A Vehicle Routing Problem with Flexible Time Windows

Duygu Taş [a,b,*], Ola Jabali [b], Tom Van Woensel [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 HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, H3T 2A7, Montréal, Canada

Available online 18 July 2014

1. Introduction

Carrier companies are faced with the daily challenge of delivering goods to customers in a cost-effective manner. Often, these companies must adhere to customer service requirements. In this environment, customer service requirements are mainly reflected by the Vehicle Routing Problem with Time Windows (VRPTW). This problem can be observed in bank deliveries, postal deliveries, and school bus routing (see Hashimoto et al. [19]). Given a set of customers, the VRPTW consists of finding least cost routes such that each customer is visited within a predetermined time window by a single vehicle. Furthermore, a vehicle must deliver a quantity not exceeding its capacity, the vehicle should also start and end its route at a given depot. The vehicle is permitted to arrive before the opening of the time window, and wait at no cost until service becomes possible, but it is not permitted to arrive after the time window closes (see Bräysy and Gendreau [4]).

The definition of the VRPTW implies that time windows are treated as hard constraints, the relaxation of which may lead to reducing the total distance traveled while using fewer vehicles. A form of time window relaxation is considered in the Vehicle Routing Problem with Soft Time Windows (VRPSTW). This problem assumes that some or all customer time windows are soft and can be violated by paying appropriate penalties (see Balakrishnan [1]). The penalty structure associated with soft time windows essentially allows serving a customer at any point of the planning horizon. This mechanism is due to the penalty policies, which dictate that early arriving vehicles must wait or incur a penalty, while any late arrival is permissible at a cost. Therefore, when compared to the VRPTW, the VRPSTW operates on a much larger feasible solution space.

In several real-world situations, time window constraints can be violated to a certain extent. Therefore, in this paper we aim to assess the operational gains obtained by employing a fixed relaxation of the time window constraints. Namely, we study the Vehicle Routing Problem with Flexible Time Windows (VRPFlexTW), in which vehicles are allowed to deviate from customer time windows by a given tolerance. This flexibility enables savings in the operational costs of carriers, since customers may be served before and after the earliest and latest time window bounds, respectively. As time window violations affect customers satisfaction, they are penalized. Furthermore, as in the VRPTW we allow early arriving vehicles to wait at no cost until the earliest allowable service time is reached. The VRPFlexTW is distinct from the VRPSTW in that the former considers a restriction on the feasible time window violation. Therefore, when compared to the VRPSTW, the VRPFlexTW operates on a smaller feasible solution space.

The main contributions of this paper are threefold:

1. We introduce and model the VRPFlexTW.
2. To produce high-quality solutions, we develop an effective solution procedure which comprises three phases: (i) initialization, (ii) improving, (iii) scheduling.
3. We conduct a series of numerical experiments on benchmark instances, and assess the operational gains of using flexible time windows.

* Corresponding author at: School of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands.

E-mail address: d.tas@tue.nl (D. Taş).

The remainder of this paper is organized as follows. The relevant literature is reviewed in Section 2. Section 3 introduces the model. The solution procedure is then described in Section 4. This is followed by computational results provided in Section 5 and by conclusions given in Section 6.

2. Literature review

The daily distribution task faced by many freight transports is captured by the Vehicle Routing Problem (VRP). In its classical definition, the VRP minimizes the total travel cost incurred by a set of homogeneous vehicles that deliver customer demands. Each customer is to be visited by a single vehicle, each vehicle starts and ends its route at a depot and delivers a quantity not exceeding its capacity. The VRP has been widely studied for over 50 years (see, e.g., Laporte [24]). In an attempt to better link the VRP to realistic applications, a number of extensions have been proposed in the literature (see, e.g., Toth and Vigo [32], and Golden et al. [18]). One of the most extensively studied variants of the VRP is the VRPTW, in which time windows ensure that each customer must be visited within a given interval. Over the years, a number of exact and heuristic solution procedures have been proposed for the VRPTW. Bräysy and Gendreau [4] review the literature on route construction and local search algorithms for the VRPTW. The authors also survey metaheuristics for the same problem (see Bräysy and Gendreau [5]). Baldacci et al. [3] provide a recent review of mathematical formulations, relaxations and exact methods for the VRPTW.

The VRPTW treats time windows as hard constraints. However, some practical applications imply that customer time windows can be treated as soft constraints, i.e., may be violated at a cost. This setting gives rise to the VRPSTW, which is significantly less studied than the VRPTW. The VRPSTW considers the existence of time windows, however it assumes that customers are available at any moment in time to receive their goods. Therefore, vehicles incur penalty costs for time window violations. As such, the VRPSTW is a special case of the VRPFlexTW, where the relaxation of time windows is unbounded, i.e., infinite flexible bounds. The majority of the literature on the VRPSTW considers a linear penalty function for time window deviations. Balakrishnan [1] develops three heuristics for the VRPSTW based on the nearest neighbor Clarke–Wright savings and space–time rules. Koskosidis et al. [23] propose a heuristic algorithm for the VRPSTW. Their algorithm decomposes the problem into an assignment component and a series of routing and scheduling components. Min [25] considers the VRPSTW for a single vehicle where the problem is solved for small-sized instances. Taillard et al. [30] propose a tabu search heuristic to solve the VRPSTW. Calvete et al. [6] consider a general medium-sized VRPSTW and propose a goal programming model. Aside from minimizing the operational cost and time window violations, the authors consider avoiding underutilization of vehicles and labor. The solution approach first computes feasible routes and then selects the set of best ones.

Ibaraki et al. [21] propose an efficient algorithm to deal with general time window constraints. The cost function considered for time window violations can be non-convex and discontinuous as long as it is piecewise linear. Furthermore, one or more time slots can be assigned to each customer. Building upon the model proposed in [21], Hashimoto et al. [19] define the travel time as a variable representing the difference between the starting times of services at two consecutive customers, and introduce its cost function.

A main difference of our model and those presented in the recent papers (e.g., Fagerholt [12], Chiang and Russell [7], Fu et al. [14], Figliozzi [13]) lies in the way of allowing time window violations. More specifically, our problem has a different structure in which vehicles are allowed to provide both early and late services with a penalty limit, and to wait in case of early arrival (until the enlarged lower bound) without any waiting time limit. Note that the most similar structure employed in [12] with a waiting time limit defined for the whole route (not separately for each customer) whereas the authors focus on a multi-ship pickup and delivery problem. Another algorithmic difference of our method between the recent algorithms is the application of a schedule optimization phase. In our paper, we apply a scheduling method in two ways: (i) as a last phase to improve the solution generated by the tabu search method and (ii) every iteration in the tabu search method as an improvement step. Moreover, we compare the computational results provided by our solution procedure for the classical VRPTW not only with the solutions obtained by the existing (meta)heuristic algorithms but also with the optimal/best-known solutions.

One of the underlying assumptions in VRPSTW is that the deviations from time windows are essentially unbounded, implying that any feasible VRP solution is feasible in the VRPSTW as well. The VRPFlexTW proposed in this paper bounds the lower and upper time window deviations, and hence allows a predetermined amount of flexibility in adhering to time windows. Qureshi et al. [26,27] develop a column generation based exact algorithm for the Vehicle Routing and scheduling Problem with Semi Soft Time Windows (VRPSSTW). This problem considers an upper bound on the tardiness time window deviation, and thus may be viewed as a special case of the VRPFlexTW. The solution approach is shown to be efficient on medium-sized instances. Tang et al. [31] study the VRP with fuzzy time windows where the authors consider the multi-objective problem of minimizing travel time and maximizing customer service level, similar to the VRPTW. The authors take into account a limited allowable deviation from time windows and solve their multi-objective model with a two-stage algorithm which yields Pareto solutions.

3. Model formulation

Formally, the VRPFlexTW can be represented by a connected digraph $G=(N,A)$ where $N=\{0,1,\dots,n,n+1\}$ is the set of nodes and $A=\{(i,j)|i,j\in N, i\neq j\}$ is the set of arcs. Nodes 0 and $n+1$ correspond to the starting and ending nodes of each route, respectively (the central depot). Let $C=N\backslash\{0,n+1\}$ denote the set of customers. For each customer $i\in C$, we have a positive demand q_i, a time window $[l_i,u_i]$ and fractions p_i^l and p_i^u which are used to set the maximum allowed violations, leading to the flexible time window. The time window at the depot, $[l_0,u_0]$ (or equivalently $[l_{n+1},u_{n+1}]$), corresponds to the feasible scheduling horizon for each vehicle route. For each node i, a flexible time window $[l_i',u_i']$ is generated with respect to the length of the original time window, where $l_i'=l_i-p_i^l(u_i-l_i)$, and $u_i'=u_i+p_i^u(u_i-l_i)$. Additionally, Q represents the capacity given for each vehicle $v\in V$ where V denotes a homogeneous fleet.

Associated with each arc $(i,j)\in A$, t_{ij} and d_{ij} represent the travel time and the distance along that arc, respectively. Note that the service time at node i, z_i is included in t_{ij}. A fixed cost c_f is incurred for using a vehicle. Time window violations, i.e., serving a customer within $[l_i',l_i]$ or $[u_i,u_i']$ are penalized by c_e and c_d for one unit of earliness and one unit of delay, respectively. Moreover, c_t is the cost paid for one unit of distance. In the early servicing case, service at the customer starts between the flexible earliest time and the original earliest time. In the late servicing case, service takes place between the original latest time and the flexible latest time. Note that vehicles wait at customers (at least) until the flexible time window is reached if they arrive early, and they cannot serve after the customer flexible time window closes. Following the commonly used assumption in the classical VRPTW, we

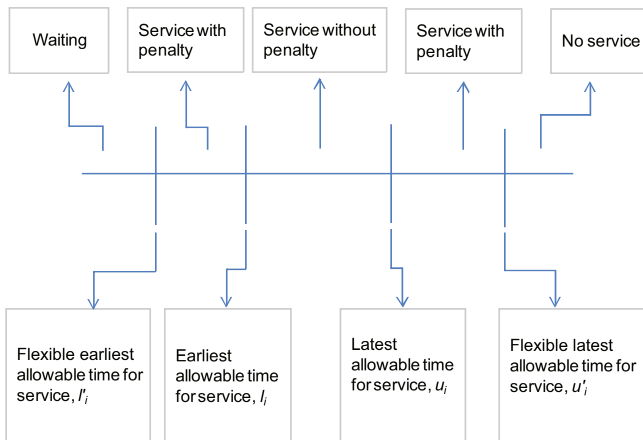

Fig. 1. Possible arrivals and their corresponding penalty cases at any customer i.

assume that waiting in the VRPFlexTW brings no penalty cost. The latter assumption enables vehicles to wait at customer locations even if they arrive within the flexible time windows, and to generate cost-efficient routes. In this way, a balance between early and late servicing is provided. Fig. 1 depicts the possible arrivals and their corresponding penalty cases at customers.

Given the previously mentioned definition, the mathematical model for the VRPFlexTW is formulated as follows:

$$\min \quad c_t \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} d_{ij} x_{ijv} + c_f \sum_{j \in C} \sum_{v \in V} x_{0jv} + c_e \sum_{i \in N} \sum_{v \in V} e_{iv} + c_d \sum_{i \in N} \sum_{v \in V} h_{iv} \tag{1}$$

subject to

$$\sum_{j \in N} \sum_{v \in V} x_{ijv} = 1, \quad i \in C, \tag{2}$$

$$\sum_{i \in N} x_{ikv} - \sum_{j \in N} x_{kjv} = 0, \quad k \in C, \ v \in V, \tag{3}$$

$$\sum_{i \in C} q_i \sum_{j \in N} x_{ijv} \le Q, \quad v \in V, \tag{4}$$

$$\sum_{j \in N \setminus \{0\}} x_{0jv} = 1, \quad v \in V, \tag{5}$$

$$\sum_{i \in N \setminus \{n+1\}} x_{i,n+1,v} = 1, \quad v \in V, \tag{6}$$

$$s_{iv} + t_{ij} - u'_0(1 - x_{ijv}) \le s_{jv}, \quad i \in N, \ j \in N, \ v \in V, \tag{7}$$

$$l'_i \le s_{iv} \le u'_i, \quad i \in N, \ v \in V, \tag{8}$$

$$e_{iv} \ge l_i - s_{iv}, \quad i \in N, \ v \in V, \tag{9}$$

$$h_{iv} \ge s_{iv} - u_i, \quad i \in N, \ v \in V, \tag{10}$$

$$e_{iv} \ge 0, \quad i \in N, \ v \in V, \tag{11}$$

$$h_{iv} \ge 0, \quad i \in N, \ v \in V, \tag{12}$$

$$x_{ijv} \in \{0, 1\}, \quad i \in N, \ j \in N, \ v \in V. \tag{13}$$

In our model, x_{ijv} is equal to 1 if vehicle v serves node j immediately after node i and 0, otherwise. s_{iv} denotes the time that vehicle v starts serving node i. Furthermore, e_{iv} and h_{iv} represent the earliness and the delay at node i in case it is served by vehicle v, respectively. The objective (1) is to minimize the total cost which consists of traveling costs, fixed costs of vehicles used for service, and penalty costs incurred for early and late servicing. The constraints (2) and (3) guarantee that exactly one vehicle arrives at each customer location and leaves it. The constraints (4) ensure that the vehicle capacity is not exceeded. The constraints (5) and (6) indicate that each vehicle route starts and terminates at the depot. The constraints (7) represent the relationship between the starting time of service at a customer and the departure time of vehicle from its predecessor. In this set of constraints, we employ the upper bound of the flexible time window at the central depot which is sufficiently large for our problem, i.e., $u'_0 \ge \max_{i \in C} u'_i$. The constraints (8) ensure that the service takes place at each customer with respect to the customer (flexible) time window. The constraints (9) link the earliness and the beginning of service; similarly, the constraints (10) link the delay and the beginning of service. The constraints (11) and (12) ensure that we have non-negative values for the earliness and the delay. The constraints (13) indicate that there is no partial servicing.

4. Solution methodology

For the VRPFlexTW formally described above, we propose a solution procedure that includes three main phases. In the first phase, an initial feasible solution is constructed. This solution is then improved by applying a tabu search metaheuristic in the second phase. These two stages lead to the assignments of vehicles and the sequences of customers in these assignments. In the first and second phases, vehicles are allowed to wait at customer locations only in case they arrive early (until the flexible time window is reached). If a vehicle arrives at a customer within its flexible time window, then service takes place without waiting. The latter situation leads to the immediate service with some penalty costs. In the third phase, the solution obtained by the tabu search algorithm is further improved by solving a Linear Programming (LP) model. This phase calculates the optimal starting time of each vehicle route from the depot, and optimal times that each vehicle should start serving the customers in its route. The objective function of the LP minimizes the total penalty cost of the vehicle route with respect to the sequence of customers given in that route.

4.1. Initial feasible solution

We apply the time-oriented nearest neighbor heuristic proposed by Solomon [28] to generate our initial routes. This heuristic first finds the customer closest to the depot where the closeness is defined by a function. At each iteration, it searches for the customer which is (i) not yet covered, (ii) feasible to be inserted, (iii) closest to the last customer in the current route. The feasibility is checked with respect to the flexible time window at the customer considered, the flexible time window at the depot and the capacity of the vehicle. If we cannot find any feasible customer for the current vehicle route, a new partial route is then initialized by inserting the customer closest to the depot. This procedure terminates when all customers are assigned to a vehicle.

We now describe the function used in the time-oriented nearest neighbor heuristic briefly explained above. Suppose that we have a partial route for vehicle v in which the last demand location is node i, and node j is any node that can be visited by that vehicle after node i. Following Solomon [28], we use a function to calculate the cost value of node j in case it is visited by vehicle v immediately after node i. This function uses three measures to evaluate the insertion of node j (with respect to the last node) into the partial route. The first measure is the distance between node i and node j. The other two measures are formally given as follows:

$$T_{jv} = s_{jv} - (s_{iv} + z_i), \tag{14}$$

and

$$r_{jv} = u'_j - (s_{iv} + t_{ij}). \tag{15}$$

In Eq. (15), r_{jv} can be thought of as the urgency of the delivery served by vehicle v to customer j. s_{jv}, which is the time that vehicle v starts serving customer j, is calculated by

$$s_{jv} = \max\{l'_j, s_{iv} + t_{ij}\}. \tag{16}$$

The cost value is then calculated by the following equation:

$$c_{jv} = \beta_1 d_{ij} + \beta_2 T_{jv} + \beta_3 r_{jv}, \tag{17}$$

where $\beta_1 \geq 0$, $\beta_2 \geq 0$, $\beta_3 \geq 0$ and $\beta_1 + \beta_2 + \beta_3 = 1$ (following Solomon [28]).

4.2. The tabu search for the VRPFlexTW

The tabu search metaheuristic has been extensively applied to the classical VRP and its extensions, such as stochastic VRP and VRPTW. Some tabu search methods proposed for the VRP are given by Gendreau et al. [16,17]. The interested reader is referred to Garcia et al. [15], Taillard et al. [30], Hertz et al. [20] and Cordeau et al. [10] for implementations of tabu search in the VRPTW. Our tabu search algorithm is based on the work of Taş et al. [29]. In this study, the authors focus on a VRP with stochastic travel times and soft time windows. Travel times on each arc are given with a known probability distribution. Soft time windows allow both early and late servicing with some penalty costs. The latter enables vehicles to start serving a customer after its time window closes. Moreover, vehicles do no wait at customer locations, leading to immediate service regardless of the arrival time. We adapt this algorithm to our problem where we have deterministic travel times and flexible time windows. Each time window is relaxed with given fractions p^l_i and p^u_i for each customer i. The flexible time boundaries can be thought of as the hard time windows defined in the classical VRPTW where vehicles are not permitted to serve before or after these intervals.

The overall tabu search procedure is described in pseudo-code as Algorithm 1. In this algorithm, y and $z(y)$ denote the current solution and its total cost value, respectively. The latter value is obtained by Algorithm 2 which calculates the total cost value of each route m in solution y. In this algorithm, penalties incurred due to violations of the original time windows are computed with respect to the Earliest Possible beginning of Service (EPS) heuristic. This heuristic allows waiting at a customer only if the assigned vehicle arrives before the flexible time window is reached. If that vehicle arrives within the customer's flexible time window, then it starts serving immediately. Note that in the tabu search method, a solution y is only taken into consideration in case all routes in this solution are feasible with respect to the flexible time window constraints.

The neighborhood of the current solution y, which is represented by $g(y)$, is generated by two types of operators: (i) changing the location of the customer within the route, (ii) removing the customer from a route and locating it into another route. Each solution y' in $g(y)$ is evaluated with respect to $c(y')$, which is calculated as follows:

$$c(y') = z(y') + \nu q(y'), \tag{18}$$

where $q(y')$ is the total demand of routes in solution y' exceeding the vehicle capacity and ν is the cost paid for one unit of excess demand. This calculation is operated in line with Cordeau et al. [10] and Taş et al. [29]. The parameter ν is adjusted after each iteration with respect to the total demand of routes in the current solution. If the current solution is feasible with respect to the vehicle capacity constraint, the value of ν is divided by $(1+\varphi)$; otherwise it is multiplied with $(1+\varphi)$.

In Eq. (18), an additional cost is added to $c(y')$ of any solution y' such that $c(y') \geq c(y)$. This mechanism provides diversification during the search. In our paper, a similar function to that given in [29] is applied to calculate additional costs. Note that the latter function uses a constant parameter (μ) to calibrate the intensity of the diversification. In [29], expected overtime of drivers working on each vehicle route is also considered to compute the total cost since there is no limit on the total route duration. This cost structure is different from that considered in our paper, and thus affects the diversification part of the algorithm (which highly depends on the total cost of routes).

In Algorithm 1, we apply two criteria following Taş et al. [29] where the algorithm terminates either because it reaches the maximum number of given tabu search iterations (θ) or because the best feasible solution did not change for a threshold number of iterations (τ). To identify the solutions that are tabu, we employ a list which includes the customers forbidden to relocate for a number of iterations. Note that the size of this list is denoted by ϑ.

The interested reader is referred to [29] for the details about medium-term memory applied in Algorithm 1, which is based on directing the search to the promising regions of the neighborhood generated by the best feasible solution. The medium-term memory, which contributes to a good solution quality in [29], is applied in this paper with a different criterion. In case the best feasible solution has not been improved for $\alpha\sqrt{\kappa}$ iterations, then the algorithm returns back to that solution to search the promising regions in the corresponding neighborhood. The value of α is determined by performing a number of preliminary experiments (see Section 5.1). Implementing the parameter α and tuning its value with respect to the sensitivity analyses make our algorithm different from that developed in [29] in which α is set to 1.0 without carrying out any parameter calibration procedure. As another difference from [29], vehicles are allowed to wait in case of early arrivals. This framework enables us to employ the EPS heuristic to compute the penalties incurred due to time window violations.

Algorithm 1. The tabu search algorithm for the VRPFlexTW.

```
Construct initial feasible solution, y_init
Set y := y_init, y* := y_init and z(y*) := z(y_init)
Set κ := 1, stop := 0
while κ ≤ θ and stop = 0 do
    Choose the first solution y' ∈ g(y) which is better than y and is
    not tabu or satisfies the aspiration criterion
    if Such a solution cannot be found then
        Choose a solution y' ∈ g(y) that minimizes c(y') value and is
        not tabu
    end
    if y' is feasible and z(y') < z(y*) then
        Set y* := y' and z(y*) := z(y')
    end
    if y* is not updated for α√κ iterations then
        Set y := y* and c(y) := c(y*)
        Update the tabu list accordingly
    end
    else
        Set y := y' and c(y) := c(y')
    end
    if q(y) = 0 then
        ν := ν/(1 + φ)
    end
    else if q(y) > 0 then
        ν := ν*(1 + φ)
    end
    if y* is not updated for τ iterations then
        Set stop := 1
    end
    Set κ := κ + 1
end
return y*
```

Algorithm 2. EPS heuristic to calculate $z(y)$.

```
foreach route m in solution y do
    foreach customer j in route m do
        Calculate the arrival time a_jv with respect to previous customer i,
        where a_jv = s_iv + t_iv
        if a_jv < l'_j then
            Set s_jv := l'_j and calculate earliness
        end
        if l'_j ≤ a_jv < l_j then
            Set s_jv := a_jv and calculate earliness
        end
        if l_j ≤ a_jv < u_j then
            Set s_jv := a_jv (no penalty)
        end
        if u_j ≤ a_jv < u'_j then
            Set s_jv := a_jv and calculate delay
        end
    end
    Calculate the total cost of route m by using its total distance,
    total earliness and total delay, and the cost of vehicle v
end
Calculate the total cost of solution y, z(y)
```

In Algorithm 2, v denotes the vehicle that operates route m. Moreover, a_{jv} and s_{jv} represent the arrival time and the beginning of service of vehicle v at node j, respectively.

4.3. Scheduling method

Recall that in the first and second phases of our solution methodology, service takes place immediately in case a vehicle arrives at a customer within its flexible time window. In the third phase, we solve the following LP model to obtain the optimal times that each vehicle starts serving the customers in its route. The interested reader is referred to Vidal et al. [33] for a classification of problems and methods addressing the scheduling and timing perspectives. The model presented below is operated for each vehicle v which is used for service in the solution generated by the tabu search algorithm. In this model, $A' \subseteq A$ is the ordered set of arcs traversed by vehicle v and $N' \subseteq N$ is the set of nodes visited in the route of that vehicle.

$$\min \quad c_e \sum_{i \in N'} e_{iv} + c_d \sum_{i \in N'} h_{iv} \tag{19}$$

$$\text{subject to} \quad s_{iv} + t_{ij} \le s_{jv}, \quad i \in N', \ j \in N', \ (i,j) \in A', \tag{20}$$

$$l'_i \le s_{iv} \le u'_i, \quad i \in N', \tag{21}$$

$$e_{iv} \ge l_i - s_{iv}, \quad i \in N', \tag{22}$$

$$h_{iv} \ge s_{iv} - u_i, \quad i \in N', \tag{23}$$

$$e_{iv} \ge 0, \quad i \in N', \tag{24}$$

$$h_{iv} \ge 0, \quad i \in N', \tag{25}$$

where the objective is to minimize the total penalty cost of the route operated by vehicle v.

Our solution procedure with three phases in which the scheduling method is applied in the last stage yields computational results given in Sections 5.2–5.6. Additionally, we perform the scheduling method every iteration in the tabu search algorithm as an improvement step. The latter implementation leads to a solution approach with two phases and its corresponding computational results are presented in Section 5.7. Note that, a scheduling method is applied only as a post-optimization phase in Taş et al. [29] since the stochastic problem is rather complicated.

5. Numerical results and insights

We perform our computational experiments on the well-known data sets given by Solomon [28]. We consider 29 problem instances with 100 customers and tight time windows (sets R1, C1 and RC1). Each instance has one depot as the central location of the homogeneous fleet of vehicles, where the vehicle capacity Q is 200 units. We set the cost coefficients (c_t, c_f, c_e, c_d) to (2.0, 400, 0.5, 1.0). For each node $i \in N$, we employ a symmetric violation with a fraction p_i ($p_i^l = p_i^u = p_i$) which is set to 0.05.

Following [28], we use four parameter sets given in Table 1 to generate the Initial Feasible Solution (IFS) for each problem instance. Among the four solutions constructed by the initialization algorithm, we select the solution with the minimum total cost value calculated with respect to the EPS heuristic, and we set the IFS as the selected solution. The algorithms proposed in our solution procedure are coded in JAVA and the LP model is solved by using IBM ILOG CPLEX 12.5 [22]. All experiments are conducted on an Intel(R) Xeon(R) CPU X5675 with 12-core 3.07 GHz and 96 GB of RAM (by using a single thread).

The rest of this section is organized as follows. Section 5.1 provides the analysis for determining the parameters of the algorithm. Further experimentation with the stopping criteria is presented in Section 5.2. Section 5.3 highlights the operational advantages of the VRPFlexTW compared to the VRPTW. Further comparison with the VRPSTW is presented in Section 5.4. Section 5.5 evaluates the solutions obtained by our solution algorithm with respect to those obtained by solving the formulation (1)–(13) within CPLEX [22]. Section 5.6 analyzes the effects of implementing larger flexibility fractions with different cost coefficients. Finally, applying the scheduling method at every iteration in the algorithm is presented in Section 5.7.

5.1. Sensitivity analyses and parameters for the tabu search

A number of tests are performed to tune the parameters employed in the tabu search algorithm. For the parameters μ, ϑ and φ, we apply a similar procedure to that given in Cordeau et al. [8] and that applied in Taş et al. [29]. In the preliminary experiments, the original VRPTW and the VRPFlexTW are considered separately since the former problem does not include any flexibility for time windows. To determine the most appropriate value of a parameter, we test its different values over an interval by keeping the other parameters unchanged. In our preliminary experiments, three main sets of results are obtained for each problem considered where the parameters μ, ϑ and φ are examined in [0.005,0.025], [2.5 $\log_{10}|N|$, 12.5 $\log_{10}|N|$] and [0.25,1.25], respectively. According to the results of the preliminary tests, for the original VRPTW we set the values of μ, ϑ and φ to 0.010, 2.5 $\log_{10}|N|$, and 0.75, respectively. For the VRPFlexTW, we set the values of μ, ϑ and φ to 0.020, 5 $\log_{10}|N|$, and 0.50, respectively.

In addition to the above procedure, we perform a number of tests to tune the parameter α used in the medium-term memory application. According to results of the preliminary tests, for the original VRPTW the value of α is set to 1.50. For the VRPFlexTW, we set the value of α to 2.00.

Table 1
Parameters used by the initialization algorithm to generate the IFS.

Tests	β_1	β_2	β_3
Test 1	0.4	0.4	0.2
Test 2	0.0	1.0	0.0
Test 3	0.5	0.5	0.0
Test 4	0.3	0.3	0.4

Table 2
Details of initial feasible solutions obtained for all problem instances where $p_i = 0.05,\ \forall i \in N$.

Ins.	Initial feasible solution				
	Del.	Dist.	Earl.	# Veh.	Obj.
C101	0.00	855.07	7.30	10	5713.78
C102	35.39	1263.54	45.70	11	6985.31
C103	159.93	1530.61	96.20	11	7669.26
C104	226.46	1789.19	45.00	11	8227.33
C105	0.00	934.36	0.00	10	5868.71
C106	6.32	1164.45	0.00	10	6335.23
C107	14.72	1033.36	11.52	10	6087.20
C108	2.60	1038.91	4.20	10	6082.52
C109	10.31	1357.11	167.55	11	7208.30
R101	0.09	1948.05	60.20	19	11,526.29
R102	63.01	1816.10	42.30	19	11,316.36
R103	135.35	1601.99	29.00	14	8953.83
R104	121.20	1387.88	14.00	12	7703.96
R105	0.15	1706.48	75.03	17	10,250.64
R106	64.38	1736.24	30.66	14	9152.19
R107	119.97	1464.20	45.53	12	7871.13
R108	86.87	1301.18	33.50	11	7105.97
R109	9.47	1487.61	90.35	13	8229.86
R110	29.17	1470.03	46.76	13	8192.61
R111	72.83	1500.50	68.65	12	7908.16
R112	55.27	1390.12	36.44	11	7253.73
RC101	1.65	2020.20	73.37	17	10,878.73
RC102	66.69	1891.19	54.00	14	9476.07
RC103	77.29	1803.36	43.20	13	8905.62
RC104	89.29	1735.92	39.75	12	8381.01
RC105	12.18	2081.24	61.75	16	10,605.55
RC106	5.50	1664.48	45.34	13	8557.13
RC107	20.88	1610.98	83.95	12	8084.82
RC108	17.62	1507.50	35.73	12	7850.48

In the tabu search algorithm, we adjust the value of the parameter ν at each iteration, which is the cost incurred for one unit of excess demand. As a reasonable starting value for this unit penalty cost, the initial value of the parameter ν is set to 1.

Recall that in the tabu search algorithm, we employ two stopping criteria represented by θ (primary criterion) and τ (secondary criterion). We obtain three sets of results by applying three different sets of values for stopping criteria, in which (θ,τ) are set to $(10^4,10^3)$, $(10^4, 5(10^3))$ and $(10^5, 2.5(10^4))$. In the next subsection, we present our results and discuss their details.

5.2. Results on stopping criteria

Table 2 provides the solutions generated by the initialization algorithm for each problem instance. As previously mentioned, the total cost of a solution involves traveling costs, fixed costs paid for vehicles used, and penalty costs incurred for early and late servicing (earliness and delay). In Table 2, we report the total delay (Del.), total distance (Dist.), total earliness (Earl.), number of vehicles activated for the service (#Veh.), and the objective function value (Obj.). Since the computational time spent by the initialization algorithm is almost 0 for each instance, we do not report this value in Table 2.

Tables 3, 4 and 5 show the solutions obtained by the tabu search algorithm, and the corresponding final solutions obtained by solving an LP model in the scheduling method. In these tables, values of (θ,τ) are equal to $(10^4,10^3)$, $(10^4, 5(10^3))$ and $(10^5, 2.5(10^4))$, respectively. For the tabu search algorithm, we report the CPU times in seconds and the improvement in total cost values in percentages (Obj. Imp%), which is calculated with respect to the IFS. For the LP model, improvement in total penalty cost incurred

Table 3
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ,τ) are $(10^4,10^3)$ and $p_i = 0.05,\ \forall i \in N$.

Ins.	Solution of the tabu search algorithm							Final solution		
	Del.	Dist.	Earl.	# Veh.	Obj.	CPU	Obj. Imp (%)	Del.	Earl.	Pen. Imp (%)
C101	0.00	828.94	0.00	10	5657.87	17	0.98	0.00	0.00	0.00
C102	0.00	953.11	10.80	10	5911.62	29	15.37	0.00	0.00	100.00
C103	0.00	934.71	18.50	10	5878.67	100	23.35	0.00	0.00	100.00
C104	0.00	994.59	41.90	11	6410.12	55	22.09	0.00	0.00	100.00
C105	0.00	828.94	0.00	10	5657.87	29	3.59	0.00	0.00	0.00
C106	0.00	828.94	0.00	10	5657.87	35	10.69	0.00	0.00	0.00
C107	0.00	828.94	0.00	10	5657.87	30	7.05	0.00	0.00	0.00
C108	0.00	828.94	0.00	10	5657.87	30	6.98	0.00	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	51	21.51	0.00	0.00	0.00
R101	0.36	1669.34	63.20	19	10,970.64	43	4.82	0.36	1.10	97.15
R102	0.52	1498.14	46.47	18	10,220.04	70	9.69	0.52	1.18	95.33
R103	0.39	1317.05	23.27	14	8246.12	36	7.90	0.39	0.50	94.67
R104	0.74	1064.94	15.00	12	6938.13	37	9.94	0.74	0.00	90.98
R105	0.00	1430.86	78.75	15	8901.09	43	13.17	0.00	0.62	99.22
R106	3.61	1286.05	54.00	13	7802.72	85	14.74	3.61	9.73	72.30
R107	9.58	1106.46	39.50	12	7042.24	49	10.53	9.58	3.75	60.96
R108	0.04	966.93	21.00	10	5944.41	54	16.35	0.04	2.45	87.96
R109	2.95	1207.36	76.33	13	7655.84	46	6.97	2.95	12.70	77.38
R110	5.20	1163.48	63.37	13	7563.85	34	7.67	5.20	8.33	74.61
R111	0.03	1060.37	65.58	11	6553.55	74	17.13	0.03	14.76	77.42
R112	3.47	998.45	26.16	10	6013.44	69	17.10	3.47	7.09	57.64
RC101	2.10	1782.00	72.56	16	10,002.39	47	8.06	2.10	4.32	88.89
RC102	5.22	1570.16	38.08	13	8364.58	83	11.73	5.49	5.44	66.17
RC103	10.26	1297.73	49.52	12	7430.48	94	16.56	10.26	4.52	64.26
RC104	4.12	1190.63	23.40	11	6797.08	52	18.90	4.12	3.00	64.47
RC105	4.21	1583.10	54.76	15	9197.80	40	13.27	4.21	5.94	77.27
RC106	1.50	1442.94	72.40	13	8123.59	40	5.07	1.50	7.41	86.19
RC107	10.89	1252.13	84.85	11	6957.58	50	13.94	10.89	18.20	62.50
RC108	3.99	1203.51	72.34	11	6847.19	28	12.78	3.99	6.69	81.73
Avg.	2.39	1170.61	38.34	12.17	7231.74	50.00	12.00	2.40	4.06	64.73

Table 4
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ,τ) are $(10^4,5(10^3))$ and $p_i=0.05$, $\forall i \in N$.

Ins.	Solution of the tabu search algorithm							Final solution		
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Obj. Imp (%)	Del.	Earl.	Pen. Imp (%)
C101	0.00	828.94	0.00	10	5657.87	44	0.98	0.00	0.00	0.00
C102	0.00	838.04	15.12	10	5683.65	104	18.63	0.00	0.00	100.00
C103	0.00	864.64	0.00	10	5729.29	189	25.30	0.00	0.00	0.00
C104	0.00	875.33	30.10	10	5765.71	171	29.92	0.00	0.00	100.00
C105	0.00	828.94	0.00	10	5657.87	130	3.59	0.00	0.00	0.00
C106	0.00	828.94	0.00	10	5657.87	149	10.69	0.00	0.00	0.00
C107	0.00	828.94	0.00	10	5657.87	139	7.05	0.00	0.00	0.00
C108	0.00	828.94	0.00	10	5657.87	139	6.98	0.00	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	151	21.51	0.00	0.00	0.00
R101	0.36	1669.34	63.20	19	10,970.64	139	4.82	0.36	1.10	97.15
R102	0.57	1517.50	43.27	17	9857.20	209	12.89	0.57	0.82	95.58
R103	0.00	1274.87	24.35	14	8161.91	190	8.84	0.00	0.00	100.00
R104	0.81	1029.86	10.00	11	6465.53	191	16.08	0.81	0.50	81.81
R105	2.15	1420.29	79.80	15	8882.63	195	13.35	2.15	7.04	86.51
R106	2.41	1292.15	37.18	13	7805.30	182	14.72	3.16	3.58	76.45
R107	0.30	1108.13	32.86	11	6633.00	181	15.73	0.30	3.75	87.00
R108	2.55	984.24	24.50	10	5983.28	178	15.80	2.55	4.28	68.30
R109	5.88	1168.69	87.81	13	7587.18	189	7.81	5.88	15.27	72.84
R110	2.12	1099.59	65.39	12	7034.01	171	14.14	2.12	10.85	78.32
R111	2.46	1077.80	56.59	12	6986.35	160	11.66	2.46	5.83	82.54
R112	5.16	1001.40	54.20	10	6035.06	147	16.80	5.16	3.10	79.20
RC101	4.92	1644.31	71.72	16	9729.40	212	10.56	4.92	7.12	79.21
RC102	10.53	1589.89	40.60	13	8410.61	220	11.24	10.80	9.13	50.18
RC103	16.01	1288.75	45.00	12	7416.01	187	16.73	16.01	7.50	48.69
RC104	5.06	1170.77	16.61	10	6354.90	176	24.18	5.06	3.00	50.92
RC105	2.53	1561.52	71.65	15	9161.39	203	13.62	2.53	7.90	83.11
RC106	1.96	1349.07	63.13	12	7531.67	181	11.98	1.96	7.67	82.71
RC107	0.33	1237.61	95.88	12	7323.49	121	9.42	0.33	4.69	94.45
RC108	12.76	1130.88	64.57	10	6306.81	180	19.66	12.76	16.97	52.83
Avg.	2.72	1143.73	37.71	11.97	7095.25	166.48	13.61	2.75	4.14	60.27

Table 5
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ,τ) are $(10^5,2.5(10^4))$ and $p_i=0.05$, $\forall i \in N$.

Ins.	Solution of the tabu search algorithm							Final solution		
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Obj. Imp (%)	Del.	Earl.	Pen. Imp (%)
C101	0.00	828.94	0.00	10	5657.87	87	0.98	0.00	0.00	0.00
C102	0.00	838.04	15.12	10	5683.65	492	18.63	0.00	0.00	100.00
C103	0.00	864.64	0.00	10	5729.29	629	25.30	0.00	0.00	0.00
C104	18.95	848.98	17.70	10	5725.77	1468	30.41	18.95	0.00	31.83
C105	0.00	828.94	0.00	10	5657.87	637	3.59	0.00	0.00	0.00
C106	0.00	828.94	0.00	10	5657.87	709	10.69	0.00	0.00	0.00
C107	0.00	828.94	0.00	10	5657.87	593	7.05	0.00	0.00	0.00
C108	0.00	828.94	0.00	10	5657.87	539	6.98	0.00	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	503	21.51	0.00	0.00	0.00
R101	0.36	1669.34	63.20	19	10,970.64	570	4.82	0.36	1.10	97.15
R102	0.07	1509.70	43.27	17	9841.11	726	13.04	0.07	0.22	99.17
R103	3.24	1232.08	23.09	14	8078.95	1539	9.77	3.24	0.48	76.45
R104	12.19	994.97	10.00	10	6007.14	1644	22.03	12.19	1.21	25.55
R105	1.06	1370.79	91.90	15	8788.59	1395	14.26	1.81	4.97	90.87
R106	1.13	1240.71	54.00	13	7709.56	1413	15.76	1.13	5.07	86.96
R107	0.26	1091.73	22.85	11	6595.14	1079	16.21	0.26	3.75	81.74
R108	2.48	952.87	18.00	10	5917.22	786	16.73	2.48	4.28	59.73
R109	3.35	1163.73	74.93	13	7568.28	1069	8.04	3.35	8.10	81.87
R110	7.12	1073.11	67.56	11	6587.13	1456	19.60	7.12	11.22	68.87
R111	2.46	1077.80	56.59	12	6986.35	460	11.66	2.46	5.83	82.54
R112	0.00	967.98	39.81	10	5955.86	673	17.89	0.00	5.41	86.41
RC101	6.73	1629.99	64.19	15	9298.79	1190	14.52	7.14	8.17	71.08
RC102	17.04	1493.60	54.59	13	8231.53	1078	13.13	17.51	5.32	54.51
RC103	17.89	1265.89	34.29	11	6966.82	775	21.77	17.89	7.29	38.54
RC104	5.06	1152.08	14.67	10	6316.55	1428	24.63	5.06	3.00	47.07
RC105	2.36	1538.94	50.78	15	9105.63	1321	14.14	2.36	8.44	76.30
RC106	1.79	1341.22	64.88	12	7516.66	777	12.16	1.79	9.25	81.26
RC107	15.58	1194.74	100.15	11	6855.14	775	15.21	15.58	28.25	54.76
RC108	12.93	1113.39	85.45	10	6282.43	1989	19.97	12.93	19.29	59.43
Avg.	4.55	1124.14	36.79	11.79	6988.46	958.62	14.84	4.61	4.85	53.52

for early and late services is given in percentages (Pen.Imp%) and this value is calculated with respect to the solution obtained by the tabu search algorithm. Moreover, we present the average values (Avg.) over all problem instances provided both by the tabu search algorithm and by the scheduling method.

Solutions in Table 3 show that in all problem instances the tabu search algorithm decreases the total distance with respect to the initial solutions. These reductions may be achieved by an increase in the delay or in the earliness, e.g., we have higher total delay in two instances (R101, RC101) and higher total earliness in ten instances (R101, R102, R104, R105, R106, R110, RC103, RC106, RC107, RC108) compared to that given by the IFS. Moreover, we use fewer vehicles in 16 problem instances. Overall, the tabu search algorithm reduces the total cost by 12.00% on average. The solutions obtained by the tabu search algorithm are further improved with the LP model by reducing an average of 64.73% of the total penalty cost. The percentage of the total penalty cost in the objective function value on average is 0.92% over the initial solutions, 0.28% over the solutions obtained by the tabu search algorithm, and 0.06% over the final solutions obtained by the scheduling method. More specifically, the tabu search algorithm provides a reduction both in the total delay (on average from 51.88 min to 2.39 min) and in the total earliness (on average from 47.83 min to 38.34 min). The scheduling method further reduces the total earliness on average to 4.06 min with a slight increase in the total delay (on average to 2.40 min).

Solutions in Table 4, which are obtained with a higher number of iterations given for the secondary terminating criterion, show that in all problem instances the tabu search algorithm decreases the total distance with respect to the initial solutions. We have higher total delay in three instances and higher total earliness in 11 instances. Moreover, we use fewer vehicles in 19 instances. Overall, the tabu search algorithm reduces the total cost by 13.61% on average, which is higher than the average value generated with a smaller number of threshold iterations to terminate the algorithm ($\tau = 1000$). The solutions obtained by the tabu search algorithm are further improved with the scheduling method by 60.27% on average according to total penalty cost. On average, the percentage of the total penalty cost in the objective function value is 0.29% over the solutions obtained by the tabu search algorithm, and 0.07% over the final solutions obtained by the scheduling method. The tabu search algorithm reduces the values of the total delay and the total earliness on average to 2.72 min and to 37.71 min, respectively. The scheduling method further decreases the total earliness on average to 4.14 min with a slight increase in the total delay (on average to 2.75 min).

In the last set of results, the tabu search algorithm operates with higher numbers of iterations both for the primary (θ) and the secondary (τ) terminating criteria. In all problem instances, the tabu search algorithm decreases the total distance with respect to the initial solutions. We have higher total delay in three instances and higher total earliness in ten instances. Moreover, we use fewer vehicles in 20 instances. Overall, the tabu search algorithm reduces the total cost by 14.84% on average, which is higher than the average value given in Table 4. The scheduling method yields a 53.52% improvement on average, which is smaller than the value given in Table 4, since the tabu search algorithm leads to better improvements for most of the problem instances. The percentage of the total penalty cost in the objective function value on average is 0.31% over the solutions obtained by the tabu search algorithm, and 0.10% over the final solutions obtained by the scheduling method. More specifically, the tabu search algorithm reduces the values of the total delay and the total earliness on average to 4.55 min and to 36.79 min, respectively. The scheduling method further decreases the total earliness on average to 4.85 min with a slight increase in the total delay (on average to 4.61 min).

The results given by Tables 3, 4 and 5 indicate that using a higher number of iterations for the primary and secondary terminating criteria in the tabu search procedure yields better final solutions. Moreover, these solutions are obtained in a reasonable amount of time. Thus, in the following experiments where different fraction values are employed, (θ,τ) are set to $(10^5, 2.5(10^4))$.

5.3. VRPFlexTW versus VRPTW

The aim of this subsection is to evaluate the benefits gained by flexible time windows compared to the hard time windows. Table 6 provides the optimal/best-known solutions for the original VRPTW (see Desaulniers et al. [11] and Baldacci et al. [2]) and the solutions of the VRPFlexTW. This table represents the total distance and the number of vehicles for the following cases: (i) the optimal/best-known solutions of the original VRPTW, (ii) the final solutions obtained by our solution procedure for the original VRPTW (VRPFlexTW with $p_i = 0,\ \forall i \in N$), (iii) the final solutions obtained by our solution procedure for the VRPFlexTW with $p_i = 0.05,\ \forall i \in N$, (iv) the final solutions obtained by our solution procedure for the VRPFlexTW with $p_i = 0.10,\ \forall i \in N$, (v) the final solutions obtained by our solution procedure for the VRPFlexTW with $p_i = 0.15,\ \forall i \in N$. We report the CPU times in seconds for cases (iii), (iv) and (v) to compare the performance of our algorithm with that of the procedure where the scheduling technique is applied in the tabu search method (see Section 5.7). Note that for the original VRPTW, the scheduling method (either applied once as a last phase or every iteration in the tabu search) does not provide any improvement since no penalties are permitted. The solutions presented by Table 6 are generated by setting (θ,τ) to $(10^5, 2.5(10^4))$ in the tabu search algorithm since these parameters yield good results as seen in Table 5. The average values of the total distance, the number of vehicles, and the CPU times over all problem instances are also given in Table 6. Note that in the formulation (1)–(13), the depot is also considered in computing the total penalty cost. In the solutions of the VRPFlexTW, relaxing the time window at the depot does not bring any earliness or lateness since that node only has a very large upper bound (leading to the scheduling horizon). Overall, the results represented by Table 6 indicate that the VRPFlexTW with a positive flexibility fraction (p_i) provides a decrease in the average number of vehicles compared to that obtained by the optimal/best-known solutions of the original VRPTW. Furthermore, the average total distance and the average number of vehicles are decreasing as p_i increases.

Results obtained for the original VRPTW (VRPFlexTW with $p_i = 0,\ \forall i \in N$) show that our solution procedure obtains good final solutions with respect to the optimal/best-known solutions. More specifically, when compared to the optimal/best-known solutions, the solutions obtained by the proposed methodology achieve a 2.22% gap in the average total distance and a 0.86% gap in the average number of vehicles. Moreover, for eight problem instances (C101, C102, C103, C105, C106, C107, C108, C109) we obtain the optimal solutions. Since our solution procedure is effective for the original problem, we first compare the final solutions obtained for the VRPFlexTW (cases (iii), (iv) and (v)) with the final solutions obtained by our solution procedure for the VRPTW (case (ii)). We also compare the final solutions obtained for the VRPFlexTW with the optimal/best-known solutions of the original VRPTW (case (i)).

Table 7 provides the analysis of the solutions obtained for the VRPFlexTW with respect to the solutions obtained by our solution procedure for the original VRPTW. To prevent redundancy, we explain one row of this table in detail. Results obtained with $p_i = 0.10,\ \forall i \in N$ show that for 13 problem instances (R102, R105, R106, R109, R111, R112, RC102, RC103, RC104, RC105, RC106, RC107, RC108) VRPFlexTW reduces both the total distance and the number of vehicles, compared to those obtained by our solution

Table 6
Comparison of the original VRPTW solutions with the final solutions obtained by our solution procedure where $(\theta,\tau)=(10^5,2.5(10^4))$.

Ins.	Opt. VRPTW		VRPFlexTW, $p_i=0$		VRPFlexTW, $p_i=0.05$			VRPFlexTW, $p_i=0.10$			VRPFlexTW, $p_i=0.15$		
	Dist.	#Veh.	Dist.	#Veh.	Dist.	#Veh.	CPU	Dist.	#Veh.	CPU	Dist.	#Veh.	CPU
C101	828.94	10	828.94	10	828.94	10	87	828.94	10	200	828.94	10	315
C102	828.94	10	828.94	10	838.04	10	492	835.08	10	467	1007.31	10	712
C103	828.06	10	828.06	10	864.64	10	629	915.23	10	769	972.41	10	617
C104	824.78	10	825.96	10	848.98	10	1468	894.95	10	1078	855.29	10	1575
C105	828.94	10	828.94	10	828.94	10	637	884.25	10	640	828.94	10	681
C106	828.94	10	828.94	10	828.94	10	709	828.94	10	683	828.94	10	672
C107	828.94	10	828.94	10	828.94	10	593	828.94	10	538	828.94	10	573
C108	828.94	10	828.94	10	828.94	10	539	828.94	10	508	828.94	10	570
C109	828.94	10	828.94	10	828.94	10	503	828.94	10	586	828.94	10	636
R101	1642.92	20	1681.52	19	1669.34	19	570	1658.78	19	792	1623.69	18	1196
R102	1471.75	18	1506.47	17	1509.70	17	726	1442.28	16	1898	1425.14	16	1236
R103	1213.62	14	1247.64	14	1232.08	14	1539	1266.84	13	1236	1210.14	14	2112
R104	976.76	11	1014.29	11	994.97	10	1644	1017.54	10	1337	980.65	10	1523
R105	1360.12	15	1408.60	15	1370.79	15	1395	1342.96	14	1928	1343.93	14	1938
R106	1239.37	13	1261.08	14	1240.71	13	1413	1249.77	13	805	1186.79	12	1435
R107	1069.09	11	1087.06	11	1091.73	11	1079	1068.63	11	1742	1050.86	11	1726
R108	936.69	10	968.92	10	952.87	10	786	1007.50	10	567	961.70	9	1390
R109	1151.89	13	1173.80	13	1163.73	13	1069	1140.28	12	816	1128.15	12	1938
R110	1072.41	12	1089.36	12	1073.11	11	1456	1095.39	12	796	1033.29	11	1723
R111	1053.50	12	1100.82	12	1077.80	12	460	1022.88	11	858	1068.23	11	1017
R112	953.44	10	986.49	11	967.98	10	673	951.69	10	1099	965.95	10	648
RC101	1623.58	15	1675.55	15	1629.99	15	1190	1611.03	15	1253	1595.05	15	1512
RC102	1461.33	14	1526.62	14	1493.60	13	1078	1439.59	13	1139	1428.71	13	2127
RC103	1261.67	11	1287.45	12	1265.89	11	775	1216.09	11	720	1279.81	12	742
RC104	1135.48	10	1174.91	11	1152.08	10	1428	1123.76	10	1245	1123.85	10	917
RC105	1517.93	15	1542.43	15	1538.94	15	1321	1469.31	14	1016	1457.73	13	1128
RC106	1376.26	12	1400.76	13	1341.22	12	777	1316.74	12	770	1298.27	11	995
RC107	1211.24	12	1276.44	12	1194.74	11	775	1202.12	11	606	1183.25	11	1895
RC108	1117.53	11	1150.82	11	1113.39	10	1989	1104.14	10	813	1094.27	10	1959
Avg.	1113.86	12.03	1138.54	12.14	1124.14	11.79	958.62	1117.98	11.62	927.76	1112.00	11.48	1224.41

Table 7
VRPFlexTW versus VRPTW with the solutions obtained by our solution procedure.

p_i	Dist.(↓), #Veh.(↓)	Dist.(↓), #Veh.(↔)	Dist.(↑), #Veh.(↓)	Dist.(↔), #Veh.(↔)	Dist.(↑), #Veh.(↔)	Dist.(↑), #Veh.(↑)
0.05	10	8	0	6	5	0
0.10	13	3	2	5	6	0
0.15	16	4	0	6	3	0

Table 8
VRPFlexTW versus VRPTW with the optimal/best-known solutions.

p_i	Dist.(↓), #Veh.(↓)	Dist.(↓), #Veh.(↔)	Dist.(↑), #Veh.(↓)	Dist.(↔), #Veh.(↔)	Dist.(↑), #Veh.(↔)	Dist.(↑), #Veh.(↑)
0.05	2	1	5	6	15	0
0.10	8	6	3	5	7	0
0.15	11	4	3	6	4	1

procedure for the original VRPTW where time windows are defined as hard time windows. For three problem instances (R101, R107, RC101), VRPFlexTW provides a reduction in the total distance with the same number of vehicles as the one given by our solutions obtained for the original VRPTW. For two problem instances (R103, R104), VRPFlexTW yields fewer vehicles; however, this brings an increase in the total distance. For five problem instances (C101, C106, C107, C108, C109), VRPFlexTW obtains the same solutions as the ones found by our solution procedure for the original VRPTW. For the remaining problem instances (C102, C103, C104, C105, R108, R110), VRPFlexTW results in an increase in the total distance with the same number of vehicles.

In Table 8, we present the analysis of the solutions obtained for the VRPFlexTW with respect to the optimal/best-known solutions of the original VRPTW. Note that in this part of the evaluation, we refer to the optimal/best-known solutions of the original VRPTW as the optimal VRPTW. To prevent redundancy, we explain one row of this table in detail. Results obtained with $p_i=0.15$, $\forall i \in N$ show that for 11 problem instances (R101, R102, R105, R106, R109, R110, RC102, RC105, RC106, RC107, RC108) VRPFlexTW reduces both the total distance and the number of vehicles, compared to those obtained by the optimal VRPTW where time windows are defined as hard time windows. For four problem instances (R103, R107, RC101, RC104), VRPFlexTW provides a reduction in the total distance with the same number of vehicles as the one given by the optimal VRPTW. For three problem instance (R104, R108, R111), VRPFlexTW yields fewer vehicles; however, this brings an increase in the total distance. For six problem instances (C101, C105, C106, C107, C108, C109), VRPFlexTW obtains the same solutions as the optimal VRPTW. For four problem instances (C102, C103, C104, R112), VRPFlexTW results in an increase in the total distance with the same number of vehicles. For the remaining problem instance (RC103), VRPFlexTW results in an increase both in the total distance and the number of vehicles.

According to the solutions analyzed in detail and to the average values given in Table 6, we conclude that the flexible time windows provide significant operational gains to the carrier companies. These gains can be observed in the total distance traveled and in the number of vehicles used, which are the two basic components in the classical VRPTW. Carrier companies can take the advantage of traversing less distance or using fewer vehicles by delivering the goods to customers with a small violation in time windows.

5.4. VRPFlexTW versus VRPSTW

This subsection aims to compare our solution procedure with existing (meta)heuristic algorithms given in the literature. We focus on the classical VRPTW since this is the standard way of testing different settings developed for soft time windows. For results presented in Table 9, we follow Figliozzi et al. [13] which is the most recent work among all related papers, and we additionally include solutions provided in Chiang and Russell [7].

The results presented in Table 9 show that our solution procedure based on the tabu search algorithm performs well on problem instances with hard time windows. For R1 and RC1 data sets, we have higher number of vehicles; however, this leads to a reduction in total distance.

Table 9
The classical VRPTW solutions obtained by selected metaheuristic algorithms and by our solution procedure.

Method	Dist.			#Veh.		
	C1	R1	RC1	C1	R1	RC1
Taillard et al. [30]	828.5	1220.4	1381.3	10.00	12.64	12.08
Chiang and Russell [7]	828.4	1219.1	1376.6	10.00	12.08	11.88
Ibaraki et al. [21]	828.4	1217.4	1391.0	10.00	11.92	11.50
Figliozzi [13]	871.8	1261.6	1419.8	10.00	12.50	12.00
VRPFlexTW	828.5	1210.5	1379.4	10.00	13.25	12.88

5.5. Solutions obtained by our algorithm versus solutions obtained by CPLEX

In this subsection, we compare the solutions obtained for small-sized problem instances by our solution procedure to those obtained by solving the formulation (1)–(13) directly within CPLEX 12.5 [22]. Three sets of experiments are conducted by selecting the first 15, 20 and 25 customers from each problem instance provided by Solomon [28]. In our solution procedure, (θ,τ,c_e,c_d) are set to $(10^5, 2.5(10^4), 0.5, 1.0)$, and $p_i=0.10$ for all $i \in N$ which corresponds to a medium flexibility among percentages considered in the main experiments. For CPLEX, a maximum of 7200 s is imposed on the solution time. This time is arranged with respect to the largest computation time required for our solution procedure where instances with 100 customers are considered (1928 s with $p_i=0.10$). Table 10 presents the solutions obtained by our solution procedure and the optimal solutions obtained by CPLEX. Results in this table indicate that our solution procedure provides the optimal solutions to 11 out of 12 instances with substantially lower computation time. For problem instance C107-15, our solution is 0.82% away from the optimality with respect to the objective function values. These results confirm that our solution procedure performs well and obtains very good results in a reasonable amount of time (not only for the classical VRPTW already shown in previous subsections, but also for the VRPFlexTW).

The interested reader is referred to Appendix A for the details of the best feasible solutions found by CPLEX within the time limit (Tables A1, A2 and A3), where the final optimality gap in percentage ($\text{Gap}_f\%$) is also given. CPLEX cannot provide a feasible solution (within the time limit) to five problem instances with 25 customers (C104-25, R103-25, R104-25, R107-25, R108-25) and thus average values (both for our solution procedure and for CPLEX) are calculated over the instances with feasible solutions. Average results given in Tables A1, A2 and A3 show that the solutions obtained by our solution procedure are better compared to those obtained by CPLEX. Moreover, the average computation time required by our solution procedure to solve these instances is much less than that required by CPLEX (which is equal to 7200 s).

5.6. Effects of implementing larger flexibility fractions

In this paper, we consider a symmetric violation with asymmetric penalty costs for each node and numerical experiments are carried out with three different flexibility fractions (0.05, 0.10 and 0.15). Note that, each customer time window is extended by

Table 10
Details of solutions obtained by our solution procedure and optimal solutions obtained by CPLEX where $p_i=0.10$, $\forall i \in N$.

Ins.	Solutions obtained by our solution procedure						Optimal solutions obtained by CPLEX					
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Dist.	Earl.	#Veh.	Obj.	CPU
C101-15	0.00	142.14	0.00	2	1084.29	1	0.00	142.14	0.00	2	1084.29	0
C106-15	0.00	142.14	0.00	2	1084.29	3	0.00	142.14	0.00	2	1084.29	30
C107-15	0.00	145.11	6.00	2	1093.22	3	0.00	142.14	0.00	2	1084.29	6875
R101-15	0.00	382.88	1.00	5	2766.25	0	0.00	382.88	1.00	5	2766.25	0
R105-15	0.00	349.88	0.00	3	1899.77	0	0.00	349.88	0.00	3	1899.77	481
RC101-15	0.00	222.50	2.58	2	1246.30	3	0.00	222.50	2.58	2	1246.30	15
C101-20	0.00	175.37	0.00	3	1550.75	10	0.00	175.37	0.00	3	1550.75	1
C105-20	0.00	175.37	0.00	3	1550.75	6	0.00	175.37	0.00	3	1550.75	5024
C106-20	0.00	175.37	0.00	3	1550.75	8	0.00	175.37	0.00	3	1550.75	591
R101-20	1.00	497.82	1.00	6	3397.15	0	1.00	497.82	1.00	6	3397.15	21
C101-25	0.00	191.81	0.00	3	1583.63	17	0.00	191.81	0.00	3	1583.63	1
R101-25	0.00	618.33	0.00	8	4436.66	18	0.00	618.33	0.00	8	4436.66	5553
Avg.	0.08	268.23	0.88	3.50	1936.98	5.75	0.08	267.98	0.38	3.50	1936.24	1549.33

Table 11
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ, τ) are $(10^5, 2.5(10^4))$, (c_e, c_d) are (0.5, 1.0) and $p_i = 0.25$, $\forall i \in N$.

Ins.	Solution of the tabu search algorithm						Final solution	
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Earl.
C101	0.00	828.94	0.00	10	5657.87	377	0.00	0.00
C102	45.39	1017.06	37.60	10	6098.31	521	45.39	0.00
C103	0.00	1022.88	164.00	10	6127.75	634	0.00	11.51
C104	8.00	933.47	0.00	10	5874.94	1741	8.00	0.00
C105	0.00	828.94	0.00	10	5657.87	532	0.00	0.00
C106	0.00	828.94	0.00	10	5657.87	597	0.00	0.00
C107	0.00	828.94	0.00	10	5657.87	496	0.00	0.00
C108	0.00	828.94	0.00	10	5657.87	689	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	511	0.00	0.00
R101	13.02	1597.96	235.83	17	10,126.86	1117	15.12	27.45
R102	5.69	1405.93	145.67	15	8890.38	664	5.69	8.89
R103	19.94	1225.99	83.24	13	7713.54	1013	19.94	9.27
R104	11.25	1022.51	44.93	10	6078.74	1013	11.25	4.49
R105	42.70	1294.36	234.39	13	7948.60	1490	43.20	123.73
R106	34.53	1207.99	153.74	12	7327.38	914	35.40	57.92
R107	13.58	1062.88	103.19	11	6590.95	902	13.58	41.50
R108	14.16	946.54	48.64	9	5531.56	1524	14.16	21.41
R109	33.89	1110.40	206.87	12	7158.12	1301	33.89	74.63
R110	14.40	1061.10	134.92	11	6604.05	559	15.51	68.12
R111	29.88	1053.24	128.64	10	6200.68	887	29.88	73.98
R112	0.85	971.28	70.76	10	5978.78	1416	0.85	2.24
RC101	36.79	1521.56	282.53	14	8821.17	1205	36.79	74.16
RC102	19.78	1395.25	189.98	13	8105.27	1835	19.78	48.87
RC103	66.45	1221.53	63.13	10	6541.07	1053	66.45	37.33
RC104	13.88	1133.50	42.76	10	6302.25	764	13.88	6.88
RC105	42.66	1425.89	175.86	13	8182.37	1173	42.66	72.58
RC106	44.31	1280.12	204.88	11	7107.00	588	45.08	110.91
RC107	54.69	1157.88	183.17	10	6462.05	1252	55.45	127.96
RC108	10.97	1127.99	122.06	10	6327.99	740	10.97	89.28
Avg.	19.89	1109.34	105.41	11.17	6760.24	948.55	20.10	37.69

Table 12
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ, τ) are $(10^5, 2.5(10^4))$, (c_e, c_d) are (0.5, 1.0) and $p_i = 0.50$, $\forall i \in N$.

Ins.	Solution of the tabu search algorithm						Final solution	
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Earl.
C101	0.00	828.94	0.00	10	5657.87	634	0.00	0.00
C102	16.27	885.72	0.00	10	5787.71	1133	16.27	0.00
C103	40.83	891.64	123.50	10	5885.85	1352	40.83	34.50
C104	12.26	992.11	0.00	10	5996.47	1632	12.26	0.00
C105	0.00	828.94	0.00	10	5657.87	698	0.00	0.00
C106	0.00	875.94	0.00	10	5751.88	576	0.00	0.00
C107	0.00	828.94	0.00	10	5657.87	1038	0.00	0.00
C108	0.00	828.94	0.00	10	5657.87	886	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	596	0.00	0.00
R101	39.86	1483.57	351.26	16	9582.63	1207	41.95	89.27
R102	25.79	1355.93	221.16	14	8448.23	1692	29.59	45.49
R103	26.80	1133.35	132.67	12	7159.84	1826	27.45	33.17
R104	9.30	1002.60	60.00	10	6044.51	850	9.30	13.48
R105	54.13	1262.53	327.66	12	7543.02	860	55.76	172.91
R106	30.79	1201.78	209.63	12	7339.16	461	34.65	89.65
R107	25.26	1051.92	150.12	11	6604.16	1147	25.26	47.80
R108	52.89	963.73	54.87	9	5607.78	654	52.89	36.12
R109	52.22	1114.68	199.41	10	6381.29	1750	63.86	153.50
R110	34.76	1066.27	175.54	10	6255.07	1500	34.76	146.71
R111	48.61	987.80	230.10	10	6139.26	741	48.61	164.66
R112	9.58	968.84	104.48	9	5599.50	1312	9.58	99.81
RC101	51.30	1481.27	335.13	13	8381.41	1146	64.65	156.89
RC102	487.92	1481.79	201.46	10	7552.24	582	487.92	156.46
RC103	395.53	1279.01	121.58	9	6614.34	808	395.53	106.58
RC104	124.42	1185.26	110.05	9	6149.96	692	124.42	73.58
RC105	112.07	1354.20	224.55	12	7732.74	1582	112.56	128.53
RC106	17.39	1226.62	296.77	11	7019.01	771	18.33	208.70
RC107	70.90	1150.99	181.28	10	6463.52	649	70.90	166.04
RC108	39.22	1076.81	71.78	10	6228.72	469	44.92	51.37
Avg.	61.31	1090.31	133.90	10.66	6570.95	1008.41	62.84	75.01

Table 13
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ,τ) are $(10^5,2.5(10^4))$, (c_e, c_d) are (2.0, 4.0) and $p_i=0.25$, $\forall i \in N$.

Ins.	Solution of the tabu search algorithm						Final solution	
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Earl.
C101	0.00	828.94	0.00	10	5657.87	425	0.00	0.00
C102	56.53	1112.59	72.46	10	6596.24	1271	56.53	20.22
C103	2.06	1063.17	33.29	10	6201.14	1349	2.06	0.00
C104	0.00	1037.85	0.00	10	6075.70	2073	0.00	0.00
C105	0.00	828.94	0.00	10	5657.87	1007	0.00	0.00
C106	0.00	866.26	0.00	10	5732.53	956	0.00	0.00
C107	0.00	828.94	0.00	10	5657.87	511	0.00	0.00
C108	0.00	828.94	0.00	10	5657.87	493	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	566	0.00	0.00
R101	10.39	1650.40	207.31	18	10,956.99	649	12.49	23.88
R102	6.80	1444.94	91.99	15	9101.05	601	7.07	23.52
R103	3.65	1208.27	63.14	13	7757.44	791	3.65	3.79
R104	2.55	1015.79	25.00	11	6491.78	806	2.55	4.38
R105	11.59	1452.68	132.00	14	8815.72	790	13.32	49.41
R106	6.38	1264.87	61.49	12	7478.23	1805	6.38	18.21
R107	3.46	1145.95	24.28	11	6754.29	769	3.46	5.15
R108	0.00	1032.60	0.00	10	6065.19	709	0.00	0.00
R109	0.00	1246.78	52.61	12	7398.78	850	0.00	4.09
R110	0.36	1166.79	29.60	11	6794.21	1407	0.36	3.96
R111	0.24	1123.55	41.00	11	6730.07	1392	0.24	10.43
R112	0.23	988.59	5.84	10	5989.79	1225	0.23	2.39
RC101	24.09	1622.05	147.16	14	9234.81	893	25.57	52.10
RC102	3.42	1491.81	71.97	13	8341.22	1911	3.42	34.23
RC103	1.72	1390.71	39.25	12	7666.79	846	1.72	18.40
RC104	1.41	1226.62	7.86	11	6874.60	1448	1.41	6.72
RC105	8.37	1542.16	54.18	13	8426.15	1325	8.37	21.55
RC106	3.20	1442.37	83.42	11	7464.36	1849	5.45	37.97
RC107	0.63	1245.44	55.41	11	7004.24	1705	0.63	15.20
RC108	4.86	1180.71	16.39	11	6813.67	1155	4.86	4.57
Avg.	5.24	1176.13	45.37	11.52	7070.84	1088.86	5.51	12.42

0.10, 0.20 and 0.30, and relaxing a time window by these values with respect to its duration provides sufficient flexibility (especially for the case where customers have tight time windows and thus delivery bounds should not be violated more than a certain extent to limit the negative impacts on customer satisfaction). This subsection aims to analyze the effects of considering larger flexibility fractions in terms of the performance of the proposed solution procedure. Tables 11 and 12 present the solutions obtained for the VRPFlexTW with $p_i=0.25$ for all $i \in N$ and with $p_i=0.50$ for all $i \in N$, respectively. In these tables, values of (θ,τ) are equal to $(10^5, 2.5(10^4))$ and penalty cost coefficients (c_e, c_d) are set to the original values, (0.5, 1.0). When we compare the average results given in Tables 11 and 12 to those represented in Table 5 with $p_i=0.05$ for all $i \in N$, we observe that implementing larger flexibility fractions leads to less total distance by using fewer vehicles where the values of the total delay and the total earliness are higher (both over the solutions obtained by the tabu search algorithm and over the corresponding final solutions obtained by the scheduling method). We have a similar observation for the comparison of average results given in Table 11 (with $p_i=0.25$ for all $i \in N$) to those represented in Table 12 (with $p_i=0.50$ for all $i \in N$).

In the second part of the analysis, we solve the same problem instances with higher penalty cost coefficients where (c_e, c_d) are equal to (2.0, 4.0). Tables 13 and 14 present the results obtained for the VRPFlexTW with $p_i=0.25$ for all $i \in N$ and with $p_i=0.50$ for all $i \in N$, respectively. In these tables, values of (θ,τ) are equal to $(10^5, 2.5(10^4))$. When we compare the average results given in Table 13 (Table 14) to those represented in Table 11 (Table 12), we observe that implementing larger penalty coefficients leads to higher total distance by using more vehicles where the values of the total delay and the total earliness are smaller (both over the solutions obtained by the tabu search algorithm and over the corresponding final solutions obtained by the scheduling method). Moreover, increasing flexibility from $p_i=0.25$ to $p_i=0.50$ within the same cost structure (from Table 13 to Table 14) leads to fewer vehicles. However, the values of the total distance, the total delay and the total earliness are higher. In other words, using fewer vehicles can be compensated with an increase not only in the total penalty cost but also in the total distance traveled. The main reason behind this circumstance is solving the problem instances by a solution procedure which may obtain local optimal solutions. The proposed solution algorithm terminates either because it reaches the maximum number of given iterations or because the best feasible solution has not been improved for a predetermined number of iterations. These properties may lead to solutions reflecting the observed outcome (in terms of an increase both in the total penalty cost and in the total distance).

5.7. *Effects of scheduling method*

In our solution approach, we apply a scheduling method to the solution obtained by the tabu search algorithm. This method improves the solution on-hand by obtaining the optimal departure time of each route from the depot and the optimal times to begin service at each customer. The aim of this section is to evaluate the impact of applying the scheduling method every iteration in the tabu search algorithm. Table 15 presents the optimal/best-known solutions for the original VRPTW and the solutions of the VRPFlexTW (with different flexibility fractions) obtained by inserting the scheduling method into the tabu search. Note that we refer to the solutions of the VRPFlexTW given by

Table 14
Details of solutions obtained by the tabu search algorithm and the scheduling method, where (θ,τ) are $(10^5, 2.5(10^4))$, (c_e, c_d) are (2.0, 4.0) and $p_i = 0.50$, $\forall i \in N$.

Ins.	Solution of the tabu search algorithm						Final solution	
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Earl.
C101	0.00	828.94	0.00	10	5657.87	599	0.00	0.00
C102	1.18	1353.45	68.42	11	7248.46	506	1.18	25.57
C103	8.16	1443.38	21.49	11	7362.39	1447	8.16	0.00
C104	0.00	1146.37	25.00	10	6342.74	535	0.00	0.00
C105	0.00	828.94	0.00	10	5657.87	561	0.00	0.00
C106	0.00	995.12	3.25	10	5996.73	970	0.00	0.00
C107	0.00	863.70	0.00	10	5727.39	562	0.00	0.00
C108	0.00	861.28	0.00	10	5722.56	1179	0.00	0.00
C109	0.00	828.94	0.00	10	5657.87	1040	0.00	0.00
R101	30.13	1588.37	270.20	16	10,237.65	724	35.46	92.00
R102	10.70	1474.76	156.82	15	9305.94	1102	10.70	44.98
R103	10.55	1305.03	77.72	13	8007.69	592	10.81	26.44
R104	4.27	1075.71	42.60	10	6253.71	1701	4.27	11.00
R105	11.74	1463.45	101.05	13	8375.94	1473	11.74	41.87
R106	5.29	1305.71	42.87	12	7518.30	1750	5.29	7.67
R107	4.63	1149.67	25.18	11	6768.20	1320	4.63	7.00
R108	2.51	994.27	7.13	10	6012.81	1556	2.51	7.00
R109	1.41	1277.89	42.88	12	7447.17	1576	1.41	28.09
R110	1.47	1195.50	66.34	10	6529.54	1132	1.47	54.21
R111	2.72	1136.02	22.18	11	6727.28	1310	2.72	9.76
R112	6.69	1015.32	32.26	9	5721.93	1538	6.69	24.13
RC101	23.06	1607.46	198.05	13	8903.25	622	23.06	120.23
RC102	12.79	1505.21	52.99	12	7967.55	1808	12.79	40.72
RC103	6.74	1382.56	5.99	12	7604.06	805	6.74	0.00
RC104	0.00	1209.11	12.44	11	6843.09	1174	0.00	11.02
RC105	6.43	1520.16	104.26	13	8474.54	544	6.43	43.09
RC106	8.45	1418.80	46.51	12	7764.41	666	8.45	30.20
RC107	41.68	1332.21	106.30	10	7043.76	448	42.89	98.05
RC108	3.42	1177.87	19.37	11	6808.17	744	3.42	15.97
Avg.	7.03	1216.73	53.49	11.31	7092.72	1033.93	7.27	25.48

Table 15
Comparison of the original VRPTW solutions with the final solutions obtained by the procedure where the scheduling method is applied in the tabu search and $(\theta,\tau) = (10^5, 2.5\ (10^4))$.

Ins.	Opt. VRPTW		VRPFlexTW, $p_i=0$		VRPFlexTW, $p_i=0.05$			VRPFlexTW, $p_i=0.10$			VRPFlexTW, $p_i=0.15$		
	Dist.	#Veh.	Dist.	#Veh.	Dist.	#Veh.	CPU	Dist.	#Veh.	CPU	Dist.	#Veh.	CPU
C101	828.94	10	828.94	10	828.94	10	127	828.94	10	289	828.94	10	374
C102	828.94	10	828.94	10	828.94	10	1321	832.77	10	1014	903.11	10	1173
C103	828.06	10	828.06	10	838.08	10	1054	881.44	10	1234	904.97	10	2600
C104	824.78	10	825.96	10	870.37	10	861	889.63	10	1822	856.06	10	2588
C105	828.94	10	828.94	10	828.94	10	884	884.25	10	898	828.94	10	883
C106	828.94	10	828.94	10	828.94	10	973	828.94	10	938	828.94	10	891
C107	828.94	10	828.94	10	828.94	10	813	828.94	10	752	828.94	10	727
C108	828.94	10	828.94	10	828.94	10	772	828.94	10	743	828.94	10	722
C109	828.94	10	828.94	10	828.94	10	1099	828.94	10	869	828.94	10	732
R101	1642.92	20	1681.52	19	1642.86	18	2808	1628.25	18	3069	1627.23	18	1449
R102	1471.75	18	1506.47	17	1459.51	17	3900	1441.83	17	1570	1416.62	16	2021
R103	1213.62	14	1247.64	14	1240.14	14	2419	1259.56	13	1183	1220.26	14	3443
R104	976.76	11	1014.29	11	1001.56	10	2937	978.88	10	2226	968.28	10	1468
R105	1360.12	15	1408.60	15	1356.41	15	3908	1353.15	14	2728	1331.21	14	2207
R106	1239.37	13	1261.08	14	1264.65	13	3046	1252.96	13	3602	1207.38	13	1763
R107	1069.09	11	1087.06	11	1073.32	11	1641	1106.19	11	2173	1056.39	11	1409
R108	936.69	10	968.92	10	964.71	10	2686	947.74	9	1214	955.52	9	2830
R109	1151.89	13	1173.80	13	1153.37	12	2281	1156.76	12	1816	1129.36	12	3266
R110	1072.41	12	1089.36	12	1074.11	12	1992	1064.36	11	1941	1032.60	11	1653
R111	1053.50	12	1100.82	12	1044.96	11	1577	1018.77	11	1373	1032.87	11	2047
R112	953.44	10	986.49	11	974.18	11	2277	961.72	10	1847	953.11	10	2619
RC101	1623.58	15	1675.55	15	1637.41	15	3052	1622.54	15	2283	1616.06	15	1861
RC102	1461.33	14	1526.62	14	1456.53	13	3078	1499.83	13	3765	1425.42	13	2090
RC103	1261.67	11	1287.45	12	1313.96	12	1261	1220.15	11	2369	1211.70	11	3738
RC104	1135.48	10	1174.91	11	1176.52	11	920	1130.74	10	2392	1109.92	10	2848
RC105	1517.93	15	1542.43	15	1540.09	15	2608	1484.79	14	2558	1495.81	13	3390
RC106	1376.26	12	1400.76	13	1375.37	13	1433	1318.51	12	1134	1268.01	11	2233
RC107	1211.24	12	1276.44	12	1201.91	11	1029	1197.22	11	3563	1186.35	11	1571
RC108	1117.53	11	1150.82	11	1102.22	10	2688	1103.56	10	1869	1091.66	10	1497
Avg.	1113.86	12.03	1138.54	12.14	1122.92	11.86	1911.90	1116.56	11.55	1835.66	1102.53	11.48	1934.24

Table 6 in which the scheduling method is applied only once to the solution generated by the tabu search algorithm as original final solutions.

When the scheduling method is applied every iteration in the tabu search algorithm, for the case with $p_i = 0.05$, $\forall i \in N$ the procedure reduces the total distance on average compared to that obtained by the original final solutions. However, this comes with an increase in the total number of vehicles on average. For the case with $p_i = 0.10$, $\forall i \in N$ the procedure reduces both the total distance and the number of vehicles on average compared to those obtained by the original final solutions. For the case with $p_i = 0.15$, $\forall i \in N$ the procedure provides a reduction in the total distance with the same number of vehicles on average compared to those given by the original final solutions. For all cases, we observe higher computational times.

Cordeau et al. [9] summarize some of the best available and the well-known classical heuristics, and provide a comparison with respect to some essential criteria. Both speed and accuracy are important features of a solution procedure based on heuristics. Solutions given by Tables 6 and 15 confirm that our solution procedure where the scheduling method is applied when the tabu search algorithm terminates obtains very good final solutions in a reasonable computational time.

6. Conclusions

In this paper, we introduce the VRPFlexTW which enables serving customers outside their original time boundaries with respect to a given tolerance. Compared to the VRPTW, the VRPFlexTW permits fixed deviations from customer time windows at a cost. Furthermore, when compared to the VRPSTW, the VRPFlexTW operates on a far more restricted solution space.

Our solution procedure comprises three main components: initialization, routing and scheduling. The time-oriented nearest neighbor heuristic is used in the initialization component. The routing component is handled via a tabu search algorithm, while the scheduling component is performed by solving an LP model. We validate our solution algorithm on benchmark instances and test the performance of the solution procedure with various stopping criteria values. Furthermore, we compare the solutions of the VRPTW with those of the VRPFlexTW. In many instances, we observe that the VRPFlexTW results in operational gains when compared to the VRPTW. These gains are achieved by a reduction in the total distance traveled or by a reduction in the number of vehicles used or by a reduction both in the total distance and in the number of vehicles.

We model a practical problem and develop an efficient solution framework to handle it. Our solution approach can effectively be used by carrier companies trying to assess the added value of allowing a certain extent of customer service flexibility. Generally, relaxing time windows improves the total cost. However, this relaxation brings some violations. This trade-off might be balanced with respect to the preferences of carrier companies and to the concerns of their customers. Further research may focus on handling uncertainties in travel times and on exploring more complex penalty functions.

Appendix A. Best feasible solutions obtained by CPLEX within the time limit

See (Tables A1–A3).

Table A1
Details of the best feasible solutions obtained for instances with 15 customers by our solution procedure and by CPLEX where $p_i = 0.10$, $\forall i \in N$.

Ins.	Solutions obtained by our solution procedure						Solutions obtained by CPLEX					
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Dist.	Earl.	#Veh.	Obj.	Gap_f(%)
C102-15	0.00	142.14	0.00	2	1084.29	4	0.00	141.07	0.00	2	1082.13	72.45
C103-15	0.00	141.48	0.00	2	1082.97	4	0.00	141.48	0.00	2	1082.97	74.92
C104-15	0.00	141.48	0.00	2	1082.97	4	0.00	137.78	0.00	2	1075.56	81.22
C105-15	0.00	142.14	0.00	2	1084.29	3	0.00	142.14	0.00	2	1084.29	50.91
C108-15	10.03	137.07	20.90	2	1094.62	3	0.00	142.14	0.00	2	1084.29	78.30
C109-15	0.00	148.47	0.00	2	1096.93	4	0.00	136.70	0.00	2	1073.39	87.02
R102-15	0.40	332.12	1.00	3	1865.14	4	0.40	336.25	1.00	3	1873.40	67.57
R103-15	0.40	332.12	1.00	3	1865.14	3	0.40	332.12	1.00	3	1865.14	70.85
R104-15	0.00	288.86	0.00	3	1777.73	4	0.00	288.86	0.00	3	1777.73	72.71
R106-15	0.00	310.10	2.02	3	1821.21	4	0.00	310.10	2.02	3	1821.21	71.31
R107-15	0.00	310.10	2.02	3	1821.21	3	0.00	315.74	0.00	3	1831.48	74.73
R108-15	0.00	278.22	0.00	3	1756.45	4	0.00	285.22	0.00	3	1770.44	73.24
R109-15	3.81	316.09	4.00	3	1837.99	3	3.81	316.09	4.00	3	1837.99	69.80
R110-15	0.00	284.73	0.00	3	1769.46	3	3.81	281.42	13.80	3	1773.55	76.08
R111-15	0.00	304.64	0.00	3	1809.28	6	0.00	326.19	0.00	3	1852.38	74.85
R112-15	11.05	276.64	16.36	2	1372.51	0	0.00	278.22	0.00	3	1756.45	75.69
RC102-15	0.00	198.64	2.00	2	1198.28	3	0.00	202.27	0.00	2	1204.55	71.01
RC103-15	0.00	198.64	2.00	2	1198.28	3	0.00	198.64	2.00	2	1198.28	77.69
RC104-15	0.00	192.93	0.00	2	1185.87	3	0.00	192.18	0.00	2	1184.37	86.91
RC105-15	5.00	260.38	1.83	3	1726.67	3	5.00	260.38	1.83	3	1726.67	60.23
RC106-15	0.00	206.95	9.60	2	1218.70	2	0.00	202.60	6.00	2	1208.19	87.98
RC107-15	0.00	186.94	1.00	2	1174.39	3	0.00	186.94	1.00	2	1174.39	88.18
RC108-15	0.00	186.94	0.00	2	1173.89	3	0.00	187.71	0.00	2	1175.42	89.00
Avg.	1.33	231.21	2.77	2.43	1439.05	3.30	0.58	232.27	1.42	2.48	1457.14	75.33

Table A2
Details of the best feasible solutions obtained for instances with 20 customers by our solution procedure and by CPLEX where $p_i = 0.10$, $\forall i \in N$.

Ins.	Solutions obtained by our solution procedure						Solutions obtained by CPLEX					
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Dist.	Earl.	#Veh.	Obj.	Gap_f(%)
C102-20	0.00	188.79	0.00	2	1177.58	7	0.00	177.43	0.00	2	1154.85	69.50
C103-20	0.00	162.65	0.00	2	1125.30	11	0.00	162.65	0.00	2	1125.30	73.57
C104-20	0.00	161.47	0.00	2	1122.93	8	0.00	160.79	0.00	2	1121.58	75.66
C107-20	43.74	160.82	0.00	2	1165.37	7	43.74	160.82	0.00	2	1165.37	64.79
C108-20	8.25	160.82	0.00	2	1129.88	6	8.25	160.82	0.00	2	1129.88	79.34
C109-20	0.00	160.82	0.00	2	1121.63	7	0.00	160.82	0.00	2	1121.63	84.04
R102-20	0.00	434.91	0.00	6	3269.82	10	0.00	434.91	0.00	6	3269.82	78.37
R103-20	0.40	368.49	1.36	4	2338.06	8	0.40	395.53	1.00	4	2391.97	77.72
R104-20	0.00	331.88	0.00	3	1863.76	7	0.00	378.28	0.00	4	2356.56	77.75
R105-20	3.56	436.37	6.00	4	2479.30	1	1.21	428.84	3.00	5	2860.39	72.86
R106-20	0.00	384.65	0.00	4	2369.30	8	0.00	397.81	2.80	4	2397.03	77.81
R107-20	0.50	373.12	5.02	3	1949.24	6	9.08	358.94	6.00	4	2329.95	78.96
R108-20	0.00	310.70	2.46	3	1822.64	7	0.00	352.45	0.00	3	1904.90	74.92
R109-20	4.02	435.27	4.59	3	2076.85	6	0.00	379.79	2.99	4	2361.08	61.30
R110-20	1.46	337.69	26.65	3	1890.17	5	0.00	365.38	0.00	4	2330.76	78.12
R111-20	2.51	368.06	2.90	3	1940.08	6	0.00	365.31	0.00	4	2330.62	78.33
R112-20	0.00	310.70	0.00	3	1821.41	7	0.00	310.70	0.00	3	1821.41	73.33
RC101-20	1.80	334.33	4.16	3	1872.54	10	1.80	334.40	3.77	3	1872.49	21.92
RC102-20	2.22	306.67	2.39	3	1816.75	7	0.00	306.79	3.00	3	1815.07	71.34
RC103-20	0.00	302.85	2.00	3	1806.70	7	0.00	306.22	0.00	3	1812.44	87.44
RC104-20	3.53	293.77	3.00	3	1792.57	7	0.00	309.82	0.00	3	1819.64	86.17
RC105-20	5.00	367.65	1.83	4	2341.22	7	5.00	367.65	1.83	4	2341.22	84.47
RC106-20	0.00	314.71	12.06	3	1835.45	6	0.00	312.49	3.07	3	1826.53	86.97
RC107-20	0.00	283.73	1.00	3	1767.96	6	0.00	283.73	1.00	3	1767.96	81.71
RC108-20	0.00	283.73	0.00	3	1767.46	7	0.00	296.13	0.00	3	1792.25	82.76
Avg.	3.08	302.99	3.02	3.04	1826.56	6.96	2.78	306.74	1.14	3.28	1928.83	75.17

Table A3
Details of the best feasible solutions obtained for instances with 25 customers by our solution procedure and by CPLEX where $p_i = 0.10$, $\forall i \in N$.

Ins.	Solutions obtained by our solution procedure						Solutions obtained by CPLEX					
	Del.	Dist.	Earl.	#Veh.	Obj.	CPU	Del.	Dist.	Earl.	#Veh.	Obj.	Gap_f(%)
C102-25	0.00	191.81	0.00	3	1583.63	20	0.00	190.74	0.00	3	1581.48	77.34
C103-25	0.00	191.81	0.00	3	1583.63	12	0.00	195.62	0.00	3	1591.23	83.38
C104-25	0.00	191.81	0.00	3	1583.63	14	No integer solution					
C105-25	0.00	191.81	0.00	3	1583.63	11	0.00	191.81	0.00	3	1583.63	35.49
C106-25	0.00	191.81	0.00	3	1583.63	13	0.00	191.81	0.00	3	1583.63	42.01
C107-25	0.00	192.18	6.00	3	1587.35	12	0.00	191.81	0.00	3	1583.63	84.42
C108-25	0.00	191.81	0.00	3	1583.63	13	0.00	219.91	0.00	3	1639.81	86.24
C109-25	0.00	192.18	0.00	3	1584.35	17	0.00	195.43	0.00	3	1590.85	88.76
R102-25	0.00	548.11	0.00	7	3896.22	16	0.00	565.57	0.00	7	3931.13	80.41
R103-25	12.40	470.42	2.00	4	2554.25	13	No integer solution					
R104-25	12.00	411.46	1.00	4	2435.42	13	No integer solution					
R105-25	2.35	529.41	4.42	5	3063.39	4	1.21	527.90	3.00	6	3458.51	73.07
R106-25	2.21	494.08	1.28	4	2591.01	10	0.00	514.90	0.00	5	3029.81	77.12
R107-25	8.80	414.48	2.18	4	2438.85	14	No integer solution					
R108-25	2.00	390.28	5.46	4	2385.29	13	No integer solution					
R109-25	0.00	460.52	0.00	4	2521.04	21	0.00	509.67	0.00	5	3019.34	77.72
R110-25	1.46	418.28	14.48	4	2445.25	17	2.06	506.55	4.30	6	3417.32	82.98
R111-25	4.25	417.93	0.00	4	2440.12	13	1.07	463.92	6.50	5	2932.16	79.69
R112-25	24.36	390.45	2.42	3	2006.47	6	18.07	652.37	36.46	5	3341.05	83.02
RC101-25	3.19	365.23	6.77	3	1937.03	17	3.19	365.23	6.77	3	1937.03	63.74
RC102-25	3.60	337.57	5.00	3	1881.25	12	1.39	337.61	6.00	3	1879.61	81.16
RC103-25	0.00	330.55	2.00	3	1862.10	11	3.53	331.52	3.00	3	1868.08	88.30
RC104-25	3.53	305.03	3.00	3	1815.10	13	0.00	430.74	0.66	4	2461.81	91.60
RC105-25	6.77	400.51	2.83	4	2409.20	11	5.97	412.90	2.83	4	2433.19	82.10
RC106-25	2.18	330.76	12.67	3	1870.05	9	2.18	337.07	6.00	3	1879.32	87.16
RC107-25	0.00	296.83	1.00	3	1794.16	10	0.00	299.20	3.66	3	1800.22	88.16
RC108-25	0.00	296.83	0.00	3	1793.66	10	0.00	297.79	0.00	3	1795.59	89.22
Avg.	2.45	330.25	2.81	3.50	2064.36	12.64	1.76	360.46	3.60	3.91	2288.11	78.32

References

[1] Balakrishnan N. Simple heuristics for the vehicle routeing problem with soft time windows. J Oper Res Soc 1993;44:279–87.

[2] Baldacci R, Mingozzi A, Roberti R. New route relaxation and pricing strategies for the vehicle routing problem. Oper Res 2011;59:1269–83.

[3] Baldacci R, Mingozzi A, Roberti R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. Eur J Oper Res 2012;218:1–6.

[4] Bräysy O, Gendreau M. Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transp Sci 2005;39:104–18.

[5] Bräysy O, Gendreau M. Vehicle routing problem with time windows, part II: metaheuristics. Transp Sci 2005;39:119–39.

[6] Calvete H, Galé C, Oliveros M, Sánchez-Valverde B. A goal programming approach to vehicle routing problems with soft time windows. Eur J Oper Res 2007;177:1720–33.

[7] Chiang W, Russell R. A metaheuristic for the vehicle-routeing problem with soft time windows. J Oper Res Soc 2004;55:1298–310.

[8] Cordeau J, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 1997;30:105–19.

[9] Cordeau J, Gendreau M, Laporte G, Potvin J, Semet F. A guide to vehicle routing heuristics. J Oper Res Soc 2002;53:512–22.

[10] Cordeau J, Laporte G, Mercier A. A unified tabu search heuristic for vehicle routing problems with time windows. J Oper Res Soc 2001;52:928–36.

[11] Desaulniers G, Lessard F, Hadjar A. Tabu search, partial elementarity, and generalized *k*-path inequalities for the vehicle routing problem with time windows. Transp Sci 2008;42:387–404.

[12] Fagerholt K. Ship scheduling with soft time windows: an optimisation based approach. Eur J Oper Res 2001;131:559–71.

[13] Figliozzi M. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transp Res Part C 2010;18:668–79.

[14] Fu Z, Eglese R, Li L. A unified tabu search algorithm for vehicle routing problems with soft time windows. J Oper Res Soc 2008;59:663–73.

[15] Garcia B, Potvin J, Rousseau J. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. Comput Oper Res 1994;21:1025–33.

[16] Gendreau M, Hertz A, Laporte G. A tabu search heuristic for the vehicle routing problem. Manag Sci 1994;40:1276–90.

[17] Gendreau M, Laporte G, Séguin R. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. Oper Res 1996;44:469–77.

[18] Golden B, Raghavan S, Wasil E. The vehicle routing problem: latest advances and new challenges. New York: Springer; 2008.

[19] Hashimoto H, Ibaraki T, Imahori S, Yagiura M. The vehicle routing problem with flexible time windows and traveling times. Discrete Appl Math 2006;154:2271–90.

[20] Hertz A, Laporte G, Mittaz M. A tabu search heuristic for the capacitated arc routing problem. Oper Res 2000;48:129–35.

[21] Ibaraki T, Imahori S, Kubo M, Masuda T, Uno T, Yagiura M. Effective local search algorithms for routing and scheduling problems with general time window constraints. Transp Sci 2005;39:206–32.

[22] IBM, ILOG CPLEX optimizer 12.5, ⟨http://www-01.ibm.com/software/integration/optimization/cplex-optimizer⟩; 2014.

[23] Koskosidis Y, Powell W, Solomon M. An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. Transp Sci 1992;26:69–85.

[24] Laporte G. Fifty years of vehicle routing. Transp Sci 2009;43:408–16.

[25] Min H. A multiobjective vehicle routing problem with soft time windows: the case of a public library distribution system. Socio-Econ Plan Sci 1991;25:179–88.

[26] Qureshi A, Taniguchi E, Yamada T. An exact solution approach for vehicle routing and scheduling problems with soft time windows. Transp Res Part E: Logist Transp Rev 2009;45:960–77.

[27] Qureshi A, Taniguchi E, Yamada T. Exact solution for the vehicle routing problem with semi soft time windows and its application. Proc—Soc Behav Sci 2010;2:5931–43.

[28] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 1987;35:254–65.

[29] Taş D, Dellaert N, Van Woensel T, de Kok T. Vehicle routing problem with stochastic travel times including soft time windows and service costs. Comput Oper Res 2013;40:214–24.

[30] Taillard E, Badeau P, Gendreau M, Guertin F, Potvin J. A tabu search heuristic for the vehicle routing problems with soft time windows. Comput Oper Res 1997;31:170–86.

[31] Tang J, Pan Z, Fung R, Lau H. Vehicle routing problem with fuzzy time windows. Fuzzy Sets Syst 2009;160:683–95.

[32] Toth P, Vigo D. The vehicle routing problem, SIAM monographs on discrete mathematics and applications, vol. 9, Society for Industrial and Applied Mathematics, Philadelphia; 2002.

[33] Vidal T, Crainic T, Gendreau M, Prins C. A unifying view on timing problems and algorithms. Technical report 2011-43, CIRRELT; 2011.