informs
http://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Articles in Advance, pp. 1–20
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Q: 1, 2

Vehicle Routing Problems with Synchronized Visits and Stochastic Travel and Service Times: Applications in Healthcare

Q: 3, 4

Hossein Hashemi Doulabi,[a,b] **Gilles Pesant,**[b,c] **Louis-Martin Rousseau**[b,d]

[a] Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, Canada; [b] Interuniversity Research Center on Enterprise Networks, Logistics and Transportation, Montreal, Canada; [c] Department of Computer and Software Engineering, Polytechnique Montreal, Montreal, Canada; [d] Department of Mathematics and Industrial Engineering, Polytechnique Montreal, Montreal, Canada

Contact: hossein.hashemi@concordia.ca, https://orcid.org/0000-0002-7385-1274 (HHD); gilles.pesant@polymtl.ca (GP); louis-martin.rousseau@polymtl.ca (L-MR)

Received: July 12, 2018
Revised: July 24, 2019
Accepted: November 4, 2019
Published Online in Articles in Advance: ■■ ■■, 2019

https://doi.org/10.1287/trsc.2019.0956

Abstract. This paper, for the first time, studies vehicle routing problems with synchronized visits (VRPS) and stochastic travel and service times. In addition to considering a home healthcare scheduling problem, we introduce an operating room scheduling problem with stochastic durations as a novel application of VRPS. We formulate VRPS with stochastic times as a two-stage stochastic integer programming model that, unlike the deterministic models in the VRPS literature, does not have any big-M constraints. This advantage comes at the cost of a large number of second-stage integer variables. We prove that the integrality constraints on second-stage variables can be relaxed, and therefore, we can apply the L-shaped algorithm and its branch-and-cut implementation to solve the problem. We enhance the model by developing valid inequalities and a lower bounding functional. We analyze the subproblems of the L-shaped algorithm and devise a specialized algorithm for them that is significantly faster than standard linear programming algorithms. Computational results show that the branch-and-cut algorithm optimally solves stochastic home healthcare scheduling instances with 15 patients and 10%–30% of synchronized visits. It also finds solutions with an average optimality gap of 3.57% for instances with 20 patients. Furthermore, the branch-and-cut algorithm optimally solves stochastic operating room scheduling problems with 20 surgeries.

Supplemental Material: The online appendix is available at https://doi.org/10.1287/trsc.2019.0956.

Keywords: vehicle routing with synchronized visits • stochastic travel and service times • two-stage stochastic integer programming

Q: 5

Q: 6

1. Introduction

Q: 7, 8, 9

Q: 10

In the literature of vehicle routing problem (VRP), a large number of studies have taken into account the scheduling of services to customers in addition to the routing of vehicles. Such routing problems are generally referred to as routing and scheduling problems. The most well-known problem in this area is vehicle routing problem with time windows (VRPTW), where a number of vehicles must serve customers with minimum cost while satisfying some time windows constraints. Recently, there has been an emergent interest in vehicle routing problem with synchronized visits (VRPS) in which two or more vehicles of different types must be simultaneously available at the customer's location for serving. VRPS has a wide range of applications in home healthcare scheduling (Bredström and Rönnqvist 2008; Di Mascolo, Espinouse, and Ozkan 2014), raw milk collection (Drexl and Sebastian 2007), staff scheduling (Lim, Rodrigues, and Song 2004; Li, Lim, and Rodrigues 2005), garbage collection (De Rosa et al. 2002), forest management (Paraskevopoulos et al. 2016), and telecommunication (Jaumard et al. 2016). From a modeling perspective, considering the vehicle synchronization is an essential assumption in many of the mentioned applications. In the absence of this feature, some of these problems, such as the milk collection and home healthcare scheduling problems, are oversimplified and do not represent what happens in the real world. Moreover, from a solution method viewpoint, VRPS is significantly more complicated than the classic VRPTW because the scheduling of vehicles is interdependent, and therefore, finding a quality solution satisfying all time windows constraints is more challenging. The interdependent scheduling of vehicles in VRPS is even more challenging when travel and service times are stochastic.

In routing and scheduling problems, uncertainty in travel and service times is one of the main factors that significantly increase problems complexity. It has been dealt with in the literature using stochastic and robust optimization approaches. Some researchers studied traveling salesman problems with independent and normally distributed travel times and developed dynamic programming algorithms to maximize the probability that the tour completes by a deadline (Kao 1978; Sniedovich 1981; Carraway, Morin, and Moskowitz 1989). In some papers, authors proposed two-stage stochastic programming methods to formulate the problems and applied branch-and-cut algorithms (Laporte, Louveaux, and Mercure 1992; Kenyon and Morton 2003; Adulyasak and Jaillet 2015). Column generation is another prevalent approach to formulate routing and scheduling problems with stochastic travel and service times. In this approach, the uncertainty of travel and service times is encapsulated in the column definition and handled in the subproblem (Taş et al. 2014, Yuan, Liu, and Jiang 2015, Errico et al. 2016).

In another category of papers, researchers developed chance-constrained programming models for routing and scheduling problems and solved them either optimally or heuristically (Laporte, Louveaux, and Mercure 1992; Li, Tian, and Leung 2010; Zhang, Chaovalitwongse, and Zhang 2012; Chen et al. 2014; Miranda and Conceição 2016). In these models, chance constraints ensure that time windows constraints or constraints restricting the maximum durations of tours are satisfied probabilistically. In addition, some researchers assumed that uncertain travel and service times belong to an uncertainty set and then applied robust optimization methods to find reliable routes. Such robust solutions either satisfy the time windows constraints for all possible realizations of uncertain parameters (Lee, Lee, and Park 2012; Agra et al. 2013) or minimize a measure index representing the amount of constraint violations for the worst case scenario (Han, Lee, and Park 2013; Souyris et al. 2013; Adulyasak and Jaillet 2015; Jaillet, Qi, and Sim 2016; Zhang et al. 2019). We refer interested readers to a recent survey by Oyola, Arntzen, and Woodruff (2017) that covers routing with stochastic travel and service times comprehensively.

Although there is a rich literature on VRP with synchronized visits and on VRP with stochastic travel and service times, to the best of our knowledge, there is no paper addressing these aspects simultaneously. The main contributions of this research are as follows.

- For the first time, we study VRP with synchronized visits and stochastic travel and service times. In addition to considering a home healthcare scheduling problem, we introduce an operating room scheduling problem with stochastic durations as a novel application of VRPS. We then formulate the problem as a two-stage stochastic integer programming model that, unlike the deterministic models in the VRPS literature, does not have any big-M constraints for the scheduling part of the problem.
- We considerably enhance the quality of the proposed L-shaped algorithm by developing some valid inequalities for both first- and second-stage models. We also develop a lower bounding functional for the second-stage cost and add it to the master problem (MP) of the L-shaped algorithm. Moreover, we propose a specialized algorithm for the subproblems of the L-shaped algorithm that is much faster than standard linear programming algorithms.
- We report extensive computational results on the VRP with synchronized visits and stochastic travel and service times.

We organize the remainder of this paper as follows. In Section 2, we introduce a VRP with synchronized visits and stochastic travel and service times. We also discuss the applications of this problem in home healthcare and operating room scheduling problems. In Sections 3 and 4, we propose a two-stage stochastic programming model and some valid inequalities to improve it, respectively. In Section 5, we develop an L-shaped algorithm and present the master problem and subproblems. We develop a lower bounding functional in Section 6. In Section 7, we propose a specialized algorithm for the subproblems of the L-shaped algorithm. We also present extensive computational results on home healthcare and operating room scheduling problems in Section 8. A case study on the home healthcare scheduling application is given in Section 9. Finally, we give some concluding remarks and future research directions in Section 10. We provide the proofs of all lemmas and theorems (except for Theorem 1) in the online appendix. We have provided a content list at the beginning of the online appendix for ease of finding the proofs of lemmas and theorems.

2. Problem Definition and Applications

We consider the following VRP with synchronized visits. Two or more fleets of homogeneous vehicles are available at a depot to serve a number of customers within a day. For serving each customer, a specific set of vehicles of different types must be available at the customer's location. If some vehicles arrive to the customer's location earlier than other required vehicles, they must wait until others arrive before service starts. Vehicles may require using some amounts of a limited resource, such as time or the available space for delivering/picking up items requested by customers.

The service times of homogeneous vehicles are the same, but they can differ for each vehicle type. When a vehicle's service to a customer finishes, the vehicle either travels to the next customer or finishes its tour by returning to the depot regardless of whether other vehicles are still serving the customer. We suppose that travel and service times are stochastic, and a number of scenarios representing the uncertainty are available. Moreover, for each customer, there is a time window with a hard earliest start time constraint and a soft latest start time constraint; if violated, some penalties are incurred to the objective function. This assumption already exists in the literature (Taillard et al. 1997; Gendreau et al. 1999; Qureshi, Taniguchi, and Yamada 2009). It is more realistic in some applications, such as home healthcare scheduling, where the patient cannot be fed for lunch or be injected earlier than a time threshold, whereas too much delay is penalized.

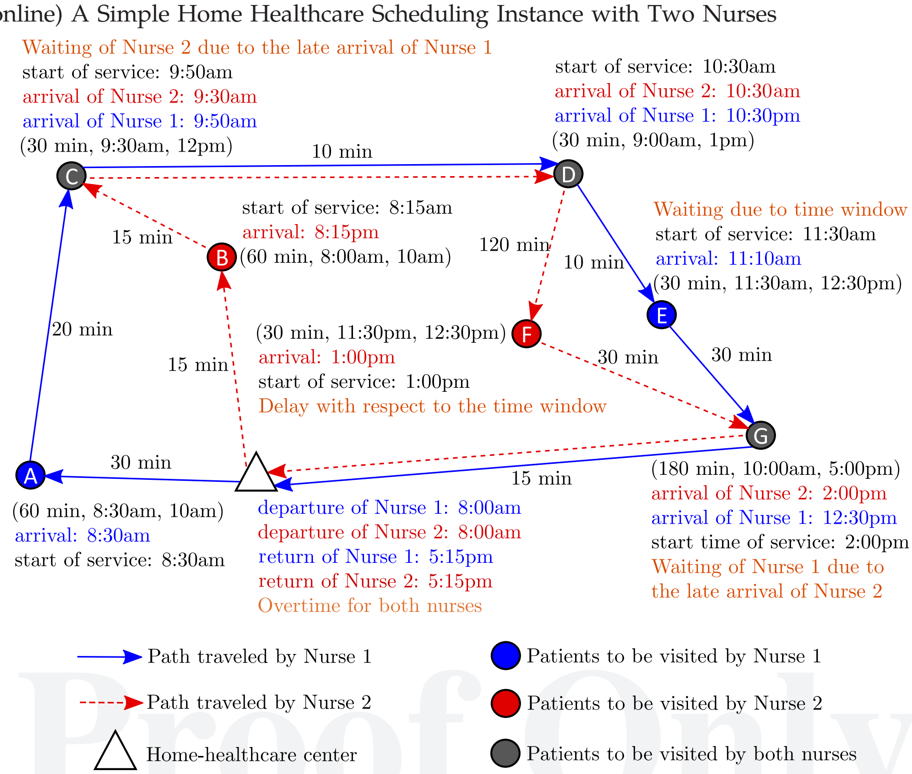
Furthermore, the time available to complete the tours is limited (e.g., 11 hours). There is also a time threshold (e.g., nine hours) for the start of the overtime period in which drivers are paid in addition to their fixed daily payments. Service to customer can start in overtime periods but must finish before the end of total available time limit. The decision maker must decide on the number of vehicles of different types to hire, routing of vehicles, and departure times of vehicles form the depot. The objective function includes the fixed costs of vehicles, travel costs, waiting costs, overtime costs, and the penalty of delays in serving customers with respect to given latest start times.

We consider two applications of the above problem in healthcare. The first application is a home healthcare scheduling problem. Two types of nurses that we refer to as Registered Nurses (RNs) and Home Health Aides (HHA) must serve patients at their homes. An RN is allowed to provide a wide range of nursing services, such as wound dressing, ostomy care, intravenous therapy, administering medication, monitoring the general health of the patient, pain control, and other health support. However, HHAs can help patients with only their basic personal needs, such as walking, feeding, and dressing. In this problem, we divide patients into three categories: patients to be cared by an RN, patients to be visited by an HHA, and patients with needs to be served simultaneously by an RN and an HHA. Di Mascolo, Espinouse, and Ozkan (2014) studied this problem in the absence of uncertainty for travel and service times and proposed a mixed integer programming model to minimize the total waiting costs. Figure 1 shows a very simple deterministic home healthcare scheduling problem with six patients that are visited by two nurses. Beside each patient, there is a data triplet, where the entries determine the service duration and the earliest and latest start times of the time window, respectively. In this figure, travel times are mentioned on arcs. For the given routing, we have computed the arrival times of nurses and start times of the services. Moreover, we have mentioned the occurrence of delays, waiting, and overtimes wherever they happen.

The second application is an operating room scheduling problem with stochastic surgery, anesthesia, and cleaning times, where each surgery is equivalent to a customer in the VRP with synchronized visits. To perform each surgery, we require two

Q: 18

Figure 1. (Color online) A Simple Home Healthcare Scheduling Instance with Two Nurses



servers (vehicles) to be simultaneously available. These servers are surgeons and operating rooms. We suppose that the number of surgeons is given, but the number of operating rooms, which is identical, is to be determined. We also assume that a unique surgeon is assigned to each surgery and that each surgeon has already fixed the sequence of surgeries to operate. We can divide the total time to complete a surgery in an operating room into three parts: (1) the preparation and anesthesia part during which the surgeon is not necessarily present in the operating room, (2) the main part of the surgery in which the surgeon operates, and (3) the cleaning part during which the operating room is busy, but the surgeon is not and may have left the room in order to rest or reach her or his next surgery in another operating room. For surgeons and operating rooms, the service time to perform a surgery is equal to the duration of the main part of the surgery. Also, for operating rooms as one of the available servers (vehicles), the travel time from surgery i to surgery j is equal to the sum of cleaning time after surgery i and the duration of the preparation and anesthesia part of surgery j. The durations of all three parts of surgeries are stochastic.

In order to match the operating room scheduling problem to the VRP with synchronized visits, we assume that servers (operating rooms and surgeons) leave a dummy depot to perform surgeries and return to it at the end of their routes. The operating room scheduling problem previously explained is still slightly different from the VRP with synchronization. For example, surgeons are not homogeneous, and surgeries are already assigned to surgeons. To match the operating room scheduling problem with our VRPS, we consider surgeons as homogeneous vehicles, and then, for this vehicle type, we consider the set of allowed arcs for traveling based on the given sequences of surgeries for surgeons. The decision maker must decide on the number of operating rooms and their routings in order to minimize the total cost that includes fixed costs of operating rooms, waiting costs of surgeons, and overtime costs of operating rooms and surgeons. Figures 2–4 show a Gantt chart for a deterministic operating room scheduling problem and the corresponding routing for surgeons and operating rooms.

Figure 2. (Color online) An Operating Room (OR) Schedule for Three Surgeons in Five Operating Rooms

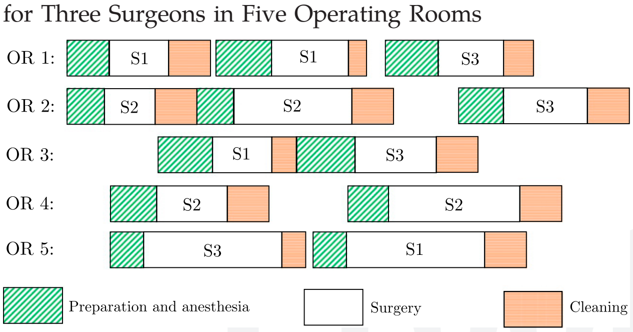
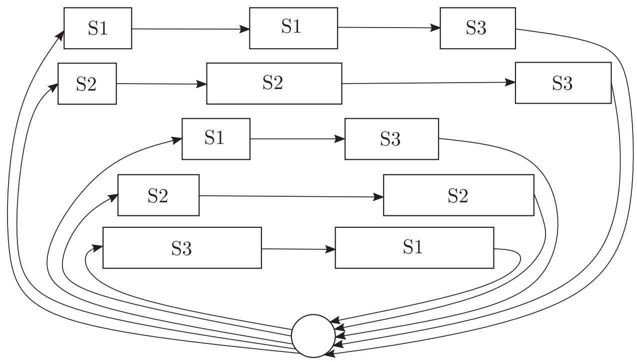


Figure 3. The Routing of Operating Rooms in the Schedule of Figure 2

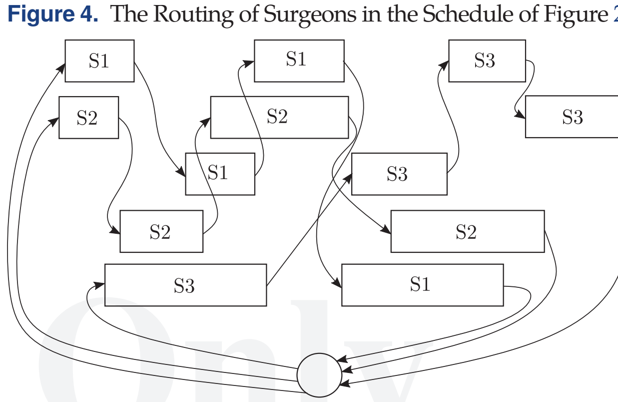


Batun et al. (2011) viewed a similar operating room scheduling problem as a scheduling problem rather than a VRP with synchronized visits. They formulated the problem as a two-stage stochastic program and proposed an L-shaped solution algorithm. Although their model is one of the most interesting papers in the field of stochastic operating room scheduling, we observed the following shortcomings of their work.

1. The scheduling part of their model suffers from big-M constraints. It is well known that models with such constraints are weak owing to poor linear programming relaxation. This issue is even worse in this model because big-M constraints are in the second-stage model, and therefore, big-M values appear as coefficients of variables in L-shaped cuts. As a result, the cuts are weak and cannot approximate the second-stage cost effectively.

2. Authors proposed two sets of symmetry-breaking constraints to deal with symmetries in their model. Although these constraints prevent the model from obtaining the same optimal solution with different presentations, they are not very effective in improving the quality of linear programming relaxation, and therefore, branch-and-bound algorithms may extend a large number of nodes before these constraints become binding.

Figure 4. The Routing of Surgeons in the Schedule of Figure 2



We have overcome these issues by developing a model in the next section that is free of big-M constraints and any symmetries.

3. Two-stage Stochastic Integer Programming Model

We propose a two-stage stochastic integer programming model for the VRPS defined in Section 2. In the first-stage model, the decision maker decides about the number of vehicles of different types to hire, routing of vehicles, and their departure times. Then, after the realization of uncertain travel and service times, the second-stage model computes the start times of services to customers. The assumption that all uncertain travel and service times reveal before deciding on second-stage variables is reasonable and valid from a modeling perspective. This is because with respect to Lemma 1, which we will provide later, the second-stage model does not use information about future uncertainties while fixing the start time of the service to any customer.

We present the first- and second-stage formulations in the two following sections separately.

3.1. First-stage Model

We use the notation shown in Table 1 for sets, parameters, and variables in the first-stage model.

Based on the given notation, we formulate the first-stage model as follows. In Model (S1), index 0 stands for the depot:

$$\text{(S1)} \quad \min_{x,y,m} \left(\sum_{r\in\mathcal{R}} c_r^v m_r + \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in\mathcal{I}_r\cup\{0\}:\\(i,j)\in\mathcal{A}_r}} c_{rij}^t x_{rij} + Q(x,y) \right) \quad (1)$$

Subject to :

$$\sum_{i\in\mathcal{I}_r:(0,i)\in\mathcal{A}_r} x_{r0i} = m_r \qquad r\in\mathcal{R} \quad (2)$$

$$\sum_{i\in\mathcal{I}_r\cup\{0\}:(i,j)\in\mathcal{A}_r} x_{rij} = 1 \qquad r\in\mathcal{R}, j\in\mathcal{I}_r \quad (3)$$

$$\sum_{i\in\mathcal{I}_r\cup\{0\}:(i,j)\in\mathcal{A}_r} x_{rij} = \sum_{i\in\mathcal{I}_r\cup\{0\}:(j,i)\in\mathcal{A}_r} x_{rji} \qquad r\in\mathcal{R}, j\in\mathcal{I}_r\cup\{0\} \quad (4)$$

$$\sum_{i,j\in\mathcal{S}:(i,j)\in\mathcal{A}_r} x_{rij} \le |\mathcal{S}| - q_{rs} \qquad r\in\mathcal{R}, \mathcal{S}\subseteq\mathcal{I}_r : |\mathcal{S}|\ge 2 \quad (5)$$

$$\sum_{t\in\mathcal{T}_{rj}^{dep}} y_{rjt} = x_{r0j} \qquad r\in\mathcal{R}, j\in\mathcal{I}_r \quad (6)$$

$$x_{rij}\in\{0,1\} \qquad r\in\mathcal{R},\ i,j\in(\mathcal{I}_r\cup\{0\}) : (i,j)\in\mathcal{A}_r \quad (7)$$

$$y_{rjt}\in\{0,1\} \qquad r\in\mathcal{R}, j\in\mathcal{I}_r, t\in\mathcal{T}_{ri}^{dep} \quad (8)$$

$$m_r \ge 0, \text{integer} \qquad r\in\mathcal{R}. \quad (9)$$

Table 1. Notation

Q: 19

Notation	Description
Sets	
$\mathcal{R}$	The set of all types of available vehicles
$\mathcal{R}_i$	The set of vehicles required for serving customer i
$\mathcal{I}$	The set of all customers
$\mathcal{I}_r$	The set of customers requiring a type r vehicle. Customers in this set may also need to be visited by vehicles of other types
$\mathcal{A}_r$	The set of allowed arcs for type r vehicles
$\mathcal{T}$	The set of available time slots within the scheduling horizon that includes the normal and overtime periods. (We suppose that the available time is split into smaller time slots with equal lengths. In the remainder of this paper, wherever we state that an event happens at time slot t, we mean that it occurs at the beginning of the time slot)
$\mathcal{T}_{ri}^{dep}$	The set of time slots at which a type r vehicle may depart the depot to serve customer i with the condition that it can return to the depot before the end of the scheduling horizon. We have presented the computation of this set by given data in Online Appendix EC.1
Parameters	
c_{rij}^t	Travel cost for a type r vehicle for traveling from customer i to customer j
c_r^v	The fixed cost of hiring a type r vehicle
d_{ri}	The required amount of the resource (if there is any) that is consumed by a type r vehicle while serving customer i
C_r	The capacity of type r vehicles for the resource that they consume while serving customers
q_{rs}	A lower bound on the minimum number of type r vehicles required for serving customers in set $S\subseteq\mathcal{I}_r$. We compute it by $q_{rs} = max\{\lceil\sum_{i\in\mathcal{S}} d_{ir}/C_r, 1\rceil\}$
Variables	
m_r	The number of type r vehicles to hire
x_{rij}	1 if a type r vehicle visits customer j immediately after customer i; 0 otherwise
y_{rjt}	1 if a type r vehicle departs the depot at time slot t to visit customer j; 0 otherwise

Objective function (1) consists of the fixed costs of vehicles, the travel costs, and the second-stage cost $Q(x,y)$ that is a function of first-stage decision variables x_{rij} and y_{rjt}. We define the second-stage cost in Section 3.2. Constraints (2) and (3) are degree constraints for the depot and customers, respectively. Constraint (4) is the flow conservation constraint. Constraint (5) guarantees that, for each type of vehicles, the capacity constraints are respected and no subtour is allowed. Constraint (6) determines the departure time of vehicles from the depot. Constraints (7)–(9) represent the integrality constraints for first-stage decision variables. In Model (S1), constraint (5) eliminates subtours composed of arcs traversed by vehicles of the same type for different types of vehicles separately but not together. We illustrate it by an example depicted in Figure 5. In this figure, there is no subtour for type 1 and type 2 vehicles

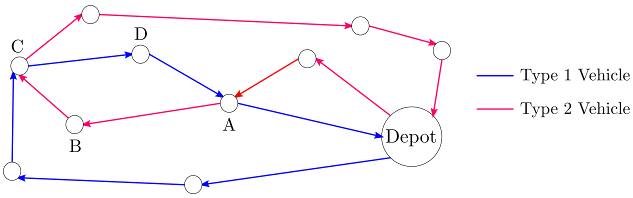


Figure 5. (Color online) A Subtour Formed by Two Different Types of Vehicles

separately, but path A-B-C-D formed by both types of vehicles is a subtour. In our problem, this type of subtour is also illegal and results in infeasibility in the scheduling of customers visits. The infeasibility happens because the start times of services to any pair of customers (i_1, i_2) in the subtour must satisfy $start_{i_1} < start_{i_2}$ and $start_{i_1} > start_{i_2}$, which is a contradiction. Therefore, although infeasibility cuts from the subproblems of L-shaped algorithm will remove such subtours, we should avoid them beforehand in the first-stage model. Doing so, we call the subproblem for infeasible solutions fewer times, and therefore, the algorithm will spend more time finding and evaluating feasible solutions. As discussed later, we address this issue by adding valid inequalities and also, a lower bounding functional developed in Sections 4.1 and 6, respectively.

3.2. Second-stage Model

To formulate the second-stage model, we use the variables, sets, and parameters in Table 2.

An assumption in modeling the second stage is that travel and service times are multiples of the time slots length. In the previous notation, we suppose that $s_{ri\omega}$, $t_{rij\omega}$, and also, all elements of $\xi(\omega)$ are given in terms of time slots length. We formulate the second-stage model for scenario $\omega \in \Omega$ as follows:

$$\text{(S2)} \quad Q(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega)) = \min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}} \left(\sum_{r\in\mathcal{R}} \sum_{\substack{i\in\mathcal{I}_r:\\ (i,0)\in\mathcal{A}_r}} \sum_{t\in\mathcal{T}_{ri0t\omega}} c^{o}_{rit\omega} u_{ri0t\omega} + \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{T}_{i\omega}} c^{d}_{it} v_{it\omega} + \sum_{r\in\mathcal{R}} c^{w}_{r} w_{r\omega} \right) \tag{10}$$

Subject to :

$$\sum_{t\in\mathcal{T}_{rij\omega}} u_{rijt\omega} = x_{rij} \qquad r\in\mathcal{R}, i,j\in(\mathcal{I}_r\cup\{0\}) : (i,j)\in\mathcal{A}_r \tag{11}$$

$$\sum_{\substack{i\in(\mathcal{I}_r\cup\{0\}):\\ (i,j)\in\mathcal{A}_r}} \sum_{\substack{t'\in\mathcal{T}_{rij\omega}:\\ t'+s_{ri\omega}+t_{rij\omega}\le t}} u_{rijt'\omega} \ge \sum_{\substack{k\in(\mathcal{I}_r\cup\{0\}):\\ (j,k)\in\mathcal{A}_r \,\&\, t\in\mathcal{T}_{rjk\omega}}} u_{rjkt\omega} \qquad r\in\mathcal{R}, j\in\mathcal{I}_r, t\in\mathcal{T}_{j\omega} \tag{12}$$

$$\sum_{\substack{j\in(\mathcal{I}_r\cup\{0\}):\\ (i,j)\in\mathcal{A}_r \,\&\, t\in\mathcal{T}_{rij\omega}}} u_{rijt\omega} = v_{it\omega} \quad r\in\mathcal{R}, i\in\mathcal{I}_r, t\in\mathcal{T}_{i\omega} \tag{13}$$

$$w_{r\omega} = \sum_{\substack{i\in\mathcal{I}_r:\\ (i,0)\in\mathcal{A}_r}} \sum_{t\in\mathcal{T}_{ri0\omega}} f_{rit\omega} u_{ri0t\omega} - \sum_{\substack{i,j\in\mathcal{I}_r\cup\{0\}:\\ (i,j)\in\mathcal{A}_r}} g_{rij\omega} x_{rij} - \sum_{i\in\mathcal{I}_r} \sum_{t\in\mathcal{T}^{dep}_{ri}} t y_{rit} \quad r\in\mathcal{R} \tag{14}$$

$$u_{r0it\omega} = y_{rit} \qquad r\in\mathcal{R}, i\in\mathcal{I}_r : (0,i)\in\mathcal{A}_r, t\in\mathcal{T}^{dep}_{ir} \tag{15}$$

$$u_{rijt\omega} \in \{0,1\} \qquad r\in\mathcal{R}, i,j\in(\mathcal{I}_r\cup\{0\}) : (i,j)\in\mathcal{A}_r, \; t\in\mathcal{T}_{rij\omega} \tag{16}$$

$$v_{it\omega} \in \{0,1\} \qquad i\in\mathcal{I}, t\in\mathcal{T}_{i\omega}. \tag{17}$$

Objective function (10) represents the second-stage cost in scenario ω and includes overtime, delay, and waiting costs. The second-stage cost $Q(\boldsymbol{x}, \boldsymbol{y})$ in objective function (1) is calculated by $Q(\boldsymbol{x}, \boldsymbol{y}) = E_{\omega\in\Omega}[Q(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega))]$, where $E_{\omega\in\Omega}[.]$ computes the expected value over scenarios $\omega\in\Omega$. Constraint (11) links first- and second-stage decision variables x_{rij} and $u_{rijt\omega}$. Constraint (12) is the no overlap constraint and indicates that, if the service to customer j starts at time slot t, then the service to customer i visited immediately before customer j by the same vehicle must have started by time slot $t - s_{ri\omega} - t_{rij\omega}$. Constraint (13) is the synchronization constraint and ensures that required vehicles start serving the customer at the same time. Constraint (14) computes total waiting times for different types of vehicles. We used auxiliary variables $w_{r\omega}$ for the ease of presenting objective function (10). We can simply substitute $w_{r\omega}$ in objective function (10). In this case, it would be more reasonable to transfer the expected value of the second and third terms on the right-hand side of constraint (14) to objective function (1) in the first-stage model. Constraint (15) introduces the departure times from the depot to the second-stage model. Constraints (16) and (17) represent integrality constraints for second-stage variables.

4. Valid Inequalities

We develop some valid inequalities for the first- and second-stage models. We add the first two valid inequalities to the first-stage model, whereas the third class of valid inequalities is for the second-stage model. The main motivation of adding these valid inequalities is to improve the linear programming relaxation of the first-stage model that results in more efficiency of the proposed L-shaped algorithm.

Table 2. Notation

Notation	Description
Sets	
$\mathcal{T}_{i\omega}$	The set of time slots at which the service to customer i may start in scenario ω
$\mathcal{T}_{rij\omega}$	The set of times slots at which a type r vehicle may start serving customer i immediately before customer j in scenario ω. How to compute sets $\mathcal{T}_{i\omega}$ and $\mathcal{T}_{rij\omega}$ is described in Online Appendix EC.1
Ω	The set of random scenarios
Original parameters	
L	The length of the normal working hours for which no overtime cost is considered. In our model, we consider this parameter in terms of time slots length. We also note that $L \leq \vert\mathcal{T}\vert$ holds because set $\mathcal{T}$ defined in Section 3.1 includes some additional time slots for overtime periods
e_i	The earliest start time in the time window of customer i
l_i	The latest start time in the time window of customer i
$s_{ri\omega}$	The duration of the service provided by a type r vehicle to customer i in scenario ω
$t_{rij\omega}$	The travel time of a type r vehicle from customer i to a customer j in scenario ω
$c^o_{rit\omega}$	The overtime cost of a type r vehicle if it starts serving customer i at time slot t in scenario ω and then immediately returns to the depot. We compute it by $c^o_{rit\omega} = c'_{overtime} \times max\{t + s_{ri\omega} + t_{ri0\omega} - L, 0\}$, where $c'_{overtime}$ is the overtime cost for a single time slot beyond the session length L. We remind that the service to a customer can start in overtime periods but must finish before the end of total available time $\vert\mathcal{T}\vert$
c^d_{it}	The delay cost of serving customer i when the service to the customer starts at time slot t. We compute it by $c^d_{it} = c'_{delay} \times max\{t - l_i, 0\}$, where c'_{delay} is the delay cost for a single time slot beyond the latest start time l_i
c^w_r	The waiting cost of resource r for each unit time slot
$\xi(\omega)$	The vector of uncertain parameters including travel and service times in scenario ω
Auxiliary parameters	
$f_{rit\omega}$	The time slot at which a type r vehicle arrives in the depot immediately after starting to serve customer i at time slot t in scenario ω. We have $f_{rit\omega} = t + s_{ri\omega} + t_{ri0\omega}$
$g_{rij\omega}$	The amount of time in scenario ω that a type r vehicle is involved in serving customer i and also in traveling to the next customer j. We have $g_{rij\omega} = s_{ri\omega} + t_{rij\omega}$
Variables	
$u_{rijt\omega}$	1 if a type r vehicle starts serving customer i at time slot t immediately before serving customer j in scenario ω; 0 otherwise
$v_{it\omega}$	1 if the service to customer i starts at time slot t in scenario ω; 0 otherwise
$w_{r\omega}$	The total waiting time of all type r vehicles in scenario ω

4.1. Subtour Elimination Constraints

As illustrated by Figure 1, constraint (5) does not eliminate subtours formed by arcs traversed by vehicles of different types. The following valid inequalities (18)–(21) avoid these subtours.

A new variable is z_{ij} (one if any vehicle serves customer j immediately after customer i; zero otherwise):

$$z_{ij} \geq x_{rij} \qquad r \in \mathcal{R}, i,j \in (\mathcal{I}_r \cup \{0\}) : (i,j) \in \mathcal{A}_r \tag{18}$$

$$z_{ij} \leq \sum_{r\in\mathcal{R}:(i,j)\in\mathcal{A}_r} x_{rij} \qquad i,j \in (\mathcal{I}_r \cup \{0\}) : (i,j) \in \bigcup_{r\in\mathcal{R}_i} \mathcal{A}_r \tag{19}$$

$$\sum_{i,j\in\mathcal{S}:(i,j)\in \bigcup_{r\in\mathcal{R}_i}\mathcal{A}_r} z_{ij} \leq |\mathcal{S}| - 1 \qquad \mathcal{S} \subseteq \mathcal{I} : |\mathcal{S}| \geq 2 \tag{20}$$

$$0 \leq z_{ij} \leq 1 \qquad i,j \in (\mathcal{I} \cup \{0\}) : (i,j) \in \bigcup_{r\in\mathcal{R}_i} \mathcal{A}_r. \tag{21}$$

Constraints (18) and (19) indicate that, for a fixed arc (i, j), z_{ij} takes one if at least one of the variables x_{rij} is equal to one, and it takes zero if all variables x_{rij} are equal to zero. Constraint (20) is the subtour elimination constraint defined on z_{ij} variables. Because x_{rij} variables are binary, we do not need to consider integrality constraints for z_{ij} variables. Although constraint (20) looks similar to the subtour elimination constraint for the asymmetric traveling salesman problem (ATSP), we cannot optimally separate constraint (20) as in ATSP. We have explained this issue in Online Appendix EC.5 and proposed a heuristic to partially separate this constraint. This heuristic does not necessarily detect all violated subtours. However, the lower bounding functional introduced in Section 6 guarantees that all subtours are avoided.

4.2. Capacity Constraints for Service Times

As discussed for the first-stage model, if vehicles consume a limited resource while serving customers, constraint (5) is essential to ensure that the capacity constraints are satisfied. In healthcare applications explained in Section 2, there is not any physical resource required while serving customers. However, in routing and scheduling problems, we can consider "time" as a resource and impose constraint (5) to guarantee that the total service time in each tour does not exceed the maximum available time in the scheduling horizon. Therefore, we can add the following valid inequality to the first-stage model:

$$\sum_{i,j\in\mathcal{S}:(i,j)\in\mathcal{A}_r} x_{rij} \le |\mathcal{S}| - \left\lceil \sum_{i\in\mathcal{S}} s_{ri\omega}/|\mathcal{T}| \right\rceil \qquad r\in\mathcal{R}, \omega\in\Omega, \mathcal{S}\subseteq\mathcal{I} : |\mathcal{S}|\ge 2. \tag{22}$$

The above cut is known as the rounded capacity inequality in the literature (Lysgaard, Letchford, and Eglese 2004). We also add some other valid inequalities by considering "time" as a consumable resource while serving customers. These constraints are framed capacity, strengthened comb, homogeneous multistar, and hypotour inequalities. We refer readers for more information about these valid inequalities to Lysgaard, Letchford, and Eglese (2004).

4.3. Improved No Overlap Constraints

The following theorem shows an improvement on constraint (12).

Theorem 1. *Constraints (23) and (24) are valid inequalities for the second-stage model:*

$$\sum_{\substack{i\in(\mathcal{I}_r\cup\{0\}):\\(i,j)\in\mathcal{A}_r}} \sum_{\substack{t'\in\mathcal{T}_{rij\omega}:\\ t'+s_{ri\omega}+t_{rij\omega}\le t}} u_{rijt'\omega} \ge \sum_{\substack{k\in(\mathcal{I}_r\cup\{0\}):\\(j,k)\in\mathcal{A}_r}} \sum_{\substack{t'\in\mathcal{T}_{rjk\omega}:\\ t'\le t}} u_{rjkt'\omega} \qquad r\in\mathcal{R}, j\in\mathcal{I}_r, t\in\mathcal{T}_{j\omega} \tag{23}$$

$$\sum_{\substack{i\in(\mathcal{I}_r\cup\{0\}):(i,j)\in\mathcal{A}_r\\ \&\,(t-s_{ri\omega}-t_{rij\omega})\in\mathcal{T}_{rij\omega}}} u_{rij(t-s_{ri\omega}-t_{rij\omega})\omega} = \sum_{\substack{k\in(\mathcal{I}_r\cup\{0\}):\\(j,k)\in\mathcal{A}_r\,\&\,t\in\mathcal{T}_{rjk\omega}}} u_{rjkt\omega} \qquad r\in\mathcal{R}, j\in\mathcal{I}_r : |\mathcal{R}_j| = 1, \; t\in\mathcal{T}_{j\omega} : t > e_j. \tag{24}$$

Proof. The validity of constraint (23) originates from the definition of variables $u_{rijt\omega}$. We have obtained constraint (23) by lifting the right-hand side of constraint (12). Constraint (23) indicates that, if service to customer j starts at time t or earlier, then the vehicle must have arrived to this customer at time t or earlier. Constraint (24) also implies that, for any customer j requiring a single vehicle (condition $\mathcal{R}_j = \{r\}$ in (24)), the service starts as soon as the vehicle arrives to the customer if the arrival time is after the earliest start time of the corresponding time window (condition $t\in\mathcal{T}_{j\omega} : t > e_j$ in (24)). Constraint (24) is valid because as stated later by Lemma 1, there is no advantage to postpone the service to a customer when all required vehicles are available at the customer's location. □

Theorem 2. *Clique inequalities (25) and (26) are equivalent to constraints (23) and (24):*

$$\sum_{\substack{k\in(\mathcal{I}_r\cup\{0\}):\\(j,k)\in\mathcal{A}_r\,\&\,t'\in\mathcal{T}_{rjk\omega}}} \sum_{\substack{t'\in\mathcal{T}_{rjk\omega}:\\ t'\le t}} u_{rjkt\omega} + \sum_{\substack{i\in(\mathcal{I}_r\cup\{0\}):\\(i,j)\in\mathcal{A}_r}} \sum_{\substack{t'\in\mathcal{T}_{rij\omega}:\\ t'+s_{ri\omega}+t_{rij\omega}>t}} u_{rijt'\omega} \le 1 \qquad r\in\mathcal{R}, j\in\mathcal{I}_r, t\in\mathcal{T}_{j\omega} \tag{25}$$

$$\sum_{\substack{k\in(\mathcal{I}_r\cup\{0\}):\\(j,k)\in\mathcal{A}_r\,\&\,t\in\mathcal{T}_{rjk\omega}}} u_{rjkt\omega} + \sum_{\substack{i\in(\mathcal{I}_r\cup\{0\}):\\(i,j)\in\mathcal{A}_r}} \sum_{\substack{t'\in\mathcal{T}_{rij\omega}:\\ t'+s_{ri\omega}+t_{rij\omega}\ne t}} u_{rijt'\omega} = 1 \qquad r\in\mathcal{R}, j\in\mathcal{I}_r : |\mathcal{R}_j| = 1, \; t\in\mathcal{T}_{j\omega} : t > e_j. \tag{26}$$

Although constraints (25) and (26) and constraints (23) and (24) are equivalent, the former are computationally more effective because mixed integer programming solvers benefit from their clique structure. Therefore, for the lower bounding functional introduced in Section 6, we use constraints (25) and (26).

5. L-shaped Algorithm

The L-shaped algorithm is applicable to stochastic programming models with continuous recourse decision variables. However, the second-stage variables of the model that we developed in Section 3 are integer. The following theorem handles this issue.

Theorem 3. *The second-stage Model* (S2) *is still valid if we relax integrality constraints* (16) *and* (17) *provided that all subtours are prevented by the lower bounding functional presented later in Section* 6.

As a result of Theorem 3, we can apply the L-shaped algorithm rather than the integer L-shaped algorithm. Proving the validity of this relaxation is a very important achievement because integer recourse variables significantly make two-stage stochastic programs more difficult. We refer readers to Sherali and Zhu (2009) and Ahmed (2011) as surveys on challenging two-stage stochastic integer programs. The MP of the proposed L-shaped algorithm is as follows:

$$(\text{MP}) \min_{x,y,m,\theta} \Bigg[\sum_{r\in\mathcal{R}} c_r^v m_r + \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in\mathcal{I}_r\cup\{0\}:\\(i,j)\in\mathcal{A}_r}} c_{rij}^t x_{rij} - \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in(\mathcal{I}_r\cup\{0\}):\\(i,j)\in\mathcal{A}_r}} c_r^w \left(\sum_{\omega\in\Omega} p_\omega g_{rijt\omega} \right) x_{rij} + - \sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{I}_r} \sum_{t\in\mathcal{T}_{ri}^{dep}} c_r^w t y_{rit} + \theta \Bigg] \quad (27)$$

Subject to :

$$(2)-(9) \quad (28)$$

$$\theta \geq \sum_{\omega\in\Omega} p_\omega \theta_\omega \quad (29)$$

$$\theta_\omega \geq \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in\mathcal{I}_r\cup\{0\}:\\(i,j)\in\mathcal{A}_r}} \pi_{crij\omega}^{(1)} x_{rij} + \sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{I}_r:(i,0)\in\mathcal{A}_r} \sum_{t\in\mathcal{T}_{ri}^{dep}} \pi_{crit\omega}^{(2)} y_{rit} \quad \omega\in\Omega, c\in\mathcal{C}_\omega^o \quad (30)$$

$$\sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in\mathcal{I}_r\cup\{0\}:\\(i,j)\in\mathcal{A}_r}} \sigma_{crij\omega}^{(1)} x_{rij} + \sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{I}_r:(i,0)\in\mathcal{A}_r} \sum_{t\in\mathcal{T}_{ri}^{dep}} \sigma_{crit\omega}^{(2)} y_{rit} \leq 0 \quad \omega\in\Omega, c\in\mathcal{C}_\omega^f \quad (31)$$

We also formulate subproblem (SP_ω) for scenarios $\omega \in \Omega$ in the L-shaped algorithm as follows:

$$(\text{SP}_\omega) \quad Q'(x,y,\xi(\omega)) = \min_{u,v} \Bigg(\sum_{r\in\mathcal{R}} \sum_{\substack{i\in\mathcal{I}_r:\\(i,0)\in\mathcal{A}_r}} \sum_{t\in\mathcal{T}_{ri0t\omega}} c_{rit\omega}^o \, u_{ri0t\omega} + \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{T}_{i\omega}} c_{it}^d v_{it\omega} + \sum_{r\in\mathcal{R}} \sum_{\substack{i\in\mathcal{I}_r:\\(i,0)\in\mathcal{A}_r}} \sum_{t\in\mathcal{T}_{ri0t\omega}} c_r^w f_{rit\omega} u_{ri0t\omega} \Bigg). \quad (32)$$

Subject to

$$(11), (13), (15), (23) \quad (33)$$

$$u_{rijt\omega} \geq 0 \quad r\in\mathcal{R}, i,j\in(\mathcal{I}_r\cup\{0\}) : (i,j)\in\mathcal{A}_r, t\in\mathcal{T}_{rij\omega} \quad (34)$$

$$v_{it\omega} \geq 0 \quad i\in\mathcal{I}, t\in\mathcal{T}_{i\omega}. \quad (35)$$

In objective function (27), the third and fourth terms are the negative parts of the waiting cost in the second-stage Model (S2). We have assumed that, in the second-stage Model (S2), $w_{r\omega}$ in objective function (10) is substituted using constraint (14), and then, the negative parts of $w_{r\omega}$ are transferred to the objective function of first-stage Model (S1). In the third term of (27), p_ω represents the probability of scenario ω. Also, θ is the approximation of the second-stage cost without the negative parts of the waiting cost. In constraint (29), θ_ω stands for the approximation of the second-stage cost in scenario ω without the negative parts of the waiting cost. Constraints (30) and (31) are the optimality and feasibility cuts that are iteratively generated after solving subproblems (SP_ω). In these constraints, $\mathcal{C}_\omega^o$ and $\mathcal{C}_\omega^f$ are the set of generated optimality and feasibility cuts for scenario ω. In constraint (30), $\pi_{crij\omega}^{(1)}$ and $\pi_{crit\omega}^{(2)}$ are the dual variables of constraints (11) and (15) when generating optimality cut $c \in \mathcal{C}_\omega^o$. Likewise, in constraint (31), $\sigma_{crij\omega}^{(1)}$ and $\sigma_{crit\omega}^{(2)}$ are the extreme rays of constraints (11) and (15) when generating feasibility cut $c \in \mathcal{C}_\omega^f$. In the objective function (32), the third term is obtained after substituting $w_{r\omega}$ in objective function (10) using constraint (14). In constraints (34) and (35) of the subproblem, we have removed that the upper bound constraints obtained from relaxing the integrality constraints (16) and (17) because they are trivial with respect to constraints (3), (11), (13), (34), and (35).

In the L-shaped algorithm, after solving the MP, we fix the first-stage solution in subproblems and solve them for all scenarios $\omega \in \Omega$. We then generate optimality and feasibility cuts from optimally solved and infeasible subproblems, respectively. The algorithm iterates until it reaches the maximum acceptable optimality gap or the time limit. We also implement the branch-and-cut version of the L-shaped algorithm, where we solve the master problem once by a branch-and-cut algorithm. In the branch-and-bound tree, whenever we find a first-stage feasible solution, we solve subproblems and add optimality and feasibility cuts (30) and (31) to the branch-and-bound tree. In the branch-and-cut algorithm, we solve subproblems only when we find a first-stage feasible solution and do not solve them for first-stage solutions that are fractional or include infeasible subtours in any node of the branch-and-bound tree.

More implementation details are provided in Online Appendix EC.19.

6. Lower Bounding Functional

In the following, we develop a lower bounding functional for the proposed L-shaped algorithm.

Lemma 1. *For a fixed first-stage solution, in any scenario $\omega \in \Omega$, services to customers start as soon as all required vehicles are available at the customers' locations.*

With respect to Lemma 1, it is clear that the second-stage model does not use information on future uncertain travel and service times while deciding about the start time of the service to any customer. This is the reason for the validity of our assumption on making second-stage decisions after the revelation of all uncertain times.

Lemma 2. *For a fixed first-stage solution, if customers are served as soon as the required vehicles are available at customers' locations, the finish time of service to each customer is convex in terms of $\xi(\omega)$.*

As discussed in Section 3.2, $Q(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega))$ that is formulated by relations (10)–(17) computes the second-stage cost for a fixed first-stage solution $(\boldsymbol{x}, \boldsymbol{y})$ in scenario ω assuming that service and travel times in this scenario are multiples of the time slots length. Similarly, $Q'(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega))$, defined by (32)–(35), calculates the second-stage cost without negative parts of the waiting cost for scenario ω if all travel and service times are multiples of the time slots length. Let us define $Q^{general}(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega))$ as a function that computes the second-stage cost without negative parts of the waiting cost for a fixed first-stage solution $(\boldsymbol{x}, \boldsymbol{y})$ in scenario ω without any condition on service and travel times (i.e., these times are not necessarily multiples of the time slots length).

Lemma 3. *For a fixed first-stage solution $(\boldsymbol{x}, \boldsymbol{y})$, $Q^{general}(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega))$ is convex in terms of $\xi(\omega)$.*

Lemma 4. $Q'(\boldsymbol{x}, \boldsymbol{y}, \lfloor \xi(\omega) \rfloor) \le Q^{general}(\boldsymbol{x}, \boldsymbol{y}, \xi(\omega))$ *holds where $\lfloor \xi(\omega) \rfloor$ denotes a vector of travel and service times rounded down to largest multiple of time slot length in scenario ω.*

Theorem 4. $\theta \ge Q'(\boldsymbol{x}, \boldsymbol{y}, \lfloor \xi(\bar{\omega}) \rfloor)$ *is a valid inequality for MP where $\xi(\bar{\omega})$ is the vector of travel and service times for the average scenario (i.e., $\xi(\bar{\omega}) = \sum_{\omega \in \Omega} p_\omega \xi(\omega)$).*

The validity of Theorem 4 relies on Lemmas 3 and 4 as well as Jensen's Inequality (Jensen 1906). This theorem shows that we can add $\theta \ge Q'(\boldsymbol{x}, \boldsymbol{y}, \lfloor \xi(\bar{\omega}) \rfloor)$ as a lower bounding functional to the MP. To consider this lower bounding functional in our model, we should impose the following constraints to the master problem:

$$\theta \ge \sum_{r \in \mathcal{R}} \sum_{\substack{i \in \mathcal{I}_r: \\ (i,0) \in \mathcal{A}_r}} \sum_{t \in \mathcal{T}_{ri0t\bar{\omega}}} c^{o}_{rit\bar{\omega}} u_{ri0t\bar{\omega}} + \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}_{i\bar{\omega}}} c^{d}_{it} v_{it\bar{\omega}} + \sum_{r \in \mathcal{R}} \sum_{\substack{i \in \mathcal{I}_r: \\ (i,0) \in \mathcal{A}_r}} \sum_{t \in \mathcal{T}_{ri0t\bar{\omega}}} c^{w}_{r} f_{rit\bar{\omega}} u_{ri0t\bar{\omega}} \tag{36}$$

(11), (13), (15), (23), (34) − (35) for scenario $\bar{\omega}$ rather than scenario ω. (37)

We note that, in constraints (36) and (37), $\bar{\omega}$ denotes the scenario corresponding to the realization $\lfloor \xi(\bar{\omega}) \rfloor$. In this lower bounding functional, we make a copy of the second-stage variables for scenario $\bar{\omega}$ and add the corresponding second-stage constraints and the objective function by (36) and (37). We emphasize that we add lower bounding constraints (36) and (37) for a single scenario $\bar{\omega}$ not all available scenarios. Therefore, this lower bounding functional includes significantly fewer constraints and variables compared with the extensive form of the second-stage model (10)–(17). The above lower bounding functional is very effective, and it is a vital part of the L-shaped algorithm developed in this paper. This is because it provides a strong lower bound for approximating the second-stage cost.

There are some points that can improve the proposed lower bounding functional. First, as stated by Theorem 3, integrality constraints on the second-stage variables are trivial, and we can relax them. Also, as discussed in Section 5, we can remove the obtained upper bound constraints $u_{rijt\omega} \le 1$ and $v_{it\omega} \le 1$ from the subproblem. However, we noticed that Q: 11 CPLEX finds feasible solutions more easily when we declare these variables as integer variables. This is perhaps because CPLEX applies some heuristics to find feasible solutions that are more effective in the case of integer variables. Therefore, in the lower bounding functional (36) and (37), we replace constraints (34) and (35) by constraints (16) and (17) and consider variables $v_{it\bar{\omega}}$ and $u_{rijt\bar{\omega}}$ as binary variables. These constraints are valid for the lower bounding functional because the proof of validity $\theta \ge Q'(\boldsymbol{x}, \boldsymbol{y}, \lfloor \xi(\bar{\omega}) \rfloor)$ in Theorem 4 remains unchanged with constraints (16) and (17) instead of constraints (34) and (35) in subproblem (SP_ω). Second, because of the clique structure of constraint (25), it is more effective than constraint (23) when we have integrality constraint on $v_{it\bar{\omega}}$ and $u_{rijt\bar{\omega}}$ variables. Therefore, in the lower bounding functional, we replace constraint (23) by constraint (25) in (37). Moreover, we improve the lower bounding functional by adding constraint (26) for scenario $\bar{\omega}$.

Theorem 5. *The proposed lower bounding functional eliminates all subtours.*

This theorem is very important considering that we can only heuristically separate the subtour elimination constraints (21).

7. Analysis of Subproblems

To generate the optimality and feasibility cuts, we need to solve subproblems for all scenarios $\omega \in \Omega$ and extract dual values or infeasibility extreme rays. This step of the algorithm is computationally demanding, especially for our subproblems that include a large number of variables and constraints. This issue is even worse in the case of the branch-and-cut implementation of the L-shaped algorithm, where we must solve subproblems whenever we find a first-stage feasible solution in any node of the branch-and-bound tree. We develop a specialized algorithm for subproblems that is much faster than standard linear programming algorithms.

In Online Appendix EC.14, we provide an algorithm that, for a fixed first-stage solution $(\hat{x}, \hat{y})$, computes the start times of services to customers in scenario ω. The idea of this algorithm is that for any demand point i, for which the required servers are available at its location, it sets the start time of service to the maximum of the earliest start time e_i and the latest arrival times of its servers. After fixing the start time of a demand point, the algorithm computes the arrival times of servers to the next demand points in their visit lists and similarly calculates the start time of service to the new demand point. This step of the algorithm continues until the all start times are computed. Using this algorithm, if we find that the completion times of all tours are within the scheduling horizon $|\mathcal{T}|$, the subproblem is feasible, and we use the formula presented in Section 7.1 to compute the dual values $\pi^{(1)}_{crij\omega}$ and $\pi^{(2)}_{crit\omega}$. We then generate an optimality cut (30) based on the calculated dual values. However, if we realize that any vehicle completes its tour after the end of the scheduling horizon, we generate an infeasibility cut as explained in Section 7.2.

7.1. Optimality Cuts

In this section, assuming that the subproblem is feasible, we analyze its dual formulation in order to compute the optimal values of dual variables. As stated in Lemma 1, subproblem (SP$_\omega$) has a special structure, and for a fixed first-stage solution, we can find the optimal second-stage solution by serving customers as soon as all required vehicles are available. The special structure of subproblem (SP$_\omega$) motivated us to analyze the dual formulation of the subproblem in order to see if there is any shortcut to find the optimal dual solution.

Before writing the dual formulation of subproblem (SP$_\omega$) given by (32)–(35), we note that, with respect to constraint (11), constraint (15) is redundant in the case that $\hat{x}_{r0i} = 0$ holds. Considering this point, we write the dual formulation of the subproblem (SP$_\omega$) as follows:

$$(\mathrm{D}_\omega)\quad \max_{\pi} \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in\mathcal{I}_r\cup\{0\}:\\(i,j)\in\mathcal{A}_r}} \hat{x}_{rij}\pi^{(1)}_{rij\omega} + \sum_{r\in\mathcal{R}} \sum_{\substack{i\in\mathcal{I}_r:\\ \hat{x}_{r0i}=1}} \sum_{t\in\mathcal{T}^{dep}_{ri}} \hat{y}_{rit}\pi^{(2)}_{rit\omega} \tag{38}$$

Subject to:

$$\pi^{(1)}_{r0j\omega} + \sum_{\substack{t'\in\mathcal{T}_{j\omega}:\\ t+t_{r0j\omega}\le t'}} \pi^{(3)}_{rjt'\omega} + \mathbb{1}_{(\hat{x}_{r0j}=1)}\pi^{(2)}_{rjt\omega} \le 0 \qquad r\in\mathcal{R}, j\in\mathcal{I}_r:(0,j)\in\mathcal{A}_r, t\in\mathcal{T}_{r0j\omega} \tag{39}$$

$$\pi^{(1)}_{rij\omega} + \sum_{\substack{t'\in\mathcal{T}_{j\omega}:\\ t+s_{ri\omega}+t_{rij\omega}\le t'}} \pi^{(3)}_{rjt'\omega} - \sum_{\substack{t'\in\mathcal{T}_{i\omega}:\\ t\le t'}} \pi^{(3)}_{rit'\omega} + \pi^{(4)}_{rit\omega} \le 0 \qquad r\in\mathcal{R}, i,j\in\mathcal{I}_r:(i,j)\in\mathcal{A}_r, t\in\mathcal{T}_{rij\omega} \tag{40}$$

$$\pi^{(1)}_{ri0\omega} - \sum_{\substack{t'\in\mathcal{T}_{i\omega}:\\ t\le t'}} \pi^{(3)}_{rit'\omega} + \pi^{(4)}_{rit\omega} \le \lambda_{rit\omega} \qquad r\in\mathcal{R}, i\in\mathcal{I}_r:(i,0)\in\mathcal{A}_r, t\in\mathcal{T}_{ri0\omega} \tag{41}$$

$$-\sum_{r\in\mathcal{R}} \pi^{(4)}_{rit\omega} \le c^d_{it} \qquad i\in\mathcal{I}, t\in\mathcal{T}_{i\omega} \tag{42}$$

$$\pi^{(3)}_{rjt\omega} \ge 0 \qquad r\in\mathcal{R}, j\in\mathcal{I}_r, t\in\mathcal{T}_{j\omega}. \tag{43}$$

In Model (D$_\omega$), $\pi^{(1)}_{rij\omega}$, $\pi^{(2)}_{rjt\omega}$, $\pi^{(3)}_{rjt\omega}$, and $\pi^{(4)}_{rit\omega}$ denote the dual variables corresponding to constraints (11), (15), (23), and (13), respectively. To simplify the model and also the analysis that follows, we have defined a new parameter $\lambda_{rit\omega}$ by $\lambda_{rit\omega} = c^o_{rit\omega} + c^w_r f_{rit\omega}$. In constraint (39) and also in the remainder of the paper, we note that $\mathbb{1}_{(.)}$ is an indicator that is equal to one if condition (.) is satisfied and zero otherwise.

In Online Appendix EC.15, we present some complicated formulas as a parametric solution for Model (D$_\omega$).

Lemma 5. *The dual solution proposed in Online Appendix* EC.15 *is a feasible dual solution for Model (D$_\omega$).*

Lemma 6. *The objective value of the dual solution proposed in Online Appendix* EC.15 *is equal to the optimal objective value of subproblem (SP$_\omega$).*

Theorem 6. *The dual solution obtained proposed in Online Appendix* EC.15 *is the optimal solution of Model (D$_\omega$).*

The validity of Theorem 6 is based on the strong duality theorem in linear programming. Lemmas 5 and 6 demonstrate that the proposed dual solution obtained in Online Appendix EC.15 is feasible and has an objective value that is equal to the optimal objective value of the primal subproblem. Theorem 6 indicates that, instead of using simplex or interior point algorithms, we can simply use relations in Online Appendix EC.15 in order to find the values of dual variables $\pi^{(1)}_{crij\omega}$ and $\pi^{(2)}_{crit\omega}$ in optimality cut (30).

7.2. Feasibility Cuts

The only way that, for a first-stage solution, subproblem (SP_ω) may turn out infeasible is that, for at least one vehicle, the tour does not complete within the scheduling horizon $|\mathcal{T}|$. The first idea to prevent from revisiting first-stage solutions with infeasible subproblems is to use the following simple no good cut:

$$\sum_{r\in\mathcal{R}} \sum_{\substack{i\in\mathcal{I}_r:\\ \exists t\in\mathcal{T}^{dep}_{ri}:\hat{y}_{rit}=1}} y_{ri(dep_{ri})} + \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in(\mathcal{I}_r\cup\{0\}):\\ (i,j)\in\mathcal{A}_r \,\&\, \hat{x}_{rij}=1}} x_{rij} \le n-1. \quad (44)$$

In (44), n is the sum of the number of $\hat{x}_{rij}$ and $\hat{y}_{rit}$ variables that are equal to one. It is clear that, if the subproblem is infeasible for a given first-stage solution, modified solutions obtained by postponing the departure times are also infeasible. Therefore, we can enhance (44) as follows:

$$\sum_{r\in\mathcal{R}} \sum_{\substack{i\in\mathcal{I}_r:\\ \exists t\in\mathcal{T}^{dep}_{ri}:\hat{y}_{rit}=1}} \sum_{\substack{t'\in\mathcal{T}^{dep}_{ri}:\\ t'\ge dep_{ri}}} y_{rit'} + \sum_{r\in\mathcal{R}} \sum_{\substack{i,j\in(\mathcal{I}_r\cup\{0\}):\\ (i,j)\in\mathcal{A}_r \,\&\, \hat{x}_{rij}=1}} x_{rij} \le n-1. \quad (45)$$

No good cuts are generally known as weak cuts. Therefore, in the following, we explain how to develop stronger feasibility cuts. In order to propose the new feasibility cuts, we first need to define the notions of "path" and "critical path." We define a path P by $P=(P_{nodes}, P_{resources})$, where $P_{nodes}=\langle a_v\rangle_{v=1 to |P_{nodes}|}$ is a sequence of $|P_{nodes}|$ nodes in $\mathcal{I}\cup\{0\}$ visited on the path and $P_{resources}=\langle r_v\rangle_{v=1 \text{ to } |P_{nodes}|-1}$ is series of vehicles types corresponding to arcs (a_v, a_{v+1}) for $v=1$ to $|P_{nodes}|-1$. The vehicles types are not necessarily the same for different arcs. The destination of the last arc in a path is the depot (i.e., $a_{|P_{nodes}|}=0$), whereas the origin may or may not be the depot. For a given first-stage solution, we refer to a path as a "critical path" in scenario ω if the two following conditions are satisfied.

1. The vehicle corresponding to each arc on the path is a critical vehicle for the customer at the arc tail (we defined the notion of "critical vehicle" in Section 7.1).
2. The sum of travel and service times on the path plus the start time of the service to the first customer on the path violates the scheduling horizon's time limit.

The reason for infeasibility of subproblem (SP_ω) for a given first-stage solution is the existence of at least one critical path. In the following, we present two types of no good cuts in order to prevent the part of the first-stage solution resulting in critical paths. We devise the first cut for critical paths originating from the depot. For this type of critical paths, we propose the following cut using the notation $P=(P_{nodes}, P_{resources})$ explained above:

$$\sum_{t\in\mathcal{T}^{dep}_{r_1a_2}: t\ge dep_{r_1a_2}} y_{r_1a_2t} + \sum_{v=1}^{|P_{nodes}|-1} x_{r_va_va_{v+1}} \le |P_{nodes}|-1. \quad (46)$$

We propose the second type of cuts for critical paths not originating from the depot. In this case, all vehicles visiting the first customer are noncritical, and the service to the customer starts when his time window opens. We can write the no good cut as follows without any knowledge about customers visited before the first customer on the path and any departure time:

$$\sum_{v=1}^{|P_{nodes}|-1} x_{r_va_va_{v+1}} \le |P_{nodes}|-2. \quad (47)$$

In Online Appendix EC.18, we provide an algorithm to extract critical paths for a given first-stage solution in scenario ω.

8. Computational Results

We implemented the proposed algorithms in C++ and used IBM ILOG CPLEX Optimization Studio V12.6 to solve mixed integer programming models. We ran experiments on a computer with two Intel Xeon X5650 Westmere processors, 2.67 GHz, and a total of 12 cores. We used a single core for running each test instance. We have provided more implementation details of the proposed L-shaped and branch-and-cut algorithms in Online Appendix EC.19.

8.1. Home Healthcare Scheduling Instances

In this section, we explain how we generated a set of home healthcare scheduling instances with stochastic travel and service times. To generate most of data in these instance sets, we used the data generation approach proposed by Di Mascolo, Espinouse, and Ozkan (2014). We slightly modified the proposed approach in order to consider stochasticity for travel and service times. In these instances, two groups of nurses, including RNs and HHAs, must serve patients who are uniformly dispersed in a square area with a side length of 40 km. The home healthcare center is located at the center of this area. For stochastic instances, we set the number of patients to {10, 15, 20}. Moreover, we define synchronization rate as the percentage of patients requiring a simultaneous service by an RN and an HHS and set it to {10, 20, 30, 40}. To determine the type of required nurses for patients without any synchronization, we randomly divided them to two groups of the same size and supposed that these groups must by served by RNs and HHAs separately. For each patient, we generated the earliest start time and also the length of the time window from [0, 120 minutes] and [60, 180 minutes] randomly and then, fixed the latest start time to the sum of time window's earliest start time and its length.

For stochastic instances, we generated 100 random scenarios for travel and service times. For each scenario, we randomly generated the service times from [20, 180 minutes]. Also, we generated the travel time between every pair of customers i and j from a normal distribution with a mean $\mu_{ij} = d_{ij}$ and a standard deviation $\sigma_{ij} = d_{ij}/6$, where d_{ij} is the Euclidean distance between customers i and j.

We supposed that the normal session length for nurses is nine hours, after which overtime penalties are incurred. Moreover, we set the maximum available time for completing tours to 11 hours. Based on http://www.payscale.com, we estimated the fixed costs of hiring an HHA and an RN to be \$94.41 and \$216 per day, respectively, which are equivalent to \$10.49 and \$24 per hour, respectively. We also supposed that nurses are paid with double rates for working beyond the normal session length. Also, we set the per hour delay cost for serving a patient to \$15.73, which is equal to 1.5 times an HHA's salary rate. Because we let the model decide about the number of nurses and we pay fixed costs for hiring RNs and HHA, the model tends to minimize waiting times to visit patients by a few nurses. Therefore, we did not consider waiting costs in the home healthcare scheduling problem. However, in the case that the number of nurses is fixed a priori, one can consider waiting costs. We obtained 120 home healthcare scheduling instances by generating 10 instances for each combination of the number of patients and the synchronization rate.

8.2. Results for Home Healthcare Scheduling Instances

We report the results of the home healthcare scheduling problem with stochastic travel and service times in Table 3. In this table, we report the results of the proposed algorithms under columns L-shaped algorithm and branch-and-cut algorithm. L-shaped algorithm refers to the master-subproblem implementation of the proposed algorithm. In this case, we iteratively solve the master problem and generate optimality and feasibility cuts in each iteration by solving the subproblems after convergence of the master problem. In the branch-and-cut algorithm, we solve the master problem only once. Within its branch-and-bound tree of this single master problem, whenever we find a first-stage feasible solution, we solve subproblems and add optimality and feasibility cuts to the tree. For L-shaped algorithm and branch-and-cut algorithm, we have included all proposed enhancements, including the lower bounding functional, valid inequalities, and the specialized algorithm for subproblems. Moreover, under column branch-and-cut algorithm without LBF, we have presented the computational results of the branch-and-cut algorithm with all enhancements except the lower bounding functional in order to evaluate the effect of this feature of the algorithm. In absence of the lower bounding functional, we add the MTZ subtour elimination constraint (Miller, Tucker, and Zemlin 1960) based on x_{rij} variables. This constraint ensures elimination of all subtours but is generally weaker than other subtour elimination constraints.

In Table 3, each row represents the average of over 10 instances. Under Data Info., Pat. No. and Syn (%) give the number of patients and the percentage of customers requiring synchronized visits, respectively. Under L-shaped algorithm and branch-and-cut algorithm, we report the results of the proposed algorithms. Because the branch-and-cut algorithm outperforms the L-shaped algorithm, we give more details for the branch-and-cut algorithm. Time (sec) gives the computational time of algorithms in seconds. $LB_.$ and $UB_.$ indicate the best lower and upper bounds of algorithms, respectively, and $Gap_.$ computes the gap between these bounds. The subscripts of $LB_.$, $UB_.$, and $Gap_.$ are L, B, and N, which represent L-shaped algorithm, branch-and-cut algorithm, and branch-and-cut algorithm without LBF, respectively. Also, under L-shaped algorithm, *Ite* gives the number of times that the L-shaped algorithm has iterated between the master problem and subproblems. Under branch-and-cut algorithm, we also have the following columns. Nodes No. gives the number of nodes that are examined in the branch-and-cut algorithm. Feas. Cut No. and Opt. Cut No. indicate the numbers of generated feasibility and optimality cuts, respectively. We set a time limit of 24 hours for running instances. However, in order to show that the proposed branch-and-cut algorithm finds high-quality upper bounds in less computational time, we report column $\Delta^{UB}_{4h,24h}(\%)$, which computes the gap between the upper bounds obtained after 4 and 24 hours. VSS indicates the value of the stochastic solution in percentage. We obtain this value by $VSS = 100(UB_{det} - UB_B)/UB_{det}$, where UB_{det} indicates the objective value of the stochastic problem for the solution obtained by solving the "mean-value" problem. By "mean-value" problem, we refer to the problem with a single scenario that is the average of all scenarios. For the fixed "mean-value" solution, if the stochastic problem turns out infeasible in at least one scenario, then we have $UB_{det} = \infty$ and $VSS = 100\%$.

Table 3. Computational Results of the Branch-and-Cut and L-shaped Algorithms for the Home Healthcare Scheduling Problem with Stochastic Travel and Service Times

Data info.		L-shaped algorithm					Branch-and-cut algorithm									Branch-and-cut algorithm without LBF			
Pat. no.	Syn. (%)	Ite.	Time (sec)	LB_L	UB_L	Gap_L(%)	Node No.	Feas. Cut No.	Opt. Cut No.	Time (sec)	LB_B	UB_B	Gap_B(%)	$\Delta^{UB}_{4h,24h}$(%)	VSS(%)	Time (sec)	LB_N	UB_N	Gap_N(%)
10	0.1	25	3,083	928	928	0.00	157	23	1,984	109	928	928	0.00	0.00	100	2,150	928	928	0.00
10	0.2	37	54,356	955	958	0.29	918	46	3,872	248	958	958	0.00	0.00	100	29,215	958	958	0.00
10	0.3	30	86,400	938	957	1.95	8,680	652	9,001	4,193	957	957	0.00	0.00	100	63,095	722	970	48.05
10	0.4	32	86,400	988	1,018	2.84	15,837	2,379	11,801	10,016	1,016	1,016	0.00	0.00	100	86,400	735	1,045	65.14
Average		31	57,559	952	965	1.27	6398	775	6,664	3,641	965	965	0.00	0.00	100	45,215	835	975	28.30
15	0.1	42	86,400	1,276	1,291	1.15	23,347	285	9,299	6,343	1,291	1,291	0.00	0.00	100	86,400	671	1,345	181.54
15	0.2	39	86,400	1,299	1,314	1.11	7,242	103	9,291	3,377	1,314	1,314	0.00	0.00	100	86,400	410	1,454	328.79
15	0.3	22	86,400	1,412	1,445	2.30	46,217	2,013	21,401	62,179	1,435	1,444	0.63	0.00	100	86,400	321	1,729	441.36
15	0.4	22	86,400	1,448	1,492	2.95	49,339	4,341	22,244	79,729	1,465	1,491	1.77	0.97	100	86,400	294	1,770	507.46
Average		31	86,400	1,359	1,385	1.88	31,536	1,685	15,558	37,907	1,376	1,385	0.60	0.24	100	86,400	424	1,575	364.79
20	0.1	32	86,400	1,580	1,628	3.05	43,560	3,127	34,686	86,406	1,585	1,628	2.74	0.83	100	86,400	322	1,981	522.58
20	0.2	18	86,400	1,650	1,732	4.96	53,276	20,942	20,202	86,400	1,651	1,719	4.12	0.81	100	86,400	318	2,488	694.75
20	0.3	20	86,400	1,833	1,956	5.87	37,652	4,750	31,542	86,400	1,842	1,901	3.24	0.01	100	86,400	575	3,338	913.27
20	0.4	9	86,400	1,870	2,307	12.92	40,040	8,875	22,970	86,400	1,880	1,958	4.19	1.36	100	86,400	300	4,100	1,258.57
Average		19	86,400	1,734	1,906	6.70	43,632	9,424	27,350	86,400	1,739	1,802	3.57	0.75	100	86,400	379	2,977	847.29

In Table 3, the average values of Gap_B are 0.00%, 0.60%, and 3.57% for instances with 10, 15, and 20 patients, respectively, whereas the averages of Gap_L for the same size instances are 1.27%, 1.88%, and 6.70%. These values demonstrate that the branch-and-cut algorithm significantly outperforms the L-shaped algorithm, especially in larger-sized instances. We can see that, as the size of instances increases, the problem gets more difficult, and average values of Gap_B increase. Small values of $\Delta^{UB}_{4h,24h}$ demonstrate that the branch-and-cut algorithm finds high-quality upper bounds in the first four hours of computational time. Moreover, in Table 3, we observe that all average values of VSS are 100% that demonstrate that considering stochasticity in modeling the home healthcare scheduling problem is very important and solutions obtained by solving the "mean-value" problem are infeasible in the stochastic problem.

Moreover, under column branch-and-cut algorithm without LBF, the average values of Gap_N are 28.30%, 364.79%, and 847.29%. Comparison of these values with those of the branch-and-cut algorithm demonstrates that the lower bounding functional is extremely vital for the effectiveness of the branch-and-cut algorithm. We also observe that, for those instances with Gap_N equal to 0.00%, computational times are significantly larger than those of the branch-and-cut algorithm with the lower bounding functional.

Table 4 presents the computational results of the branch-and-bound algorithm with different solution methods for subproblems. In this table, each row gives the average of results for 40 instances with different synchronization rates. Under Proposed method, we provide results for the case that subproblems are solved using the method proposed in Section 7. Primal simplex, Dual simplex, and Interior point present computational results for cases that we used standard linear programming algorithms to solve subproblems. Furthermore, Ave. time (sec) shows the average solution time for solving a single subproblem, and Nodes no. indicates the number of nodes explored within a computational time of 30 minutes. Also, under columns Primal simplex, Dual simplex, and Interior point, Time ratio gives the ratio of the corresponding Ave. time (sec) to the Ave. time (sec) of our proposed method. Average values of Time ratio demonstrate that our subproblem analysis method is 169, 483, and 196 times faster than primal simplex, dual simplex, and interior point methods, respectively. Moreover, our proposed method explores a significantly higher number of nodes and provides lower optimality gaps within the time limit.

8.3. Operating Room Scheduling Instances

For the operating room scheduling problem, we generated a set of instances with stochastic surgery, anesthesia, and cleaning times. We set the number of surgeries to {11, 15, 20, 25}. For each instance, we generated 500 random scenarios. We generated surgery and anesthesia durations using distributions provided in table Q: 12
1 of Gul et al. (2011). Gul et al. (2011) extracted these distributions from the data of 4,034 patients at Mayo Clinic in the first 21 weeks of 2006. Because no data are available in Gul et al. (2011) for cleaning times, we generated them from [0, 15 minutes] uniformly. After transforming the generated durations to travel and service times in the equivalent VRPS as explained in Section 2, we rounded travel and service times to the closest multiple of five minutes, which is the length of time slots in our model.

To generate operating room scheduling instances, we introduce a parameter ρ that denotes the average working time of a surgeon. We set ρ to {5, 7, 9} hours. We set the number of surgeons to $\lceil \gamma/\rho \rceil$, where γ denotes the sum of surgeries durations averaged over all scenarios. We assigned surgeries to surgeons as follows. The idea of the following procedure is to make balanced workloads for surgeons. We first sorted surgeries in a decreasing order of the average surgery duration. Then, we assigned surgeries one by one from the sorted list to surgeons. To assign each surgery, among all surgeons, we choose the one with the lowest sum of assigned surgeries durations. After assigning all surgeries, we sequenced surgeries randomly

Q: 20 **Table 4.** Comparison of Different Solution Methods for Subproblems in Stochastic Home Healthcare Scheduling Instances Within a Time Limit of 30 Minutes

Data info.	Proposed method			Primal simplex				Dual simplex				Interior point			
Pat. no.	Ave. time (sec)	Nodes no.	Gap_B (%)	Ave. time (sec)	Time ratio	Nodes no.	Gap_B (%)	Ave. time (sec)	Time ratio	Nodes n.	Gap_B (%)	Ave. time (sec)	Time ratio	Nodes no.	Gap_B (%)
10	0.007	1,835	1.23	0.596	86	403	3.64	1.563	220	60	8.28	0.653	94	194	6.24
15	0.012	2,227	2.55	1.884	161	32	INF	4.581	378	0	INF	1.912	164	9	INF
20	0.016	802	24.56	4.291	259	1	INF	14.832	850	0	INF	5.516	330	0	INF
Average	0.012	1,621	9.45	2.257	169	145	INF	6.992	483	20	INF	2.694	196	68	INF

Table 5. Computational Results of the Branch-and-Cut and L-shaped Algorithms for the Operating Room Scheduling Problem with Stochastic Durations

Data info.		L-shaped algorithm					Branch-and-cut algorithm									Branch-and-cut algorithm without LBF			
Pat. no.	Sur. time limit	Ite.	Time (sec)	LB_L	UB_L	Gap_L(%)	Node no.	Feas. cut no.	Opt. cut no.	Time (sec)	LB_B	UB_B	Gap_B(%)	$\Delta^{UB}_{4h,24h}$(%)	VSS(%)	Time (sec)	LB_N	UB_N	Gap_N(%)
11	5	16	54,059	25,898	26,016	0.44	1,284	0	6,164	3,473	26,016	26,016	0.00	0.00	6.47	20,431	26,016	26,016	0.00
11	7	17	2,796	23,459	23,459	0.00	509	1	4,785	1,772	23,459	23,459	0.00	0.00	8.45	6,899	23,459	23,459	0.00
11	9	15	19,489	24,384	24,678	1.17	512	0	3,719	1,446	24,484	24,484	0.00	0.00	9.38	8,791	24,484	24,484	0.00
Average		16	25,448	24,580	24,717	0.54	768	0	4,889	2,231	24,653	24,653	0.00	0.00	8.10	12,040	24,653	24,653	0.00
15	5	14	80,136	32,625	33,969	4.19	6,232	0	13,430	10,790	33,127	33,127	0.00	0.07	7.41	86,400	14,146	44,945	383.19
15	7	11	70,304	26,983	27,972	4.05	5,016	7	7,668	4,200	27,494	27,494	0.00	0.00	9.95	77,909	25,736	28,078	11.38
15	9	12	43,979	23,821	24,692	3.73	759	54	4,367	1,064	24,396	24,396	0.00	0.00	100.00	2,493	24,396	24,396	0.00
Average		12	64,807	27,810	28,878	3.99	4,002	20	8,488	5,351	28,339	28,339	0.00	0.02	39.12	55,601	21,426	32,473	131.52
20	5	13	86,400	44,237	46,472	5.09	9,280	0	11,693	11,251	44,832	44,832	0.00	0.00	5.88	86,400	16,496	66,512	572.95
20	7	8	36,093	33,678	35,179	4.88	20,125	1	9,513	8,753	34,262	34,262	0.00	0.00	11.06	86,400	14,967	48,267	754.99
20	9	11	35,910	30,866	32,323	4.93	14,176	161	9,945	8,177	31,821	31,935	0.34	0.00	100.00	35,451	31,412	32,042	1.90
Average		11	52,801	36,261	37,991	4.96	14,527	54	10,384	9,393	36,972	37,010	0.11	0.00	38.98	69,417	20,958	48,940	443.28
25	5	8	86,400	57,045	60,082	5.35	16,782	0	39,141	84,461	57,736	59,577	3.23	0.43	3.32	79,426	20,207	109,297	867.49
25	7	7	86,400	40,581	45,347	11.96	19,005	6	27,569	86,400	42,110	43,519	3.37	0.78	16.86	86,400	9,937	69,614	1,155.35
25	9	4	48,794	37,642	40,502	8.15	18,906	1,717	9,503	7,329	39,956	39,956	0.00	0.00	100.00	24,667	39,956	39,956	0.00
Average		7	73,865	45,089	48,644	8.49	18,231	574	25,404	59,397	46,600	47,684	2.20	0.40	40.06	63,497	23,366	72,955	674.28

Proof Only

for each surgeon. We supposed that the normal session length, during which no overtime penalty is paid, is nine hours. We also considered the possibility of having overtime for at most two hours. Because it does not make sense to consider time windows for the start time surgeries, we set earliest and latest start times to the beginning and end of the day, respectively. We also supposed that surgeons are available at the beginning of the scheduling horizon.

We used all cost coefficients provided by Batun et al. (2011). The fixed cost of opening an operating room is $4,437. There is no waiting cost for operating rooms because we consider fixed cost for them. However, because there is no fixed cost for surgeons, we considered the waiting cost to be $88.74 per minute for them. We also set the overtime cost for surgeons and operating rooms to $133.11 and $12.37 per minute, respectively. We generated 10 instances for each combination of ρ and the number of surgeries for a total of 120 instances.

8.4. Results for Operating Room Scheduling Instances

We report the results of the operating room scheduling problem with stochastic durations in Table 5. In this table, each row represents the average over 10 instances. Under Data info., Pat. no. and Sur. time limit, we give the number of patients and the value of parameter ρ used for the generation of instances, respectively. Other columns of this table are the same as similar columns in Table 3.

In Table 5, we observe that the values of Gap_B are considerably less than those of Gap_L. This observation demonstrates that the branch-and-cut algorithm strongly dominates the L-shaped algorithm in the operating room scheduling context too. In Table 5, the average values of Gap_B are 0.00%, 0.00%, 0.11%, and 2.20% for instances with 11, 15, 20, and 25 surgeries, respectively. We observe that most of the instances with up to 20 surgeries are optimally solved. Furthermore, our branch-and-cut algorithm can solve instances with 25 surgeries and average surgeon time limits of nine hours optimally. These results demonstrate that our branch-and-cut algorithm is significantly more effective than the algorithm proposed by Batun et al. (2011), which can solve instances with up to 10 and 11 surgeries optimally.

Moreover, in Table 5, the average values of VSS are 8.10%, 39.12%, 38.98%, and 40.06%, which show that value of stochastic solution for instances with more surgeries is higher and that the application of our algorithm is more justifiable and beneficial in such cases. In addition, we observe that VSS increases as the average surgeon time limit increases. This is because in instances with higher average surgeon time limits (ρ), it is more likely that the mean-value solution results in unexpected overtime or infeasibility in the stochastic problem. The other noticeable point is that the values of $\Delta^{UB}_{4h,24h}$ are less than 0.78%, which shows that our branch-and-cut algorithm improves the upper bound values within the first four hours of computational time. It is also noteworthy that the average of computational time to obtain optimal solutions of instances with 11 surgeries is 2,231 seconds, whereas the solution times of the algorithm proposed by Batun et al. (2011) for two sets of instances with the same size are 4,866 and 9,992 seconds.

Comparison of the average values of Gap_N and Gap_B for instances with 15, 20, and 25 patients shows that the lower bounding functional plays an important role in the proposed branch-and-cut algorithm. We also observe that, for instances with 10 patients, the average computational times of branch-and-cut algorithm without LBF are considerably larger than those of branch-and-cut algorithm.

Similar to Table 4, Table 6 provides the computational results of the branch-and-bound algorithm for stochastic operating room scheduling instances with different solution methods of subproblems. In all cases, we set the time limit to 30 minutes. We observe that our proposed specialized algorithm for subproblems is 78, 76, and 59 times faster than primal

Table 6. Comparison of Different Solution Methods for Subproblems in Stochastic Operating Room Scheduling Instances Within a Time Limit of 30 Minutes

Data info.	Proposed method			Primal simplex				Dual simplex				Interior point			
Pat. no.	Ave. time (sec)	Nodes no.	Gap_B (%)	Ave. time (sec)	Time ratio	Nodes no.	Gap_B (%)	Ave. time (sec)	Time ratio	Nodes no.	Gap_B (%)	Ave. time (sec)	Time ratio	Nodes no.	Gap_B (%)
11	0.053	307	2.79	1.91	37	0	24.64	1.96	37	0	23.86	1.97	38	0	24.31
15	0.058	280	6.62	3.35	58	0	INF	3.76	61	0	INF	2.93	49	0	INF
20	0.088	114	10.03	7.38	84	0	INF	9.23	91	0	INF	5.79	67	0	INF
25	0.107	89	12.81	17.57	132	0	INF	15.28	114	1	INF	9.20	80	0	INF
Average	0.077	198	8.06	7.55	78	0	INF	7.56	76	0	INF	4.97	59	0	INF

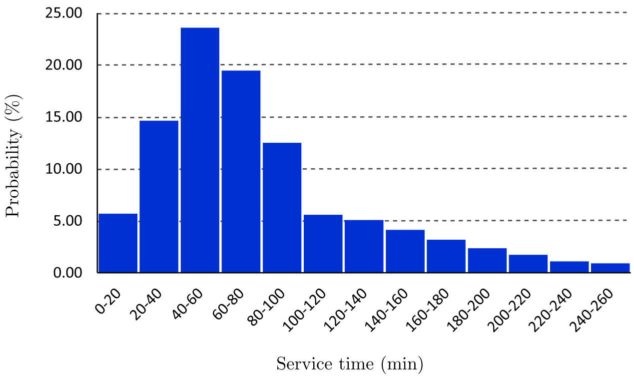


Figure 6. (Color online) Histogram of the Service Times in the Real-World Home Healthcare Scheduling Instances

simplex, dual simplex, and interior point methods, respectively. We can also observe that our method explores 198 nodes within the time limit, whereas other methods time out in the root node of the branch-and-bound tree. It is also noteworthy that the value of Time ratio increases in terms of the number of patients.

9. Case Study

In addition to the generated instances, we tested our proposed algorithm on a set of real-world instances for a home healthcare company located in Canada. We had access to the record of visits in the first three months of 2018. To create each test instance, on each day, we chose 10, 15, or 20 visits randomly. This resulted in 90 instances with a varied number of synchronized visits. The earliest and latest start times of the time windows are in the ranges of [8:00 a.m., 3:00 p.m.] and [8:30 a.m., 6:00 p.m.], respectively. Figure 6 shows the histogram of service times that vary between 10 and 260 minutes; 80% of service times are less than or equal to two hours. After discussing with the company's specialists, we decided to consider the fixed cost of RN and HHA and the length of the scheduling horizon as explained in Section 8.

In Table 7, we report the computational results of the home healthcare instances with real data. Under Data info., Syn. no. and Inst. no. show the number of visits requiring synchronization and the number of instances for that category, respectively. This table shows that our proposed algorithm finds quality solutions with reasonable optimality gaps within a time limit of four hours. As in Table 3, for almost all of the new instances, VSS was 100% because of the infeasibility of the expected value solution in the stochastic problem. Therefore, in Table 7, we report $RVSS$, a revised version of VSS, instead of VSS. $RVSS$ provides a better insight on the value of considering stochasticity in the synchronized VRP. We use $RVSS = 100(UB_H - UB_B)/UB_H$. In this formula, UB_H represents a finite upper bound obtained using a heuristic algorithm that solves some modified expected value problems in several iterations until it obtains a feasible solution for the stochastic problem. $RVSS$ (%) implies the amount of cost saving resulted from our proposed algorithm compared with this reasonable heuristic that one may use in the practice without taking the uncertainty into account. The details of this heuristic are provided in Online Appendix EC.20. Table 7 shows that the average $RVSS$ (%) is around 8.50%. The details of the cost improvement in $RVSS$ (%) are given in the last four columns of the table. For instance, under Hiring cost impr. (%), the values of $100(UB_H^{hiring} - UB_B^{hiring})/UB_H$ are computed, where UB_H^{hiring} and UB_B^{hiring} are the hiring costs of nurses in the heuristic solution and our branch-and-cut solution, respectively. Compared with the heuristic, our algorithm results in solutions that save between 17.60% and 22.40% on hiring costs of nurses on average. This saving comes at the expense of

Table 7. Computational Results of the Branch and Cut for the Home Healthcare Scheduling Problem with Real Data

Data info.			Branch-and-cut algorithm								
Pat. no.	Syn. no.	Inst. no.	Time (sec)	LB_B	UB_B	Gap_B(%)	$RVSS$(%)	Hiring cost impr (%)	Travel cost impr (%)	Overtime cost impr (%)	Delay cost impr (%)
10	1–2	16	76	787	787	0.00	10.39	15.67	0.47	–1.02	–4.73
10	3–4	14	333	894	894	0.00	8.42	19.54	0.63	–3.86	–7.89
Average			204	841	841	0.00	9.47	17.60	0.55	–2.44	–6.31
15	1–2	10	331	1,039	1,039	0.00	9.76	22.28	0.84	–4.42	–8.94
15	3–4	14	1,002	1,162	1,162	0.00	7.86	21.82	1.16	–5.54	–9.58
15	5–6	6	5,150	1,176	1,179	0.31	9.05	23.95	0.02	–5.02	–9.90
Average			1,608	1,124	1,125	0.06	8.73	22.40	0.82	–5.06	–9.43
20	1–2	9	1,805	1,241	1,241	0.00	8.03	23.92	1.06	–5.45	–11.50
20	3–4	6	6,138	1,358	1,360	0.14	6.62	18.68	0.71	–4.83	–7.94
20	5–6	7	7,749	1,499	1,502	0.21	6.00	18.14	0.63	–4.91	–7.86
20	7–8	8	12,370	1,561	1,572	0.73	8.08	19.01	1.11	–4.47	–7.57
Average			6,876	1,410	1,414	0.27	7.28	20.21	0.90	–4.94	–8.89

the deterioration of overtime and delay cost that are justifiable considering the total saving.

10. Conclusion

In this paper, we studied a vehicle routing problem with synchronized visits and stochastic travel and service times. In addition to considering a home healthcare scheduling problem, we cast an operating room scheduling problem with stochastic durations as a VRPS. We developed a two-stage stochastic integer programming model to formulate VRPS with stochastic times. In contrast to the deterministic models in the VRPS literature, our proposed formulation is free of big-M constraints. We obtained this advantage by splitting the available time into smaller time slots that resulted in a large number of second-stage integer variables. We proved that the integrality constraints on second-stage variables can be relaxed. Having continuous variables in the second stage, we applied the L-shaped algorithm and its branch-and-cut implementation as solution methods. Moreover, we improved the proposed approach by devising valid inequalities and a lower bounding functional. We also analyzed the subproblems of the L-shaped algorithm and proposed a specialized algorithm for them that is significantly faster than standard linear programming algorithms (60 to 480 times). These enhancements are general and applicable to a wide range of stochastic routing and scheduling problems with or without synchronization.

Computational experiments revealed that, in the stochastic home healthcare scheduling problem, the branch-and-cut algorithm solves instances with 15 patients and 10%–30% of synchronized visits to optimality. In addition, it finds solutions with an average optimality gap of 3.57% for instances with 20 patients. In the stochastic operating room scheduling problem, the branch-and-cut algorithm is capable of finding optimal solutions for instances with 20 surgeries. This is a considerable improvement over the state-of-art algorithm that reports on instances with 11 surgeries.

In this research, we supposed that vehicles of the same type are identical and considered an implicit assignment of them to customers. A possible future research direction would be to explore the possibility of the explicit assignment of identical vehicles with different costs in the objective function. Moreover, developing robust and chance constraints for VRPS with uncertain travel and service times should be of interest.

References

Adulyasak Y, Jaillet P (2015) Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Sci.* 50(2):608–626.

Agra A, Christiansen M, Figueiredo R, Hvattum LM, Poss M, Requejo C (2013) The robust vehicle routing problem with time windows. *Comput. Oper. Res.* 40(3):856–866.

Ahmed S (2011) Two-stage stochastic integer programming: A brief introduction. *Wiley Encyclopedia of Operations Research and Management Science.* Q: 13

Batun S, Denton BT, Huschka TR, Schaefer AJ (2011) Operating room pooling and parallel surgery processing under uncertainty. *INFORMS J. Comput.* 23(2):220–237.

Bredström D, Rönnqvist M (2008) Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *Eur. J. Oper. Res.* 191(1):19–31.

Carraway RL, Morin TL, Moskowitz H (1989) Generalized dynamic programming for stochastic combinatorial optimization. *Oper. Res.* 37(5):819–829.

Chen L, Hà MH, Langevin A, Gendreau M (2014) Optimizing road network daily maintenance operations with stochastic service and travel times. *Transportation Res. Part E: Logist. Transportation Rev.* 64:88–102. Q: 14

De Rosa B, Improta G, Ghiani G, Musmanno R (2002) The arc routing and scheduling problem with transshipment. *Transportation Sci.* 36(3):301–313.

Di Mascolo M, Espinouse M-L, Ozkan CE (2014) Synchronization between human resources in home healthcare context. *Proc. Internat. Conf. Healthcare Systems Engrg* (Springer), 73–86. Q: 15

Drexl M, Sebastian H-J (2007) On some generalized routing problems. Technical report, Deutsche Post Lehrstuhl für Optimierung von Distributionsnetzwerken (NN). Q: 16

Errico F, Desaulniers G, Gendreau M, Rei W, Rousseau L-M (2016) A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *Eur. J. Oper. Res.* 249(1):55–66.

Gendreau M, Guertin F, Potvin J-Y, Taillard E (1999) Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Sci.* 33(4):381–390.

Gul S, Denton BT, Fowler JW, Huschka T (2011) Bi-criteria scheduling of surgical services for an outpatient procedure center. *Production Oper. Management* 20(3):406–417.

Han J, Lee C, Park S (2013) A robust scenario approach for the vehicle routing problem with uncertain travel times. *Transportation Sci.* 48(3):373–390.

Jaillet P, Qi J, Sim M (2016) Routing optimization under uncertainty. *Oper. Res.* 64(1):186–200.

Jaumard B, Pouya H, Fahim R, Barrios A (2016) Planning network migration. *Proc. 2016 IEEE Internat. Conf. Comm. (ICC)* (IEEE), 1–6.

Jensen JLWV (1906) Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Math.* 30(1):175–193.

Kao EPC(1978) A preference order dynamic program for a stochastic traveling salesman problem. *Oper. Res.* 26(6):1033–1045.

Kenyon AS, Morton DP (2003) Stochastic vehicle routing with random travel times. *Transportation Sci.* 37(1):69–82.

Laporte G, Louveaux F, Mercure H (1992) The vehicle routing problem with stochastic travel times. *Transportation Sci.* 26(3): 161–170.

Lee C, Lee K, Park S (2012) Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *J. Oper. Res. Soc.* 63(9):1294–1306.

Li X, Tian P, Leung SCH (2010) Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *Internat. J. Production Econom.* 125(1): 137–145.

Li Y, Lim A, Rodrigues B (2005) Manpower allocation with time windows and job-teaming constraints. *Naval Res. Logist.* 52(4): 302–311.

Lim A, Rodrigues B, Song L (2004) Manpower allocation with time windows. *J. Oper. Res. Soc.* 55(11):1178–1186.

Lysgaard J, Letchford AN, Eglese RW (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Programming* 100(2):423–445.

Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problems. *J. ACM* 7(4): 326–329.

Miranda DM, Conceição SV (2016) The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Systems Appl.* 64:104–116.

Oyola J, Arntzen H, Woodruff DL (2017) The stochastic vehicle routing problem, a literature review. Part II. Solution methods. *EURO J. Transportation Logist.* 6(4):349–388.

Paraskevopoulos DC, Laporte G, Repoussis PP, Tarantilis CD (2016) Resource constrained routing and scheduling: Review and research prospects. Report, CIRRELT.

Qureshi AG, Taniguchi E, Yamada T (2009) An exact solution approach for vehicle routing and scheduling problems with soft time windows. *Transportation Res. Part E: Logist. Transportation Rev.* 45(6):960–977.

Sherali HD, Zhu X (2009) Two-stage stochastic mixed-integer programs: Algorithms and insights. *Adv. Appl. Math. Global Optim.* 17:1–31.

Sniedovich M (1981) Technical note—analysis of a preference order traveling salesman problem. *Oper. Res.* 29(6):1234–1237.

Souyris S, Cortés CE, Ordóñez F, Weintraub A (2013) A robust optimization approach to dispatching technicians under stochastic service times. *Optim. Lett.* 7(7):1549–1568.

Taillard É, Badeau P, Gendreau M, Guertin F, Potvin J-Y (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Sci.* 31(2):170–186.

Taş D, Gendreau M, Dellaert N, Van Woensel T, De Kok AG (2014) Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *Eur. J. Oper. Res.* 236(3):789–799.

Yuan B, Liu R, Jiang Z (2015) A branch-and-price algorithm for the home healthcare scheduling and routing problem with stochastic service times and skill requirements. *Internat. J. Production Res.* 53(24):7450–7464.

Zhang T, Chaovalitwongse WA, Zhang Y (2012) Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Comput. Oper. Res.* 39(10):2277–2290.

Zhang Y, Baldacci R, Sim M, Tang J (2019) Routing optimization with time windows under uncertainty. *Math. Programming* 175(1-2):263–305. **Q: 17**

AUTHOR QUERIES

DATE 12/3/2019

JOB NAME TRSC

ARTICLE 20190956

QUERIES FOR AUTHOR Hashemi Doulabi, Pesant, and Rousseau

THIS QUERY FORM MUST BE RETURNED WITH ALL PROOFS FOR CORRECTIONS

Q: 1_Please provide all funding information, if applicable, including the names of the institutions as well as any grant numbers associated with the funding, or confirm that no funding was received.

Q: 2_Your color figures appear in color in your page proof to simulate their presentation in color online, but they will be converted to grayscale in the print version of your article because you have not requested color processing for print. Please (a) check your figures to confirm that they will be sufficiently clear in grayscale presentation in the print version of your article and (b) modify your figure legends as necessary to remove any references to specific colors in the figure.

Q: 3_Please confirm that the article title, author names, affiliations, and email addresses are set correctly. If applicable, please provide author ORCID numbers.

Q: 4_Please provide department or section in affiliation b and postal code in affiliations a-d.

Q: 5_Please confirm that keywords are correct as set.

Q: 6_Please confirm that heading levels are correct as set.

Q: 7_Please verify that all displayed equations and in-text math notations are set correctly.

Q: 8_Per INFORMS style, built-up fractions included within paragraph text should be converted to their in-line form, i.e., with a solidus rule, as in $[a + (b/x)]^{1/2}$, whenever possible. Long or complex in-line formulas (within paragraph text) should be set as display equations. Please review all in-text equations and modify as necessary to be represented accurately as either display or in-line equations.

Q: 9_Per INFORMS style, all variables should be italic and all vectors should be bold. Please confirm that all terms have been formatted properly throughout.

Q: 10_Per style, sentences that begin with mathematical notation should be avoided. Please rewrite sentences that begin with mathematical notation throughout the text.

Q: 11_Please spell out CPLEX and MTZ in the text.

Q: 12_Per journal style, if your paper includes any citations to specific figures, tables, appendices, sections, or enumerations from other publications, they are presented in lowercase to avoid confusion with citations to similar components of the current paper (which are capitalized). Please confirm throughout.

Q: 13_Please provide editors, page range, and publisher's name, city, state, and country in ref. Ahmed.

Q: 14_Please provide issue number for refs. Chen et al., Miranda and Conceição, and Sherali and Zhu.

Q: 15_Please provide editors and publisher's city, state, and country in refs. Di Mascolo et al. and Jaumard et al.

Q: 16_Please provide institution's city, state, and country in refs. Drexl and Sebastian 2007 and Paraskevopoulos et al.

Q: 17_Reference Zhang et al. 2019 has been updated. Please verify.

Q: 18_Please check that all tables and figures, including their titles and notes (which may contain edits) are set correctly.

Q: 19_The unnumbered tables in text have been numbered per INFORMS style. Table numbers have been edited accordingly. Please verify.

Q: 20_Please define INF in Tables 4 and legends.