European Journal of Operational Research 236 (2014) 789–799

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach

D. Taş [a,*], M. Gendreau [b], N. Dellaert [a], T. van Woensel [a], A.G. de Kok [a]

[a] School of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
[b] CIRRELT and MAGI, École Polytechnique, C.P. 6079, Succursale Centre-ville, Montréal, Québec, Canada H3C 3A7

A R T I C L E I N F O

Article history:
Received 13 September 2012
Accepted 15 May 2013
Available online 25 May 2013

Keywords:
Routing
Stochastic travel times
Column generation
Branch-and-price

A B S T R A C T

We study a vehicle routing problem with soft time windows and stochastic travel times. In this problem, we consider stochastic travel times to obtain routes which are both efficient and reliable. In our problem setting, soft time windows allow early and late servicing at customers by incurring some penalty costs. The objective is to minimize the sum of transportation costs and service costs. Transportation costs result from three elements which are the total distance traveled, the number of vehicles used and the total expected overtime of the drivers. Service costs are incurred for early and late arrivals; these correspond to time-window violations at the customers. We apply a column generation procedure to solve this problem. The master problem can be modeled as a classical set partitioning problem. The pricing subproblem, for each vehicle, corresponds to an elementary shortest path problem with resource constraints. To generate an integer solution, we embed our column generation procedure within a branch-and-price method. Computational results obtained by experimenting with well-known problem instances are reported.

1. Introduction

The Vehicle Routing Problem (VRP), sometimes referred to as capacitated VRP, aims to find a set of feasible routes that start and end at the depot to serve a set of customers. Each customer, given with a known demand, is visited exactly once by one vehicle. Each route can service a total demand that cannot exceed the vehicle capacity. The objective is to minimize the total cost, traditionally derived from the sum of distances traveled or the number of vehicles used or a combination of these. The interested reader is referred to Toth and Vigo [32], and Laporte [22,23] for comprehensive literature surveys about the VRP. This problem is extended by considering different customer service aspects such as starting the service at each customer within a given time interval, called the Vehicle Routing Problem with Time Windows (VRPTW). Time windows are called soft when they can be violated with some penalty costs. They are called hard when violations are not permitted, i.e., vehicles are allowed to wait with no cost if they arrive early and they are prohibited to serve if they arrive late. For reviews on the VRPTW, the reader is referred to Bräysy and Gendreau [4,5], Gendreau and Tarantilis [17], and Kallehauge [21].

In the classical formulation of the VRP, all problem elements are deterministic. However, carrier companies have to deal with various types of uncertainty in real-life applications. Service quality may become quite poor if uncertainties are disregarded at the planning level, since routes may be inefficient or even infeasible in some cases. To overcome the inefficiency incurred at the operational level, stochastic variants of the VRP have been introduced (see Gendreau et al. [16] for a review on stochastic routing problems). Common parameters considered in these variants are stochastic demands, stochastic customers and stochastic travel times. In this research, we study a version of the VRP where we focus on stochastic travel times with a known probability distribution. Using stochastic travel times enables us to construct both reliable and efficient routes. In addition to the cost effectiveness, we also consider customer service aspects where each customer has a soft time window that allows early and late servicing.

The motivation for focusing on the problem described in this paper is that in real-life contexts vehicles operate on a traffic that may be congested, which leads to uncertain travel times. We consider these stochastic travel times at the planning level to accurately evaluate arrivals of vehicles at customer locations by means of stochastic performance measures incorporated in our problem. In practice, these evaluations are carried out with respect to the customers' delivery time intervals. The latter are potentially soft in real-life applications since travel times are uncertain and thus cannot exactly be predicted. Note that customers can often be provided service outside their time windows.

* Corresponding author. Tel.: +31 40 2474388; fax: +31 40 2465949.
E-mail address: d.tas@tue.nl (D. Taş).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.05.024

For our problem, we consider the formulation introduced in Taş et al. [30], where the authors focus on modeling aspects and on solving the problem effectively with a new solution procedure based on metaheuristics. The study conducted in [30] extends existing models, which are generated for stochastic routing problems, by proposing a one-stage formulation in which the objective function copes with time-window violations and expected overtime, and all constraints are linear (see Ando and Taniguchi [1], Russell and Urban [27], and Li et al. [24] for existing models). In this formulation, the objective is to minimize the sum of transportation costs and service costs. Transportation costs result from three elements which are the total distance traveled, the number of vehicles used and the total expected overtime of the drivers. Service costs are incurred for early and late arrivals; these correspond to time-window violations at the customers. Various solution options can be provided to carrier companies by generating different combinations of two main cost components. In our study, we optimally solve this model.

As mentioned above, the formulation introduced by Taş et al. [30] divides the cost components into transportation costs and service costs. It is important to consider this distinction for real-life applications since carrier companies focus on aspects which are usually different from those that concern their customers. In practice, service providers aim to serve all customers by employing the least (operational) cost vehicle routes. For customers, deliveries need to be satisfied by these routes as reliably as possible (within the predefined time windows as promised). The transportation cost component includes true costs paid by carrier companies. Note that by definition, this component is directly related to efficiency of operations. Service costs are obtained by confronting arrival times with soft time windows at customers and thus directly associated with reliability of operations. In this paper, the uncertainty in travel times is taken into account in both cost calculations.

To obtain the optimal solution of our model, we apply a column generation procedure (see Lübbecke and Desrosiers [25], and Desaulniers et al. [9] for comprehensive surveys on column generation). In our procedure, the master problem can be modeled as a classical set partitioning problem. The pricing subproblem, for each vehicle, corresponds to an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). This column generation procedure is embedded within a branch-and-price scheme to obtain integer solutions. Branch-and-price method has proved to be a very successful exact approach for tackling deterministic and stochastic variants of vehicle routing problems. Some applications can be found in Desrochers et al. [11], Fischetti et al. [15], Chabrier [6], Irnich and Villeneuve [19], and Christiansen and Lysgaard [7]. As far as we know, no one has yet studied exact methods to solve the VRP with soft time windows and stochastic travel times. Our paper extends the related literature by providing such a procedure.

The remainder of this paper is organized as follows. The problem and the formulation used in this paper are introduced in Section 2. The column generation procedure and its pricing problem are presented respectively in Sections 3 and 4. Then, we describe our branch-and-price algorithm in Section 5 and report numerical results on Solomon's problem instances [29] in Section 6. Finally, we provide our conclusions in Section 7.

2. Problem description and formulation

Let $G = (N,A)$ be a connected digraph where $N = \{0, 1, \ldots, n\}$ is the set of nodes and A is the set of arcs. In this graph, node 0 denotes the depot, and nodes 1 to n represent customers. A distance d_{ij}, and a travel time T_{ij} with a known probability distribution are defined for each arc (i,j), where $i \neq j$. With each customer $i \in N \backslash \{0\}$ is associated a positive demand q_i, a positive service time s_i, and a soft time window $[l_i, u_i]$ where l_i and u_i are non-negative parameters. Soft time windows enable to serve customers outside their time windows, but some penalty costs must be incurred for the company for early or late servicing. The scheduling horizon for the problem is represented by $[l_0, u_0]$, which is the time window given for the depot. Furthermore, a homogeneous fleet of vehicles of equal capacity (Q) is located at the depot. These vehicles, which belong to set V, are not allowed to wait at customer locations in case of early arrival; service must take place immediately.

In this paper, we focus on the mathematical formulation introduced in Taş et al. [30]. We first summarize the notation used in this formulation in Table 1.

The model, which is solved by applying exact algorithms in our paper, can then be stated as follows:

$$\min \sum_{v \in V} \left[\rho \frac{1}{C_1} \left(c_d \sum_{j \in N} D_{jv}(\mathbf{x}) + c_e \sum_{j \in N} E_{jv}(\mathbf{x}) \right) + (1-\rho) \frac{1}{C_2} \left(c_t \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijv} + c_f \sum_{j \in N \backslash \{0\}} x_{0jv} + c_o O_v(\mathbf{x}) \right) \right] \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in N} \sum_{v \in V} x_{ijv} = 1, \qquad i \in N \setminus \{0\}, \quad (2)$$

$$\sum_{i \in N} x_{ikv} - \sum_{j \in N} x_{kjv} = 0, \qquad k \in N \setminus \{0\}, v \in V, \quad (3)$$

$$\sum_{j \in N} x_{0jv} = 1, \qquad v \in V, \quad (4)$$

$$\sum_{i \in N} x_{i0v} = 1, \qquad v \in V, \quad (5)$$

$$\sum_{i \in N \backslash \{0\}} q_i \sum_{j \in N} x_{ijv} \leqslant Q, \qquad v \in V, \quad (6)$$

$$\sum_{i \in B} \sum_{j \in B} x_{ijv} \leqslant |B| - 1, \qquad B \subseteq N \setminus \{0\}, v \in V, \quad (7)$$

$$x_{ijv} \in \{0, 1\}, \qquad i \in N,\ j \in N,\ v \in V. \quad (8)$$

The objective function (1) minimizes the total weighted cost which has two main components, service costs and transportation costs. Constraints (2) ensure that each customer is visited exactly once. Constraints (3) are the flow conservation constraints at each customer for each vehicle. Constraints (4) and (5) indicate that every vehicle route must start from and end at the depot. Constraints (6) state that the load of each vehicle cannot exceed its capacity. Constraints (7) are subtour elimination constraints, and (8) are the integrality constraints. Parameter ρ is needed to obtain

Table 1
Notation used in the mathematical model.

x_{ijv}	Equal to 1 if vehicle v covers arc (i,j), 0 otherwise
$\mathbf{x}$	Vector of vehicle assignments and customer sequences in these Vehicle routes, where $\mathbf{x} = \{x_{ijv} \mid i,j \in N, v \in V\}$
$D_{jv}(\mathbf{x})$	*Expected* delay at node j when it is served by vehicle v
$E_{jv}(\mathbf{x})$	*Expected* earliness at node j when it is served by vehicle v
$O_v(\mathbf{x})$	*Expected* overtime of the driver working on route of vehicle v
c_d	Penalty cost paid for one unit of delay
c_e	Penalty cost paid for one unit of earliness
c_t	Cost paid for one unit of distance
c_o	Cost paid for one unit of overtime
c_f	Fixed cost paid for each vehicle used for servicing

various combinations of the two main cost components by adjusting their values with scaling parameters C_1 and C_2. The interested reader is referred to Taş et al. [30] for details about these parameters and their calculations.

The above formulation can also be modeled as a Multi-Objective Problem (MOP), which aims to find the complete set of Pareto optimal solutions (referred to as the Pareto frontier). However, obtaining the optimal Pareto frontier may be difficult or even impossible when NP-hard problems are considered, especially for large-sized instances (see Jozefowiez et al. [20]). Moreover, we deal with a complex pricing subproblem due to the stochastic nature of the problem. This would become much harder to handle in case the problem is formulated as a MOP. In this paper, we apply exact methods to our problem by focusing on the model which is proposed as a single-objective optimization problem in Taş et al. [30]. The MOP variant to this problem is left for future research.

To make the problem tractable, it is assumed that the random traversal time spent for one unit of distance follows a suitable Gamma distribution with shape parameter α and scale parameter λ. This approach leads to Gamma distributed arc traversal times where shape parameters are obtained by scaling α with respect to the length of the corresponding arc. Since vehicles do not wait at customer locations, the arrival time of a vehicle at a node along its route can be defined as the sum of travel times on arcs covered by the vehicle until that node. The latter calculation requires an adjustment to the time window at the visited node by taking into account the cumulative service time. Gamma distributed arrival times are then derived where shape parameters are obtained by scaling α with respect to the total length of the arcs covered. These definitions enable us to compute expected delay, expected earliness and expected overtime values exactly by using an approach similar to that given in Dellaert et al. [8]. Note that expected delay and expected earliness values, and thus the total service cost of a route, are computed with respect to the optimal starting time of that route from the depot (see Section 4.1 for the related calculations).

3. Column generation

The interested reader is referred to Desrosiers et al. [12] for details about the column generation method. In the following, we present the master problem and the pricing subproblem of the column generation method proposed for the model presented in Section 2.

The Master Problem: The master problem, which corresponds to constraints (2) in the original model, can be formulated as a set partitioning problem as follows:

$$\min \quad \sum_{p\in P} K_p y_p \tag{9}$$

$$\text{s.t.} \quad \sum_{p\in P} a_{ip} y_p = 1, \quad i \in N\setminus\{0\}, \tag{10}$$

$$y_p \in \{0,1\}, \quad p \in P, \tag{11}$$

where P is the set of all feasible vehicle routes that start from and end at the depot. Here, K_p is the total weighted cost of route p, and a_{ip} is equal to 1 if customer i is served by route p and 0, otherwise. The decision variable y_p is equal to 1 if route p is selected in the solution and 0, otherwise.

The Pricing Subproblem: The pricing subproblem for each vehicle v, which corresponds to constraints (3)–(8) in the original formulation, is an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). Note that in the problem at hand, the only resource is vehicle capacity. The subproblem for a given vehicle v can then be written as follows:

$$\min \quad \overline{K}_p \tag{12}$$

$$\text{s.t.} \quad (3)-(8), \tag{13}$$

where p corresponds to the route of vehicle v and $\overline{K}_p$ is the reduced cost of route p. The latter is computed by:

$$\begin{aligned}\overline{K}_p = K_p - \sum_{i\in N\setminus\{0\}} a_{ip}u_i &= \rho\frac{1}{C_1}\left(c_d\sum_{j\in N} D_{jv}(\mathbf{x}) + c_e\sum_{j\in N} E_{jv}(\mathbf{x})\right)\\ &+ (1-\rho)\frac{1}{C_2}\left(c_t\sum_{i\in N}\sum_{j\in N} d_{ij}x_{ijv} + c_f\sum_{j\in N\setminus\{0\}} x_{0jv} + c_o O_v(\mathbf{x})\right)\\ &- \sum_{i\in N\setminus\{0\}} a_{ip}u_i,\end{aligned} \tag{14}$$

where u_i, $i \in N\setminus\{0\}$ is the dual price associated with covering constraints (10).

In a column generation algorithm, one starts by solving a Restricted Linear Programming Master Problem (RLPMP), in which constraints (11) are relaxed and only the vehicle routes of an initial feasible solution are included. These initial routes constitute a subset of all feasible vehicle routes in the formulation (9)–(11). One then solves the pricing subproblem by using the optimal dual values obtained by solving the RLPMP. If a new vehicle route with negative reduced cost is found by the pricing subproblem, it is added to the RLPMP and this problem is re-optimized to obtain new optimal dual values. Otherwise, the algorithm terminates since an optimal solution of the linear programming relaxation of the formulation (9)–(11) has been found. For the first step of the algorithm, we construct an initial feasible solution by using the initialization algorithm introduced in Taş et al. [30]. This algorithm modifies the I1 insertion heuristic of Solomon [29] by taking into account expected violations of the time windows computed with respect to the stochastic travel times. The interested reader is referred to Taş et al. [30] for details about this initialization algorithm.

4. Elementary shortest path problem with resource constraints

We solve our pricing subproblem with the algorithm of Feillet et al. [13] by applying the state space augmentation (decremental state space relaxation) technique of Boland et al. [3], and Righini and Salani [26]. The algorithm proposed in [13] to solve the ESPPRC is based on the label correcting reaching algorithm of Desrochers [10]. In the latter algorithm, labels are used to denote the partial paths. More precisely, each label at a node represents a path from the depot to that node; it specifies the cost of the path and the consumption of the resources along the path. The label correcting reaching algorithm repeatedly processes the nodes at which some unprocessed labels can be found: each new label is extended to each possible successor node. If the algorithm cannot generate any new labels, then it terminates. Feillet et al. [13] extended this classical label correcting algorithm, which had been developed for the non-elementary shortest path problem with resource constraints, by including node resources to solve the ESPPRC optimally. Beasley and Christofides [2] were the first to propose the idea of adding a binary resource for each node in the graph, but they did not conduct any computational experiments for that. Feillet et al. [13] apply the idea of having a resource associated with each node, where each resource has a limit of one. When a node cannot be visited in any extension of a partial path (either because it has already been visited by that path or because its addition would violate at least one resource constraint), one unit of the corresponding resource is consumed. This consumption results in reaching the resource limit, which makes that node unreachable for the partial path.

In the state space augmentation algorithm, the problem is relaxed where multiple visits are forbidden only for the nodes in a given set $S \subseteq N\backslash\{0\}$. If the optimal solution of the relaxed form of the ESPPRC is elementary, then it is also optimal for the ESPPRC. Otherwise, the state space is augmented by adding to S nodes which appear more than once (i.e., they are not elementary) in the optimal solution of the relaxed problem.

In our problem, a state $\left(W_p^1,\ldots,W_p^R,a_p^S,\mathbf{V}_p^S\right)$ is associated with each path p from the depot to node i. In that state, $\left(W_p^1,\ldots,W_p^R\right)$ represents the consumption of each of the R resources along the path p. a_p^S denotes the number of nodes in S which are unreachable by path p. $\mathbf{V}_p^S$ is the vector of unreachable nodes in S, which is defined by $V_p^b=1$ if node $b \in S$ is unreachable by path p and 0, otherwise. We represent each path p by a label $(L_p, \overline{K}_p)$ where $L_p = \left(W_p^1,\ldots,W_p^R,a_p^S,\mathbf{V}_p^S\right)$ and $\overline{K}_p$ is the reduced cost of path p, which is computed by Eq. (14) with respect to the optimal starting time of path p from the depot (see Section 4.1 for the calculation of the optimal departure time). Let p and p^* be two distinct paths from the depot to node i where each path starts from the depot at the optimal departure time of its corresponding vehicle. In addition, suppose that these two paths arrive at node i at different times (different expected arrival times). In order to obtain a meaningful dominance relationship, the starting time of one path (path p) from the depot must be adjusted to make this path arrive at node i at the same time as the other path (path p^*). The reduced cost of path p computed with respect to this adjusted starting time from the depot is denoted by $\overline{K}_{p_{p^*}}$. The dominance relationship is then defined as follows:

Definition 4.1. If p and p^* are two distinct paths from the depot to node i with labels $(L_p, \overline{K}_p)$ and $(L_{p^*}, \overline{K}_{p^*})$, respectively, then path p dominates path p^* if and only if $W_p^r \leqslant W_{p^*}^r$ for $r = 1, \ldots, R$, $a_p^S \leqslant a_{p^*}^S, V_p^b \leqslant V_{p^*}^b$ for all $b \in S$, $\overline{K}_p \leqslant \overline{K}_{p^*}$, $\overline{K}_{p_{p^*}} \leqslant \overline{K}_{p^*}$ and $(L_p, \overline{K}_p) \neq (L_{p^*}, \overline{K}_{p^*})$.

That is, path p dominates path p^* if (i) it consumes fewer resources for each resource considered, (ii) every unreachable node is also unreachable for path p^*, and (iii) it is less costly when it starts from the depot at its optimal departure time and when it arrives at node i at the same time as path p^*. A path is called efficient if its corresponding label is non-dominated. The method applied to solve our pricing subproblem is described in Algorithm 1. In that algorithm, Π_i, I, and H_{ij} denote the list of labels on node i, the list of nodes that will be treated and the set of labels extended from node i to node j, respectively. Moreover, $EFF(\Pi)$ is the procedure that removes dominated labels and keeps only the non-dominated ones in Π. In our problem, we have only one resource constraint, which is the capacity of vehicles (Q). Therefore, we have only W_p^1 in the labels which corresponds to the consumption of the capacity resource along path p. In addition, w_{ij}^1 represents the consumption of capacity along arc (i,j) which is equal to the demand value at node j (q_j). Formally, the extension of a label at node i to node j is defined in Algorithm 2. In this algorithm, $\pi_{p'}$ represents the resulting label obtained by extending label π_p from node i to node j.

Algorithm 1. Algorithm with state space augmentation technique to solve the ESPPRC

```
S ← ∅
S' ← ∅
repeat
    S ← S ∪ S'
    S' ← ∅
    Π_0 = {(0,0,0,0)}
    forall the i ∈ N \ {0} do
        Π_i ← ∅
    end
    I = {0}
    repeat
        Choose i ∈ I
        forall the (i, j) ∈ A do
            H_ij ← ∅
            forall the π_p = (W_p^1, a_p^S, V_p^S, K̄_p) ∈ Π_i do
                if (j ∉ S) or (j ∈ S and V_p^j = 0) then
                    if Extend(i, π_p, j) ≠ FALSE then
                        H_ij ← H_ij ∪ {Extend(i, π_p, j)}
                    end
                end
            end
            if j ∈ N \ {0} then
                Π_j ← EFF(Π_j ∪ H_ij)
                if Π_j has changed and j ∉ I then
                    I ← I ∪ {j}
                end
            end
        end
        I ← I \ {i}
    until I = ∅;
    Π_0 ← EFF(Π_0)
    if There is at least one elementary path on the depot with
    negative reduced cost then
        Send such paths to the RLPMP
    else
        if The minimum reduced cost is negative then
            Select the customer with the highest multiplicity in the
            solution with the minimum reduced cost
            S' ← {selected customer}
        end
    end
until S' = ∅;
```

Algorithm 2. Extend (i,π_p,j)

```
if W_p^1 + w_ij^1 > Q then
    return FALSE
else
    compute W_p'^1 and K̄_p'
    a_p'^S ← a_p^S
    V_p'^S ← V_p^S
    if j ∈ S then
        a_p'^S ← a_p'^S + 1
        V_p'^j ← 1
    end
    foreach b ∈ S and (j, b) ∈ A such that W_p'^1 + w_jb^1 > Q do
        a_p'^S ← a_p'^S + 1
        V_p'^b ← 1
    end
    return π_p' = (L_p', K̄_p')
end
```

Note that in Algorithm 1, the procedure $EFF(\Pi)$ is applied to the list of paths on the ending depot after all nodes are treated. At this step, the elementary paths on the ending depot which are efficient ones with non-negative reduced costs and dominated ones regardless of their reduced costs are kept in an Intermediate Column Pool (ICP), which is similar to the application of the buffer column pool proposed in Savelsbergh and Sol [28]. Instead of solving the ESPPRC immediately after re-optimizing the RLPMP, we first search the ICP after re-computing column reduced costs with respect to new optimal dual values. The columns with negative reduced costs are then sent from the ICP to the RLPMP. The ESPPRC is solved if we cannot find any such columns in the ICP. At each iteration, we check the size of the ICP to clean it if it is needed. Cleaning takes place when the number of columns in the ICP is larger than a threshold value. In this situation, all columns which have been kept for more than a pre-determined number of iterations are removed from the ICP. This way, we prevent the ICP from becoming too large and, thus, too costly to explore.

In addition to the strategy described above, we apply one more accelerating method which is related to the pricing subproblem in our solution procedure. Each time that we extend a label to the ending depot, we compare this recently added path with all other efficient paths at that node by employing the dominance relation. This way, we determine the number of efficient elementary paths with negative reduced costs at the ending depot. If this number is larger than a threshold value, we stop the ESPPRC and then send all these paths, which are efficient elementary ones with negative reduced costs, to the RLPMP. The interested reader is referred to Feillet et al. [13,14] for similar implementations of this technique.

4.1. Service cost component

For the expected values considered in service cost component, we summarize the related notation in Table 2. The equations given in this table are constructed with respect to the random traversal time which is Gamma distributed with shape parameter α and scale parameter λ. Recall that we have Gamma distributed arc traversal times and Gamma distributed arrival times by means of the definitions provided in Section 2. The interested reader is referred to Taş et al. [30] for further explanations about these definitions.

The expected delay and expected earliness at node j when it is served by vehicle v are then computed as follows when vehicle v departs from the depot at time 0:

$$D_{jv}(\mathbf{x}) = \begin{cases} \alpha_{jv}\lambda_{jv}\left(1-\Gamma_{\alpha_{jv}+1,\lambda_{jv}}\left(u'_j\right)\right) - u'_j(1-\Gamma_{\alpha_{jv},\lambda_{jv}}(u'_j)), & \text{if } u_j > s_{jv} \\ E[Y_{jv}] + s_{jv} - u_j, & \text{otherwise} \end{cases}$$

$$E_{jv}(\mathbf{x}) = \begin{cases} l'_j\Gamma_{\alpha_{jv},\lambda_{jv}}\left(l'_j\right) - \alpha_{jv}\lambda_{jv}\Gamma_{\alpha_{jv}+1,\lambda_{jv}}\left(l'_j\right), & \text{if } l_j > s_{jv} \\ 0, & \text{otherwise} \end{cases}$$

As we already mentioned in Section 2, the total service cost of a path is computed with respect to the optimal starting time of that path from the depot. We determine the optimal departure time of each vehicle from the depot with the Golden Section Search method. The Golden Section Search technique can be applied to find the minimum (or the maximum) value of a unimodal function. In order to be able to use the Golden Section Search method, we need to prove that the total service cost of a path is a unimodal function of its corresponding vehicle's departure time from the depot. Since a convex function is also unimodal, we prove instead that the total service cost component is convex in the following proposition.

Proposition 4.1. *For all routes, the total service cost is a convex function of the corresponding vehicle's departure time from the depot.*

Proof of Proposition 4.1. For a given vehicle v, the total service cost of its route is equal to $\left(c_d\sum_{j\in N}D_{jv}(\mathbf{x}) + c_e\sum_{j\in N}E_{jv}(\mathbf{x})\right)$, where $c_d \geqslant 0$ and $c_e \geqslant 0$. For any node $j \in N$ which is visited by vehicle v, the service cost Z_{jv} is computed by:

$$Z_{jv} = c_d D_{jv}(\mathbf{x}) + c_e E_{jv}(\mathbf{x}). \tag{15}$$

Let y denote the departure time of vehicle v from the depot. Since a sum of convex functions is also a convex function, we need to show that Z_{jv} is convex. Exploiting the fact that Z_{jv} is continuously differentiable, we must therefore show that $\frac{\partial^2 Z_{jv}}{\partial y^2} \geqslant 0$ to prove that the total service cost of the route of vehicle v $\sum_{j\in N} Z_{jv}$ is a convex function of y. We distinguish between three cases:

Case 1. $l'_j - y \geqslant 0$.

Z_{jv} is then computed as follows:

$$\begin{aligned} Z_{jv} &= c_d\alpha_{jv}\lambda_{jv}\left(1-\Gamma_{\alpha_{jv}+1,\lambda_{jv}}\left(u'_j - y\right)\right) \\ &\quad - c_d\left(u'_j - y\right)\left(1-\Gamma_{\alpha_{jv},\lambda_{jv}}\left(u'_j-y\right)\right) + c_e\left(l'_j - y\right)\Gamma_{\alpha_{jv},\lambda_{jv}}\left(l'_j - y\right) \\ &\quad - c_e\alpha_{jv}\lambda_{jv}\Gamma_{\alpha_{jv}+1,\lambda_{jv}}\left(l'_j - y\right). \end{aligned}$$

We know that

$$\Gamma_{\alpha,\lambda}(q) = \frac{1}{\Gamma(\alpha)}\int_0^q \frac{z^{\alpha-1}e^{-z/\lambda}}{\lambda^\alpha}dz = \frac{1}{\Gamma(\alpha)}\int_0^{\frac{q}{\lambda}} t^{\alpha-1}e^{-t}dt = \frac{1}{\Gamma(\alpha)}\gamma\left(\alpha,\frac{q}{\lambda}\right),$$

where $\gamma(\alpha,\frac{q}{\lambda})$ represents the lower incomplete gamma function with parameters α and $\frac{q}{\lambda}$ ($\alpha \geqslant 0$, $q \geqslant 0$ and $\lambda > 0$). The first and second derivatives of this function with respect to q are given as follows:

$$\frac{\partial\gamma(\alpha,\frac{q}{\lambda})}{\partial q} = \frac{1}{\lambda}\left(\frac{q}{\lambda}\right)^{\alpha-1}e^{-\frac{q}{\lambda}} \quad \text{and} \tag{16}$$

$$\frac{\partial^2\gamma(\alpha,\frac{q}{\lambda})}{\partial q^2} = \frac{1}{\lambda^2}\left(\frac{q}{\lambda}\right)^{\alpha-2}e^{-\frac{q}{\lambda}}\left(\alpha-1-\frac{q}{\lambda}\right). \tag{17}$$

Then, $\frac{\partial^2 Z_{jv}}{\partial y^2}$ is computed as follows:

$$\begin{aligned} \frac{\partial^2 Z_{jv}}{\partial y^2} &= -c_d\alpha_{jv}\lambda_{jv}\frac{1}{\Gamma(\alpha_{jv}+1)}\frac{\partial^2\gamma\left(\alpha_{jv}+1,\frac{u'_j-y}{\lambda_{jv}}\right)}{\partial y^2} \\ &\quad + c_d\left(u'_j-y\right)\frac{1}{\Gamma(\alpha_{jv})}\frac{\partial^2\gamma\left(\alpha_{jv},\frac{u'_j-y}{\lambda_{jv}}\right)}{\partial y^2} - 2c_d\frac{1}{\Gamma(\alpha_{jv})} \\ &\quad \times \frac{\partial\gamma\left(\alpha_{jv},\frac{u'_j-y}{\lambda_{jv}}\right)}{\partial y} + c_e\left(l'_j-y\right)\frac{1}{\Gamma(\alpha_{jv})}\frac{\partial^2\gamma\left(\alpha_{jv},\frac{l'_j-y}{\lambda_{jv}}\right)}{\partial y^2} - 2c_e \\ &\quad \times \frac{1}{\Gamma(\alpha_{jv})}\frac{\partial\gamma\left(\alpha_{jv},\frac{l'_j-y}{\lambda_{jv}}\right)}{\partial y} - c_e\alpha_{jv}\lambda_{jv}\frac{1}{\Gamma(\alpha_{jv}+1)} \\ &\quad \times \frac{\partial^2\gamma(\alpha_{jv}+1,\frac{l'_j-y}{\lambda_{jv}})}{\partial y^2}. \end{aligned} \tag{18}$$

Table 2
Notation used for the calculation of service cost component.

$\Gamma(\alpha)$	Gamma function, where $\Gamma(\alpha) = \int_0^\infty e^{-r}r^{\alpha-1}dr$
$\Gamma_{\alpha,\lambda}(\delta)$	Cumulative Gamma distribution function, where $\delta \geqslant 0$ and $\Gamma_{\alpha,\lambda}(\delta)=\int_0^\delta \frac{(e^{-z/\lambda})(z)^{\alpha-1}}{\Gamma(\alpha)\lambda^\alpha}dz$
Y_{jv}	Arrival time of vehicle v at node j
A_{jv}	Set of arcs covered by vehicle v before visiting node j
α_{jv}	Shape parameter of Y_{jv}, where $\alpha_{jv} = \alpha\sum_{(l,k)\in A_{jv}} d_{lk}$
λ_{jv}	Scale parameter of Y_{jv}, where $\lambda_{jv} = \lambda$
s_{jv}	Total service time spent by vehicle v for servicing until node j
u'_j	Upper bound of the time window at node j shifted by s_{jv}, i.e., $u'_j = u_j - s_{jv}$
l'_j	Lower bound of the time window at node j shifted by s_{jv}, i.e., $l'_j = l_j - s_{jv}$
$E[T_{ij}]$	Mean of travel time on arc (i,j), i.e., $E[T_{ij}] = \alpha\lambda d_{ij}$
$E[Y_{jv}]$	Mean of Y_{jv}, when node j is visited immediately after node i, i.e., $E[Y_{jv}] = E[Y_{iv}] + E[T_{ij}]$

By using Eqs. (16) and (17), Eq. (18) can be written as follows:

$$
\begin{aligned}
\frac{\partial^2 Z_{jv}}{\partial y^2} = & -c_d\alpha_{jv}\lambda_{jv}\frac{1}{\Gamma(\alpha_{jv}+1)}\frac{1}{\lambda_{jv}^2}\left(\frac{u'_j-y}{\lambda_{jv}}\right)^{\alpha_{jv}-1}e^{-\left(\frac{u'_j-y}{\lambda_{jv}}\right)}\left(\alpha_{jv}-\left(\frac{u'_j-y}{\lambda_{jv}}\right)\right)\\
& + c_d\left(u'_j-y\right)\frac{1}{\Gamma(\alpha_{jv})}\frac{1}{\lambda_{jv}^2}\left(\frac{u'_j-y}{\lambda_{jv}}\right)^{\alpha_{jv}-2}e^{-\left(\frac{u'_j-y}{\lambda_{jv}}\right)}\left(\alpha_{jv}-1-\left(\frac{u'_j-y}{\lambda_{jv}}\right)\right)\\
& + 2c_d\frac{1}{\Gamma(\alpha_{jv})}\frac{1}{\lambda_{jv}}\left(\frac{u'_j-y}{\lambda_{jv}}\right)^{\alpha_{jv}-1}e^{-\left(\frac{u'_j-y}{\lambda_{jv}}\right)} + c_e\left(l'_j-y\right)\frac{1}{\Gamma(\alpha_{jv})}\\
& \times\frac{1}{\lambda_{jv}^2}\left(\frac{l'_j-y}{\lambda_{jv}}\right)^{\alpha_{jv}-2}e^{-\left(\frac{l'_j-y}{\lambda_{jv}}\right)}\left(\alpha_{jv}-1-\left(\frac{l'_j-y}{\lambda_{jv}}\right)\right) + 2c_e\frac{1}{\Gamma(\alpha_{jv})}\\
& \times\frac{1}{\lambda_{jv}}\left(\frac{l'_j-y}{\lambda_{jv}}\right)^{\alpha_{jv}-1}e^{-\left(\frac{l'_j-y}{\lambda_{jv}}\right)} - c_e\alpha_{jv}\lambda_{jv}\frac{1}{\Gamma(\alpha_{jv}+1)}\\
& \times\frac{1}{\lambda_{jv}^2}\left(\frac{l'_j-y}{\lambda_{jv}}\right)^{\alpha_{jv}-1}e^{-\left(\frac{l'_j-y}{\lambda_{jv}}\right)}\left(\alpha_{jv}-\left(\frac{l'_j-y}{\lambda_{jv}}\right)\right),
\end{aligned}
$$

where $\Gamma(\alpha_{jv} + 1) = \alpha_{jv}\Gamma(\alpha_{jv})$. The above equation leads to:

$$
\begin{aligned}
\frac{\partial^2 Z_{jv}}{\partial y^2} = & c_d\frac{1}{\Gamma(\alpha_{jv})}\frac{\left(u'_j-y\right)^{\alpha_{jv}-1}}{\lambda_{jv}^{\alpha_{jv}}}e^{-\left(\frac{u'_j-y}{\lambda_{jv}}\right)} + c_e\frac{1}{\Gamma(\alpha_{jv})}\\
& \times\frac{\left(l'_j-y\right)^{\alpha_{jv}-1}}{\lambda_{jv}^{\alpha_{jv}}}e^{-\left(\frac{l'_j-y}{\lambda_{jv}}\right)}.
\end{aligned}
$$

So, $\frac{\partial^2 Z_{jv}}{\partial y^2} \geqslant 0$.

Case 2. $u'_j - y \geqslant 0$ and $l'_j - y \leqslant 0$.

Since $l_j \leqslant s_{jv} + y$, $E_{jv}(\mathbf{x})$ is equal to 0. Z_{jv} and its second derivative are then computed as follows:

$$Z_{jv} = c_d\alpha_{jv}\lambda_{jv}\left(1-\Gamma_{\alpha_{jv}+1,\lambda_{jv}}\left(u'_j-y\right)\right) - c_d\left(u'_j-y\right)\left(1-\Gamma_{\alpha_{jv},\lambda_{jv}}\left(u'_j-y\right)\right) \quad \text{and,}$$

$$\frac{\partial^2 Z_{jv}}{\partial y^2} = c_d\frac{1}{\Gamma(\alpha_{jv})}\frac{\left(u'_j-y\right)^{\alpha_{jv}-1}}{\lambda_{jv}^{\alpha_{jv}}}e^{-\left(\frac{u'_j-y}{\lambda_{jv}}\right)}.$$

So, $\frac{\partial^2 Z_{jv}}{\partial y^2} \geqslant 0$.

Case 3. $u'_j - y \leqslant 0$.

In this case, $E_{jv}(\mathbf{x})$ is again equal to 0 since $l_j \leqslant s_{jv} + y$. We then compute Z_{jv} and its second derivative as follows:

$$Z_{jv} = c_d(E[Y_{jv}] + s_{jv} + y - u_j) \quad \text{and,}$$

$$\frac{\partial^2 Z_{jv}}{\partial y^2} = 0.$$

These three cases yield that for any node $j \in N$ which is visited by vehicle v, Z_{jv} is a convex function of the departure time of v from the depot. Therefore, we can conclude that for any vehicle v, the total service cost of its route is a convex function of the departure time of v from the depot.

The above proof allows us to use the Golden Section Search method to compute the optimal departure time of each vehicle from the depot. We consider the departure times over the interval $[g,h]$ where initially $g = l_0$ and $h = u_0$. Note that $[l_0,u_0]$ is the time window at the depot, which corresponds to the scheduling horizon of the problem. An error tolerance fraction (ϵ) is used to terminate the algorithm, which is set to 10^{-6} in our study.

5. Branch-and-price method

To generate an integer solution, we embed our column generation procedure within a branch-and-price method, in which the strategy applied is branching on arcs (see Feillet et al. [13] and Tagmouti et al. [31] for details about this method). Since we have a homogeneous fleet of vehicles of equal capacity, we compute sum of flows (f_{ij}) on each arc (i,j) as follows:

$$f_{ij} = \sum_{v \in V} x_{ijv}. \tag{19}$$

We force arc (i,j) into the solution by setting $\sum_{v\in V} x_{ijv} = 1$, and we exclude arc (i,j) from the solution by imposing $\sum_{v\in V} x_{ijv} = 0$. If we have several fractional flow variables, we choose the arc (i,j) on which the value of f_{ij} is the closest one to the midpoint (0.5). If there are several closest variables, the first one found is chosen.

To make the branching process computationally efficient, all distinctive columns obtained at the root node are stored in a separate pool. Feasible columns with respect to branching rules at a child node are then taken from that pool and used by the column generation algorithm. At each child node, we keep an extra column which serves all customers with a very high total weighted cost. By this way, it is ensured that we have an initial feasible solution at all nodes in the branch-and-price tree.

In this procedure, each problem instance is solved by employing both Depth-First (DF) and Breadth-First (BF) methods in the branch-and-price tree. In DF method, the Upper Bound (UB) of the root node corresponds to the initial feasible solution generated at that node, thus providing a starting value for UB. To proceed into the next level, we select the child node which has the minimum Lower Bound (LB) value. In BF method, we solve an integer programming over all columns obtained by column generation algorithm at the root node. This solution is then assigned as the starting value of UB.

6. Numerical results

We use Solomon's problem instances [29] for testing our exact solution approach based on column generation and branch-and-price. We focus both on problem instances with tight time windows (type 1) and on problem instances with wide time windows (type 2). Recall that the soft time windows in our problem setting, which allow early and late servicing at customers with some penalty costs, do not lead to the inclusion of any resource constraint in the ESPPRC. This structure makes our problem similar to the capacitated VRP, and thus very sensitive to the capacity of vehicles, which is the only resource in the pricing subproblem. We first ran a number of preliminary tests to determine the most appropriate value of the vehicle capacity that enables us to obtain results in a reasonable amount of time. Two sets of preliminary tests were then conducted to determine the most appropriate values for the parameters used by accelerating methods.

In preliminary tests, we first considered the original vehicle capacities given by Solomon [29] where type 1 problem instances have a capacity equal to 200. When using this value, even the optimal solution of the linear programming relaxation of the formulation (9)–(11) at the root node cannot be provided, especially for the problem instances of R1 and RC1 sets with 20 customers, due to the huge amount of CPU time required. Therefore, we iteratively reduced the vehicle capacity by 15 units until the solution of the relaxed problem at the root node could be reached in a reasonable amount of time. According to these preliminary results, we set the capacity of all vehicles to 50.

We then computed the average number of columns obtained by the column generation algorithm at the root node over the problem instances considered to generate preliminary results. This average value is used to have both an efficient and a manageable Intermediate Column Pool (ICP). A threshold value for the size of the ICP is accordingly arranged where the pool contains no larger

Table 3
Results of problem instances in RC set with 20 customers.

Inst.	RootLB	Breadth-first method						Depth-first method					
		RootUB	BestLB	BestUB	CPU	Gap%	Tree	RootUB	BestLB	BestUB	CPU	Gap%	Tree
RC101	2184.44	2196.64	2185.71	2196.64	50.1	0.50	3	2422.11	2186.59	2196.93	725.8	0.47	12
RC102	2178.20	2191.19	2179.78	2190.65	100.7	0.50	5	2412.26	2178.61	2189.38	289.6	0.49	10
RC103	2174.45	2188.98	2175.57	2186.42	68.1	0.50	4	2223.97	2175.57	2186.42	941.1	0.50	15
RC104	2173.48	2188.90	2180.47	2186.26	619.6	0.27	8	2220.13	2175.60	2186.26	2004.7	0.49	19
RC105	2179.18	2191.56	2179.87	2190.59	71.1	0.49	4	2231.68	2180.55	2191.32	1035.9	0.49	15
RC106	2175.17	2189.84	2177.42	2187.59	78.1	0.47	4	2225.12	2178.97	2187.59	1971.5	0.40	16
RC107	2172.33	2187.41	2174.97	2185.29	135.2	0.47	5	2220.75	2174.47	2184.52	1508.4	0.46	16
RC108	2172.11	2188.83	2173.95	2184.04	115.2	0.46	5	2215.73	2174.30	2184.04	818.5	0.45	18
RC201	2216.92	2218.81	2216.92	2218.81	11.0	0.09	0	2315.64	2217.82	2219.63	43.1	0.08	4
RC202	2188.57	2199.02	2188.57	2199.02	6.4	0.48	0	2274.88	2188.57	2199.02	42.8	0.48	5
RC203	2176.18	2190.30	2177.09	2187.52	143.1	0.48	6	2264.96	2177.85	2187.52	493.9	0.44	15
RC204	2175.16	2189.77	2176.06	2186.32	153.0	0.47	6	2242.22	2176.82	2186.32	426.2	0.44	15
RC205	2193.60	2201.14	2193.60	2201.14	10.5	0.34	0	2278.38	2193.86	2202.13	44.3	0.38	6
RC206	2191.51	2200.85	2191.51	2200.85	7.6	0.43	0	2271.94	2192.26	2202.62	37.9	0.47	5
RC207	2182.12	2196.91	2183.35	2193.96	47.2	0.49	3	2236.98	2183.35	2193.96	753.2	0.49	17
RC208	2171.98	2186.15	2173.82	2183.96	161.1	0.47	5	2215.05	2174.09	2183.96	1564.3	0.45	17

Table 4
Results of problem instances in C and R sets with 20 customers.

Inst.	RootLB	Breadth-first method					Depth-first method				
		RootUB	BestLB	BestUB	Gap%	Tree	RootUB	BestLB	BestUB	Gap%	Tree
C101	1681.27	1852.10	1696.46	1852.10	9.17	11	1925.96	1688.76	1886.41	11.70	48
C102	1665.08	1828.85	1666.06	1828.85	9.77	11	1885.19	1665.18	1843.84	10.73	63
C103	1664.75	1828.85	1666.40	1828.85	9.75	10	1870.66	1665.05	1836.54	10.30	58
C104	1664.71	1828.84	1665.74	1828.84	9.79	10	1846.33	1664.81	1832.36	10.06	67
C105	1675.17	1847.31	1682.74	1847.31	9.78	10	1902.05	1677.12	1881.33	12.18	53
C106	1679.18	1849.95	1689.69	1849.95	9.48	11	1916.44	1686.15	1878.39	11.40	45
C107	1672.46	1835.47	1676.08	1835.47	9.51	9	1882.23	1673.58	1862.11	11.26	61
C108	1666.24	1829.23	1669.32	1829.23	9.58	9	1836.62	1667.14	1836.62	10.17	53
C109	1665.51	1828.67	1667.52	1828.67	9.66	10	1832.96	1665.93	1832.96	10.03	73
C201	1781.77	1891.94	1799.48	1891.94	5.14	10	2234.47	1784.12	1922.31	7.75	41
C202	1720.95	1868.89	1727.78	1868.89	8.17	9	2151.94	1723.36	1890.14	9.68	49
C203	1701.86	1862.61	1703.45	1862.61	9.34	9	2059.85	1702.24	1885.02	10.74	61
C204	1693.88	1863.32	1694.91	1863.32	9.94	9	1865.58	1694.02	1865.58	10.13	59
C205	1753.96	1877.77	1763.46	1877.77	6.48	9	2149.62	1757.24	1966.10	11.89	51
C206	1734.51	1862.37	1741.68	1862.37	6.93	9	2086.92	1737.90	1955.14	12.50	52
C207	1717.01	1873.65	1721.28	1873.65	8.85	9	2064.22	1718.32	1884.82	9.69	58
C208	1714.84	1861.50	1717.08	1861.50	8.41	9	2029.32	1714.98	1907.96	11.25	49
R101	1319.43	1437.37	1324.36	1437.37	8.53	7	1474.19	1320.64	1474.19	11.63	65
R102	1300.74	1418.95	1305.57	1418.95	8.68	7	1459.39	1303.30	1452.40	11.44	52
R103	1292.38	1409.51	1294.59	1409.51	8.88	7	1434.82	1293.50	1434.82	10.93	55
R104	1286.63	1403.57	1291.04	1403.57	8.72	7	1429.55	1287.06	1429.55	11.07	68
R105	1303.97	1419.12	1309.47	1419.12	8.37	7	1452.84	1305.13	1452.84	11.32	56
R106	1289.52	1406.26	1295.97	1406.26	8.51	7	1437.22	1289.98	1437.22	11.41	56
R107	1287.86	1405.54	1292.52	1405.54	8.74	7	1430.65	1289.09	1408.12	9.23	53
R108	1284.77	1402.63	1288.48	1402.63	8.86	7	1428.26	1285.08	1428.26	11.14	68
R109	1291.19	1410.28	1295.96	1410.28	8.82	7	1443.29	1293.11	1443.29	11.61	54
R110	1285.22	1404.25	1288.36	1404.25	9.00	7	1440.40	1286.82	1440.40	11.93	62
R111	1286.47	1403.88	1292.00	1403.88	8.66	7	1433.58	1287.53	1425.84	10.74	55
R112	1283.48	1402.61	1285.11	1402.61	9.14	7	1427.53	1284.16	1427.53	11.16	67
R201	1367.54	1485.57	1373.25	1485.57	8.18	6	1522.29	1370.73	1522.29	11.06	71
R202	1316.23	1435.18	1318.74	1435.18	8.83	6	1470.97	1316.79	1470.97	11.71	69
R203	1304.43	1426.40	1307.88	1426.40	9.06	6	1457.82	1304.43	1457.82	11.76	62
R204	1284.61	1402.61	1287.01	1402.61	8.98	6	1425.55	1287.01	1417.64	10.15	49
R205	1334.60	1444.55	1339.86	1444.55	7.81	6	1471.69	1336.32	1471.69	10.13	70
R206	1297.19	1414.87	1301.41	1414.87	8.72	6	1445.43	1298.36	1445.43	11.33	69
R207	1293.59	1412.32	1296.14	1412.32	8.96	6	1441.51	1293.59	1441.51	11.43	74
R208	1284.40	1402.61	1288.52	1402.61	8.85	7	1425.55	1284.84	1425.55	10.95	66
R209	1289.54	1411.14	1293.54	1411.14	9.09	7	1445.48	1290.32	1445.48	12.02	65
R210	1301.79	1420.15	1305.29	1420.15	8.80	6	1451.18	1302.64	1451.18	11.40	65
R211	1283.46	1402.62	1285.26	1402.62	9.13	7	1425.57	1284.53	1425.57	10.98	73

than 70% of the average number of columns observed at the root node. More specifically, we clean the ICP in case the number of columns in that pool is larger than 150. In this situation, all columns that have been kept for more than 15 iterations are removed from the ICP. The latter parameter was set by carrying out a number of experiments in which different number of iterations were tested to obtain the most appropriate value for an effective search in the ICP at each iteration. We then obtained a set of results by testing the threshold value, which prematurely terminates the pricing subproblem, over the interval [0, 15]. According to these tests, we stop the ESPPRC if the number of efficient elementary paths with negative reduced costs at the ending depot is larger than 10. Note that a

Table 5
Results of problem instances in RC set with 25 customers.

Inst.	RootLB	Breadth-first method						Depth-first method					
		RootUB	BestLB	BestUB	CPU	Gap%	Tree	RootUB	BestLB	BestUB	CPU	Gap%	Tree
RC101	2663.82	2688.85	2669.81	2682.70	1524.3	0.48	8	2895.13	2664.57	2687.43	10800.0	0.86	34
RC102	2656.18	2678.14	2663.84	2676.58	1399.7	0.48	8	2885.91	2656.18	2678.37	10800.0	0.84	25
RC103	2652.06	2675.63	2658.32	2670.58	1464.8	0.46	8	2903.37	2652.06	2677.50	10800.1	0.96	29
RC104	2651.93	2675.57	2658.23	2670.48	4321.4	0.46	10	2902.25	2651.93	2670.48	10800.6	0.70	31
RC105	2657.93	2684.02	2664.00	2677.30	1594.5	0.50	8	2886.28	2657.93	2681.08	10800.0	0.87	25
RC106	2651.72	2674.03	2659.46	2673.26	10800.0	0.52	12	2913.33	2651.84	2679.39	10800.3	1.04	29
RC107	2648.38	2672.19	2655.92	2669.05	2113.2	0.49	8	2908.49	2648.42	2674.13	10800.2	0.97	36
RC108	2648.18	2670.92	2654.35	2667.37	1956.2	0.49	8	2901.36	2648.18	2677.44	10800.0	1.10	37
RC201	2708.91	2716.07	2708.91	2716.07	18.5	0.26	0	3050.71	2709.69	2721.21	142.3	0.43	8
RC202	2683.89	2689.12	2683.89	2689.12	17.0	0.19	0	2994.67	2683.89	2689.62	105.7	0.21	5
RC203	2662.52	2674.81	2662.52	2674.81	16.9	0.46	0	2929.48	2662.52	2674.46	3019.5	0.45	19
RC204	2660.69	2674.31	2660.69	2673.94	126.8	0.50	4	2929.48	2660.69	2673.72	549.1	0.49	16
RC205	2686.93	2697.12	2686.93	2697.12	22.3	0.38	0	2983.00	2686.93	2699.54	379.7	0.47	18
RC206	2684.94	2696.37	2684.94	2696.37	15.5	0.43	0	2989.09	2686.64	2693.69	121.4	0.26	8
RC207	2657.66	2683.02	2664.92	2680.39	10800.0	0.58	11	2937.62	2657.66	2683.87	10800.0	0.99	28
RC208	2648.18	2669.69	2654.06	2667.19	2047.8	0.49	8	2902.54	2648.18	2677.90	10800.2	1.12	40

Table 6
Results of problem instances in C and R sets with 25 customers.

Inst.	RootLB	Breadth-first method					Depth-first method				
		RootUB	BestLB	BestUB	Gap%	Tree	RootUB	BestLB	BestUB	Gap%	Tree
C101	2109.32	2287.61	2118.78	2287.61	7.97	9	2379.21	2110.64	2321.27	9.98	61
C102	2095.25	2259.02	2096.23	2259.02	7.77	9	2322.13	2095.47	2279.79	8.80	60
C103	2092.48	2259.02	2094.84	2259.02	7.84	7	2307.60	2092.62	2289.67	9.42	68
C104	2092.12	2259.00	2092.68	2259.00	7.95	8	2276.49	2092.23	2267.49	8.38	75
C105	2104.20	2277.47	2111.60	2277.47	7.86	9	2349.89	2107.12	2336.25	10.87	72
C106	2107.61	2284.53	2118.98	2284.53	7.81	9	2370.29	2111.96	2307.44	9.26	59
C107	2102.62	2265.64	2106.28	2265.64	7.57	9	2322.78	2103.11	2308.22	9.75	74
C108	2096.40	2259.40	2097.92	2259.40	7.70	8	2275.96	2097.30	2275.96	8.52	62
C109	2095.67	2258.83	2097.00	2258.83	7.72	8	2264.00	2096.10	2264.00	8.01	79
C201	2188.29	2345.52	2216.25	2345.52	5.83	8	2630.49	2198.92	2530.86	15.10	54
C202	2148.20	2313.69	2154.18	2313.69	7.40	8	2358.14	2151.05	2341.68	8.86	60
C203	2141.20	2306.83	2143.04	2306.83	7.64	8	2335.17	2141.20	2331.80	8.90	72
C204	2133.85	2302.21	2134.93	2302.21	7.84	8	2304.44	2134.53	2304.44	7.96	82
C205	2166.23	2325.72	2172.42	2325.72	7.06	7	2496.39	2168.52	2446.89	12.84	60
C206	2166.06	2325.61	2168.12	2325.61	7.26	7	2461.36	2166.58	2359.47	8.90	60
C207	2147.02	2313.40	2147.51	2313.40	7.72	7	2534.05	2147.02	2331.65	8.60	52
C208	2146.34	2311.06	2146.63	2311.06	7.66	7	2400.45	2146.41	2361.13	10.00	70
R101	1651.01	1701.87	1655.94	1701.87	2.77	5	1757.57	1653.74	1717.35	3.85	40
R102	1635.08	1692.01	1639.37	1692.01	3.21	6	1730.67	1637.71	1730.67	5.68	52
R103	1622.67	1681.17	1627.55	1681.17	3.29	5	1704.21	1623.44	1700.49	4.75	49
R104	1615.72	1679.70	1619.41	1679.70	3.72	5	1692.17	1618.09	1692.17	4.58	52
R105	1632.57	1689.16	1637.55	1689.16	3.15	6	1726.75	1634.68	1726.75	5.63	45
R106	1619.93	1679.19	1625.56	1679.19	3.30	6	1704.77	1620.22	1704.77	5.22	24
R107	1616.84	1669.92	1619.88	1669.92	3.09	5	1700.07	1617.57	1700.07	5.10	55
R108	1613.40	1672.33	1617.59	1672.33	3.38	5	1690.85	1614.43	1690.85	4.73	23
R109	1618.68	1680.79	1623.98	1680.79	3.50	5	1711.85	1620.22	1710.18	5.55	51
R110	1612.59	1671.69	1616.29	1671.69	3.43	6	1709.76	1613.74	1709.76	5.95	67
R111	1615.66	1673.12	1619.30	1673.12	3.32	5	1697.77	1617.13	1697.77	4.99	54
R112	1610.63	1667.80	1612.15	1667.80	3.45	5	1691.30	1612.14	1675.71	3.94	51
R201	1707.42	1752.16	1710.42	1752.16	2.44	4	1859.22	1709.06	1859.22	8.79	11
R202	1659.86	1710.55	1662.99	1710.55	2.86	4	1784.37	1661.25	1784.37	7.41	12
R203	1645.89	1688.08	1646.99	1688.08	2.49	4	1767.73	1646.23	1767.73	7.38	9
R204	1616.52	1668.16	1616.52	1668.16	3.19	4	1690.64	1616.74	1690.64	4.57	23
R205	1664.37	1723.91	1667.73	1723.91	3.37	5	1787.39	1665.56	1780.33	6.89	66
R206	1635.33	1682.97	1638.80	1682.97	2.70	4	1737.39	1635.33	1737.39	6.24	16
R207	1627.99	1674.31	1630.43	1674.31	2.69	5	1740.86	1629.74	1740.86	6.82	31
R208	1615.72	1669.98	1616.40	1669.98	3.31	5	1689.94	1615.72	1689.94	4.59	10
R209	1621.93	1679.28	1626.96	1679.28	3.22	5	1724.92	1624.22	1724.92	6.20	34
R210	1644.59	1692.09	1646.87	1692.09	2.75	5	1746.88	1645.32	1746.88	6.17	23
R211	1610.58	1670.37	1612.17	1670.37	3.61	5	1690.22	1611.30	1690.22	4.90	36

fixed set of parameters is used to each problem instance solved in this section to see the effect of increasing the problem size (number of customers).

To obtain our computational results, we set $\rho = 0.50$, $C_1 = 1.00$, $C_2 = 1.00$, while $(c_d, c_e, c_t, c_f, c_o)$ are equal to $(1.00, 0.10, 1.00, 400, 5/6)$, respectively (following Taş et al. [30]). The Coefficient of Variation (CV) of the travel time spent for traversing one unit distance is equal to 1.00, where $\alpha = 1.00$ and $\lambda = 1.00$. We have two stopping criteria for our solution procedure. The procedure terminates in case the gap between the best LB and the best UB is smaller than 0.005 (0.5%). In addition, we set a limit for the total CPU time which is equal to 3 hours.

In following tables, "RootLB" and "RootUB" represent the values of the LB and the UB found at the root node of the tree,

Table 7
Results of problem instances in C, R and RC sets with 50 customers.

Inst.	RootLB	Breadth-first method					Depth-first method				
		RootUB	BestLB	BestUB	Gap%	Tree	RootUB	BestLB	BestUB	Gap%	Tree
C101	3981.25	4162.32	3986.67	4162.32	4.41	4	4455.02	3988.35	4455.02	11.70	17
C102	3959.11	4130.22	3959.14	4130.22	4.32	4	4330.52	3959.28	4330.52	9.38	10
C103	3952.71	–	–	–	–	–	4184.55	3952.71	4184.55	5.87	7
C104	3946.34	–	–	–	–	–	4158.11	3946.55	4158.11	5.36	5
C105	3964.07	4140.22	3967.76	4140.22	4.35	5	4398.39	3965.38	4398.39	10.92	22
C106	3968.83	4150.76	3968.83	4150.76	4.58	0	4454.17	3970.02	4454.17	12.20	16
C107	3961.48	4138.15	3961.60	4138.15	4.46	4	4341.27	3961.60	4341.27	9.58	11
C108	3951.19	4125.79	3952.27	4125.79	4.39	3	4292.48	3952.08	4292.48	8.61	12
C109	3948.95	–	–	–	–	–	4266.58	3949.60	4266.58	8.03	4
C201	4066.61	4243.63	4068.51	4243.63	4.30	2	5646.81	4068.90	5646.81	38.78	7
C202	4042.69	4216.51	4042.69	4216.51	4.30	0	4926.45	4043.22	4926.45	21.84	4
C203	4033.78	4199.93	4034.06	4199.93	4.11	2	4450.83	4034.06	4450.83	10.33	1
C204	4020.06	4189.15	4020.20	4189.15	4.20	3	4205.53	4020.25	4205.53	4.61	4
C205	4051.12	4224.86	4051.58	4224.86	4.28	1	5418.65	4051.58	5418.65	33.74	1
C206	4047.85	4218.18	4047.85	4218.18	4.21	0	5147.35	4047.85	5147.35	27.16	1
C207	4044.95	4218.08	4044.95	4218.08	4.28	1	4966.97	4044.95	4966.97	22.79	2
C208	4043.60	4215.95	4043.60	4215.95	4.26	0	4971.79	4043.60	4971.79	22.95	2
R101	3537.25	3631.72	3538.19	3631.72	2.64	2	3684.90	3538.19	3684.90	4.15	5
R102	3503.53	3598.90	3503.53	3598.90	2.72	0	3645.16	3503.88	3645.16	4.03	2
R103	3483.89	3582.74	3483.89	3582.74	2.84	0	3621.07	3484.99	3621.07	3.90	4
R104	3457.50	3554.97	3457.50	3554.97	2.82	0	3634.73	3457.66	3634.73	5.12	2
R105	3504.74	3602.90	3506.25	3602.90	2.76	2	3681.83	3506.25	3681.83	5.01	4
R106	3482.40	3582.84	3482.40	3582.84	2.88	0	3633.09	3484.02	3633.09	4.28	3
R107	3468.94	–	–	–	–	–	3630.53	3470.19	3630.53	4.62	2
R108	3454.54	3552.22	3454.54	3552.22	2.83	0	3606.13	3454.63	3606.13	4.39	1
R109	3470.55	3570.60	3470.55	3570.60	2.88	0	3617.90	3471.18	3617.90	4.23	4
R110	3459.96	3557.78	3460.44	3557.78	2.81	1	3582.07	3460.94	3582.07	3.50	2
R111	3460.12	3558.61	3460.12	3558.61	2.85	0	3620.53	3460.74	3620.53	4.62	2
R112	3452.71	3549.79	3452.71	3549.79	2.81	1	3568.58	3452.74	3568.58	3.36	2
R201	3603.79	3677.91	3603.79	3677.91	2.06	1	3929.10	3603.79	3929.10	9.03	1
R202	3531.58	3621.23	3531.58	3621.23	2.54	0	3815.31	3531.58	3815.31	8.03	1
R204	3453.92	3561.69	3453.92	3561.69	3.12	0	3597.85	3453.92	3597.85	4.17	1
R205	3540.22	3624.99	3540.22	3624.99	2.39	0	3768.11	3540.22	3768.11	6.44	1
R206	3498.43	3597.08	3498.43	3597.08	2.82	0	3710.59	3498.43	3710.59	6.06	1
R207	3470.66	3577.27	3470.66	3577.27	3.07	0	3661.64	3470.70	3661.64	5.50	1
R208	3453.88	3553.29	3453.88	3553.29	2.88	0	3586.21	3453.88	3586.21	3.83	1
R209	3495.34	–	–	–	–	–	3632.68	3495.34	3632.68	3.93	2
R210	3489.15	–	–	–	–	–	3690.16	3489.15	3690.16	5.76	1
R211	3452.69	3548.81	3452.69	3548.81	2.78	0	3574.57	3452.69	3574.57	3.53	1
RC101	4747.43	4878.24	4749.49	4878.24	2.71	6	4952.00	4748.08	4952.00	4.29	112
RC102	4734.82	4868.81	4735.32	4868.81	2.82	6	4936.10	4734.86	4936.10	4.25	57
RC103	4727.63	4858.33	4728.60	4858.33	2.74	5	4911.10	4727.67	4911.10	3.88	39
RC104	4724.19	4855.98	4725.11	4855.98	2.77	5	4941.93	4724.19	4941.93	4.61	26
RC105	4735.41	4867.22	4736.33	4867.22	2.76	5	4956.55	4736.06	4956.55	4.66	47
RC106	4727.51	4858.99	4728.59	4858.99	2.76	5	4954.15	4728.16	4954.15	4.78	32
RC107	4721.59	4853.52	4723.14	4853.52	2.76	5	4948.95	4723.13	4948.95	4.78	21
RC108	4720.13	4852.08	4721.21	4852.08	2.77	5	4908.66	4721.06	4908.66	3.97	19
RC201	4842.82	4964.44	4846.72	4964.44	2.43	4	5271.42	4843.23	5271.42	8.84	16
RC202	4763.64	4894.51	4763.92	4894.51	2.74	4	5071.99	4763.64	5071.99	6.47	16
RC203	4737.49	4870.78	4738.49	4870.78	2.79	5	4951.02	4737.49	4951.02	4.51	24
RC204	4724.75	4857.19	4725.37	4857.19	2.79	5	4965.04	4725.16	4965.04	5.08	22
RC205	4788.26	4913.72	4789.21	4913.72	2.60	4	5094.98	4789.30	5094.98	6.38	13
RC206	4770.63	4893.91	4771.75	4893.91	2.56	5	5099.22	4771.34	5099.22	6.87	24
RC207	4741.09	4867.96	4742.00	4867.96	2.66	5	5017.94	4741.09	5017.94	5.84	17
RC208	4720.13	4852.08	4721.21	4852.08	2.77	5	4935.06	4721.06	4935.06	4.53	24

Table 8
Results of problem instances in C sets with 100 customers.

Inst.	RootLB	Breadth-first method					Depth-first method				
		RootUB	BestLB	BestUB	Gap%	Tree	RootUB	BestLB	BestUB	Gap%	Tree
C101	8549.79	8747.36	8549.79	8747.36	2.31	0	9403.58	8549.79	9403.58	9.99	0
C105	8512.86	–	–	–	–	–	9263.38	8512.86	9263.38	8.82	1

respectively. ‘‘BestLB’’ and ‘‘BestUB’’ indicate the best LB and the best UB values obtained over the tree, respectively. The percentage of the gap between the best LB and the best UB, and the size of the tree in terms of the highest level reached are also reported. We present the CPU times in seconds for only the problem instances in RC set with 20 and 25 customers since the algorithm stops by means of the limit given for the gap before it reaches the time limit for some of the instances. For all problem instances in C and R sets, the algorithm stops due to the time limit. The results of the problem instances are not reported in case the column generation

Table 9
Results of problem instances in C, R and RC sets with 100 customers obtained by DF method with 8 hour CPU limit.

Inst.	RootLB	RootUB	BestLB	BestUB	Gap%	Tree
C101	8549.79	9403.58	8550.86	9403.58	9.97	7
C102	8494.70	9104.29	8494.70	9104.29	7.18	2
C103	8472.20	8878.68	8472.20	8878.68	4.80	2
C105	8512.86	9263.38	8514.10	9263.38	8.80	8
C106	8498.13	9217.98	8501.87	9217.98	8.42	2
C107	8497.82	9140.80	8499.49	9140.80	7.55	2
C108	8481.48	9036.47	8481.48	9036.47	6.54	1
C109	8472.73	8883.03	8472.73	8883.03	4.84	1
C201	8559.23	10480.70	8559.23	10480.70	22.45	1
R109	6830.88	7104.53	6830.88	7104.53	4.01	0
RC101	8380.00	8790.81	8380.00	8790.81	4.90	0
RC102	8346.21	8545.24	8346.21	8545.24	2.38	0
RC103	8325.13	8503.30	8325.13	8503.30	2.14	0
RC106	8329.07	8536.55	8329.07	8536.55	2.49	0
RC107	8311.85	8491.95	8311.85	8491.95	2.17	0

Table 10
Average results of problem instances in C, R and RC sets with 20, 25, 50 and 100 customers obtained by BF and DF methods with 3 hour CPU limit.

Set	Method	Ave. Gap%	Set	Method	Ave. Gap%
C1–20	BF	9.61	C1–20	DF	10.87
C2–20	BF	7.91	C2–20	DF	10.45
R1–20	BF	8.74	R1–20	DF	11.14
R2–20	BF	8.77	R2–20	DF	11.18
RC1–20	BF	0.46	RC1–20	DF	0.47
RC2–20	BF	0.40	RC2–20	DF	0.40
C1–25	BF	7.80	C1–25	DF	9.22
C2–25	BF	7.30	C2–25	DF	10.15
R1–25	BF	3.30	R1–25	DF	5.00
R2–25	BF	2.97	R2–25	DF	6.36
RC1–25	BF	0.49	RC1–25	DF	0.92
RC2–25	BF	0.41	RC2–25	DF	0.55
C1–50	BF	4.42	C1–50	DF	9.07
C2–50	BF	4.24	C2–50	DF	22.78
R1–50	BF	2.80	R1–50	DF	4.27
R2–50	BF	2.71	R2–50	DF	5.63
RC1–50	BF	2.76	RC1–50	DF	4.40
RC2–50	BF	2.67	RC2–50	DF	6.07
C1–100	BF	2.31	C1–100	DF	9.40

Table 11
Average results of problem instances in C, R and RC sets with 100 customers obtained by DF method with 8 hour CPU limit.

Set	Method	Ave. Gap%	Set	Method	Ave. Gap%
C1–100	DF	7.26	C2–100	DF	22.45
R1–100	DF	4.01	RC1–100	DF	2.82

algorithm cannot obtain the optimal solution for the root node within the CPU limit. Additionally, we have no result for the problem instances if the integer programming cannot be solved at the root node within the CPU limit when we apply BF method.

Our algorithms are implemented in Visual C++, and linear programming models in our solution approach are solved by IBM ILOG CPLEX 12.2 [18]. We run all experiments on an Intel Core Duo with 2.93 GHz and 4 GB of RAM.

The results given in Tables 3–8 indicate that our solution approach with BF method provides better results than those obtained by DF method in terms of the gap between the best LB and the best UB. However, solutions of six instances with 50 customers and one instance with 100 customers, which can be provided by DF method, cannot be obtained by applying BF method due to the huge amount of CPU time or memory that it requires to solve the integer programming at the root node. Since we do not have such an obstacle in DF method, we solve the problem instances with 100 customers where the limit for the total CPU time is set to 8 hours. The related results that show how far we can go with regard to the level of the branch-and-price tree are provided in Table 9. In these results, the highest level reached in the tree is relatively small due to the size of the problem instances. The effect of this situation is seen in the UB which cannot be improved within the time limit.

The average gap values between the best LB and the best UB for the sets with 20, 25, 50 and 100 customers provided by BF and DF methods, where the CPU limit is equal to 3 hours, are presented in Table 10. Table 11 shows the results of the sets with 100 customers where the applied strategy is DF method and the CPU limit is equal to 8 hours.

7. Conclusions

In this paper, we have considered a vehicle routing problem with soft time windows and stochastic travel times. For this problem, we proposed an exact solution approach based on column generation algorithm and branch-and-price method. To solve the pricing subproblem in column generation procedure, we extended an existing elementary shortest path algorithm with resource constraints by introducing a new dominance relation and by applying a state space augmentation technique. Moreover, two separate methods were implemented for searching in the branch-and-price tree. Our numerical study was performed on well-known problem instances.

The results indicate that our solution approach can effectively be used to solve the model, in which the aim is to construct both reliable and efficient routes, for medium- and large-sized problem instances. Even though our pricing subproblem is rather complex due to the stochasticity, we have an effective column generation algorithm. Finally, future research should focus on time-dependent and stochastic formulations, that we have not studied in this paper.

References

[1] N. Ando, E. Taniguchi, Travel time reliability in vehicle routing and scheduling with time windows, Networks and Spatial Economics 6 (2006) 293–311.
[2] J.E. Beasley, N. Christofides, An algorithm for the resource constrained shortest path problem, Networks 19 (1989) 379–394.
[3] N. Boland, J. Dethridge, I. Dumitrescu, Accelerated label setting algorithms for the elementary resource constrained shortest path problem, Operations Research Letters 34 (2006) 58–68.
[4] O. Bräysy, M. Gendreau, Vehicle routing problem with time windows, part I: route construction and local search algorithms, Transportation Science 39 (2005) 104–118.
[5] O. Bräysy, M. Gendreau, Vehicle routing problem with time windows, part II: metaheuristics, Transportation Science 39 (2005) 119–139.
[6] A. Chabrier, Vehicle routing problem with elementary shortest path based column generation, Computers & Operations Research 33 (2006) 2972–2990.
[7] C.H. Christiansen, J. Lysgaard, A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands, Operations Research Letters 35 (2007) 773–781.
[8] N.P. Dellaert, A.G. de Kok, W. Wei, Push and pull strategies in multi-stage assembly systems, Statistica Neerlandica 54 (2000) 175–189.
[9] G. Desaulniers, J. Desrosiers, M.M. Solomon, Column Generation, Springer, New York, 2005.
[10] M. Desrochers, An Algorithm for the Shortest Path Problem with Resource Constraints, Technical Report G-88-27, GERAD, 1988.
[11] M. Desrochers, J. Desrosiers, M.M. Solomon, A new optimization algorithm for the vehicle routing problem with time windows, Operations Research 40 (1992) 342–354.
[12] J. Desrosiers, Y. Dumas, M.M. Solomon, F. Soumis, Time constrained routing and scheduling, in: M. Ball, T. Magnanti, C. Monma, G. Nemhauser (Eds.), Network Routing, Handbooks in Operations Research and Management Science, vol. 8, North-Holland, Amsterdam, 1995, pp. 35–139.
[13] D. Feillet, P. Dejax, M. Gendreau, C. Gueguen, An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems, Networks 44 (2004) 216–229.
[14] D. Feillet, M. Gendreau, L.-M. Rousseau, New refinements for the solution of vehicle routing problems with branch and price, INFOR 45 (2007) 239–256.

[15] M. Fischetti, P. Toth, D. Vigo, A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs, Operations Research 42 (1994) 846–859.
[16] M. Gendreau, G. Laporte, R. Séguin, Stochastic vehicle routing, European Journal of Operational Research 88 (1996) 3–12.
[17] M. Gendreau, C.D. Tarantilis, Solving Large-Scale Vehicle Routing Problems with Time Windows: The State of Art, Technical Report 2010-04, CIRRELT, 2010.
[18] IBM, ILOG CPLEX Optimizer 12.2, 2012. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>.
[19] S. Irnich, D. Villeneuve, The shortest-path problem with resource constraints and k-cycle elimination for $k \geqslant 3$, INFORMS Journal on Computing 18 (2006) 391–406.
[20] N. Jozefowiez, F. Semet, E.-G. Talbi, Parallel and hybrid models for multi-objective optimization: application to the vehicle routing problem, in: J. Guervos, P. Adamidis, H.-G. Beyer, J.-L. Fernández-Villacanas, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature (PPSN VII), Granada, Spain, LNCS, vol. 2439, Springer-Verlag, Berlin, Heidelberg, 2002, pp. 271–280.
[21] B. Kallehauge, Formulations and exact algorithms for the vehicle routing problem with time windows, Computers & Operations Research 35 (2008) 2307–2330.
[22] G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms, European Journal of Operational Research 59 (1992) 345–358.
[23] G. Laporte, What you should know about the vehicle routing problem, Naval Research Logistics 54 (2007) 811–819.
[24] X. Li, P. Tian, S.C.H. Leung, Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm, International Journal of Production Economics 125 (2010) 137–145.
[25] M.E. Lübbecke, J. Desrosiers, Selected topics in column generation, Operations Research 53 (2002) 1007–1023.
[26] G. Righini, M. Salani, New dynamic programming algorithms for the resource constrained elementary shortest path problem, Networks 51 (2008) 155–170.
[27] R.A. Russell, T.L. Urban, Vehicle routing with soft time windows and Erlang travel times, Journal of the Operational Research Society 59 (2008) 1220–1228.
[28] M. Savelsbergh, M. Sol, Drive: dynamic routing of independent vehicles, Operations Research 46 (1998) 474–490.
[29] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35 (1987) 254–265.
[30] D. Taş, N. Dellaert, T. van Woensel, T. de Kok, Vehicle routing problem with stochastic travel times including soft time windows and service costs, Computers & Operations Research 40 (2013) 214–224.
[31] M. Tagmouti, M. Gendreau, J.-Y. Potvin, Arc routing problems with time-dependent service costs, European Journal of Operational Research 181 (2007) 30–39.
[32] P. Toth, D. Vigo, The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, vol. 9, Philadelphia, 2002.