust plans for the FCTSP and the RT-CTSP. It addresses a practical setting in which there are uncertainties associated with the schedules of outbound trips by incorporating scenario sampling, a method which assumes the availability of historical data on trip schedules to which probability distributions can be fit and sampled to obtain potential scenarios. A model which optimizes the routing plan for a single inbound and multiple sampled outbound scenarios is first solved to select daily drivers and their inbound routes. The outbound routing plan is then obtained either statically for the FCTSP or dynamically for the RT-FCTSP by solving a different model on just outbound trips once their schedules have been confirmed. When applied on a real-world dataset of commute trips, the results show that plan robustness generally increases with sampled scenario count. The only drawback is that the robustness is also accompanied by an increase in vehicle count. Therefore, a method which compares the per-unit price ratio of vehicle increase to uncovered riders is proposed to best evaluate the trade-off between plan robustness and vehicle reduction.

Acknowledgement

We would like to thank Stephen Dolen from Logistics, Transportation, and Parking of the University of Michigan for his assistance in obtaining the dataset used in this research. Part of this research was funded by NSF Grant 1854684 and the Rackham Graduate Student Research Grant. This research was also supported in part through computational resources and services provided by Advanced Research Computing at the University of Michigan.

References

1. Bent, R.W., Van Hentenryck, P.: Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research **52**(6), 977–987 (2004)
2. Cordeau, J.F.: A branch-and-cut algorithm for the dial-a-ride problem. Operations Research **54**(3), 573–586 (2006)
3. Cordeau, J.F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological **37**(6), 579–594 (2003)
4. Desrochers, M.: An algorithm for the shortest path problem with resource constraints. Tech. Rep. G-88-27, Les Cahiers du GERAD (1988)
5. Dumas, Y., Desrosiers, J., Soumis, F.: The pickup and delivery problem with time windows. European Journal of Operational Research **54**(1), 7–22 (1991)
6. Farley, A.A.: A note on bounding a class of linear programming problems, including cutting stock problems. Operations Research **38**(5), 922–923 (1990)
7. Hasan, M.H., Van Hentenryck, P., Budak, C., Chen, J., Chaudhry, C.: Community-based trip sharing for urban commuting. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, pp. 6589–6597. AAAI Press, California, USA (2018)
8. Hasan, M.H., Van Hentenryck, P., Legrain, A.: The commute trip sharing problem. arXiv e-prints arXiv:1904.11017 (2019)
9. Hunsaker, B., Savelsbergh, M.W.P.: Efficient feasibility testing for dial-a-ride problems. Operations Research Letters **30**(3), 169–173 (2002)
10. Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.: A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological **20**(3), 243–257 (1986)
11. Raghunathan, A.U., Bergman, D., Hooker, J., Serra, T., Kobori, S.: The integrated last-mile transportation problem (ILMTP). In: Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, pp. 388–397. AAAI Press, California, USA (2018)
12. Raghunathan, A.U., Bergman, D., Hooker, J., Serra, T., Kobori, S.: Seamless multimodal transportation scheduling. arXiv e-prints arXiv:1807.09676 (2018)
13. Savelsbergh, M.W.P.: Local search in routing problems with time windows. Annals of Operations Research **4**(1), 285–305 (1985)

14. Serra, T., Raghunathan, A.U., Bergman, D., Hooker, J., Kobori, S.: Last-mile scheduling under uncertainty. In: L.M. Rousseau, K. Stergiou (eds.) Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pp. 519–528. Springer International Publishing, Cham (2019)
15. Srour, F.J., Agatz, N., Oppen, J.: Strategies for handling temporal uncertainty in pickup and delivery problems with time windows. Transportation Science **52**(1), 3–19 (2018)