Pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times

ARTICLE INFO

Keywords:
Pickup and delivery problem
Stochastic travel times
Hard time windows
Branch and price

ABSTRACT

Due to the uncertain nature of the traffic system, it is not trivial for delivery companies to reliably satisfy customers' time windows. To guarantee the reliability of the pickup and delivery service under stochastic and time-dependent travel times, we consider a pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. We propose a chance-constrained model where the operational cost and the service's reliability are considered. To quantify the service reliability, every node is associated with a desired node service level, and there exists a global service level, both measured by success probabilities. We present an estimation method for arrival times and success probabilities under stochastic travel and service times. We propose an exact solution approach based on a branch-price-and-cut framework, where a labeling algorithm generates columns. Computational experiments are conducted to assess the effectiveness of the solution framework, and Monte Carlo simulations are used to show that the proposed method can generate routes that satisfy both node and global service levels.

1. Introduction

The pickup and delivery problem (PDP) is an extensively studied routing problem. Customer requests consist of two parts: a pickup at one location and a delivery at another. The pickup and delivery problem has a wide range of applications, including ridesharing services (Wang et al., 2016) and meal delivery services (Aziez et al., 2020). In pickup and delivery services, an important factor that influences customer satisfaction is the punctuality of the service. Nguyen et al. (2019) states that being able to perform the delivery service within a specified time window is one of the most important attributes in customer's decision-making, given multiple service options. For ridesharing and shuttle services, punctuality is significant to passenger satisfaction. In most used meal delivery apps, like Uber Eats and Meituan, customers can request refunds or cancel the order if the service is later than a specific time. Therefore, companies must perform the service punctually and reliably.

However, it is not trivial to satisfy customers' time windows because of the uncertain nature of the traffic system. Traffic congestion is very common in urban areas where pickup and delivery services are frequently used. Consequently, travel times are usually unpredictable and fluctuate significantly with the time of day (Yazici et al., 2012). It is hard for delivery companies to find efficient routing solutions that balance operational costs and service reliability. To provide reliable pickup and delivery solutions with time window constraints in an urban setting, we propose an algorithm that solves pickup and delivery problems while considering the service's reliability under stochastic and time-dependent travel times.

In a significant part of the vehicle routing literature, the travel times between certain customers are assumed to be deterministic. Most of the existing work that studies stochastic travel times adopts the assumption that the time windows are soft, meaning that the time windows can be violated at the expense of some penalty cost in the objective function. At each customer node, a penalty cost is imposed according to how early or late the vehicle arrives. If a vehicle arrives early, it can start its service right away. A limitation of the above method is that it is impractical to measure the cost of late deliveries and determine such penalty cost function in terms of money in real-world applications. The cost can be much more than the refund, for such late deliveries could impact future sales. An alternative method is to assume time windows are hard. Under the setting of hard time windows, a vehicle must wait until the start of a time window to begin its service if it arrives early. This is common for passenger services since they may not be available before their scheduled pickup time. Arriving later than the end of a time window results in a service failure of the whole route. Compared to the soft time window assumption, the hard time window assumption can evaluate the reliability of the service with the probability of satisfying time windows. In this paper, we assume that time windows are hard and introduce the concept of service level, which indicates the desired probability that delivery is on-time. As customers may have different requirements and expectations of punctuality, we assume each node is associated with a node service level. We also assume a global service level measures the reliability of an overall routing solution. The global service level is a direct indicator for the service provider to evaluate its overall service reliability. The precise definitions of these terms are presented in later sections of the paper.

We consider a pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times (PDPHTW-STDTT). We assume the travel times between customer locations follow a specific probability distribution, and the distribution may depend on when the travel begins. The distribution is considered known and can be derived from historical data. Each customer is assumed to be associated with a hard time window and the desired node service level. There is a desired global service level. Using a chance-constrained approach, we aim to provide a routing solution that satisfies the desired service levels with the minimum operational cost.

We propose an arrival time estimation method based on the works of Jula et al. (2006) and Ehmke et al. (2015). Different from the heuristic solution method proposed by Ehmke et al. (2015), our solution method is an exact solution method based on branch-cut-and-price. We propose a set partitioning formulation of the PDPHTW-STDTT that includes a route success probability constraint. Since it is inefficient and not practical to enumerate all possible routes in the pricing problem, a labeling algorithm is developed to eliminate the less promising routes to accelerate the solution process. However, given the presence of probabilistic information in the routes, deciding if one route is more promising than the other is not trivial. To deal with this challenge, exact dominance rules that handle probabilistic information are proposed. To further accelerate the algorithm, heuristic dominance rules are also proposed. Computational experiments are performed, and Monte Carlo simulations are conducted to show the algorithm's effectiveness under stochastic travel times.

The main contributions of this paper are summarized as follows.

1. We define the pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times and propose a chance-constrained model for the problem.
2. We present an exact algorithm that is based on branch-cut-and-price to solve the proposed problem. New labeling algorithms and dominance rules are proposed in the pricing problem to deal with stochastic travel times and probabilistic information.
3. In the numerical experiments, we evaluate the trade-off between the reliability and the total cost of the routing solution. We also demonstrate the algorithm's effectiveness in guaranteeing desired service levels.

The paper is structured as follows. Section 2 presents a review of the related literature. Section 3 defines PDPHTW-STDTT and introduces the route service level estimation method. In Section 4, the solution method based on branch-cut-and-price is described. Computational results are reported in Section 5, followed by conclusions in Section 6.

2. Literature Review

The literature dealing with uncertainty can be categorized into stochastic vehicle routing problems (SVRP) and robust vehicle routing problems. Adulyasak and Jaillet (2016) summarizes that stochastic vehicle routing problems are applied to cases where known probability distributions can describe the uncertainty. In contrast, the robust vehicle routing problem is proposed to deal with scenarios where probability distributions are hard to estimate. Bertsimas et al. (2011) outlined that, compared to Stochastic VRP, the uncertainty model of robust VRP is not stochastic but deterministic and set-based, and its goal is to develop a solution that is feasible for all possible uncertainty realizations in the given set. Agra et al. (2013) proposed two new formulations for the robust vehicle routing problem with time windows and implemented a cutting-plane algorithm for the path inequalities formulation. Braaten et al. (2017) proposed an efficient heuristic for the robust vehicle routing problem with time windows based on adaptive large neighborhood search. Munari et al. (2019) proposed a compact formulation for the robust vehicle routing problem with time windows and first developed a branch-price-and-cut method to solve the set partitioning formulation of the problem.

In our work, however, we assume the travel time distribution is known since there is sufficient work that studies the distribution of travel times in urban areas using historical data (Yazici et al., 2012), and information is also available from online databases. Therefore, our work falls into the category of SVRP, and the following literature review will focus on how to deal with time windows in the SVRP context. Recent surveys of SVRP can be found in review papers by Oyola et al. (2017, 2018).

Some of the work that studies SVRP does not take time windows into account, and the objectives are usually related to the duration of the routes. In the work by Laporte et al. (1992), the VRP with stochastic travel and service time was considered for the first time. Time windows are not introduced, and the intention is to limit the maximal route duration. The authors presented three different formulations based on stochastic programming and solved the problem with a branch-and-cut approach. Kenyon and Morton (2003) proposed two models with different objective functions. The first objective function minimizes the expectation of the route completion time, while the second maximizes the probability of completing all the routes before a certain time. A solution method based on branch-and-cut is proposed. Van Woensel et al. (2008) proposed a vehicle routing problem with dynamic travel times and utilized queueing theory to determine the travel times on the arcs depending on the time. Lecluyse et al. (2009) studied a vehicle routing problem with stochastic time-dependent travel times without time windows. The proposed method adjusts the objective of the classical VRP that minimizes the expected total travel time by adding the standard deviation of the travel times into the objective function. The method's performance is evaluated on instances of up to 80 customers, which shows that the reliability of the routing solution is improved when the standard deviation of travel times is considered.

In most of the literature studying SVRP with time windows, it is assumed that the time windows are soft. Soft time windows imply that vehicles that arrive earlier than the start of time windows do not have to wait, and early and late arrivals will lead to penalty costs. Ando and Taniguchi (2006) studied the VRPTW with uncertain travel times. Late arrivals generate a penalty cost proportional to the delayed time length. The objective is to minimize the total cost, including operational costs and penalties. A genetic algorithm is used to solve the problem. Li et al. (2010) investigated the VRPTW with stochastic travel and service times. One of the two proposed formulations assumes soft time windows, and a stochastic programming model with resource formulation is given. A tabu-search-based heuristic solves the problem. Taş et al. (2013, 2014a,b) studied vehicle routing problems with stochastic travel times, including soft time windows under both time-independent and time-dependent settings. The objective function is to minimize a total weighted cost, which includes the penalty for delay and earliness. A method to estimate the expectation and variance of arrival times is proposed. In Taş et al. (2013) and Taş et al. (2014a), the problems are solved by a tabu-search based heuristic. In Taş et al. (2014b), a solution method based on column generation and branch-and-price solution is proposed.

Few papers studied the SVRP with hard time windows, which are the most closely related references to our work. In those papers, the key point is to guarantee a certain service level at each customer node. A major challenge is to deal with the truncation of the probability distributions of arrival times caused by the hard time windows, as vehicles have to wait for the start of the time window to begin service. Different methods are proposed to estimate the probability distributions of arrival times after the truncation. The work of Jula et al. (2006) is one of the first efforts that investigate the effect of hard time windows. Both travel times and service times are considered to be stochastic. A Taylor series expansion approximates the mean and variance of arrival time at each customer node. The estimation method deals with nonstationary time-varying travel time distributions. Chebyshev and Chernoff bounds determine if a route meets the required service level. The proposed problem is a traveling salesman problem involving only one vehicle. Li et al. (2010) also proposed a chance-constrained programming formulation for VRP with stochastic travel times, which differs from the above formulation. In the chance-constrained programming model, the service level at each customer node is guaranteed. The probability of route success is derived directly from Monte Carlo simulations without explicit estimation expressions. Ehmke et al. (2015) studied a VRP with stochastic travel times and hard time windows. Using extreme value theory, the authors proposed a different estimation method to approximate the mean and variance of arrival times. When estimating the route success probability after estimating the mean and variance of arrival times, it is approximated under the assumption that the distributions of the arrival times are normal. The authors defined the route feasibility check to solve the problem and integrated it into existing heuristics like tabu-search. To show the effectiveness of the proposed method, the notion of lateness is defined to evaluate the reliability of a route. Miranda and Conceição (2016) proposed a method to approximate the distribution function of arrival times using convolution functions. A metaheuristic is proposed to solve the VRP with stochastic travel times. The authors conducted numerical experiments on instances with up to 100 customers. Gutierrez et al. (2018) extended the arrival times estimation method by Ehmke et al. (2015) by integrating stochastic service times. Several different confidence levels are defined, and a multi-population memetic algorithm is proposed.

All of the work above used heuristics to solve the problem. A limitation of the above heuristic solution methods is that even though the service levels at customer nodes can be guaranteed, there is no guarantee on the global service level. Errico et al. (2018) proposed an exact solution based on a branch-cut-and-price method to solve a VRP with hard time windows. The goal is to guarantee a global service level, which is the probability that all the customer nodes are served on time. However, the stochasticity lies in service times, not travel times. Unlike the work mentioned above, the probability distribution functions of service times are assumed to be discrete. Therefore, the probability of a route's success can be computed directly, and no estimation is needed. The authors presented a set partitioning formulation with a probability constraint and developed a branch-cut-and-price solution framework. Numerical experiments are

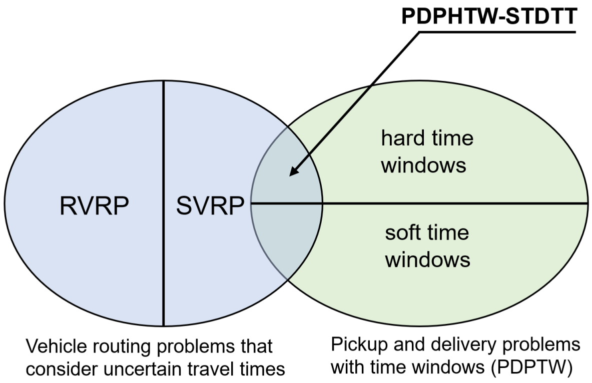

Figure 1: The position of the proposed problem in the literature

performed on instances with up to 50 customers.

Another related research area is the time-dependent vehicle routing problem (TDVRP), whose survey can be found in Gendreau et al. (2015). Ichoua et al. (2003) first introduced the "first in, first out" (FIFO) property under the assumption of time-dependent travel times and proposed a step-wise function speed model which satisfies the property. Dabia et al. (2013) proposed a branch-and-price algorithm for time-dependent vehicle routing problems with time windows, which can solve instances with up to 100 customers. Sun et al. (2018b) studied the time-dependent pickup and delivery problem with time windows and its variants where some customers can be skipped and the departure time can be flexible. The definition of time-dependent under stochastic travel times assumption is slightly different from the traditional definition used by the above papers and is given in the next section.

As for the solution method, our work adopts the branch-cut-and-price framework, which is an exact solution method. The readers interested in the branch-cut-and-price method in VRP applications are referred to a survey by Costa et al. (2019). We study the pickup and delivery problem more restrictive than the vehicle routing problem because the PDP requires the pickup node and the delivery node served by the same vehicle. Therefore, the branch-cut-and-price procedure for solving PDP varies from the VRP version to some degree. The interested readers are referred to the work by Ropke and Cordeau (2009), which provides a detailed branch-cut-and-price solution method for the PDPTW.

Figure 1 summarizes the position of the proposed problem in the literature and its relation with similar problems. The proposed method also belongs to time-dependent vehicle routing problems (TDVRP). Our work contributes to the literature and differs from the other work that studies stochastic travel times VRP with hard time windows in the following ways. First, our work integrates time-dependent travel times. Second, to our knowledge, we are the first to propose an exact solution method by proposing a labeling algorithm with probabilistic information.

3. The PDPHTW-STDTT

In this section, we first introduce the definition of the pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. Then, the method to estimate the means and variances of arrival times is presented. Finally, we show how the route success probability is estimated.

3.1. Problem description

Let $G = (N, A)$ be a directed graph, where $N = \{0, 1, ..., 2n + 1\}$ is the set of nodes and A is the set of arcs $A = \{(i, j)|i, j \in N\}$. Nodes 0 and $2n + 1$ represent the origin and destination depot of the vehicles. The set $N_P = \{1, ..., n\} \in N$ represents the set of pickup nodes and $N_D = \{n + 1, ..., 2n\} \in N$ represents the set of delivery nodes. There is a set of n requests. Each request i is associated with a pickup node i and a delivery node $i + n$. For each request, the pickup node must be visited before the delivery node, and both nodes must be visited by the same vehicle. Each node must be visited only once.

We consider a capacitated pickup and delivery problem, where each request is associated with load size, q_i. Pickup node i has a load of q_i and delivery node $n + i$ has a load of $-q_i$ (drop off). As there is no inventory at the depot, $q_0 = q_{2n+1} = 0$. Each node $i \in N_p \cup N_d$ has a time window $[e_i, l_i]$, where e_i is the start of the time window and l_i

is the end of the time window. The vehicle arrives earlier than e_i must wait until e_i to start its service at node i. d_{ij} denotes the distance between node i and node j. A travel cost c_{ij} is associated with each arc $(i, j) \in A$. For the sake of simplicity, we assume the travel cost is proportional to the travel distance and $c_{ij} = d_{ij}$ in this paper.

There is a fleet of identical vehicles, each with a maximum capacity of Q. We assume there is no limit on the number of vehicles used. However, a fixed cost C_{fixed} is paid when a vehicle is used. All the vehicles depart at the depot at time 0 and finish the route at the depot.

We assume the travel times between nodes and service times at nodes are stochastic. The travel time on each arc $(i, j) \in A$ is a Gaussian random process $X_{ij}(t)$, with t representing the time when the vehicle enters arc (i, j). We assume the mean of $X_{ij}(t)$, $E[X_{ij}(t)]$ is time-dependent, while the variance of $X_{ij}(t)$, $Var[X_{ij}]$ is time-independent, only proportional to the length of the arc, d_{ij}. Following Ehmke et al. (2015)'s notation, we define a variation coefficient c_v, so that $Var[X_{ij}] = c_v d_{ij}$. We assume that stochasticity is only in the travel times but not path choices. Therefore, there is only one path between any two nodes.

It is assumed that the travel times on an individual arc at any particular time t are statistically independent. Most of the literature that studies time-dependent travel times satisfies the "first in, first out" (FIFO) property. The FIFO property under deterministic travel times states that if two vehicles traverse the same arc, the one that leaves first will arrive first. Most literature adopts a speed profile model that consists of step-wise functions and satisfies the FIFO principle proposed by Ichoua et al. (2003). However, when it comes to stochastic and time-dependent travel times, the FIFO property can not be guaranteed at the single-vehicle level because of the uncertainty in travel times.

Nevertheless, based on real-life observations and experiences, on average, vehicles that leave early will arrive early. Therefore, the FIFO property should hold at the macro level, meaning that $E[X_{ij}(t)]$ satisfies the FIFO principle. We assume that $E[X_{ij}(t)]$ is derived from the speed profile model mentioned above. The speed profile model assumes that the scheduling horizon is divided into k intervals, and the vehicle's speed is a constant in each of the intervals. Each arc $(i, j) \in A$ is associated with a speed profile that consists of a constant speed for each interval. We assume all the arcs share the same speed profile, meaning that the speeds on all the arcs are the same at a specific time. The $E[X_{ij}(t)]$ derived by the above method is a piece-wise linear function. Details about the FIFO property and the definition of the speed profile can be found in Ichoua et al. (2003) and Sun et al. (2018a). We note that the proposed method can deal with more complicated travel time assumptions where travel speeds differ on different arcs or the travel times follow a more complicated changing pattern. For the sake of simplicity, we adopt the simplified assumption above.

The service time at each node is a random variable that is time-independent. The service time S_i at node i has mean $E[S_i]$ and variance $Var[S_i]$. It is assumed that S_i follows a normal distribution. Once the service is finished, the vehicle starts to travel to the next node on the route.

A route r is defined by a sequence of nodes $r = (v_0, v_1, ..., v_m, v_{m+1})$, where $v_1, ..., v_m \in N_p \cup N_d$ and v_0, v_{m+1} represent the depot. We define a node as successful if the vehicle arrives at the node within its time window. Node success probability is defined by the probability that a node in a route is successful when given the route and all needed distributions. We define that a route is successful if the vehicle arrives at each node on the route within their time windows. Route success probability is the probability that a given route is successful when given all needed distributions. Each node $i \in N_p \cup N_d$ is associated with a node service level θ_j $(0 \leq \theta_j < 1)$, indicating the desired node success probability by the corresponding customer. We also define a global service level Θ $(0 \leq \Theta < 1)$, indicating the desired probability that all the routes are successful. The objective is to minimize the total costs when the node and global service levels are satisfied.

3.2. Arrival times estimation

Several methods for estimating the arrival times under stochastic travel time and hard time windows assumption are mentioned in our literature review. We adopt the method proposed by Jula et al. (2006), because of its capability of dealing with time-dependent travel times. In addition, this estimation method makes it more practical to generate dominance rules for the proposed labeling algorithm, presented later in this paper. A comparison of the adopted method and Ehmke et al. (2015)'s method can be found in Appendix A.

The main idea of Jula et al. (2006)'s method is to estimate the first and second moments of the arrival times iteratively. Let A_i^r denote the arrival time at node i on route r , and D_i^r denote departure time after service at node i on route r. The means and variances of $X_{ij}(t)$ and S_i are known. When the vehicle travels from node i to node j, the expressions below give the means and variances of the arrival and departure times at node j. The estimations of the means and variances of arrival times and departure times of all the nodes on a route can be derived iteratively by repeating this process. As in Jula et al. (2006), the following equations can be derived:

$$E[A_j^r] \approx E[D_i^r] + E[X_{ij}(E[D_i^r])] \tag{1}$$

$$Var[A_j^r] \approx \{1 + E'[X_{ij}(E[D_i^r])]\}^2 Var[D_i^r] + Var[X_{ij}] \tag{2}$$

$$E(D_j^r) = g_j(E[A_j^r]) + E[S_j] \tag{3}$$

$$Var(D_j^r) \approx \frac{1}{4}\left(\int_{E[A_j^r]-\sigma[A_j^r]}^{E[A_j^r]+\sigma[A_j^r]} g_j'(x)\,dx\right)^2 + Var[S_j] \tag{4}$$

$$g_j(t) = \begin{cases} e_j & if\ t \le e_j \\ t & if\ e_j < t \le l_j \\ M(t - l_j) + l_j & if\ t > l_j \end{cases} \tag{5}$$

$$g_j'(t) = \begin{cases} 0 & if\ t \le e_j \\ 1 & if\ e_j < t \le l_j \\ M & if\ t > l_j \end{cases} \tag{6}$$

where $E'[X_{ij}(t)]$ is the derivative of $E[X_{ij}(t)]$ with respect to t. $\sigma[A_j^r]$ is the standard deviation of A_j^r. (5) and (6) show that $g_j(t)$ is a piece-wise linear function to model the effect of the hard time window at node j, and $g_j'(t)$ is the derivative of $g_j(t)$ with respect to t. M represents a very big number. Expression (1) and (2) estimate the expectation and variance of arrival time at a node based on the expectation and variance of departure time at the previous node and travel time between the nodes. Expression (3) and (4) calculate the expectation and variance of departure time at a node based on the expectation and variance of arrival time and service time at the same node.

3.3. Route success probability estimation

Given the mean and the variance of the arrival time at a node, the node success probability can be estimated. We adopt the estimation method in Ehmke et al. (2015) that assumes the arrival times conform to a normal distribution. Note that the actual distribution of arrival times may be skewed because of hard time windows, and the actual distribution can not be computed precisely. Thus, the node success probability at node i on route r, denoted by P_i^r, is estimated by

$$P_i^r = P(A_i^r \le l_i) \approx \Phi\left(\frac{l_i - E[A_i^r]}{\sqrt{Var[A_i^r]}}\right) \tag{7}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

The estimated route success probability P^r is the product of the node success probabilities of the nodes on the route and is given by the following equation:

$$P^r = \prod_{i \in r} P_i^r \tag{8}$$

A route is considered feasible when all the node service levels of the nodes on the route and all constraints in the pickup and delivery problem are satisfied. Let $\mathcal{R}$ be the set of all feasible routes. Let $\mathcal{R}_{chosen} = \{r^1, r^2, ..., r^k\}$ be the set of routes that are chosen in the routing solution. Global success probability P_{global} is the probability that all chosen routes are successful. To satisfy the desired node service levels and the desired global service level, we have the following constraints:

$$P_i^r \ge \theta_i \qquad \forall r \in \mathcal{R}_{chosen},\ \forall i \in r \tag{9}$$

$$P_{global} = \prod_{r \in \mathcal{R}_{chosen}} P^r \geq \Theta \tag{10}$$

where θ_i is the node service level of node i, and Θ is the global service level. Expression (9) states that each node's success probability satisfies its node service level. Expression (10) states that the global success probability satisfies the global service level.

4. Solution approach

This section presents a solution method for PDPHTW-STDTT based on branch-cut-and-price. We first provide the set partitioning formulation of the problem. Then, we emphasize how to conduct column generation and solve the pricing problem. Finally, we introduce the branch-cut-and-price framework, the cutting plane method, and the branching strategies.

4.1. Set partitioning formulation

We define the cost of a route $r = (v_0, v_1, ..., v_m, v_{m+1})$ as $c_r = C_{fixed} + \sum_{i=0}^{m} c_{v_i v_{i+1}}$. Let the parameter a_{ir} denotes if node i is visited on route r. $a_{ir} = 1$ if route r visits node i, otherwise $a_{ir} = 0$. The binary variable x_r indicates if route r is chosen in the solution.

The PDPHTW-STDTT can be formed in the following set partitioning formulation.

$$minimize \quad \sum_{r \in \mathcal{R}} c_r x_r \tag{11}$$

$$s.t \quad \sum_{r \in \mathcal{R}} a_{ir} x_r = 1 \qquad \forall i \in N_p \tag{12}$$

$$\sum_{r \in \mathcal{R}} x_r \, ln(P^r) \geq ln(\Theta) \tag{13}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R} \tag{14}$$

The objective function (11) minimizes the total cost, including fixed vehicle and arc costs. Constraint (12) indicates that each pickup node is visited exactly once. As we pair a pickup node with its corresponding delivery node in the routes, each delivery node is guaranteed to be visited exactly once. Constraint (13) ensures the global service level is satisfied and is derived from taking the logarithm of both sides of (10).

4.2. Column generation

The set of all feasible routes is large and cannot be explicitly enumerated. We utilize a column generation method (Desaulniers et al., 2006) to generate routes iteratively. Only a subset of $\mathcal{R}$ is maintained in the linear program. In each iteration, a relaxation of the restricted master problem (RMP) is solved, generating the dual multipliers of constraints (12) and (13), π_i ($\forall i \in N_p$) and λ respectively. Then, the pricing algorithm is called to find routes with negative reduced costs, which are added to the subset of $\mathcal{R}$ in the RMP. The iteration stops when no such route is found. The subset of $\mathcal{R}$ is initialized with the set of routes that visit only one pair of pickup and delivery nodes.

The reduced cost $\bar{c_r}$ of a route $r \in \mathcal{R}$ is calculated as:

$$\bar{c_r} = c_r - \sum_{i \in N_p} a_{ir} \pi_i + ln(P^r)\lambda \tag{15}$$

$$= C_{fixed} + \sum_{i=0}^{m} c_{v_i v_{i+1}} - \sum_{i \in N_p} a_{ir} \pi_i + \sum_{i \in r} ln(P_i^r)\lambda \tag{16}$$

4.3. Pricing algorithm

The pricing algorithm finds routes with negative reduced cost (15). The pricing problem is a variant of the shortest path problems with resource constraints (ESPPRC, Irnich and Desaulniers (2005)) and is mainly solved by a labeling algorithm (Righini and Salani, 2008). The main idea of the labeling algorithm is to represent every partial route that starts at node 0 and ends at any node i with a label L. The label contains information about the route, including the current load on the vehicle, the delivery tasks that need to be finished, the accumulated reduced cost, and the arrival times and probability estimations mentioned in Section 3, etc. Labels are extended from existing labels toward the destination node. During the extension process, we check if the new label satisfies the constraints in the assumption of the problem, like the capacity constraint, node service level constraint, etc. In other words, the routes generated by the labeling algorithm are feasible. To speed up the algorithm, dominance rules are usually proposed to eliminate routes that are not useful.

Note that a bidirectional labeling algorithm is computationally more efficient than a unidirectional labeling algorithm. We only consider a forward labeling algorithm. This is because the probability information can not be extended backward.

4.3.1. Definition of labels

A partial route $r^k = (v_0, v_1, ..., v_m)$ is a route that does not necessarily end at the destination. As the number of partial routes may become large in the labeling algorithm, we use superscript k to differentiate them. Each partial route r^k is associated with a label L^k.

$L^k = [\mathcal{N}(L^k), \mathcal{M}(L^k), \mathcal{Q}(L^k), \mathcal{C}(L^k), E_A(L^k), V_A(L^k), E_D(L^k), V_D(L^k), Prob(L^k), \mathcal{O}(L^k), \mathcal{U}(L^k), \pi(L^k)]$. The components of L^k are explained below:

- $\mathcal{N}(L^k)$ is the last node visited on partial route r^k.
- $\mathcal{M}(L^k)$ is the partial route r^k, which contains all the nodes visited in the order that they are visited.
- $\mathcal{Q}(L^k)$ is the load in the vehicle after visiting node $\mathcal{N}(L^k)$.
- $\mathcal{C}(L^k)$ is the accumulated cost along partial route r^k after visiting node $\mathcal{N}(L^k)$, without considering the fixed cost.
- For the necessary information to estimate arrival times and success probabilities,
  $E_A(L^k)$, $V_A(L^k)$ are the estimated mean and variance of the arrival time at node $\mathcal{N}(L^k)$.
  $E_D(L^k)$, $V_D(L^k)$ are the estimated mean and variance of the departure time at node $\mathcal{N}(L^k)$.
  $Prob(L^k)$ is the estimated route success probability of the partial route r^k.
- $\mathcal{O}(L^k) \subseteq N_p$ is the set of pickup nodes that have been visited in partial route r^k, whose corresponding delivery nodes have not been visited.
- $\mathcal{U}(L^k) \subseteq N_p$ is the set of *unreachable requests* . A node i is said to be *unreachable* if:
  1. node i has already been visited on partial route r^k (so that it will not be visited again), or
  2. going directly from current node $\mathcal{N}(L^k)$ to pickup node i can not satisfy the service levels, which is $Prob(L^k) \cdot P_i^{r^k \oplus i} < \theta_i$, or $Prob(L^k) \cdot P_i^{r^k \oplus i} < \Theta$. Here, $r^k \bigoplus i$ represents the new partial route after extending partial route r^k to node i.
- $\pi(L^k)$ is the sum of the dual values associated with Constraint (12) on the partial route r^k.

4.3.2. Label extension

Given a label L^k associated with partial route r^k, its extension to node i is feasible only if:

$$\mathcal{Q}(L^k) + q_i \leq Q \qquad \forall i \in N \tag{17}$$

$$i \notin \mathcal{U}(L^k) \qquad \forall i \in N_p \tag{18}$$

$$i - n \in \mathcal{O}(L^k) \qquad \forall i \in N_d \tag{19}$$

$$\mathcal{O}(L^k) = \emptyset \qquad if \;\; i = 2n+1 \tag{20}$$

$$Prob(L^k) \cdot P_i^{r^k \oplus i} \geq max\{\theta_i, \Theta\} \tag{21}$$

Condition (17) guarantees the vehicle capacity constraint will not be violated. Conditions (18) - (20) make sure the precedence constraint of the pickup and delivery problem is satisfied. Condition (21) ensures that the node and global service levels are satisfied.

If it is feasible to extend an existing label L^k to node i, a new label L^h is created, representing the new partial route $r^k \bigoplus i$. The label updating rules are:

$$\mathcal{N}(L^h) = i \tag{22}$$

$$\mathcal{O}(L^h) = \begin{cases} \mathcal{O}(L^k) \cup \{i\} & if\ i\ \in\ N_p \\ \mathcal{O}(L^k) \backslash \{i-n\} & if\ i\ \in\ N_d \end{cases} \tag{23}$$

$$\mathcal{U}(L^h) = \begin{cases} \mathcal{U}(L^k) \cup \{i\} & if\ i\ \in\ N_p \\ \mathcal{U}(L^k) & otherwise \end{cases} \tag{24}$$

$$\mathcal{Q}(L^h) = \mathcal{Q}(L^k) + q_i \tag{25}$$

$$C(L^h) = C(L^k) + c_{\mathcal{N}(L^k)i} \tag{26}$$

$$\pi(L^h) = \begin{cases} \pi(L^k) + \pi_i & if\ i\ \in\ N_p \\ \pi(L^k) & otherwise \end{cases} \tag{27}$$

$$E_A(L^h) = E_D(L^k) + E[X_{\mathcal{N}(L^k)i}(E_D(L^k))] \tag{28}$$

$$V_A(L^h) = (1 + E'[X_{\mathcal{N}(L^k)i}(E_D(L^k))])^2 V_D(L^k) + Var[X_{\mathcal{N}(L^k)i}] \tag{29}$$

$$E_D(L^h) = g_i(E_A(L^h)) + E[S_i] \tag{30}$$

$$V_D(L^h) = \frac{1}{4}(\int_{E_A(L^h)-\sqrt{V_A(L^h)}}^{E_A(L^h)+\sqrt{V_A(L^h)}} g_i'(x)\,dx)^2 + Var[S_i] \tag{31}$$

$$Prob(L^h) = Prob(L^k) P_i^{r^k \bigoplus i} = Prob(L^k) \Phi\left(\frac{l_i - E_A(L^h)}{\sqrt{V_A(L^h)}}\right) \tag{32}$$

Expressions (22) - (27) are derived from the definition of the label. Expressions (28) - (32) are derived from the conclusions in Section 3.

4.3.3. Label dominance rules

The concept of label dominance is adopted to restrict the total number of labels and speed up the algorithm. A dominance check is performed within the set of labels with the same ending node. If another dominates a label, the dominated label will be discarded.

Definition of dominance: Consider two feasible partial routes r^1 and r^2, with labels L^1 and L^2, respectively. If $\mathcal{N}(L^1) = \mathcal{N}(L^2)$, a dominance check with the following principles will be performed. We say L^1 dominates L^2 so that L^2 can be eliminated, if:

(1) If extending r^2 to any node i generates a feasible route $r^2 \bigoplus i$, then extending r^1 to node i also generates a feasible route $r^1 \bigoplus i$.

(2) For any such extensions $r^1 \bigoplus i$ and $r^2 \bigoplus i$, the reduced cost of $r^1 \bigoplus i$ is always less than or equal to the reduced cost of $r^2 \bigoplus i$.

We first propose and prove Lemma 1, which will be used to prove the exact dominance rules.

Lemma 1: For partial route r^1 associated with label L^1 and r^2 associated with label L^2, if $E_A(L^1) \leq E_A(L^2)$, $Prob(L^1) \geq Prob(L^2)$ and $(1+f'_{max})^2 V_D(L^1) \leq (1+f'_{min})^2 V_D(L^2)$, then $P_i^{r^1 \bigoplus i} \geq P_i^{r^2 \bigoplus i}$ holds for feasible extensions to any node i, where f'_{max} and f'_{min} are the maximum and minimum value of the derivative of the expected travel time with respect to time, over all arcs and over the scheduling horizon.

Proof of Lemma 1:

Assume $r^1 \bigoplus i$ and $r^2 \bigoplus i$ are feasible extensions of r^1 and r^2 to an arbitrary node i, and the associated labels are L^{1^*} and L^{2^*}, respectively.

We first show that $E_A(L^{1^*}) \leq E_A(L^{2^*})$. Given expression (30):

$$E_D(L^h) = g_i(E_A(L^h)) + E[S_i]$$

where $g_i(\cdot)$ is non-decreasing, $E_A(L^1) \leq E_A(L^2)$, and $E[S_j]$ is constant, we derive that $E_D(L^1) \leq E_D(L^2)$. With $E_D(L^1) \leq E_D(L^2)$, and expression (28):

$$E_A(L^h) = E_D(L^k) + E[X_{\mathcal{N}(L^k)i}(E_D(L^k))]$$

where we assume the FIFO property of travel time function $E[X_{ij}(t)]$, we get $E_A(L^{1^*}) \leq E_A(L^{2^*})$.

We then show that $V_A(L^{1^*}) \leq V_A(L^{2^*})$. In the Lemma statement, we have:

$$(1 + f'_{max})^2 V_D(L^1) \leq (1 + f'_{min})^2 V_D(L^2) \tag{33}$$

Because of the definition of f'_{max} and f'_{min}, we have:

$$E'[X_{\mathcal{N}(L^1)i}(E_D(L^1))] \leq f'_{max} \tag{34}$$

$$E'[X_{\mathcal{N}(L^2)i}(E_D(L^2))] \geq f'_{min} \tag{35}$$

From inequalities (33) - (35), it can be derived that

$$(1 + E'[X_{\mathcal{N}(L^1)i}(E_D(L^1))])^2 V_D(L^1) \leq (1 + E'[X_{\mathcal{N}(L^2)i}(E_D(L^2))])^2 V_D(L^2) \tag{36}$$

Given (36), and expression (29):

$$V_A(L^h) = (1 + E'[X_{\mathcal{N}(L^k)i}(E_D(L^k))])^2 V_D(L^k) + Var[X_{\mathcal{N}(L^k)i}]$$

where $Var[X_{ij}]$ is time-independent, we have $V_A(L^{1^*}) \leq V_A(L^{2^*})$.

Since $E_A(L^{1^*}) \leq E_A(L^{2^*})$ and $V_A(L^{1^*}) \leq V_A(L^{2^*})$, we have $\frac{l_i - E_A(L^{1^*})}{\sqrt{V_A(L^{1^*})}} \geq \frac{l_i - E_A(L^{2^*})}{\sqrt{V_A(L^{2^*})}}$. Since $\Phi(\cdot)$ is monotonic increasing, $P_i^{r^1 \bigoplus i} = \Phi\left(\frac{l_i - E_A(L^{1^*})}{\sqrt{V_A(L^{1^*})}}\right) \geq \Phi\left(\frac{l_i - E_A(L^{2^*})}{\sqrt{V_A(L^{2^*})}}\right) = P_i^{r^2 \bigoplus i}$. $\square$

Then, the exact pricing rule is proposed. If all eight conditions in Proposition 1 are satisfied, it is guaranteed that one label dominates the other label, according to the definition of dominance.

Proposition 1 (Exact pricing): Label L^2 is dominated by L^1, if the following 8 conditions are satisfied:

(1) $\mathcal{N}(L^1) = \mathcal{N}(L^2)$.
(2) $\mathcal{U}(L^1) \subseteq \mathcal{U}(L^2)$.
(3) $\mathcal{O}(L^1) = \mathcal{O}(L^2)$. Note that under the stochastic and time-dependent assumption, the triangle inequality for travel times does not necessarily hold, so this is different from $\mathcal{O}(L^1) \subseteq \mathcal{O}(L^2)$ proposed in Ropke and Cordeau (2009).
(4) $Q(L^1) \leq Q(L^2)$.
(5) $C(L^1) - \pi(L^1) + \lambda Prob(L^1) \leq C(L^2) - \pi(L^2) + \lambda Prob(L^2)$.
(6) $E_A(L^1) \leq E_A(L^2)$.
(7) $(1 + f'_{max})^2 V_D(L^1) \leq (1 + f'_{min})^2 V_D(L^2)$.
(8) $Prob(L^1) \geq Prob(L^2)$.

Proof of Proposition 1:

We prove that it is sufficient to derive dominance relationship between L^1 and L^2 from proposition 1. In other words, we want to show the definition of dominance is satisfied if given all the conditions in Proposition 1. Assume there exists partial routes r^1 and r^2 whose corresponding labels L^1 and L^2 satisfy conditions (1) - (8) in proposition 1, and i is an arbitrary node.

Regarding number (1) in the definition of dominance, we want to show if $r^2 \bigoplus i$ is feasible based on the definition of feasibility in (17) - (21), $r^1 \bigoplus i$ is also feasible, for any node i. For the vehicle capacity constraint, the load associated with partial route $r^1 \bigoplus i$ does not violate the capacity constraint (17) if $r^2 \bigoplus i$ satisfies the requirement because of condition (4). For the route success probability constraint, condition (8) gives $Prob(L^1) \geq Prob(L^2)$, and Lemma 1 gives $P_i^{r^1 \bigoplus i} \geq P_i^{r^2 \bigoplus i}$ for all nodes i. Therefore, $Prob(L^1)P_i^{r^1 \bigoplus i} \geq Prob(L^2)P_i^{r^2 \bigoplus i}$ states that $r^1 \bigoplus i$ does not violate the probability constraint (21) provided that $r^2 \bigoplus i$ satisfies the constraints. As for precedence constraints of the pickup and delivery problem, conditions (2) and (3) guarantee that if $r^2 \bigoplus i$ satisfies (18) - (20), $r^1 \bigoplus i$ will also satisfy. Thus, for any node i, if $r^2 \bigoplus i$ is feasible, $r^1 \bigoplus i$ is also feasible.

Regarding number (2) in the definition, let $\bar{c}(L^k)$ be the reduced cost of L^k as defined in (15). We assume the labels for new feasible partial routes $r^1 \bigoplus i$ and $r^2 \bigoplus i$ are L^{1^*} and L^{2^*}. We show that $\bar{c}(L^{1^*}) \leq \bar{c}(L^{2^*})$ for all feasible node i.

$$\begin{aligned}
\bar{c}(L^{1^*}) &= C(L^{1^*}) - \pi(L^{1^*}) + \lambda \cdot ln(Prob(L^{1^*})) + C_{fixed} && (37)\\
&= [C(L^1) + c_{\mathcal{N}(L^1)i}] - [\pi(L^1) + \pi_i] + \lambda \cdot ln(Prob(L^1) \cdot P_i^{r^1 \bigoplus i}) + C_{fixed} && (38)\\
&= [C(L^1) - \pi(L^1) + \lambda \cdot ln(Prob(L^1))] + \lambda \cdot ln(P_i^{r^1 \bigoplus i}) + c_{\mathcal{N}(L^1)i} - \pi_i + C_{fixed} && (39)\\
&\leq [C(L^2) - \pi(L^2) + \lambda \cdot ln(Prob(L^2))] + \lambda \cdot ln(P_i^{r^1 \bigoplus i}) + c_{\mathcal{N}(L^1)i} - \pi_i + C_{fixed} && (40)\\
&\leq [C(L^2) - \pi(L^2) + \lambda \cdot ln(Prob(L^2))] + \lambda \cdot ln(P_i^{r^2 \bigoplus i}) + c_{\mathcal{N}(L^1)i} - \pi_i + C_{fixed} && (41)\\
&= C(L^{2^*}) - \pi(L^{2^*}) + \lambda \cdot ln(Prob(L^{2^*})) + C_{fixed} && (42)\\
&= \bar{c}(L^{2^*}) && (43)
\end{aligned}$$

Dual multipliers λ and π_i for $i \in N_p$ are defined in Section 4.2. For $i \notin N_p$, we define $\pi_i = 0$. Inequality (40) is derived from condition (5), and inequality (41) is derived from Lemma 1. □

4.3.4. Heuristic Pricing

With the exact dominance rules in proposition 1, we eliminate labels carefully to guarantee that the exact optimal solution can be found, which could be very time-consuming. To speed up the algorithm, we propose heuristic pricing rules which are less restrictive than proposition 1. As condition (7) in Proposition 1 may be very strict when the travel times change rapidly with time, we loosen it in the heuristic pricing rules. The algorithm may end up with a sub-optimal solution, and the performance of the heuristic pricing is evaluated in Section 5.3.

Proposition 2 (Heuristic pricing): Label L^2 is dominated by L^1 heuristically, if:

(1) $\mathcal{N}(L^1) = \mathcal{N}(L^2)$
(2) $\mathcal{U}(L^1) \subseteq \mathcal{U}(L^2)$
(3) $\mathcal{O}(L^1) = \mathcal{O}(L^2)$
(4) $\mathcal{Q}(L^1) \leq \mathcal{Q}(L^2)$
(5) $C(L^1) - \pi(L^1) + \lambda Prob(L^1) \leq C(L^2) - \pi(L^2) + \lambda Prob(L^2)$
(6) $E_A(L^1) \leq E_A(L^2)$
(7) $V_D(L^1) \leq V_D(L^2)$
(8) $Prob(L^1) \geq Prob(L^2)$

4.4. Branch-cut-and-price framework

The branch-cut-and-price algorithm is a branch-and-bound-based method to solve integer programs, which integrates column generation where variables are generated dynamically.

Algorithm 1 describes the structure of the proposed branch-cut-and-price algorithm. The interested readers are referred to Barnhart et al. (1998) for more details of a branch-and-price algorithm.

The column generation and pricing algorithm used that generate new variables are introduced in Section 4.2 and 4.3 in detail. The following subsections describe the cutting planes and branching strategies used.

4.4.1. Cutting planes

Cuts or valid inequalities strengthen the lower bound after column generation is finished at a given node of the branch-and-bound tree. Jepsen et al. (2008) introduce subset-row cuts (SRC) defined over routing variables x_i, which are:

$$\sum_{r \in \mathcal{R}} \lfloor p \sum_{i \in S} a_{ir} \rfloor x_r \leq \lfloor p|S| \rfloor \qquad \forall S \subseteq N_p, 0 < p < 1 \tag{44}$$

SRCs are efficient in strengthening the linear relaxation. However, adding the above cut to the RMP makes the pricing problem more challenging, as extra information is needed in the labels to deal with the new constraints. Pecin

Algorithm 1 Branch-cut-and-price algorithm

while stopping criteria is not satisfied and there exists unprocessed nodes **do**
  Select an unprocessed node from the branch-and-bound tree.
  $Flag \leftarrow True$
  while True **do**
    Solve the linear relaxation of the selected node's restricted master problem (RMP).
    Find routes with negative reduced costs by solving the pricing problem repeatedly until no such route can be found. Add the found routes to the RMP.
    if the current objective value of the RMP is greater than the current upper bound **then**
      $Flag \leftarrow False$
      break
    end if
    Generate cuts using the method introduced in Section 4.4.1.
    if new cuts are found **then**
      Add the constraints to the RMP.
    else
      break
    end if
  end while
  if $Flag$ is $False$ **then**
    continue
  end if
  if the solution to the relaxation of the RMP is an integer solution **then**
    Update the upper bound.
  else
    Generate child nodes using the branching strategies introduced in 4.4.2. Add the generated nodes to the branch-and-bound tree.
  end if
end while

et al. (2017) propose a generalization of SRCs named limited-memory subset-row cuts (lm-SRC), which have less impact on the pricing problem and make the pricing process faster. We apply the limited-memory subset-row cuts in the following form:

$$\sum_{r\in\mathcal{R}} \alpha(C, M, p, r)x_r \leq \lfloor p|C| \rfloor \qquad \forall C \subseteq M \subseteq N_p, 0 < p < 1 \tag{45}$$

where M is an additional memory set, $C \subseteq M \subseteq N_p$, and α is a function of C, M, p, r. The algorithm to compute coefficient α and decide memory set M is explained in Pecin et al. (2017) in detail and is not included here. Even though each label should have an additional dimension for each lm-SRC, the coefficients do not need to be stored in the labels. Therefore, the structure of the pricing problem is not changed, and only minor changes are needed. The interested readers are referred to Pecin et al. (2017) and Sun et al. (2018b). In this paper, we generate cuts with $|C| = 3$ and $p = 0.5$.

4.4.2. Branching strategies

When the solution to the linear relaxation of the RMP is not integral, we follow the branch-and-bound framework and divide the feasible region of the current node into two regions, generating two child nodes of the current node. The following branching rules are applied. First, we branch on the total number of vehicles used, $n_v = \sum_{r\in\mathcal{R}} x_r$. If n_v is fractional, constraints $\sum_{r\in\mathcal{R}} x_r \leq \lfloor n_v \rfloor$ and $\sum_{r\in\mathcal{R}} x_r \geq \lceil n_v \rceil$ are added to the child nodes. The pricing problem will add a dual variable associated with the vehicle number constraint. Then, we branch on the arc-flow variables. The arc (i, j) whose flow is closest to 0.5 is selected. On the first branch, we set the flow on this arc to 0 by removing arc (i, j) from arc set A of this child node. Also, the routes containing arc (i, j) are removed from the RMP of the child node.

Table 1: Speed profiles

speed profile number	speed in interval (0, 120)	speed in interval (120,600)	speed in interval (600,720)	speed in interval (720,840)
1	1	1	1	1
2	0.9	1.1	0.9	1.1
3	0.75	1.2	0.8	1.25
4	0.67	1.33	0.88	1.33

On the second branch, we set the flow to 1, and all other arcs starting from i: $(i, k), k \neq j$ are removed from arc set A.

5. Computational experiments

In this section, we first introduce the design of the computational experiments. Instances for the proposed problem are generated based on the instances introduced by Ropke et al. (2007) for the PDPTW. Then, the performance of the heuristic pricing is evaluated. Finally, we present the results of the experiments and show the proposed method is adequate to satisfy the desired service levels. Besides, the importance of modeling time-dependent travel times is evaluated, and the impact of service levels is analyzed.

5.1. Experiment design

Ropke et al. (2007) introduced a set of instances for the PDPTW, where the coordinates of the pickup and delivery nodes are randomly chosen according to a uniform distribution over a [0, 50]×[0, 50] square. The depot is located at $(25, 25)$. Our instances adopt the same node coordinates, maximum capacity of vehicles, load quantities, and start times of time windows as those in Ropke et al. (2007). The maximum capacity of a vehicle Q is set to 15. Ropke et al. (2007) assumes the width of time windows is 60 in some instances and 120 in others. We modify Ropke's assumptions on time window width, making the width of the windows range from 60 to 100. For the fixed cost, we adopt the same parameter as Ropke and Cordeau (2009) that sets a vehicle's fixed cost F_{fixed} to 10000. We assume the cost of traversing an arc equals the arc's length ($c_{ij} = d_{ij}$).

For the parameters regarding stochastic travel and service times, the mean of the service time at each node is randomly chosen from values $[8, 10, 12]$, and the variance of the service time at each node is randomly chosen from values $[1, 2, 3]$. The variance of travel times is decided by variation coefficient c_v, so that $Var[X_{ij}] = c_v d_{ij}$. In the experiments, c_v is an adjustable parameter ranging from 0 to 0.4. The means of travel times are decided by the speed profile model (Ichoua et al., 2003), which assumes the travel speed is constant in a time interval on all arcs. We divide the scheduling horizon (0,840) into 4 time intervals: $[(0, 120), (120, 600), (600, 720), (720, 840)]$, representing four periods in a day, as suggested in Sun et al. (2018b). Four different speed profiles are used in the experiments, one at a time, presented in Table 1. The mean of the travel times of each arc $E[X_{ij}(t)]$ can be computed according to the speed profiles, and $E[X_{ij}(t)]$ are piece-wise linear functions.

As for service levels, we focus on studying the global service level, rather than node service levels, in our main experiment. This is because some computational experiments studying node service levels can be found in previous work, including Miranda and Conceição (2016) and Gutierrez et al. (2018). The estimation algorithms show that the node service levels can be guaranteed to a large extent. However, no computational study focuses on the global service level. Here, we evaluate how well the global service level can be satisfied using our estimation methods, and we analyze the impact of the different global service levels Θ. The desired node service levels are set to 70% in the experiments. The instance set can be found in the following link: `bit.ly/3uX5zF9`.

We use Monte Carlo simulations to determine the actual success probabilities as benchmarks. We evaluate the algorithm's effectiveness by checking if the simulated success probabilities satisfy the desired service levels. We also compare the difference between the estimated success probabilities and simulated success probabilities. In each Monte Carlo simulation, travel and service times are sampled from the assumed probability distributions. We compute the simulated success probability after 1000 runs.

We compare the proposed algorithm's routes with those generated by a deterministic routing algorithm that does not consider the stochasticity in travel times and service times (based on Ropke and Cordeau (2009)). The deterministic

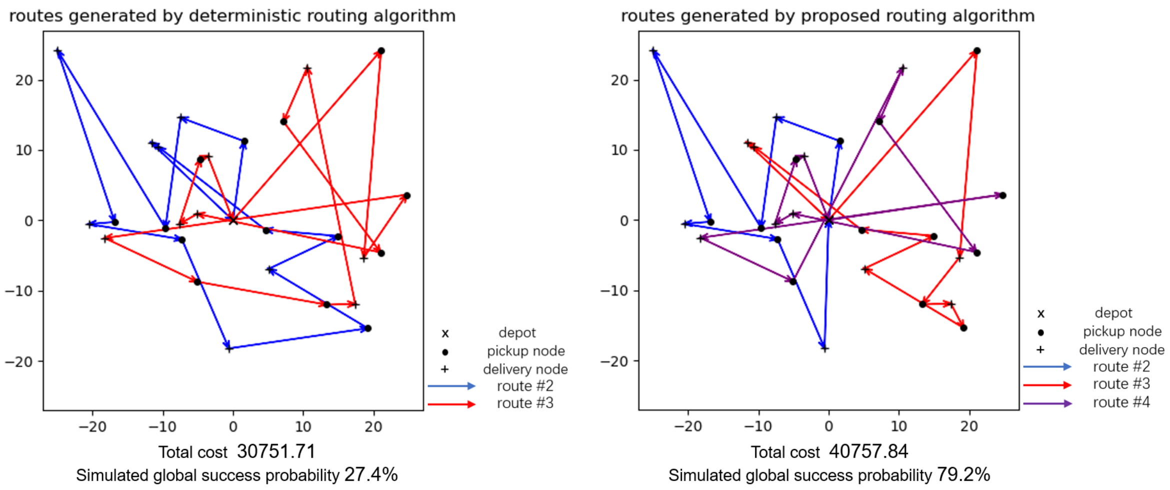

Figure 2: Comparison of deterministic routing algorithm with proposed routing algorithm

routing algorithm is introduced in Section 5.2. In this way, the difference in the simulated success rate illustrates the effectiveness of the proposed algorithm.

The proposed algorithm is implemented in Java with linear program solver Gurobi 9.0.2. The experiments are performed on a Windows 10 computer with AMD 3.4 GHz CPU and 64 GB RAM.

5.2. An illustrative example

In this section, an example is given to show the difference in generated routes between the proposed algorithm and the deterministic routing algorithm. In the example, there are 20 pickup nodes, with the variation coefficient $c_v = 0.15$, and the width of the time windows is set to 70. The arc travel times follow speed profile number 4 in Table 1.

For the deterministic routing algorithm, the travel times are deterministic and computed based on speed profile number 4. Service times are also deterministic, and service time $s_i = E[S_i]$. Service levels are not considered in the deterministic algorithm, and the goal is to minimize the total cost. The proposed algorithm's desired global service level is set to 70%.

The routing solutions generated by the above two algorithms are shown in Figure 2. Three vehicles are needed in the solution generated by the deterministic routing algorithm, while four vehicles are needed in the solution generated by the proposed routing algorithm. Note that routes #1 in the two generated solutions are identical and are therefore omitted in Figure 2. Route #2 (blue) and route #3 (red) in the deterministic algorithm are split into three different routes (blue, red, purple) in the proposed algorithm. In the proposed algorithm, one more vehicle is needed to guarantee the desired service rate, leading to a higher total cost. However, compared to the deterministic routing algorithm, the reliability of the routing solution improves significantly, and the simulated global success probability is greater than the desired service level. Some "compact" routes in the deterministic routing algorithm are split into some more "reliable" routes in the proposed algorithm.

5.3. Performance of heuristic pricing

As shown in section 4.3.4, when the travel times are not time-dependent, the exact pricing method is identical to the heuristic pricing method, and it is efficient to obtain the exact solution. However, the faster the means of the travel times change throughout the day, the slower the exact pricing algorithm gets. In this section, we evaluate the time used to solve the problem to optimality and the quality of the solutions provided by the heuristic pricing algorithm. The experiment is conducted on smaller instances with 20 pickup nodes, with $c_v = 0.3$ and the width of time windows set to 80.

Table 2 compares the performance of the exact pricing algorithm and the heuristic pricing algorithm. It is demonstrated that when the travel time is time-independent, the time used by the exact and heuristic pricing algorithm is almost the same. The time used by the exact pricing algorithm grows significantly with f'_{max} and $|f'_{min}|$, which is consistent with the conjecture. In the worst cases, the run time of the exact pricing algorithm is more than 100 times the run

Table 2: Comparison of exact pricing with heuristic pricing

speed profile	f_max	f_min	desired global service level	exact pricing		heuristic pricing	
				cost	time(ms)	cost	time(ms)
1	0	0	0.6	30783.95	71178	30783.95	70129
			0.7	30790.93	116086	30790.93	113668
			0.8	40772.44	72428	40772.44	71255
			0.9	40773.78	1397	40773.78	1439
2	0.22	-0.18	0.6	30780.71	304674	30780.71	119490
			0.7	30790.93	128771	30790.93	117126
			0.8	40766.95	5970	40766.95	3399
			0.9	40772.44	1866	40772.44	1160
3	0.5	-0.375	0.6	40759.56	26141	40759.56	2807
			0.7	40759.56	24417	40759.56	2602
			0.8	40759.56	14792	40759.56	1802
			0.9	40766.95	15898	40766.95	1487
4	0.511	-0.496	0.6	40753.20	231522	40753.20	2515
			0.7	40757.64	414671	40757.64	4690
			0.8	40759.56	548095	40759.56	5986
			0.9	40766.95	320102	40766.95	10297
Average				38269.69	143625.5	38269.69	29740.75

Table 3: Base instance

number of total nodes	c_v	time window width	desired global service level	number of vehicles	cost	estimated global success probability	simulated global success probability	time(ms)
60	0.3	80	0.6	5	51124.92	63.99%	78.3%	46997
			0.7	5	51134.51	70.36%	79.6%	38488
			0.8	5	51158.50	80.47%	83.2%	117882
			0.9	5	51174.49	90.12%	98.7%	79416

time of the heuristic pricing algorithm. It can be seen that the heuristic pricing algorithm generates the same solutions as the exact pricing method. We conduct extensive tests on small instances up to 25, as the exact algorithm cannot solve some larger instances. It is concluded that the heuristic pricing algorithm can generate high-quality solutions and even optimal solutions in small instances.

5.4. Model performance

We next test the sensitivity of the algorithm to the following parameters: the number of nodes, the width of time windows, and the variation coefficient. To avoid the confusion caused by too many sets of different parameters, we generate a base instance as a baseline and change one parameter based on the baseline at a time. In each instance, we set the desired global service rate to 60%, 70%, 80%, and 90% and observe the number of vehicles used, total cost, estimated global success probability, simulated global success probability, and running time used. Heuristic pricing is used in all experiments in this section, and speed profile number 4 is adopted.

For the base instance, the following parameters are chosen: number of pickup nodes = 30 , $c_v = 0.3$, and window width = 80. From the results shown in Table 3, we see that when a higher desired global service level is required, the solution ends up with a higher total cost, which is intuitive. The estimated global success probability is always greater than the desired service level, which conforms to the probability constraint in the problem formulation. The simulated global success probability is greater than the desired global service level, which shows the algorithm's effectiveness.

Table 4 shows the model performance when the number of pickup nodes varies from 20 to 45. The number of needed vehicles and the total cost grows with the number of pickup nodes. Ten vehicles are needed in the last instance in Table 4 because the problem is very close to an infeasible problem regarding the probability constraints. Therefore, more vehicles are needed to guarantee global success probability. Table 5 shows the model performance when the variance coefficient of travel times varies from 0.1 to 0.4. The number of needed vehicles and the cost grows with the variance of the travel times. An instance is not solved when it is impossible to satisfy the desired global service level, meaning that the estimated global success probability is still less than the desired global service level even if each route visits only a single pair of requests. The last two instances in the table show that when the variation coefficient gets large, it is impossible to maintain a high global service level. Table 6 shows the model performance when the window width varies from 60 to 90. When all other conditions remain the same, the tighter the time windows are, the less possible it is to satisfy a high desired global service level. From all the above experiments, we observe that there is no apparent relationship between the model parameters and the estimation error of global success probability, and the performance of the algorithm is not sensitive to the number of pickup nodes, the variance coefficient of travel times, and the widths of time windows.

We compare the above results generated by the proposed algorithm with routing solutions generated by the deterministic routing algorithm introduced in Section 5.2 in Table 7. The desired global service level of the proposed algorithm is set to 0.6 in different experiment settings in Table 7. As the deterministic routing algorithm's optimization goal is to minimize the total cost regardless of the service reliability, the simulated global success probabilities are below 10% in most cases. The proposed algorithm improves the reliability of the pickup and delivery service significantly and guarantees the global success probabilities are above 60%. Taking the stochastic information about traffic into account is vital for ensuring the reliability of the service. However, the improvement in reliability comes at the cost of more needed vehicles and increased operational costs. With the proposed algorithm, 1 to 2 more vehicles are needed compared to the deterministic algorithm in Table 7.

We conclude by mentioning a limitation of the proposed algorithm. It is observed that there is a gap between the estimated global success probability and the simulated global success probability. The gap exists because there is no closed form of the actual probability distribution of arrival times, and the estimation method has an estimation error. The average gap between the estimated global success probability and the simulated global success probability of the instances listed in this section is 9.9%. As the experiments show, in the vast majority of the cases, the estimation method gives a lower bound of the estimated probability. Therefore, the gap makes the simulated success probability even higher than the desired service level. The proposed algorithm is thus more reliable and more robust to fluctuations than required. In a real-life setting, the decision maker can balance the actual success probability and operational cost by adjusting the desired service level slightly.

5.5. The importance of considering time-dependent travel times

In this section, we analyze the importance of considering time-dependent travel times. We assume the real travel speed follows speed profile number 4. The proposed algorithm can model the changes in travel times during the day by incorporating speed profile number 4. We compare the proposed algorithm with an algorithm considering stochastic but time-independent travel times. The time-independent algorithm assumes the travel speed does not change during the day and has the value of the average travel speed of speed profile number 4, which is 1.17 in time intervals (0, 840).

In Table 8, the differences in cost and simulated global success probability between the time-dependent and time-independent solutions are evaluated. The simulation sampled travel times between nodes based on speed profile number 4. As Table 8 shows, not considering time-dependent travel time may lead to a lower cost but also result in a much lower success probability. Sometimes, there is little chance that some time windows can be satisfied. We conclude that modeling the change in travel time during the day is instrumental in developing routes for improving delivery reliability.

5.6. The impact of node and global service levels

In the previous experiments, for the sake of simplicity, it is assumed that the desired node service levels are fixed at 0.7. In this section, we assume that the desired node service levels are not the same, as some of the nodes require more reliable service. These nodes are referred to as special nodes, whose desired node service levels are set to 0.9. The special nodes are chosen randomly from the node-set. We adjust the proportion of the special nodes in the data set and the desired global service level and observe the operational costs under different combinations of node and global service levels. In the experiments, the number of total nodes is 60, with the time window width being 80 and $c_v = 0.3$.

Table 9 shows the operational costs under different node and global service level settings. The operational cost increases monotonically with the desired global service level and the proportion of special nodes. It is intuitive, as

Table 4: Model performance on instances with different numbers of nodes

number of total nodes	c_v	time window width	desired global service level	number of vehicles	cost	estimated global success probability	simulated global success probability	time(ms)
40	0.3	80	0.6	4	40753.20	61.12%	77.7%	2515
			0.7	4	40753.42	73.96%	80.8%	4690
			0.8	4	40759.56	85.9%	93.6%	5986
			0.9	4	40766.95	91.26%	98.0%	10297
50			0.6	4	40980.69	69.07%	79.6%	22012
			0.7	4	40984.27	70.47%	78.0%	21440
			0.8	4	41011.97	80.43%	80.6%	48496
			0.9	5	50997.64	93.15%	99.5%	32665
70			0.6	5	51298.32	63.34%	78.2%	919768
			0.7	5	51299.06	74.28%	91.2%	474662
			0.8	5	51323.53	85.60%	90.7%	1167582
			0.9	6	61325.10	90.03%	95.2%	167783
80			0.6	6	61484.20	62.86%	68.3%	550447
			0.7	6	61490.87	73.57%	89.7%	132978
			0.8	6	61499.46	84.41%	95.1%	357319
			0.9	6	61546.67	90.73%	94.8%	323820
90			0.6	6	61689.53	62.18%	79.9%	597232
			0.7	6	61691.90	74.24%	80.7%	1355956
			0.8	6	61705.41	81.21%	87.3%	1032443
			0.9	10	101698.51	90.06%	94.7%	1509237

Table 5: Model performance on instances with different variation coefficient

number of total nodes	c_v	time window width	desired global service level	number of vehicles	cost	estimated global success probability	simulated global success probability	time(ms)
60	0.1	80	0.6	4	41122.46	79.11%	98.0%	96064
			0.7	4	41122.46	79.11%	97.8%	120711
			0.8	4	41128.76	93.15%	98.5%	55734
			0.9	4	41128.76	93.15%	98.5%	74231
	0.15		0.6	4	41128.76	73.04%	89.7%	65609
			0.7	4	41128.76	73.04%	88.2%	77473
			0.8	4	41132.32	86.45%	91.0%	36828
			0.9	5	51113.35	98.47%	98.1%	71536
	0.2		0.6	4	41132.32	61.07%	76.6%	31061
			0.7	5	51113.35	87.64%	89.7%	27885
			0.8	5	51113.35	87.64%	87.7%	13200
			0.9	5	51121.74	92.60%	93.1%	15215
	0.4		0.6	5	51174.49	61.00%	87.2%	1046310
			0.7	6	61170.89	71.02%	97.7%	6028135
			0.8	not solved				
			0.9	not solved				

Table 6: Model performance on instances with different time window width

number of total nodes	c_v	time window width	desired global service level	number of vehicles	cost	estimated global success probability	simulated global success probability	time(ms)
60	0.3	60	0.6	not solved				
			0.7	not solved				
			0.8	not solved				
			0.9	not solved				
		70	0.6	5	51189.82	61.06%	89.50%	440004
			0.7	not solved				
			0.8	not solved				
			0.9	not solved				
		90	0.6	4	41148.06	61.21%	84.8%	867478
			0.7	4	41165.97	70.98%	83.9%	500526
			0.8	5	51113.35	83.97%	88.7%	37562
			0.9	5	51124.92	91.81%	91.6%	69572

Table 7: Comparison of the proposed algorithm and deterministic routing algorithm

number of total nodes	c_v	time window width	deterministic routing algorithm			proposed algorithm		
			number of vehicles	cost	simulated global success probability	number of vehicles	cost	simulated global success probability
40	0.3	80	3	30751.72	46.5%	4	40753.20	77.7%
50	0.3	80	3	30992.84	9.2%	4	40980.69	79.4%
60	0.3	80	3	31141.39	6.5%	5	51124.92	78.3%
70	0.3	80	4	41273.51	1.5%	5	51298.32	78.2%
80	0.3	80	4	41493.92	3.7%	6	61484.20	68.3%
90	0.3	80	4	41678.40	0.4%	6	61689.53	79.9%
60	0.1	80	3	31141.39	19.1%	4	41122.46	98.0%
60	0.15	80	3	31141.39	13.2%	4	41128.76	89.7%
60	0.2	80	3	31141.39	8.4%	4	41132.32	76.6%
60	0.4	80	3	31141.39	2.5%	5	51174.49	87.2%
60	0.3	70	4	41120.98	9.0%	5	51189.82	89.5%
60	0.3	90	3	31108.07	6.9%	4	41148.06	84.8%

more reliable service leads to more operational costs. The cost grows almost linearly with the enhancement of the proportion of special nodes and achieves a plateau when the proportion is higher than a threshold.

Overall, the number of vehicles needed and the operational cost increase with desired node and global service levels and decrease with the length of the time windows. Therefore, to achieve reliable and punctual service with a lower cost, it is practical to give the customers wider time windows if possible.

6. Conclusions

The paper studies the pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. We focus on satisfying customers' expectations of the punctuality of the delivery, measured by node success probabilities and global success probability. We present an estimation method for arrival times and success probabilities under stochastic travel and service times. In most cases, the estimated success probabilities provide a lower bound for the actual success probabilities. A solution approach based on a branch-price-and-cut framework

Table 8: Comparison of time-dependent solution and the time-independent solution

number of total nodes	c_v	time window width	desired global service level	proposed algorithm cost	proposed algorithm simulated global success probability	time independent algorithm cost	time independent algorithm simulated global success probability
30	0.1	70	0.8	20561.75	87.2%	20586.85	85.3%
40				40753.20	86.1%	30758.08	0.0%
50				40977.37	93.9%	31129.29	0.6%
60				51121.74	93.2%	41133.21	0.2%
30	0.15			20590.70	95.8%	20586.85	85.9%
40				40759.56	98.5%	30766.05	0.2%
50				40978.28	85.8%	40984.92	3.3%
60				51126.76	89.4%	41139.14	57.3%
30	0.2			30551.70	98.4%	30550.87	34.3%
40				40759.56	93.6%	30766.05	0.7%
50				40989.72	95.0%	40989.72	96.5%
60				51151.28	97.9%	41148.76	44.9%
Average				39193.47	92.9%	33379.15	34.1%

Table 9: Costs under different node and global service levels settings

cost / global service level / proportion of special nodes	0.5	0.6	0.7	0.8	0.9
0	51116.53	51124.92	51134.51	51158.50	51174.49
0.2	51131.43	51131.43	51140.44	51162.18	61147.06
0.4	51148.65	51148.65	51148.65	51162.18	61147.06
0.6	51151.02	51151.02	51151.02	51162.18	61147.06
0.8	51151.02	51151.02	51151.02	51162.18	61147.06

is proposed, including exact and heuristic labeling algorithms that integrate the probability distributions' information. Computational experiments on smaller instances show that the heuristic pricing algorithm can provide high-quality solutions in less time. The results of computational experiments indicate that the heuristic pricing algorithm can solve instances with up to 50 pickup nodes. Monte Carlo simulations show that the proposed method can ensure the desired service levels. The comparison between the proposed method and the deterministic routing algorithm that assumes travel times are constant further illustrates the advantage of the proposed method.

A possible direction for future research is to integrate Monte Carlo simulations into the route construction phase, namely the pricing algorithm, so that the estimation error can be reduced. Research can also consider other approaches to solve this problem, such as a robust optimization approach. Furthermore, currently, we assume that the travel time variance is proportional to the travel distance, and there is only one path between two nodes, which is restrictive. These assumptions can be relaxed in future research.

Acknowledgements

Funding: This work was supported by the National Science Foundation Awards CPS #1545130 and CNS-1932615.

References

Adulyasak, Y., Jaillet, P., 2016. Models and algorithms for stochastic and robust vehicle routing with deadlines. Transportation Science 50, 608–626.

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L.M., Poss, M., Requejo, C., 2013. The robust vehicle routing problem with time windows. Computers & operations research 40, 856–866.

Ando, N., Taniguchi, E., 2006. Travel time reliability in vehicle routing and scheduling with time windows. Networks and Spatial Economics 6, 293–311.

Aziez, I., Côté, J.F., Coelho, L.C., 2020. Exact algorithms for the multi-pickup and delivery problem with time windows. European Journal of Operational Research 284, 906–919.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46, 316–329.

Bertsimas, D., Brown, D.B., Caramanis, C., 2011. Theory and applications of robust optimization. SIAM review 53, 464–501.

Braaten, S., Gjønnes, O., Hvattum, L.M., Tirado, G., 2017. Heuristics for the robust vehicle routing problem with time windows. Expert Systems with Applications 77, 136–147.

Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science 53, 946–985.

Dabia, S., Ropke, S., Van Woensel, T., De Kok, T., 2013. Branch and price for the time-dependent vehicle routing problem with time windows. Transportation Science 47, 380–396.

Desaulniers, G., Desrosiers, J., Solomon, M.M., 2006. Column generation. volume 5. Springer Science & Business Media.

Ehmke, J.F., Campbell, A.M., Urban, T.L., 2015. Ensuring service levels in routing problems with time windows and stochastic travel times. European Journal of Operational Research 240, 539–550.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., Rousseau, L.M., 2018. The vehicle routing problem with hard time windows and stochastic service times. EURO Journal on Transportation and Logistics 7, 223–251.

Gendreau, M., Ghiani, G., Guerriero, E., 2015. Time-dependent routing problems: A review. Computers & Operations Research 64, 189–197.

Gutierrez, A., Dieulle, L., Labadie, N., Velasco, N., 2018. A multi-population algorithm to solve the VRP with stochastic service and travel times. Computers & Industrial Engineering 125, 144–156.

Ichoua, S., Gendreau, M., Potvin, J.Y., 2003. Vehicle dispatching with time-dependent travel times. European Journal of Operational Research 144, 379–396.

Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints, in: Column generation. Springer, pp. 33–65.

Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D., 2008. Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research 56, 497–511.

Jula, H., Dessouky, M., Ioannou, P.A., 2006. Truck route planning in nonstationary stochastic networks with time windows at customer locations. IEEE Transactions on Intelligent Transportation Systems 7, 51–62.

Kenyon, A.S., Morton, D.P., 2003. Stochastic vehicle routing with random travel times. Transportation Science 37, 69–82.

Laporte, G., Louveaux, F., Mercure, H., 1992. The vehicle routing problem with stochastic travel times. Transportation Science 26, 161–170.

Lecluyse, C., Van Woensel, T., Peremans, H., 2009. Vehicle routing with stochastic time-dependent travel times. 4OR 7, 363–377.

Li, X., Tian, P., Leung, S.C., 2010. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. International Journal of Production Economics 125, 137–145.

Miranda, D.M., Conceição, S.V., 2016. The vehicle routing problem with hard time windows and stochastic travel and service time. Expert Systems with Applications 64, 104–116.

Munari, P., Moreno, A., De La Vega, J., Alem, D., Gondzio, J., Morabito, R., 2019. The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method. Transportation Science 53, 1043–1066.

Nguyen, D.H., De Leeuw, S., Dullaert, W., Foubert, B.P., 2019. What is the right delivery option for you? consumer preferences for delivery attributes in online retailing. Journal of Business Logistics 40, 299–321.

Oyola, J., Arntzen, H., Woodruff, D.L., 2017. The stochastic vehicle routing problem, a literature review, part ii: solution methods. EURO Journal on Transportation and Logistics 6, 349–388.

Oyola, J., Arntzen, H., Woodruff, D.L., 2018. The stochastic vehicle routing problem, a literature review, part i: models. EURO Journal on Transportation and Logistics 7, 193–221.

Pecin, D., Pessoa, A., Poggi, M., Uchoa, E., 2017. Improved branch-cut-and-price for capacitated vehicle routing. Mathematical Programming Computation 9, 61–100.

Righini, G., Salani, M., 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks: An International Journal 51, 155–170.

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

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

Sun, P., Veelenturf, L.P., Dabia, S., Van Woensel, T., 2018a. The time-dependent capacitated profitable tour problem with time windows and precedence constraints. European Journal of Operational Research 264, 1058–1073.

Sun, P., Veelenturf, L.P., Hewitt, M., Van Woensel, T., 2018b. The time-dependent pickup and delivery problem with time windows. Transportation Research Part B: Methodological 116, 1–24.

Taş, D., Dellaert, N., Van Woensel, T., De Kok, T., 2013. Vehicle routing problem with stochastic travel times including soft time windows and service costs. Computers & Operations Research 40, 214–224.

Taş, D., Dellaert, N., Van Woensel, T., De Kok, T., 2014a. The time-dependent vehicle routing problem with soft time windows and stochastic travel times. Transportation Research Part C: Emerging Technologies 48, 66–83.

Taş, D., Gendreau, M., Dellaert, N., Van Woensel, T., De Kok, A., 2014b. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. European Journal of Operational Research 236, 789–799.

Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N., 2008. Vehicle routing with dynamic travel times: A queueing approach. European Journal of Operational Research 186, 990–1007.

Wang, X., Dessouky, M., Ordonez, F., 2016. A pickup and delivery problem for ridesharing considering congestion. Transportation letters 8,

Table 10: Percentage estimation errors of the two estimation methods

c_v	proposed algorithm compared with simulation		Ehmke's estimation method compared with simulation	
	mean	variance	mean	variance
0.1	-0.6%	+4.6%	+0.2%	+6.1%
0.15	-0.9%	+12.6%	+0.5%	+10.7%
0.2	-1.6%	+18.1%	+0.8%	+14.2%
0.3	-2.9%	+23.7%	+1.4%	+23.5%
0.4	-4.4%	+24.9%	+1.5%	+35.1%

259–269.

Yazici, M.A., Kamga, C., Mouskos, K.C., 2012. Analysis of travel time reliability in new york city based on day-of-week and time-of-day periods. Transportation research record 2308, 83–95.

A. Comparison of the proposed and Ehmke's estimation methods

In this section, we compare the proposed arrival time estimation method based on Jula et al. (2006) with the arrival time estimation method proposed by Ehmke et al. (2015).

The proposed method and Ehmke's method both first estimate the means and the variances of node arrival times, and then calculate the node success probabilities according to Expression (7). To compare the estimation accuraries of the two methods, we compare the estimated means and variances of arrival times with simulation result. In the simulation, travel times are sampled from normal distributions where c_v is the variance coefficient. The simulated mean and variance of arrival time at a node are calculated after 1000 runs. We study the arrival times of 458 nodes along a set of 54 routes derived from routing solutions in Section 5.4. For each node, the percentage errors of estimated mean and variation for each estimation method are calculated. Table 10 shows the average percentage estimation errors of the two methods over the 458 nodes.

The proposed method underestimates the mean while Ehmke's method overestimates the mean. The two methods are both effective on estimating the means of arrival times, since the estimation errors are less than 5% of arc travel times on average. Both of the two methods overestimate the variance of arrival times, and the estimation errors grow with the variance of travel times. Overall, the two estimation methods have similar performance on estimating node arrival times.

In Table 11, we applied Ehmke's method to the proposed branch-cut-and-price solution framework using the same heuristic pricing rules and compared the generated solutions with the proposed method. The result shows that the two estimation methods have the same performance in most of the cases. In the last three cases, the proposed method outperforms Ehmke's methods in terms of cost. A possible reason for this is that Ehmke's method overestimates the needed travel times.

Table 11: Quality of routing solutions generated by the two estimation methods

number of total nodes	c_v	time window width	desired global service level	proposed algorithm			Ehmke's estimation method		
				number of vehicles	cost	simulated global success probability	number of vehicles	cost	simulated global success probability
60	0.15	80	0.6	4	41128.76	89.7%	4	41128.76	88.5%
60	0.15	80	0.7	4	41128.76	88.2%	4	41128.76	89.1%
60	0.15	80	0.8	4	41132.32	91.0%	4	41132.32	91.6%
60	0.15	80	0.9	5	51113.35	98.1%	5	51113.35	97.5%
60	0.2	80	0.6	4	41132.32	76.6%	4	41132.32	78.0%
60	0.2	80	0.7	5	51113.35	89.7%	5	51113.35	88.1%
60	0.2	80	0.8	5	51113.35	87.7%	5	51113.35	88.4%
60	0.2	80	0.9	5	51121.74	93.1%	5	51121.74	92.3%
60	0.3	80	0.6	5	51124.92	78.3%	5	51124.92	79.4%
60	0.3	80	0.7	5	51134.51	79.6%	6	61128.32	85.8%
60	0.3	80	0.8	5	51158.50	83.2%	6	61129.49	87.1%
60	0.3	80	0.9	5	51174.49	98.7%	6	61147.06	98.6%