A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times

F. Errico [*], G. Desaulniers [†], M. Gendreau [‡], W. Rei [§], L.-M. Rousseau [¶]

Abstract

The vehicle routing problem with hard time windows and stochastic service times (VRPTW-ST) introduced by Errico et al. (2013) in the form of a chance-constrained model, mainly differs from other vehicle routing problems with stochastic service or travel times considered in literature by the presence of *hard* time windows. This makes the problem extremely challenging. In this paper, we model the VRPTW-ST as a two-stage stochastic program and define two recourse policies to recover operations feasibility when the first stage plan turns out to be infeasible. We formulate the VRPTW-ST as a set partitioning problem and solve it by exact branch-cut-and-price algorithms. Specifically, we developed efficient labeling algorithms by suitably choosing label components, determining extension functions, and developing lower and upper bounds on partial route reduced cost to be used in the column generation step. Results on benchmark data show that our methods are able to solve instances with up to 50 customers for both recourse policies.

Keywords: Vehicle routing problem, service time, stochastic programming, a-priori optimization, column generation

1 Introduction

In this paper we consider the vehicle routing problem with hard time windows and stochastic service times (VRPTW-ST) that was introduced in Errico et al. (2013). A hard time window allows to arrive earlier than its lower bound and wait, but forbids to arrive later than its upper bound. The VRPTW-ST appears in various practical applications where the drivers (e.g., technicians or repairmen) of the vehicles perform specific services at the visited customers. The central issue of

[*]CIRRELT Interuniversity Research Center on Enterprise Networks, Logistics and Transportation, and Département de génie de la construction, École de Technologie Supérieure de Montréal. fausto.errico@cirrelt.ca

[†]GERAD Group for Research in Decision Analysis, and Département de mathématiques et génie industriel, École Polytechnique de Montréal.

[‡]CIRRELT Interuniversity Research Center on Enterprise Networks, Logistics and Transportation, and Département de mathématiques et génie industriel, École Polytechnique de Montréal.

[§]CIRRELT Interuniversity Research Center on Enterprise Networks, Logistics and Transportation, and Département de management et de technologie, Université du Québec à Montréal.

[¶]CIRRELT Interuniversity Research Center on Enterprise Networks, Logistics and Transportation, and Département de mathématiques et génie industriel, École Polytechnique de Montréal.

the problem is to determine vehicle routes that respect the customer's time windows when the details of the service to be performed at each customer are unknown beforehand, yielding stochastic service times. Originally, Errico et al. (2013) proposed to formulate the VRPTW-ST as a chance-constrained optimization model that includes a probabilistic constraint. Specifically, this constraint ensures that the probability that the planned routes cannot meet the time windows when the service times are observed does not exceed a prefixed threshold. Although this first model enables a manager to analyze the impact of stochastic service times on the vehicle routes, it does not take into account what is actually done in the case where the planned routes turn out to be infeasible during the operations. In many cases, corrective actions entailing extra costs are applied to retrieve route feasibility. Considering how costly these corrective actions can be, it then becomes an important aspect of the problem considered.

In this paper, we propose to study the VRPTW-ST as an a priori optimization problem that involves two stages. In the first stage when service times are unknown, an *a priori plan* composed of a set of planned routes (also called the *first-stage decisions*), is computed. In the second stage, all data are revealed and the a priori plan is modified according to a given *recourse* policy. In this setting, the VRPTW-ST is modeled as a two-stage stochastic program in which the objective is to minimize the total expected cost that includes expected travel costs and costs incurred by the second stage recourse actions.

A variety of stochastic vehicle routing problems have been addressed previously. The most common sources of uncertainty encountered are demand volumes (see e.g. Bertsimas 1992; Laporte et al. 2002), customers presence (see e.g. Gendreau et al. 1995), and stochastic travel and service times. The papers dealing with stochastic travel and service times have exploited different modeling frameworks such as dynamic programming, chance-constrained programming and robust optimization. Laporte et al. (1992) address a non-capacitated vehicle routing problem (VRP) with stochastic travel and service times and a soft constraint on route duration. The authors present one chance-constrained model and two models with recourse, where the recourse action consists in the payment of a penalty. Lambert et al. (1993) present a variant of the same problem where customer time windows must be met in all scenarios. Kenyon and Morton (2003) study a problem setting similar to that of Laporte et al. (1992) but with different objective functions: One model minimizes the expected completion time, the other the probability to violate a given completion deadline. In Wang and Regan (2001), a non-capacitated VRP with soft time windows and stochastic travel times is considered where the service is skipped (but not the visit itself) when a customer is visited after its deadline. To evaluate the expected load of a route, the authors propose an algorithm of exponential complexity with respect to the number of customers. A heuristic approach combining robust optimization and stochastic programming for a particular version of the non-capacitated VRP with soft time windows, a route duration constraint, and uncertainty in the customer presence and service times is proposed in Sungur et al. (2010). Heuristic approaches for versions of the capacitated VRP with soft time windows, a soft duration limit, and stochastic travel and service times are proposed in Li et al. (2010) and Lei et al. (2012). The recourse action considered in these papers consists in the simple payment of a penalty. A similar setting is adopted in Taş et al. (2013) where a three-phase tabu search algorithm is developed to minimize a combination of expected earliness and lateness at customers, and generalized operational costs. A robust optimization approach for the VRP with travel and demand uncertainty, and customer deadlines is presented in Lee et al. (2012). Jula et al. (2006) and Chang et al. (2009) develop heuristics for the traveling salesman problem (TSP) with hard time windows and stochastic travel and service times. Other authors considered sources of uncertainty different from service or travel times in combination with hard time window constraints. Erera et al. (2009) propose a two-stage approach for the VRP with stochastic demands, a capacity constraint, and hard time windows. Campbell and Thomas (2008) investigate the TSP with customer deadlines and uncertainty in the presence of customers.

In the context of the VRP, exact solution methods addressing a combination of service time uncertainty and hard time windows have been considered for the first time in Errico et al. (2013). With respect to the latter, the present work states the VRPTW-ST as a two-stage stochastic program, including also the definition of two new recourse strategies. In fact, recourses consisting only in the payment of a penalty, as common in the vehicle routing literature with stochastic times, are not applicable in our case because time windows are hard and no late service is allowed. Adopting a two-stage stochastic model gives rise to a very challenging problem requiring the development of new solution methods.

This paper makes several contributions toward designing exact algorithms to solve recourse-based models for the VRPTW-ST. The two problem variants studied (one for each recourse policy) are formulated as set partitioning models and the procedure of Errico et al. (2013) is adopted to compute the probability that a route is feasible. To solve the proposed models, we implement state-of-the-art branch-price-and-cut algorithms. The major methodological challenges are encountered in the solution of the column generation subproblems. We devise label-setting algorithms in which labels have components to properly account for the reduced costs that depend on the service time probability distributions. Furthermore, we show how to compute the expected cost of a complete route and of a partial route, and derive from these computations label extension functions. We also propose several upper and lower bounds that are used to define efficient dominance rules. These techniques are all dependent on the recourse policy used and ad hoc developments are necessary. An extensive computational study is performed using the instance set proposed in Errico et al. (2013). The results show that our methods are able to efficiently solve instances with up to 50 customers for both recourse policies.

The reminder of the paper is organized as follows. In Section 2, we formally describe the problem, including the two recourse policies, and formulate its two variants as set partitioning models. In addition, we show how to compute the expected cost of a route. In Section 3, we detail our branch-price-and-cut algorithms. The results of our computational study are reported in Section 4. Finally, we conclude in Section 5.

2 The VRPTW-ST

We first provide a general statement of the VRPTW-ST in Section 2.1 and introduce the recourse policies in Section 2.2. Then we present a generic set partitioning model for the VRPTW-ST in Section 2.3. This model is adapted afterwards for each policy by specifying the feasible route sets

(Section 2.4), and how to compute the probability that a route is operationally feasible (Section 2.5) and the route expected costs (Section 2.6).

2.1 Problem statement

Consider a directed graph $G = (V, A)$, where $V = \{0, 1, \ldots, n\}$ is the node set and $A = \{(i, j) \mid i, j \in V\}$ the arc set. Node 0 represents a depot where a fleet of homogeneous vehicles is initially located, and $V_c = \{1, \ldots, n\}$ is the customer set. A time window $[a_i, b_i]$ and a stochastic service time are associated with each customer $i \in V_c$. In particular, service times are uncertain at the planning stage (i.e., the first stage of the problem), but their probability distributions are assumed to be known and mutually independent. The actual value of a service time is only observed once a vehicle arrives at the associated customer's location. We assume that a fixed time t^{eval} (independent of the customer) is needed to evaluate this service time. This evaluation must start within the customer's time window and its time is not part of the service time. Without loss of generality, we associate with node 0 an unconstraining time window $[a_0, b_0]$ and a constant service time s_0 equal to 0 (seen as a stochastic service time with a single possible value). Denote by t_i^{eval} the time required to evaluate the service time at node $i \in V$, where $t_i^{eval} = t^{eval}$ if $i \in V_c$ and $t_0^{eval} = 0$. A non-negative travel cost c_{ij} and a travel time t_{ij} are associated with each arc $(i, j) \in A$. Furthermore, a "no-service" penalty π is given.

An a priori plan for the VRPTW-ST is a set of a priori vehicle routes such that each route starts and ends at node 0 and all the customers are assigned to exactly one route. An a priori route is said *operationally feasible* (op-feasible, for short) if, at every customer visited along this route, the evaluation of the service time starts within the customer's time window and service is performed. However, vehicles can arrive at customers before the beginning of their time window. In this case the beginning of the service time evaluation must be postponed until the time window opens. Vehicles are not allowed to arrive after the end of a time window. It should be observed that the revealed service times directly determine if a route is op-feasible. Whenever the actual service times induce a route to be op-infeasible, recourse actions have to be taken to regain feasibility. For the models proposed in this paper, we define recourse actions that are based on skipping given customers along the route. The choice of which customer to skip is determined by the particular recourse policy. Two such policies are used to define the proposed two-stage stochastic models. In both cases, each skipped customer entails a penalty of π that may represent the dissatisfaction of a customer whose service is postponed to another day or the cost of an emergency service. Therefore, the VRPTW-ST consists of finding a set of a priori vehicle routes such that the total expected cost, obtained as the sum of the expected travel distance and penalty costs, is minimized.

The little attention that the VRPTW-ST has received in the literature is perhaps due to its complexity. As shown in Errico et al. (2013), the computation of the probability that a route is op-feasible requires the knowledge of the vehicle arrival time probability distribution at the customer locations. However, the presence of hard time windows generally implies the truncation of these distributions, thus preventing the straightforward application of convolution properties when summing the random variables. Furthermore, the present paper accounts for new recourse strategies and this adds levels of complexity given that it becomes extremely difficult and computationally expensive to provide good bounds on the cost of a solution. For this reason, instead of addressing the problem in its full generality we impose two practically reasonable conditions:

C.1: The probability that each route is op-feasible must be greater than or equal to a given "reliability" threshold $\alpha \in (0, 1)$.

C.2: No route requires more than one recourse action.

Both conditions have a practical justification as they avoid routes being op-infeasible systematically. Notice that conditions C.1 and C.2 have the same spirit as the restrictions on the expected vehicle load imposed in the majority of the works addressing capacitated VRP with stochastic demand (see Laporte et al. 2002, for example).

2.2 Recourse policies

We consider two alternative recourse policies. In both cases, vehicles first follow their a priori routes. When a route becomes op-infeasible, one of the following recourse actions is applied:

- **Skip-Current Recourse (C)**. Once at a customer the actual service time is revealed after its evaluation. If the revealed service time induces infeasibility with the next customer's time window, the service at the current customer is skipped and the no-service penalty π is paid.
- **Skip-Next Recourse (N)**. The service at the current customer is always performed. If this implies infeasibility with the next customer's time window, the visit to the next customer is skipped and the no-service penalty π is charged.

In different contexts, other researchers considered similar recourse policies. Wang and Regan (2001) address a variant of the VRP with stochastic travel times where the recourse policy skipped the current customer when the vehicle is late at the current customer. Such a policy can be seen as a variant of the recourse C, where the skipping action is performed if the service at the current customer causes op-infeasibility at the next customer. We chose the latter because it seems more suitable when considering stochastic service times (in this case, a vehicle has no practical reasons to arrive late at a customer). Campbell and Thomas (2008) considered recourse N when addressing the TSP with customer deadlines combined with uncertainty about customer's presence.

It should be noticed that it is generally possible to consider more complex recourse strategies than the proposed ones, which could potentially provide more efficient route plans and operational policies. Tractability is, however, a major obstacle to their development. As it will be clearly shown in the remainder of this paper, even the relatively simple recourses C and N turn out to be very challenging from the computational point of view.

2.3 Formulation

Consider a route r defined as a sequence of nodes $r = (v_0, v_1, \ldots, v_q, v_{q+1})$, where $v_1, \ldots, v_q \in V_c$ and v_0 and v_{q+1} represent the depot 0. As previously mentioned, conditions C.1 and C.2 restrict the set of feasible routes. To account for C.1, let us denote $\mathcal{R}_\alpha$ the set of routes whose probability to be op-feasible is greater than or equal to the reliability threshold α. Let us also denote $\overline{\mathcal{R}}^C$ and $\overline{\mathcal{R}}^N$ the sets of routes satisfying condition C.2 for recourses C and N, respectively. We then define the feasible route set $\mathcal{R}^l = \mathcal{R}_\alpha \bigcap \overline{\mathcal{R}}^l$ for each recourse policy $l \in \{C, N\}$. With each route r is associated an expected cost $\tilde{c}_r^l$, $l \in \{C, N\}$, that is recourse-dependent and accounts for both the normal routing costs $c_r = \sum_{i=0}^{q} c_{v_i,v_{i+1}}$ and the costs incurred by a recourse action if service times imply op-infesibility. The expected recourse costs is a function of the no-service penalty and the expected travel cost variation. Furthermore, we associate binary parameters a_{ir}, $i \in V_c$, that take value 1 if route r visits customer i and 0 otherwise.

Considering the binary variables x_r, $\forall r \in \mathcal{R}^l$, which take value 1 if route r is chosen in the a priori plan, and 0 otherwise, the VRPTW-ST can be formulated, for each recourse policy $l \in \{C, N\}$, as follows:

$$\min \quad \sum_{r \in \mathcal{R}^l} \tilde{c}_r^l x_r \tag{1}$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}^l} a_{ir} x_r = 1 \qquad \forall i \in V_c \tag{2}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R}^l, \tag{3}$$

where the objective function (1) minimizes the total expected costs, set partitioning constraints (2) ensure that each customer is visited once in the a priori solution, and (3) impose the binary requirements on the decision variables.

2.4 Sets $\overline{\mathcal{R}}^C$ and $\overline{\mathcal{R}}^N$

We now formally define sets $\overline{\mathcal{R}}^C$ and $\overline{\mathcal{R}}^N$. As in Errico et al. (2013), we assume for the rest of this paper that the service time probability distributions are discrete with finite support. Let s_i^{max} be the maximum service time at node $i \in V$, and consider again a route $r = (v_0, v_1, \dots, v_q, v_{q+1})$.

- **Set** $\overline{\mathcal{R}}^C$. For each $h \in \{1, 2, \dots, q-1\}$ and each $i \in \{1, 2, \dots, q\}$, we define

$$\hat{t}_{v_i}^h := \begin{cases} \max\{\hat{t}_{v_{i-1}}^h + t_{v_{i-1}}^{eval} + s_{v_{i-1}}^{max} + t_{v_{i-1},v_i}, a_{v_i}\} & \text{if } i \neq h+1, \\ \max\{\hat{t}_{v_{i-1}}^h + t_{v_{i-1}}^{eval} + t_{v_{i-1},v_i}, a_{v_i}\} & \text{if } i = h+1, \end{cases} \tag{4}$$

  where $\hat{t}_{v_i}^h$ represents the worst case arrival time at node v_i if the service is skipped at customer v_h (without loss of generality, we set $\hat{t}_{v_0}^h = 0$). We say that $r \in \overline{\mathcal{R}}^C$ if and only if $\hat{t}_{v_i}^h \leq b_{v_i}$ for all $h \in \{1, 2, \dots, q-1\}$ and $i \in \{1, 2, \dots, q\}$.

- **Set** $\overline{\mathcal{R}}^N$. For each $h \in \{2, 3, \dots, q\}$ and each $i \in \{1, 2, \dots, q\}$ with $i \neq h$, we define

$$\hat{t}_{v_i}^h := \begin{cases} \max\{\hat{t}_{v_{i-1}}^h + t_{v_{i-1}}^{eval} + s_{v_{i-1}}^{max} + t_{v_{i-1},v_i}, a_{v_i}\} & \text{if } i \neq h+1, \\ \max\{\hat{t}_{v_{i-2}}^h + t_{v_{i-2}}^{eval} + s_{v_{i-2}}^{max} + t_{v_{i-2},v_i}, a_{v_i}\} & \text{if } i = h+1, \end{cases} \tag{5}$$

where we define the values $\hat{t}^h_{v_i}$ as previously. We say that $r \in \overline{\mathcal{R}}^N$ if and only if $\hat{t}^h_{v_i} \leq b_{v_i}$ for all $h \in \{2, 3, \ldots, q\}$ and $i \in \{1, 2, \ldots, q\}$ with $i \neq h$.

2.5 Probability that a route is op-feasible

To properly present our solution methodology, we recall in this subsection the method proposed by Errico et al. (2013) to determine the probability that a route is op-feasible using an updated notation. As previously shown by the authors, if the service time distributions of the customers are assumed to be discrete with finite support, then the probability of a route being op-feasible can be determined recursively.

For a given route $r = (v_0, \ldots, v_q, v_{q+1})$, let $r_i = (v_0, \ldots, v_i)$, $1 \leq i \leq q$, be the subroutes of r starting at the depot and ending at node v_i. Let t_i and s_i be random variables representing respectively the vehicle arrival time, provided that no recourse action was taken, and the service time at node $i \in V$. We also define m^s_i as the probability mass function associated with the service time distribution at node $i \in V$ (i.e., $m^s_i(k) = \Pr\{s_i = k\}, \forall k \in \mathbb{N}$).

Given that the depot does not have time window restrictions, we can write the probability of route r being op-feasible as:

$$\Pr\{r \text{ is op-feasible}\} = \Pr(\{t_{v_q} \leq b_{v_q}\} \cap \{r_{q-1} \text{ is op-feasible}\}).$$

Let $m^t_{v_i}(z)$ be the probability mass function associated with the event $t_{v_i} = z$, provided that r_{i-1} is op-feasible, which is formulated as follows:

$$m^t_{v_i}(z) := \Pr(\{t_{v_i} = z\} \cap \{r_{i-1} \text{ is op-feasible}\}). \tag{6}$$

Given (6), we can then write

$$\Pr\{r \text{ is op-feasible}\} = \sum_{z \leq b_{v_q}} m^t_{v_q}(z). \tag{7}$$

Hence, the computation of the probability that a route is op-feasible requires the computation of the mass probability function of the vehicle arrival time at the last visited node. As developed in Errico et al. (2013), this can be done recursively.

Considering that the vehicle performing route r is never allowed to arrive later than b_{v_i} at node v_i, we define the random variable $\bar{t}_{v_i}$ with values in the interval $[a_{v_i}, b_{v_i}]$ for $1 \leq i \leq q$ as follows:

$$\bar{t}_{v_i} = \begin{cases} a_{v_i} & \text{if } t_{v_i} < a_{v_i} \\ t_{v_i} & \text{if } a_{v_i} \leq t_{v_i} \leq b_{v_i}. \end{cases}$$

This random variable indicates the earliest time at which the evaluation of the service time at node

v_i can start. Its probability mass function $\overline{m}^t_{v_i}(z)$ is given by:

$$\overline{m}^t_{v_i}(z) = \begin{cases} 0 & \text{if } z < a_{v_i} \\ \sum_{l \le a_{v_i}} m^t_{v_i}(l) & \text{if } z = a_{v_i} \\ m^t_{v_i}(z) & \text{if } a_{v_i} < z \le b_{v_i} \\ 0 & \text{if } z > b_{v_i}. \end{cases} \tag{8}$$

With this notation, we obtain the relation $t_{v_i} = \bar{t}_{v_{i-1}} + s_{v_{i-1}} + t^{eval}_{v_{i-1}} + t_{v_{i-1},v_i}$. Substituting the above definitions into (6) and by the independence of the random variables t_i and s_i, it is possible to write distribution $m^t_{v_i}$ in terms of $\overline{m}^t_{v_{i-1}}$ and $m^s_{v_{i-1}}$ as follows:

$$\begin{aligned} m^t_{v_i}(z) &= \Pr(\{\bar{t}_{v_{i-1}} + t^{eval}_{v_{i-1}} + s_{v_{i-1}} + t_{v_{i-1},v_i} = z\} \cap \{r_{i-1} \text{ is op-feasible}\}) \\ &= \sum_{k \in \mathbb{N}} m^s_{v_{i-1}}(k) \overline{m}^t_{v_{i-1}}(z - t_{v_{i-1},v_i} - t^{eval}_{v_{i-1}} - k). \end{aligned} \tag{9}$$

Substituting the above expression into (7) yields:

$$\Pr\{r \text{ is op-feasible}\} = \sum_{z \le b_{v_q}} \sum_{k \in \mathbb{N}} m^s_{v_{q-1}}(k) \overline{m}^t_{v_{q-1}}(z - t_{v_{q-1},v_q} - t^{eval}_{v_{q-1}} - k). \tag{10}$$

Expression (10) can be applied recursively to obtain the probability that route r is op-feasible.

2.6 Route expected costs $\tilde{c}^C_r$ and $\tilde{c}^N_r$

Considering route $r = (v_0, \ldots, v_q, v_{q+1})$, we denote $P_r := \Pr\{r \text{ is op-feasible}\}$, and we remind that a penalty π is paid whenever the service at a customer is skipped. Furthermore, for any arc (v_{l-1}, v_l) in route r, we denote $\overline{p}_{v_{l-1},v_l} = \Pr\{r \text{ is successful up to } v_{l-1}\} - \Pr\{r \text{ is successful up to } v_l\}$ the decrease in the probability that route r is op-feasible caused by the arc (v_{l-1}, v_l). Condition C.2 considerably simplifies the calculation of the expected cost of a route. In fact it is straightforward to prove the following propositions.

Proposition 2.1. (Expected cost, recourse C). *The expected cost of route r can be expressed as*

$$\tilde{c}^C_r = c_r + \pi(1 - P_r),$$

where $c_r := \sum_{i=1}^{q+1} c_{v_{i-1},v_i}$.

Proposition 2.2. (Expected cost, recourse N). *The expected cost of route r can be expressed as*

$$\tilde{c}^N_r = c_r + \sum_{l=1}^{q-1} (\pi + c_{v_l,v_{l+2}} - c_{v_l,v_{l+1}} - c_{v_{l+1},v_{l+2}}) \overline{p}_{v_l,v_{l+1}}$$

$$= c_r + \sum_{l=1}^{q-1} \pi_{v_{l+1}}^{v_l,v_{l+2}} \ \overline{p}_{v_l,v_{l+1}} \tag{11}$$

where $c_r := \sum_{i=1}^{q+1} c_{v_{i-1},v_i}$ *and* $\pi_{v_{l+1}}^{v_l,v_{l+2}} := \pi + c_{v_l,v_{l+2}} - c_{v_l,v_{l+1}} - c_{v_{l+1},v_{l+2}}$.

It should be observed that, in the rest of the paper, we make the assumption that $\pi_{v_{l+1}}^{v_l,v_{l+2}} \geq 0$. Otherwise, it would be systematically more profitable to travel directly from v_l to v_{l+2} without servicing v_{l+1} which is undesirable for practical settings.

3 Branch-price-and-cut algorithms for the VRPTW-ST

Branch-price-and-cut is a branch-and-bound algorithm where a lower bound is computed by column generation at each node of the search tree. The lower bounds are tightened by strengthening the corresponding linear relaxation through the dynamic inclusion of valid inequalities (i.e., like in a cutting-plane method). In Section 3.1 we provide the details of our column generation procedure and, in particular, we investigate the mathematical properties that are useful in the solution of the corresponding subproblem for each recourse policy. In Section 3.2 we briefly describe the acceleration strategies implemented in our code, while cutting planes and branching strategies are discussed in Section 3.3.

3.1 Column generation

We begin this subsection by describing the general solution approach developed to solve the models proposed in this paper. As previously mentioned, a branch-price-and-cut algorithm produces a search tree where at each node the linear relaxation of problem (1)-(3), instantiated by the branching decisions defining the specific node and all previously added cutting planes, is solved to obtain a lower bound. Given the extremely large number of variables in (1)-(3), this linear relaxation cannot be solved directly and an iterative column generation procedure (see, e.g., Lübbecke and Desrosiers 2005) is invoked to do so. An iteration of column generation begins by solving a restricted master problem (RMP) defined as model (1)-(3) restricted to a subset of the routes included in $\mathcal{R}^l$, $l \in \{C, N\}$. Let $\bar{x}$ define the solution to this RMP. Then, the iteration pursues by solving a subproblem to verify the optimality of $\bar{x}$ with respect to the linear relaxation of (1)-(3) at the current node. When optimality is not observed, a set of new routes is added to the RMP and a new iteration begins. Otherwise, the column generation process stops.

Verifying if solution $\bar{x}$ is optimal entails searching for routes (i.e. variables) with negative reduced costs. Let $\gamma_i \geq 0$ define the dual multipliers associated with constraints (2). Therefore, for each recourse policy $l \in \{C, N\}$, the reduced cost $\bar{c}_r^l$ of a given route $r \in \mathcal{R}^l$ can be expressed as follows:

$$\bar{c}_r^l := \tilde{c}_r^l - \sum_{i \in V_c} a_{ir} \gamma_i. \tag{12}$$

For a specific policy $l \in \{C, N\}$, the column generation subproblem thus consists of minimizing (12) over the set of feasible routes $\mathcal{R}^l$. At each iteration of the algorithm, the subproblem is simply initialized by using the dual multiplier values associated with the solution of the RMP. The resulting subproblem belongs to the family of elementary shortest path problems with resource constraints (ESPPRC). The particularity in this case is that the time resource is probabilistic. As shown in Errico et al. (2013) that considered a similar subproblem, dynamic programming can be used to efficiently solve these ESPPRC. In the present context, the challenges to address for applying dynamic programming reside both in dealing with the recourse actions considered in the stochastic models and in handling assumption C.2. In particular, the recourse costs imply added complexity when computing valid lower and upper bounds for the expected total costs associated with partial routes.

In the rest of this subsection, we describe the labeling algorithms developed to solve the subproblems defined under the two recourse policies. Specifically, we detail the labels, extension functions and the dominance rules adopted in the proposed algorithms.

3.1.1 A labeling algorithm for recourse C

Labeling algorithms (see Irnich and Desaulniers 2005) are used to solve various types of shortest path problems. Such an algorithm represents partial paths originating from the source node in a network using vectors called labels. Starting with an initial label E_0 at the source node 0, the algorithm enumerates partial paths by propagating labels forwardly through the network G using extension functions. The paths ending at a same node are compared using a dominance rule in order to eliminate paths for which it can be proven that they cannot yield an optimal source-to-sink path.

The labeling algorithm developed in this section inherits the structure proposed in Errico et al. (2013). However, given the different cost structure and assumption C.2 that imposes new constraints on the routes, the label components, the extension functions and the dominance rules of the algorithm require important modifications.

In the following, given a route r, we denote $\mathcal{N}(r)$ and $\mathcal{A}(r)$ the set of nodes and arcs in r, respectively. When referring to a partial route, we add a subscript to specify the last visited customer. For example, $r_i = (0, \ldots, i)$ denotes a generic partial route ending at node $i \in V$. The notation $\mathcal{N}(\cdot)$ and $\mathcal{A}(\cdot)$ naturally generalizes to partial routes.

Label choice. We represent a partial route $r_i = (0, \ldots, i)$ by a label with the following $4 + n + b_i - a_i$ components:

- n components $V_i^1, \ldots, V_i^n$ indicating the number of visits to each node (0 or 1 for elementary paths). V_i^l can also be set to 1 if node l is unreachable (see definition below).
- $b_i - a_i + 1$ components $\overline{M}_i^t(a_i), \ldots, \overline{M}_i^t(b_i)$, one for each $z \in \mathbb{N} \cap [a_i, b_i]$, representing the distribution of the earliest time at which the service time evaluation can start provided the route is op-feasible, where we define

$$\overline{M}_i^t(t) := \sum_{l \leq t} \overline{m}_i^t(l). \tag{13}$$

Note that $\overline{M}_i^t(t)$ is also defined for $t \notin [a_i, b_i]$. However, given that $\overline{M}_i^t(t) = 0$ for all $t < a_i$ and $\overline{M}_i^t(t) = \overline{M}_i^t(b_i)$ for all $t > b_i$, we store only the values in the interval $[a_i, b_i]$. Observe that (7) and (13) enable to write:

$$P_i = \overline{M}_i^t(b_i), \tag{14}$$

where $P_i := \Pr\{r_i \text{ is op-feasible}\}$. These components allow to handle condition C.1 and to compute the route reduced cost.

- One component C_i representing the reduced cost of partial route r_i. Specifically, using (14) we define the expected cost of a partial route r_i as

$$\begin{aligned}\tilde{c}_{r_i}^C &= c_{r_i} + \pi(1 - P_i)\\ &= c_{r_i} + \pi(1 - \overline{M}_i^t(b_i)).\end{aligned}$$

  Hence, the reduced cost component C_i can be expressed in the following way:

$$C_i := \bar{c}_{r_i}^C = \tilde{c}_{r_i}^C - \sum_{l \in \mathcal{N}(r_i)} \gamma_l.$$

- Two components T_i^{max} and T_i^{rec} to account for condition C.2. T_i^{max} represents the latest possible arrival time in i if no recourse action was performed before reaching i, while T_i^{rec} represents the latest possible arrival time in i if a recourse action was taken. These times are possibly adjusted to meet the time window lower bound.

The number of label components at each node is node-dependent and varies according to the node's time window width. The initial label $E_0 = (C_0, V_0^1, \ldots, V_0^n, \overline{M}_0^t(a_0), \ldots, \overline{M}_0^t(b_0), T_0^{max}, T_0^{rec})$ at node 0 is such that $C_0 = 0$, $V_0^l = 0$, $\forall l \in V_c$, $\overline{M}_0^t(z)$, $\forall z \in \mathbb{N} \cap [a_i, b_i]$, $T_0^{max} = T_0^{rec} = a_0$.

Extension functions. Considering a node $i \in V$ and an associated label $E_i = (C_i, V_i^1, \ldots, V_i^n, \overline{M}_i^t(a_i), \ldots, \overline{M}_i^t(b_i), T_i^{max}, T_i^{rec})$, we now define how to extend this label along an arc $(i,j) \in A$ to create a new label $E_j = (C_j, V_j^1, \ldots, V_j^n, \overline{M}_j^t(a_j), \ldots, \overline{M}_j^t(b_j), T_j^{max}, T_j^{rec})$. For the components $V^1, \ldots, V^n$, the extension is straightforward: $V_j^j = V_i^j + 1$ and $V_j^l = V_i^l$ for all $l \in V_c$. However, for any $l \in V_c$ with $V_j^l = 0$ and such that any subsequent extension from j to l cannot satisfy the time window upper bound at l, then V_j^l is set to 1, i.e., l is considered unreachable (see Feillet et al. 2004).

To compute the label components $\overline{M}_j^t(z_j)$, $z_j \in \mathbb{N} \cap [a_j, b_j]$, we follow the method developed in Errico et al. (2013). Each of these components can be expressed as a function of the $b_i - a_i + 1$ label components $\overline{M}_i^t(z_i)$, one for each $z_i \in \mathbb{N} \cap [a_i, b_i]$. In fact, observe that by combining definition (13) with expression (9), we can write

$$\overline{M}_j^t(z_j) = \sum_{k \in \mathbb{N}} m_i^s(k) \sum_{z \leq z_j} \overline{m}_i^t(z - t_{ij} - t_i^{eval} - k)$$

$$= \sum_{k \in \mathbb{N}} m_i^s(k) \overline{M}_i^t(z_j - t_{ij} - t_i^{eval} - k) \tag{15}$$

for all $z_j \in \mathbb{N} \cap [a_j, b_j]$.

For the extension of the reduced cost, consider that

$$\begin{aligned} C_j = \bar{c}^C_{r_j} &= \tilde{c}^C_{r_j} - \sum_{l \in \mathcal{N}(r_j)} \gamma_l \\ &= c_{r_j} + \pi(1 - \overline{M}_j^t(b_j)) - \sum_{l \in \mathcal{N}(r_j)} \gamma_l \\ &= c_{r_i} + c_{ij} + \pi(1 - \overline{M}_i^t(b_i)) + \pi \overline{p}_{ij} - \sum_{l \in \mathcal{N}(r_i)} \gamma_l - \gamma_j \\ &= \bar{c}^C_{r_i} - \gamma_j + c_{ij} + \pi \overline{p}_{ij} \\ &= C_i - \gamma_j + c_{ij} + \pi \overline{p}_{ij}, \end{aligned}$$

which implicitly defines the corresponding extension function.

We now focus on the computation of the label components T_j^{max} and T_j^{rec}. The extension function used to compute T_j^{max} is given by

$$T_j^{max} = \min\{\max\{a_j, T_i^{max} + t_i^{eval} + s_i^{max} + t_{ij}\}, b_j\}.$$

In this function, $T_i^{max} + t_i^{eval} + s_i^{max} + t_{ij}$ corresponds to the latest arrival time at j if its time window is not considered. The minimum and maximum functions ensure the respect of this time window. In particular, the minimum involving b_j is required because a value greater than b_j would mean that a recourse action should have been taken before arriving at j.

For the computation of T_j^{rec}, three cases need to be considered: no recourse action was performed prior to j, a recourse action was performed in i or a recourse action occurred prior to i. In the first case, the value of T_j^{rec} is not meaningful and can be set to 0. In the second case, the latest arrival time at j following a recourse action is given by $T_i^{max} + t_i^{eval} + t_{ij}$. The third case results in the time $T_i^{rec} + t_i^{eval} + s_i^{max} + t_{ij}$. Depending on the service time realizations observed along the route, the second and third cases might need to be considered simultaneously in the label extension. Consequently, these cases yield the following extension function:

$$T_j^{rec} = \begin{cases} 0 & \text{if } \overline{M}_j^t(b_j) = 1 \\ \max\{a_j, T_i^{max} + t_i^{eval} + t_{ij}\} & \text{if } \overline{M}_j^t(b_j) < 1 \text{ and } \overline{M}_i^t(b_i) = 1 \\ \max\{a_j, T_i^{rec} + t_i^{eval} + s_i^{max} + t_{ij}, T_i^{max} + t_i^{eval} + t_{ij}\} & \text{otherwise,} \end{cases}$$

where possible waiting before the time window lower bound a_j is taken into account with the maximum function. The first part of this function corresponds to the first case, the second to the second case where, additionally, no recourse action was performed prior to i, and the third to a mix of the second and third cases.

The label E_j resulting from these extension functions is deemed feasible if $V_j^l \leq 1$ for all $l \in V_c$ (elementary requirements), $\overline{M}_j^t(b_j) \geq \alpha$ (condition C.1) and $T_j^{rec} \leq b_j$ (condition C.2). Note that the time window constraints are implicitly considered in the definition of the extension functions and, because $\alpha > 0$, condition C.1 ensures that they are satisfied for certain service time realizations. When label E_j is not feasible, it is discarded.

As a final remark, observe that the extension functions associated with the label components $V^1, \ldots, V^n$, T^{max} and T^{rec} are non-decreasing, a property exploited by the dominance rule.

Dominance. Consider two feasible partial routes r_i^h, $h = 1, 2$, both ending in a node $i \in V_c$ and represented by the labels $E_i^h = (C_i^h, V_i^{1,h}, \ldots, V_i^{n,h}, \overline{M}_i^{t,h}(a_i), \ldots, \overline{M}_i^{t,h}(b_i), T_i^{max,h}, T_i^{rec,h})$, $h = 1, 2$. We say that E_i^1 dominates E_i^2 if

1) any feasible extension e of r_i^2 ending at a given node j is also feasible for r_i^1, and

2) for any such extension e, the inequality $C_j^1 \leq C_j^2$ holds, where C_j^h is the reduced cost of the route obtained by extending route r_i^h, $h = 1, 2$.

In this case, the dominated label E_i^2 can be discarded. When these labels dominate each other, at most one of them can be deleted. Because conditions 1) and 2) are difficult to verify, we derive below sufficient conditions to identify dominated labels. In the following we make repeatedly use of the following lemma.

Lemma 3.1. (Stochastic dominance, Errico et al. 2013). *If the routes r_i^1 and r_i^2 are such that their labels satisfy $\overline{M}_i^{t,1}(z) \geq \overline{M}_i^{t,2}(z)$ for all $z \in \mathbb{N} \cap [a_i, b_i]$, then for any common feasible extension e ending at a given node j, the extended labels satisfy $\overline{M}_j^{t,1}(z) \geq \overline{M}_j^{t,2}(z)$ for all $z \in \mathbb{N} \cap [a_j, b_j]$.*

To simplify the notation, we express the reduced cost $\bar{c}_{r_i}$ of a partial route r_i ending in node $i \in V_c$ as $\bar{c}_{r_i}^C = \rho_{r_i} + \pi(1 - \overline{M}_i^t(b_i))$ where $\rho_{r_i} := c_{r_i} - \sum_{l \in \mathcal{N}(r_i)} \gamma_l$. Superscripts might be added to this notation to identify the routes.

Proposition 3.1. (Dominance rule, recourse C). *If r_i^1 and r_i^2 are such that*

(i) $\rho_i^1 \leq \rho_i^2$,

(ii) $V_i^{l,1} \leq V_i^{l,2}$ *for all* $l \in V_c$,

(iii) $\overline{M}_i^{t,1}(z) \geq \overline{M}_i^{t,2}(z)$ *for all* $z \in \mathbb{N} \cap [a_i, b_i]$,

(iv) $T_i^{max,1} \leq T_i^{max,2}$, *and*

(v) $T_i^{rec,1} \leq T_i^{rec,2}$,

then r_i^1 dominates r_i^2 in the sense specified by conditions 1) *and* 2).

Proof. We need to show that conditions 1) and 2) are satisfied under the hypotheses $(i) - (v)$. Consider an extension e ending at customer j of routes r_i^1 and r_i^2, resulting in routes denoted $r_i^1 \oplus e$ and $r_i^2 \oplus e$, respectively. Assume that $r_i^2 \oplus e$ is feasible, that is, it is elementary, its probability of being op-feasible $\overline{M}_j^{t,2}(b_j)$ is greater than or equal to α, and the latest arrival time if a recourse action is taken $T_j^{rec,2}$ is less than or equal to b_j. From hypotheses $(ii) - (v)$, Lemma 3.1, and the fact that the extension functions of the $V^1, \dots, V^n$, T^{max} and T^{rec} components are non-decreasing, it follows that $r_i^1 \oplus e$ is also feasible and, thus, condition 1) is met.

Now, we can write:

$$\begin{aligned}
C_j^1 &= \bar{c}^C_{r_i^1 \oplus e} = \rho_j^1 + \pi(1 - \overline{M}_j^{t,1}(b_j)) \\
&= c_{r_i^1} + \sum_{(i,j)\in\mathcal{A}(e)} c_{ij} - \sum_{l\in\mathcal{N}(r_i^1)} \gamma_l - \sum_{l\in\mathcal{N}(e)} \gamma_l + \pi(1 - \overline{M}_j^{t,1}(b_j)) \\
&= \rho_i^1 + \sum_{(i,j)\in\mathcal{A}(e)} c_{ij} - \sum_{l\in\mathcal{N}(e)} \gamma_l + \pi(1 - \overline{M}_j^{t,1}(b_j)) \\
&\leq \rho_i^2 + \sum_{(i,j)\in\mathcal{A}(e)} c_{ij} - \sum_{l\in\mathcal{N}(e)} \gamma_l + \pi(1 - \overline{M}_j^{t,1}(b_j)) \\
&= \rho_j^2 + \pi(1 - \overline{M}_j^{t,1}(b_j)) \\
&\leq \rho_j^2 + \pi(1 - \overline{M}_j^{t,2}(b_j)) = \bar{c}^C_{r_i^2 \oplus e} = C_j^2
\end{aligned}$$

where the first inequality follows from (i), and the second from Lemma 3.1 together with the fact that $\pi \geq 0$. Thus, condition 2) is satisfied. □

3.1.2 A labeling algorithm for recourse N

One of the main differences between recourses N and C appears in the cost structure. First of all, as it will be made clear later on, there is no unique way to define the cost of a partial route. In order to overcome such difficulty, we introduce below the concept of *incomplete cost* of a partial route. Secondly, because performing recourse N implies a shortcut in the a priori route (the visit to a customer is skipped), the actual routing cost depends on the customer where the recourse action occurs, as opposite to recourse C where such a cost is customer-independent. Therefore, this requires more involved techniques to bound route costs and prove the dominance rule.

The structure of the label setting algorithm proposed in this section is similar to that presented in the previous one. However, certain label component definitions and extension functions, as well as the dominance rule have been modified. We slightly modify the notation introduced in the previous section as we use subscripts to make explicit the visit order of the customers. For example, a partial route is denoted r_{v_i} where $r_{v_i} = (v_0, \dots, v_i)$ and v_0 is the depot 0. In the following we describe our algorithm by underlining the differences with the previous one.

Label choice. A partial route $r_{v_i} = (0, \dots, v_i)$ is represented by a label with the following $4 + n + b_{v_i} - a_{v_i}$ components:

- n components $V_{v_i}^1, \dots, V_{v_i}^n$ as for recourse C.
- $b_{v_i} - a_{v_i} + 1$ components $\overline{M}_{v_i}^t(a_{v_i}), \dots, \overline{M}_{v_i}^t(b_{v_i})$, one for each $z \in \mathbb{N} \cap [a_{v_i}, b_{v_i}]$ as for recourse C.
- One component C_{v_i} is used to account for the reduced cost. Differently from recourse C, there is no unique way to determine the cost of a partial route. In fact, if we consider a partial route ending at a generic customer v_i, the corresponding penalty $\pi_{v_i}^{v_{i-1},v_{i+1}}$ depends on v_{i+1} which,

at this point, is unknown. To overcome this difficulty, we introduce the *incomplete* expected cost of a partial route r_{v_i}:

$$\tilde{c}^N_{r_{v_i}} := c_{r_{v_i}} + \sum_{l=1}^{i-2} \pi_{v_{l+1}}^{v_l, v_{l+2}} \, \overline{p}_{v_l, v_{l+1}}. \tag{16}$$

Note that because the time window at the depot 0 is unconstraining, the above expression gives the actual expected cost (11) if $v_i = 0$. Hence, we define the label component C_{v_i} as the *incomplete reduced cost* of a partial route

$$C_{v_i} := \overline{c}^N_{r_{v_i}} := \tilde{c}^N_{r_{v_i}} - \sum_{l=1}^{i} \gamma_{v_l}.$$

- Two components $T^{max}_{v_i}$ and $T^{rec}_{v_i}$ as for recourse C. However, the extension functions used to compute these components differ from those presented for recourse C.

The initial label E_0 is the same as for recourse C.

Extension functions. Given a node $v_i \in V$, we now show how the associated label $E_{v_i} = (C_{v_i}, V^1_{v_i}, \ldots, V^n_{v_i}, \overline{M}^t_{v_i}(a_{v_i}), \ldots, \overline{M}^t_{v_i}(b_{v_i}), T^{max}_{v_i}, T^{rec}_{v_i})$ is extended along an arc $(v_i, v_{i+1}) \in A$ to yield a label $E_{v_{i+1}} = (C_{v_{i+1}}, V^1_{v_{i+1}}, \ldots, V^n_{v_{i+1}}, \overline{M}^t_{v_{i+1}}(a_{v_{i+1}}), \ldots, \overline{M}^t_{v_{i+1}}(b_{v_{i+1}}), T^{max}_{v_{i+1}}, T^{rec}_{v_{i+1}})$. Label components $V^1_{v_{i+1}}, \ldots, V^n_{v_{i+1}}$, $\overline{M}^t_{v_{i+1}}(\cdot)$, and $T^{max}_{v_{i+1}}$ are computed using the extension functions described for recourse C.

For the extension of the incomplete reduced cost, consider that

$$\begin{aligned}
\overline{c}^N_{r_{v_{i+1}}} &= \tilde{c}^N_{r_{v_{i+1}}} - \sum_{l=1}^{i+1} \gamma_{v_l} \\
&= c_{r_{v_{i+1}}} + \sum_{l=1}^{i-1} \pi_{v_{l+1}}^{v_l, v_{l+2}} \overline{p}_{v_l, v_{l+1}} - \sum_{l=1}^{i+1} \gamma_{v_l} \\
&= c_{r_{v_i}} + c_{v_i, v_{i+1}} + \sum_{l=1}^{i-2} \pi_{v_{l+1}}^{v_l, v_{l+2}} \overline{p}_{v_l, v_{l+1}} + \pi_{v_i}^{v_{i-1}, v_{i+1}} \overline{p}_{v_{i-1}, v_i} - \sum_{l=1}^{i} \gamma_{v_l} - \gamma_{v_{i+1}} \\
&= \overline{c}^N_{r_{v_i}} + c_{v_i, v_{i+1}} + \pi_{v_i}^{v_{i-1}, v_{i+1}} \overline{p}_{v_{i-1}, v_i} - \gamma_{v_{i+1}},
\end{aligned}$$

and, in terms of label components,

$$C_{v_{i+1}} = C_{v_i} + c_{v_i, v_{i+1}} + \pi_{v_i}^{v_{i-1}, v_{i+1}} \overline{p}_{v_{i-1}, v_i} - \gamma_{v_{i+1}}.$$

For the extension function of the label component $T^{rec}_{v_i}$, we need to consider again the three cases: no recourse action was performed prior to v_{i+1}, a recourse action was performed in v_i or a recourse action occurred prior to v_i. Only the treatment of the second case differs from that considered for recourse C. Indeed, here, the visit at customer v_i is skipped, resulting in the latest arrival time at v_{i+1} equal to $T^{max}_{v_{i-1}} + t^{eval}_{v_{i-1}} + s^{max}_{v_{i-1}} + t_{v_{i-1}, v_{i+1}}$. Consequently, the following extension function is:

$$T^{rec}_{v_{i+1}} = \begin{cases} 0 & \text{if } \overline{M}^t_{v_{i+1}}(b_{v_{i+1}}) = 1 \\ \max\{a_{v_{i+1}}, T^{max}_{v_i} + t^{eval}_{v_i} + t_{v_i,v_{i+1}}\} & \text{if } \overline{M}^t_{v_{i+1}}(b_{v_{i+1}}) < 1 \text{ and } \overline{M}^t_{v_i}(b_{v_i}) = 1 \\ \max\{a_{v_{i+1}}, T^{rec}_{v_i} + t^{eval}_{v_i} + s^{max}_{v_i} + t_{v_i,v_{i+1}}, T^{max}_{v_{i-1}} + t^{eval}_{v_{i-1}} + s^{max}_{v_{i-1}} + t_{v_{i-1},v_{i+1}}\} & \text{otherwise.} \end{cases}$$

As for recourse C, label $E_{v_{i+1}}$ is declared feasible if $V^l_{v_{i+1}} \leq 1$ for all $l \in V_c$, $\overline{M}^t_{v_{i+1}}(b_{v_{i+1}}) \geq \alpha$ and $T^{rec}_{v_{i+1}} \leq b_{v_{i+1}}$. Again, we observe here that the extension functions for the label components $V^1, \ldots, V^n$, T^{max} and T^{rec} are non-decreasing.

Dominance. Consider two feasible partial routes $r^h_{v_i}$, $h = 1, 2$, both ending in a given node $v_i \in V_c$ and represented by the labels $E^h_{v_i} = (C^h_{v_i}, V^{1,h}_{v_i}, \ldots, V^{n,h}_{v_i}, \overline{M}^{t,h}_{v_i}(a_{v_i}), \ldots, \overline{M}^{t,h}_{v_i}(b_{v_i}), T^{max,h}_{v_i}, T^{rec,h}_{v_i})$, $l = 1, 2$. The definition of dominance is very similar to the one used for recourse C. We say that $E^1_{v_i}$ dominates $E^2_{v_i}$ if

1) Any feasible extension $e_{v_{q+1}} = (v_i, \ldots, v_{q+1})$, where v_{q+1} represents the depot 0, of $r^2_{v_i}$ is also feasible for $r^1_{v_i}$, and

2) For any such extension $e_{v_{q+1}}$, the inequality $C^1_{v_{q+1}} \leq C^2_{v_{q+1}}$ holds, where $C^h_{v_{q+1}}$ is the reduced cost of the route obtained by extending route $r^h_{v_i}$, $h = 1, 2$.

Formally, the main difference between the present dominance definition and the one previously stated for recourse C is that we only consider here extensions ending in the depot 0. The main reason for this choice comes from the fact that our label components C_{v_k} represent incomplete reduced costs that only coincide with the actual reduced cost in v_{q+1}, i.e., the depot. In this case, dominance properties are better proved by first providing in Lemma 3.2 parametric upper and lower bounds on the reduced cost of a generic route $r_{v_i} \oplus e_{v_{q+1}}$, by giving in Proposition 3.2 parametric sufficient conditions for label dominance, and finally by deriving in Corollary 3.1 the actual sufficient conditions for dominance.

To simplify the notation, we set

$$\rho_{r_{v_i}} := \sum_{l=0}^{i-1} (c_{v_l,v_{l+1}} - \lambda_{v_{l+1}}), \tag{17}$$

$$\rho_{e_{v_{q+1}}} := \sum_{l=i}^{q} (c_{v_l,v_{l+1}} - \lambda_{v_{l+1}}). \tag{18}$$

where $\lambda_{v_{q+1}} = 0$. Hence, for example, the incomplete reduced cost for route r_{v_i} can be written $\overline{c}^N_{r_{v_i}} = \rho_{r_{v_i}} + \sum_{l=1}^{i-2} \pi^{v_l,v_{l+2}}_{v_{l+1}} \overline{p}_{v_l,v_{l+1}}$. We also define

$$\pi^{max} := \max_{i,l \in V_c,\ k \in V,\ l \neq k \neq i,\ l \neq i} \{\pi^{lk}_i\},$$

$$\pi^{min} := \min_{i,l \in V_c,\ k \in V,\ l \neq k \neq i,\ l \neq i} \{\pi^{lk}_i\},$$

and, for any $l, i \in V_c$,

$$\delta_{li}^{max} = \max_{k \in V}\{c_{lk} - c_{ik}\}, \tag{19}$$

$$\delta_{li}^{min} = \min_{k \in V}\{c_{lk} - c_{ik}\}. \tag{20}$$

Finally, given that the time window at the depot v_{q+1} is unconstraining, we get that the probability that a route r_{v_q} is op-feasible equals that of $r_{v_{q+1}} := r_{v_q} \cup \{(v_q, v_{q+1})\}$ and, therefore, $\overline{M}^t_{v_{q+1}}(b_{v_{q+1}}) = \overline{M}^t_{v_q}(b_{v_q})$.

Lemma 3.2. (Upper and lower bounds, recourse N) *The incomplete reduced cost* $\overline{C}_{v_{q+1}}$ *of route* $r_{v_i} \oplus e_{v_{q+1}}$ *admits the following upper and lower bounds:*

$$\overline{C}_{v_{q+1}} \leq \overline{C}_{v_i} + (\pi - c_{v_{i-1},v_i} + \delta^{max}_{v_{i-1},v_i})\overline{p}_{v_{i-1},v_i} + \rho_e + \pi^{max}(\overline{M}^t_{v_i}(b_{v_i}) - \overline{M}^t_{v_q}(b_{v_q})) =: \overline{B}(\overline{M}^t_{v_q}(b_{v_q})) \tag{21}$$

$$\overline{C}_{v_{q+1}} \geq \overline{C}_{v_i} + (\pi - c_{v_{i-1},v_i} + \delta^{min}_{v_{i-1},v_i})\overline{p}_{v_{i-1},v_i} + \rho_e + \pi^{min}(\overline{M}^t_{v_i}(b_{v_i}) - \overline{M}^t_{v_q}(b_{v_q})) =: \underline{B}(\overline{M}^t_{v_q}(b_{v_q})). \tag{22}$$

Proof. Given that the techniques to prove the validity of the upper and the lower bound are similar, we only focus on the upper bound. We rewrite the incomplete reduced cost of $r_{v_i} \oplus e_{v_{q+1}}$ by underlying the different components

$$\overline{C}_{v_{q+1}} = \overline{C}_{r_{v_i}} + \pi^{v_{i-1},v_{i+1}}_{v_i}\overline{p}_{v_{i-1},v_i} + \rho_e + \sum_{l=i}^{q-1} \pi^{v_l,v_{l+2}}_{v_{l+1}}\overline{p}_{v_l,v_{l+1}}. \tag{23}$$

Consider now that

$$\pi^{v_{i-1},v_{i+1}}_{v_i} \leq \pi - c_{v_{i-1},v_i} + \delta^{max}_{v_{i-1},v_i},$$

and that

$$\pi^{v_l,v_{l+2}}_{v_{l+1}} \leq \pi^{max} \text{ for all } l \in \{i, \ldots, q-1\}.$$

Finally, observing that

$$\sum_{l=i}^{q-1} \overline{p}_{v_l,v_{l+1}} = \overline{M}^t_{v_i}(b_{v_i}) - \overline{M}^t_{v_q}(b_{v_q}),$$

and substituting this relation and the above inequalities in (23), the lemma is proved. □

The following proposition provides a framework specifying how parametric lower and upper bounds on the cost of a solution can be used to identify dominated labels.

Proposition 3.2. *Suppose that a lower bound* $\underline{B}^h(\overline{M}^{t,h}_{v_q}(b_{v_q}))$ *and an upper bound* $\overline{B}^h(\overline{M}^{t,h}_{v_q}(b_{v_q}))$ *are available on the incomplete reduced cost* $C^h_{v_{q+1}}$ *of generic routes* $r^h_{v_i} \oplus e_{v_{q+1}}$*,* $h = 1, 2$*, and suppose that* $\overline{B}^1(\overline{M}^{t,1}_{v_q}(b_{v_q}))$ *is a non-increasing function. If two partial routes* $r^1_{v_i}$ *and* $r^2_{v_i}$ *are such that*

(i) $\overline{B}^1(\overline{M}^{t,2}_{v_q}(b_{v_q})) \leq \underline{B}^2(\overline{M}^{t,2}_{v_q}(b_{v_q}))$,

(ii) $V^{l,1}_{v_i} \leq V^{l,2}_{v_i}$ *for all* $l \in V_c$,

(iii) $M^{t,1}_{v_i}(z) \geq M^{t,2}_{v_i}(z)$ *for all* $z \in \mathbb{N} \cap [a_{v_i}, b_{v_i}]$,

(iv) $T^{max,1}_{v_i} \leq T^{max,2}_{v_i}$, *and*

(v) $T^{rec,1}_{v_i} \leq T^{rec,2}_{v_i}$,

then r^1 *dominates* r^2.

Proof. We need to show that conditions 1) and 2) of the dominance definition are satisfied under the hypotheses $(i) - (v)$. To prove that condition 1) is satisfied we use the same reasoning as in Proposition 3.1. In fact, consider any extension $e_{v_{q+1}}$ of $r^1_{v_i}$ and $r^2_{v_i}$ to obtain routes $r^1_{v_i} \oplus e_{v_{q+1}}$ and $r^2_{v_i} \oplus e_{v_{q+1}}$, respectively. Assume that $r^2_{v_i} \oplus e_{v_{q+1}}$ is feasible. From hypotheses $(ii) - (v)$, Lemma 3.1, and the fact that the extension functions of the $V^1, \ldots, V^n$, T^{max} and T^{rec} components are non-decreasing, it follows that $r^1_{v_i} \oplus e_{v_{q+1}}$ is also feasible and condition 1) is met.

Now, we can deduce that

$$\begin{aligned}\overline{C}^1_{v_{q+1}} &\leq \overline{B}^1(\overline{M}^{t,1}_{v_q}(b_{v_q})) \\ &\leq \overline{B}^1(\overline{M}^{t,2}_{v_q}(b_{v_q})) \\ &\leq \underline{B}^2(\overline{M}^{t,2}_{v_q}(b_{v_q})) \leq \overline{C}^2_{v_{q+1}}.\end{aligned}$$

The second inequality ensues from the facts that hypothesis (iii) and Lemma 3.1 imply $\overline{M}^{t,1}_{v_q}(b_{v_q}) \geq \overline{M}^{t,2}_{v_q}(b_{v_q})$ and that $\overline{B}^1(\cdot)$ is a non-increasing function. The third inequality is valid from hypothesis (i). This shows that condition 2) is also satisfied. □

Corollary 3.1. (Dominance rule) *Consider the two partial routes* $r^1_{v_i}$ *and* $r^2_{v_i}$ *and any common extension* $e_{v_{q+1}}$*. If these routes satisfy the conditions* $(ii) - (v)$ *of Proposition 3.2 and the following condition:*

$$\begin{aligned}\overline{C}^1_{v_i} \leq \overline{C}^2_{v_i} - (\pi - c_{v_{i-1},v_i} + \delta^{1\ max}_{v_{i-1},v_i})\overline{p}^1_{v_{i-1},v_i} + (\pi - c_{v_{i-1},v_i} + \delta^{1\ min}_{v_{i-1},v_i})\overline{p}^2_{v_{i-1},v_i} + \\ -\overline{M}^{1t}_{v_i}(b_{v_i})\pi^{max} + \overline{M}^{2t}_{v_i}(b_{v_i})\pi^{min} + \alpha(\pi^{max} - \pi^{min}),\end{aligned} \tag{24}$$

then r^1 *dominates* r^2.

Proof. Given that the values π^{hk}_i are positive (see the assumption at the end of Section 2.6), the bound $\overline{B}^1(\overline{M}^{t,1}_{v_q}(b_{v_q}))$ in Lemma 3.2 is a non-increasing function of $\overline{M}^{t,1}_{v_q}(b_{v_q})$. Hence, Proposition

3.2 is applicable with the bounds of Lemma 3.2. With these bounds, condition (i) of Proposition 3.2 can be rewritten as follows:

$$\overline{C}^1_{v_i} \leq \overline{C}^2_{v_i} - (\pi - c_{v_{i-1},v_i} + \delta^{1\ max}_{v_{i-1},v_i})\bar{p}^1_{v_{i-1},v_i} + (\pi - c_{v_{i-1},v_i} + \delta^{2\ min}_{v_{i-1},v_i})\bar{p}^2_{v_{i-1},v_i} + \\ -\overline{M}^{1t}_{v_i}(b_{v_i})\pi^{max} + \overline{M}^{2t}_{v_i}(b_{v_i})\pi^{min} + \overline{M}^{2t}_{v_q}(b_{v_q})(\pi^{max} - \pi^{min}). \tag{25}$$

The corollary is proved by observing that, because $\overline{M}^{t,2}_{v_q}(b_{v_q}) \geq \alpha$, one can replace $\overline{M}^{t,2}_{v_q}(b_{v_q})$ by α in inequality (25) to yield the sufficient condition (24). □

3.2 Acceleration strategies

Several known acceleration techniques from the literature have been adapted and implemented in our code. In particular we relax elementarity by adopting a combination of *ng*-path (Baldacci et al. 2011) and decremental state space relaxations (Boland et al. (2006) and Righini and Salani (2008)). We also perform heuristic dynamic programming by: 1) Temporarily eliminating unpromising arcs from the arc set A, and gradually recovering the full arc set when no negative reduced cost route is found; 2) applying an aggressive dominance rule to eliminate a large amount of labels, and progressively weakening it when no negative reduced cost route is found. Finally, we apply a tabu search heuristic to the columns in the current basis, as described in Desaulniers et al. (2008), to attempt to generate rapidly negative reduced cost columns. See Errico et al. (2013) for a more detailed description of the above acceleration strategies.

3.3 Cutting planes and branching strategies

We strengthen the lower bounds obtained at the end of the column generation procedure by adding to the problem a family of valid inequalities introduced in Jepsen et al. (2008) called *subset-row* inequalities. In the general case, subset-row inequalities are expressed (for $l \in \{N, C\}$) as

$$\sum_{r \in \mathcal{R}^l} \left\lfloor \frac{1}{k} \sum_{i \in S} a_{ir} \right\rfloor x_r \leq \left\lfloor \frac{|S|}{k} \right\rfloor, \quad \forall S \subseteq V_c,\ 2 \leq k \leq |S|.$$

However, similarly to Jepsen et al. (2008), we consider only the cuts obtained with $|S| = 3$ and $k = 2$ because they are easier to find. In this case, the cuts can be rewritten as

$$\sum_{r \in \mathcal{R}^l_S} x_r \leq 1, \quad \forall S \subseteq V_c\ : |S| = 3, \tag{26}$$

where $\mathcal{R}^l_S$ is the subset of paths visiting at least two customers in S.

When branching is required in the branch-and-bound search tree, two types of branching decisions can be applied. If the total number of vehicles used in the current fractional solution is fractional, we branch on this number by adding a constraint in the master problem (or modifying its right-hand side if it was previously added). The associated dual variable must then be subtracted from the reduced cost of every variable, that is, this dual value is added as a constant in the subproblem objective function. When the total number of vehicles is integer, we branch on an arc-flow variable that is selected using the farthest from the nearest integer rule which, in the present context, entails selecting the arc (i, j) whose flow is the closest to 0.5. On one branch, the flow on the selected arc is set to 0 by simply removing (i, j) from set A. On the other branch, the flow on this arc is set to 1 by removing all arcs (i, k), $k \neq j$, and all arcs (k, j), $k \neq i$, from A. The columns in the RMP that are in conflict with the imposed decision are deleted. Finally, the enumeration process applies a best-first search strategy to explore the search tree.

4 Computational experiments

We performed a series of experiments to investigate different aspects of our work, namely, to analyze the performance of the algorithms developed for each recourse and to compare the two recourses in terms of computational efficiency and solution costs. The experiments were performed on a machine running Linux (Suse) and an Intel Core i7-2600 3.40 GHz CPU with 16 G of RAM. On all tests performed, a maximum computing time of five hours was allowed to solve each instance.

4.1 Instance set

To perform our experiments, we adapted to our context the instances of the benchmark set introduced in Errico et al. (2013) which are based on the well-known datasets proposed in Solomon (1987) for the VRP with time windows. These datasets are grouped in six instance classes: R1, RC1, C1, R2, RC2, and C2. Each instance contains 100 customers that are distributed in a 100×100 square. Each class is first defined according to how the customers are located within the square (R for randomly, C for clustered and RC for a mix of randomly and clustered). Classes are then differentiated according to the average time window width compared to the average traveling time (1 for narrow time windows and 2 for wide time windows). A total of 56 instances are available, each class having between 8 and 12 instances. In Errico et al. (2013), the instances used were obtained by disregarding vehicle capacity and customer demands from the Solomon's datasets. Stochastic service times were also formulated using discrete triangular distributions. Several distribution families were considered including:

1. **Basic**: Symmetric triangular distributions with a median equal to the corresponding deterministic service time, i.e., 100 for R and RC classes and 900 for C classes, and support intervals of $[80, 120]$ for R and RC classes and $[700, 1100]$ for C classes.

2. **Large-support**: Similar to the Basic family, but the support of the probability distributions is considerably larger: $[50, 150]$ for R and RC classes, and $[450, 1350]$ for C classes.

In this paper, we provide the results for Basic and Large-support distributions and instances with narrow time windows (R1, RC1, C1), given that less recourse actions are needed in presence of wide time windows. Our algorithms were able to solve very few 100-customer instances. We consequently followed the common practice of including in our experimentation, instances with reduced size obtained from the original Solomon's instances by considering only their first 25 and 50 customers. All results were obtained by setting $\pi = 200$, $\alpha = 95\%$ and $t^{eval} = 10$.

It should be noticed that we performed experiments on a wider set of instances than what is reported here. In fact, we also experimented with positively-skewed discrete triangular distributions, discrete uniform distributions, as well as several values of π, α and t^{eval}. However, the obtained results were not significantly different in terms of solution quality and algorithmic performances than what is shown here. In consequence, we omit these results for reasons of conciseness.

4.2 Results

The results of our algorithm for recourse C, when addressing the Basic and the Large-support instances, are given in Table 1. These results are grouped by instance class and number of customers (the results for the 100-customer instances are reported at the end of this section). For each group of instances, Columns 1 and 2 identify the number of customers (n) and the instance class (class). Columns 3 to 7 concern the Basic instances. In particular, Column 3 indicates the ratio between the number of instances solved to optimality within 5 hours of computing time and the number of instances included in the group. Columns 4 and 5 report the mean and the standard deviation of the number of vehicles used in the optimal solutions obtained within this time limit. Columns 6 and 7 provide the mean and standard deviation of the computing times (in seconds) taken by the algorithm to solve the corresponding instances. Columns 8 to 12 report the results for Large-support instances and their meaning is the same as for the Basic case. Each column reports the aggregated results for a particular instance group identified by the first two columns. For example, row 1 reports aggregated results for instances belonging to the C1 family with 25 customers. A value *all* in one of the first two columns indicates that the results have been aggregated at a higher level. For example, row 4 reports the aggregated results for all the instances in C1, R1, and RC1 that include 25 customers, while row 9 reports the aggregated results over all the instances.

		Basic					Large-support				
		Instances	No. vehicles		Time (s)		Instances	No. vehicles		Time (s)	
n	Class	Solved	Avg	Std	Avg	Std	Solved	Avg	Std	Avg	Std
25	C1	9/9	3.2	0.4	2309.5	2403.8	3/9	4.7	0.5	4861.8	5965.1
	R1	12/12	5.0	1.3	18.8	31.3	12/12	5.8	1.6	32.7	50.9
	RC1	8/8	3.3	0.4	34.1	45.9	8/8	3.6	0.7	596.6	520.6
	all	29/29	4.0	1.2	733.9	1706.3	23/29	4.9	1.6	858.8	2684.6
50	C1	6/9	5.7	0.7	6430.3	4286.4	0/9				
	R1	11/12	8.5	2.0	1540.1	2985.2	9/12	10.0	2.1	916.7	1114.6
	RC1	7/8	6.7	1.2	1774.1	1659.3	2/8	8.5	0.5	6320.6	6119.6
	all	24/29	7.3	1.9	2830.9	3716.1	11/29	9.7	2.0	1899.2	3488.5
all	*all*	53/58	5.5	2.3	1683.5	2989.3	34/58	6.4	2.9	1195.4	3008.2

Table 1: Recourse C.

Table 2 allows to compare the results reported in the previous table by considering the subset of instances solved to optimality in both cases (i.e., the Basic and Large-support cases). It includes the relative variations observed in the results when recourse C is applied on the Large-support instances compared to when it is applied to the Basic instances. The Basic instance family serves here as the reference point and the results are aggregated as before. Again, for each instance group, Columns 1

		No. instances		Var. Rout. Cost (%)		Var. No. Vehicles	
n	Class	Common	Diff	Avg	Std	Avg	Std
25	C1	3	-6	24.5	2.4	100.0	0.0
	R1	12	0	5.3	4.3	75.0	59.5
	RC1	8	0	7.7	7.2	37.5	48.4
	all	23	-6	8.6	8.2	65.2	56.0
50	C1	0	-6				
	R1	9	-2	5.2	2.7	88.9	73.7
	RC1	2	-5	5.1	3.0	50.0	50.0
	all	11	-13	5.2	2.7	81.8	71.6
all	*all*	34	-19	7.5	7.1	70.6	62.0

Table 2: Recourse C. Comparison between Basic and Large-support instances.

and 2 identify the number of customers and the instance class. Columns 3 and 4 specify the number of instances used to perform the comparison (that is, those solved for both instance families), and the difference between the numbers of instances solved in each family, respectively. Columns 5 and 6 report the mean and standard deviation of the percentage of variation in the expected total routing cost of the solutions, respectively. Columns 7 and 8 provide the same statistics for the number of vehicles used.

The results show that our algorithm can solve 53 out of 58 instances of the Basic family and 34 out of 58 instances of the Large-support one. This clearly shows that the instances in the Large-support family are more difficult to solve that the ones in the Basic family. In fact, as was observed in Errico et al. (2013), the arrival time probability distributions at the customers usually have larger supports when the service time probability distributions also have larger supports. This fact has two general consequences: 1) the memory consumption is larger; and 2) it is harder to find dominated labels because there are more conditions to be verified. Furthermore, when comparing the optimal solutions obtained in both cases, we see that, in general, an increase in the data uncertainty entails routes with higher expected total routing costs and higher vehicle utilization. Finally, we observe that, for both recourses, the 50-customer instances are much more difficult to solve than the 25-customer instances and the instances in class C1 require the most effort because the support of the service time distributions is much larger for these instances than the others (see Section 4.1).

The same experiments were repeated using recourse N. The obtained results are presented in Tables 3 and 4, which display the same information as above. When assessing how our algorithm performs on instances obtained using the two distribution families, we make the same observations as with recourse C.

Comparative results on the instances solved to optimality for both recourse policies (C and N) are reported in Table 5, where the meaning of the columns is the same as for Table 4 with the exception that a new Column 1 has been added, specifying the distribution type. In this table, the results for recourse C serve as the reference points.

These results show that when the uncertainty on the data is relatively small (Basic instances), the solutions obtained with the two recourse policies are almost equivalent, both in terms of the expected routing costs and the number of vehicles used. However, these instances are harder to solve when formulated using the recourse policy N. Indeed, 16 instances could not be solved when applying recourse N while they were solved when recourse C was considered. Such a difference is mostly

		Basic					Large-support				
		Instances	No. vehicles		Time (s)		Instances	No. vehicles		Time (s)	
n	Class	Solved	Avg	Std	Avg	Std	Solved	Avg	Std	Avg	Std
25	C1	6/ 9	3.3	0.5	3355.5	3728.8	0/9				
	R1	12/12	5.0	1.3	47.5	79.8	11/12	4.9	1.0	111.3	115.9
	RC1	8/ 8	3.3	0.4	2295.8	4101.8	7/8	3.4	0.5	2865.9	4905.3
	all	26/29	4.1	1.3	1502.7	3217.4	18/29	4.3	1.1	1182.5	3342.0
50	C1	1/9	5.0	0.0	4124.8	0.0	0/9				
	R1	7/12	9.3	1.9	1432.9	1876.4	5/12	9.2	1.5	5436.8	4797.5
	RC1	3/8	7.3	0.9	7430.5	6323.3	1/8	9.0	0.0	310.7	0.0
	all	11/29	8.4	2.1	3313.3	4480.9	6/29	9.2	1.3	4582.4	4778.0
all	*all*	37/58	5.4	2.5	2041.0	3732.0	24/58	5.5	2.4	2032.5	4031.3

Table 3: Recourse N.

		No. instances		Var. Rout. Cost (%)		Var. No. Vehicles	
n	Class	Common	Diff	Avg	Std	Avg	Std
25	C1	0	-6				
	R1	11	-1	0.7	2.0	18.2	38.6
	RC1	7	-1	2.9	5.5	14.3	35.0
	all	18	-8	1.5	3.9	16.7	37.3
50	C1	0	-1				
	R1	3	-2	6.0	2.4	33.3	47.1
	RC1	1	-2	2.5	0.0	100.0	0.0
	all	4	-5	5.1	2.5	50.0	50.0
all	*all*	22	-13	2.2	4.0	22.7	41.9

Table 4: Recourse N. Comparison between Basic and Large-support instances.

			No. instances		Var. Rout. Cost (%)		Var. No. Vehicles	
Distr.	n	Class	Common	Diff	Avg	Std	Avg	Std
Basic	25	C1	6	-3	-0.8	2.0	0.0	0.0
		R1	12	0	-0.2	0.6	0.0	0.0
		RC1	8	0	-1.6	4.0	0.0	0.0
		all	26	-3	-0.8	2.5	0.0	0.0
	50	C1	1	-5	0.0	0.0	0.0	0.0
		R1	7	- 4	0.2	0.7	0.0	0.0
		RC1	3	-4	0.5	0.9	0.0	0.0
		all	11	-13	0.2	0.7	0.0	0.0
	all	all	37	-16	-0.5	2.2	0.0	0.0
Large	25	C1	0	-3				
		R1	11	-1	-3.5	2.1	-45.5	49.8
		RC1	7	-1	-5.0	5.1	-28.6	45.2
		all	18	-5	-4.1	3.7	-38.9	48.7
	50	C1	0	0				
		R1	5	-4	-1.1	0.9	-60.0	49.0
		RC1	1	-1	0.4	0.0	0.0	0.0
		all	6	-5	-0.9	1.0	-50.0	50.0
	all	*all*	24	-10	-3.3	3.5	-41.7	49.3

Table 5: Comparison between recourses C and N.

due to the less effective dominance rule applied for recourse N. For the Large-support instances, one finds that the recourse policy N presents some advantages over the recourse policy C. In fact, improvements are observed in the quality of the solutions obtained. This is particularly evident with regards to the number of vehicles used in the solutions. These observations can be explained by the fact that when applying recourse N for a given customer along a route both the visit and the service are skipped, as opposed to recourse C where only the service is skipped. Consequently, when using recourse N, such shortcuts enable vehicles to possibly serve more customers.

For completeness, we show in Table 6 results for the 100-customer instances solved to optimality within 5 hours of computing time. Columns 1 and 2 identify the recourse action and the distribution used, Column 3 refers to the original name in the Solomon's database and Column 4 displays the computing time. We observe that not many 100-customer instances could be solved within the imposed computing time limit.

Recourse	Distr.	Inst. name	Time (s)
C	Basic	C107	12684.4
		R101	1.9
		R102	378.6
		R103	1082.7
		R105	1298.4
		RC101	768.8
		RC105	10341.8
	Large	R101	17.1
		R102	1362
		R105	10178.5
		RC101	3219.1
N	Basic	R101	19.6
		R105	2490.6
		RC101	16433.8
	Large	RC101	12077.9

Table 6: 100-customer instances.

5 Conclusions

In this paper we have introduced the VRPTW-ST in the form of a two-stage stochastic program that can be expressed using two different recourse policies, referred to as C (skipping the service at the current customer) and N (skipping the visit at the next customer). We formulate the problem as a set partitioning model and, for each proposed recourse policy, we formally define the feasible region and show how to compute the expected cost of a route. For the VRPTW-ST with both recourse policies, we developed state-of-the-art branch-cut-and-price algorithms by: defining recourse specific label choices, developing suitable extension functions, and deriving new dominance rules. Our computational results show that our algorithms are effective on the tested instances that involve up to 50 customers. Both algorithms are more efficient when solving instances with less service time uncertainty (i.e., the Basic instances). Instances formulated using the recourse policy N are generally harder to solve than those formulated using the recourse policy C. Furthermore, on the instances exhibiting less service time uncertainty, the solutions obtained using the two recourse policies seem to be similar both in terms of the expected costs and the number of vehicles used. However, on instances with higher data uncertainty, the recourse policy N shows some advantages in terms of the quality of the solutions obtained.

References

Baldacci, R., A. Mingozzi, and R. Roberti (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research 59*(5), 1269–1283.

Bertsimas, D. J. (1992). A vehicle routing problem with stochastic demand. *Operations Research 40*(3), 574–585.

Boland, N., J. Dethridge, and I. Dumitrescu (2006). Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters 34*(1), 58–68.

Campbell, A. M. and B. W. Thomas (2008). Probabilistic traveling salesman problem with deadlines. *Transportation Science 42*(1), 1–21.

Chang, T.-S., Y.-W. Wan, and W. T. Ooi (2009). A stochastic dynamic traveling salesman problem with hard time windows. *European Journal of Operational Research 198*(3), 748–759.

Desaulniers, G., F. Lessard, and A. Hadjar (2008). Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science 42*(3), 387–404.

Erera, A. L., M. Savelsbergh, and E. Uyar (2009). Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. *Networks 54*(4), 270–283.

Errico, F., G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau (2013). The vehicle routing problem with hard time windows and stochastic travel times. *Submitted for publication*.

Feillet, D., P. Dejax, M. Gendreau, and C. Gueguen (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks 44*(3), 216–229.

Gendreau, M., G. Laporte, and R. Séguin (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science 29*(2), 143–155.

Irnich, S. and G. Desaulniers (2005). *Shortest Path Problems with Resource Constraints*, Chapter 2, pp. 33–65. GERAD 25th Anniversary Series. Springer.

Jepsen, M., B. Petersen, S. Spoorendonk, and D. Pisinger (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research 56*(2), 497–511.

Jula, H., M. Dessouky, and P. A. Ionnou (2006). Truck route planning in nonstationary stochastic networks with time windows at customer locations. *IEEE Transactions of Intelligent Transportation Systems 7*(1), 51–62.

Kenyon, A. S. and D. P. Morton (2003). Stochastic vehicle routing with random travel times. *Transportation Science 37*(1), 69–82.

Lambert, V., G. Laporte, and F. V. Louveaux (1993). Designing collection routes through bank branches. *Computers & Operations Research 20*(7), 783–791.

Laporte, G., F. V. Louveaux, and L. Hamme (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research 50*(3), pp. 415–423.

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

Lee, C., K. Lee, and S. Park (2012). Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society 60*, 1294–1306.

Lei, H., G. Laporte, and B. Guo (2012). A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. *TOP 20*, 99–118.

Li, X., P. Tian, and S. C. H. Leung (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics 125*(1), 137–145.

Lübbecke, M. E. and J. Desrosiers (2005). Selected topics in column generation. *Operations Research 53*(6), 1007–1023.

Righini, G. and M. Salani (2008). New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks 51*(3), 155–170.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research 35*, 254–265.

Sungur, I., Y. Ren, F. Ordóñez, M. Dessouky, and H. Zhong (2010). A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science 44*(2), 193–205.

Taş, D., N. Dellaert, T. van Woensel, and T. de Kok (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research 40*(1), 214–224.

Wang, X. and C. A. Regan (2001). Assignment models for local truckload trucking problems with stochastic service times and time window constraints. *Transportation Research Record 1771*, 61–68.