A New Exact Algorithm to Solve the Multi-Trip Vehicle Routing Problem with Time Windows and Limited Duration

Rodolphe Giroudeau, Oliver Naud, Florent Hernandez, Dominique Feillet

A new exact algorithm to solve the Multi-trip vehicle routing problem with time windows and limited duration

Florent Hernandez
Cemagref UMR ITAP, 361 rue JF Breton, Montpellier, France , LIRMM UMR 5506, 161 rue Ada 34392, Montpellier France,
florent.hernandez@lirmm.fr,

Dominique Feillet
Ecole des Mines de Saint-Etienne, CMP Georges Charpak, F-13541 Gardanne, France,
feillet@emse.fr,

Rodolphe Giroudeau
LIRMM UMR 5506, 161 rue Ada 34392, Montpellier France,
rodolphe.giroudeau@lirmm.fr,

Olivier Naud
Cemagref UMR ITAP, 361 rue JF Breton, Montpellier, France,
olivier.naud@cemagref.fr,

Abstract

This article tackles the multi-trip vehicule routing problem with time windows and limited duration. A trip is a timed route such that a succession of trips can be assigned to one vehicle. We provide a two-phase exact algorithm to solve it. The first phase enumerates possible ordered lists of client matching trip maximum duration criterion. The second phase uses a Branch and Price scheme to generate and choose best set of trips to visit all customers. We propose a set covering formulation as the column generation master problem, where columns (variables) represent trips. The sub-problem selects appropriate timing for trips and has a pseudo-polynomial complexity. Computional results on Solomon's benchmarks are presented. The computional times obtained with our new algorithm are much lower than the ones obtained in the sole exact algorithm previously published on this problem.

Keywords : Vehicle routing, Time windows, Multi-trip, Column Generation, Dynamic programming, Branch and Price

1 Introduction

The Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW) is a variant of the classical Vehicle Routing Problem with Time Windows (VRPTW) where vehicles can be scheduled more than one trip within a workday or planning time horizon. The multi-trip feature is needed when the vehicles fleet size is limited. In this study, we consider a special case of the MTVRPTW, called MTVRPTW-LD, where trips (routes) have a limited duration. Motivations to impose this duration limit can be management issues, e.g. limiting the maximum driving time for drivers, or can follow from the nature of transported goods, e.g. delivering perishable goods.

This problem was addressed first in Azi et al. (2007) and Azi et al. (2010). In Azi et al. (2007), the authors proposed an exact method for the single vehicle case. The multi-vehicles variant is considered in Azi et al. (2010) and updated numerical results are given in Azi (2010). We describe in the present article a new exact method for this latter problem. This method is based on Azi's investigations and on our own works on MTVRPTW reported in Hernandez et al. (2009). Using the same instances as in Azi (2010), which are based on Solomon's - Solomon (1987) - we show that the new algorithm allows large improvements in terms of computing times.

This article is organized as follows. The following section is devoted to related works on Vehicle Routing Problem with Time Windows, more specifically on the multi-trips variant problem and on the strategies to solve it. The principle of column generation used in Branch and Price schemes is described in the same section. In section 4, we present our new exact method for MTVRPTW-LD, which is composed of two phases like in Azi et al. (2010). In section 5, consistently with the choice made in Azi et al. (2010) and Azi (2010), we present results for Solomon benchmark instances, including analysis of effects of size of customer time windows and trip max duration on solutions and performance, and compare these with results obtained in Azi (2010). Following sections are devoted to general discussion, conclusion, and perspectives.

2 Literature review

As Azi et al. (2007), we note that papers about VRPTW with limited fleet and multi-trip are scarce in the literature. Most of the few works reported so far on solving this problem involved metaheuristics.

2.1 Multi-trip vehicle routing problem

As far as we know, Fleischmann (1990), as cited by Battarra et al. (2009), is the first study including the multi-trip idea in the vehicle routing problem without Time Windows. The author used a savings based algorithm to construct the routes and a bin packing heuristic to combine them on vehicles. The same principles have been used in Taillard et al. (1996). In their study, routes are constructed by a Tabu search algorithm. According to Sen and Bülbül (2008), the study in Brandão and Mercer (1998) is about a rich vehicle routing problem with multi-trips and many additional considerations like time windows, heterogeneous fleet, maximum legal driving time per day for drivers, unloading time of vehicles, etc. The authors developped a tabu search to solve this problem. In Sen and Bülbül (2008) is also mentionned the multi-phase algorithm reported in Petch and Salhi (2003), which can be considered as the combination of two approaches mentionned in Taillard et al. (1996) and Brandão and Mercer (1998). Other works analysed in Sen and Bülbül (2008) include Olivera and Viera (2007), a method based on an adaptive memory procedure, Salhi and Petch (2007) which involves a genetic algorithm to solve the MTVRPTW, and Alonso et al. (2008). In this latter paper is studied the case of periodic vehicle routing problem with time windows and multi-trips, in which one customer can be served 1 to t times in a planning period, and in which split and delivery is allowed. A tabu search is applied to solve the problem. In Battarra et al. (2009), the authors decompose the MTVRPTW into two easier problems, and create two heuristics to solve them. The first heuristic deals with the creation of routes and the second with a bin packing problem. The complete algorithm is iterative and is based on a self adaptive guidance strategy which enforces the route heuristic to compute only the routes that can improve the solution.

2.2 Exact method for vehicle routing problem with time windows and multi-trips

To our knowledge, the first exact method on MTVRPTW-LD was proposed in Azi et al. (2007). The authors considered the case of the delivery of perishable goods with a single vehicle. They created an algorithm with two phases of dynamic programming. In the first phase, dynamic programming is used to generate all non-dominated routes. A graph where the nodes represent the routes obtained in phase 1 is then created. Transitions of this graph represent the possible successions of routes. Note that the size of routes' graph is bounded thanks to the limit on route duration. In the second phase, dynamic programming is used to generate the working day for the vehicle from the routes' graph and with the dominance rule given in Feillet et al. (2004). In Azi et al. (2010), the authors considered the same problem but with an homogeneous fleet instead of a single vehicle. The first phase is similar to the single vehicle case. The second phase uses the Column Generation technique to generate the working day for each vehicle. In the Column Generation scheme, the pricing problem is in fact similar to the ESPPRC used in the second phase of Azi et al. (2007), besides cost modification implied by dual variables. Note that in both single and multi-vehicle cases, the graph of non-dominated routes is generated only once.

As our work presented here is also based on Column Generation and the Branch and Price scheme, we recall hereafter how it has been applied so far to VRPTW.

2.3 Column Generation

The Branch and Price is a Branch and Bound where the lower bound is computed by Column Generation. This technique is used to solve huge mixed integer programs. Column Generation consists in decomposing the whole problem into two simpler problems, called master problem and sub-problem. These two problems are solved iteratively. This process stops when there is no longer a solution for the sub-problem.

The first application of this method on the VRPTW is given in Desrochers et al. (1992). In the VRPTW case, the master problem is a set covering problem, and the sub-problem corresponds to an elementary shortest path problem with resource constraints (ESPPRC). The ESPPRC sub-problem is defined over a graph $G' = (N \cup \{o, d\}, E)$, where $N \cup \{o, d\}$ and E are the sets of nodes and arcs, respectively. N is the set of nodes for each customer in $V \setminus \{v_0\}$. Nodes o is the node for the depot 0 at the beginning of a route and d is the node for 0 at the route end. The set E contains arcs (o, j), $\forall j \in N$; arcs (i, d), $\forall i \in N$; and arcs (i, j), $\forall i, j \in N$ such that customer j can be visited after customer i by at least one feasible route. A cost c_{ij} is associated with each arc $(i, j) \in E$.

Each feasible route is represented by a path in G'. The following $|N|+2$ resource constraints are needed on the ESPPRC, so that time windows are encountered and vehicle capacity is not exceeded: the time t, the vehicle load q, and V^i for each customer $i \in N$ indicating if the customer i has been visited along the path. The resource intervals are the customer time windows for t, $[0, Q]$ for q, where Q is the vehicle capacity, and $[0, 1]$ for each V^i. This set of resources is denoted by $R = \{t, q, V^1, \cdots, V^{|N|}\}$.

In order that the Column Generation scheme will produce an exact solution, the ESPPRC needs to be solved exactly. Since the ESPPRC is $\mathcal{NP}$-hard (see Dror (1994)), Desaulniers et al. (2008) proposed a metaheuristic alternative to find new columns. When none can be found by the metaheuristic, then the exact algorithm for ESPPRC is called.

3 Multi-trip vehicle routing problem with time windows and limited duration

Formally, the MTVRPTW-LD is defined as follows. Let $G = (V, A)$ be a directed graph where $V = \{0, \cdots, n\}$ and A is the set of arcs (i, j). 0 represents the depot and $1, \cdots, n$ the customers. A cost c_{ij} and a travel time t_{ij} are attached to each arc $(i, j) \in A$. The fleet comprises U vehicles, all with same load capacity Q. Let $[0, T]$ be the planning time horizon, and t_{max} the duration limit of a trip. For each $i \in \{1, \cdots, n\}$ is defined a demand d_i and a service time st_i. Each client must be served within a time window $[a_i, b_i]$ with $a_i, b_i \in [0, T]$. However, vehicles can arrive at a client i earlier than a_i and wait. The problem is to find a set of trips with the lowest cost and using at most U vehicles, and such that (i) all customers are served, (ii) two trips cannot be travelled at the same time by the same vehicle, (iii) loads comply with the capacity of vehicles and (iv) time constraints at the clients and depot are met. In this case, the *trip duration* is the elapsed time between the depot departure time, after the vehicle has been loaded, and the arrival time to the last customer of the trip, before the delivery. The schedule of a vehicle must also include, in the *complete trip duration*, the loading time, the service time to last customer and return time to depot.

We can model the MTVRPTW-LD with the following MIP:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \tag{1}$$

subject to

$$\sum_{\{j \in V | (i,j) \in A\}} x_{ij}^k = \delta_i^k, (i \in V \setminus \{0\}, k \in K) \tag{2}$$

$$\sum_{k \in K} \delta_i^k \geq 1, (i \in V \setminus \{0\}) \tag{3}$$

$$\sum_{\{j \in V | (i,j) \in A\}} x_{ij}^k - \sum_{\{j \in V | (i,j) \in A\}} x_{ji}^k = 0, (i \in V, k \in K) \tag{4}$$

$$\sum_{\{i \in V | (0,i) \in A\}} x_{0i}^k \leq 1, k \in K \tag{5}$$

$$\sum_{(i,j) \in A} d_i x_{ij}^k \leq Q, k \in K \tag{6}$$

$$\alpha^k = \beta \sum_{\{i \in V\}} st_i \delta_i^k, k \in K \tag{7}$$

$$S_i^k + st_i + t_{ij} - S_j^k + M x_{ij}^k \leq M, (i, j) \in A, i, j \neq 0, k \in K \tag{8}$$

$$S_i^k + st_i + t_{i0} - d_k^{back} + M x_{i0}^k \leq M, (i, 0) \in A, k \in K \tag{9}$$

$$d_k^{start} + \alpha^k + st_0 + t_{0i} - S_i^k + Mx_{0i}^k \leq M, (0,i) \in A, k \in K \tag{10}$$

$$S_i^k \leq d_k^{start} + \alpha^k + t_{max}, i \in V, k \in K \tag{11}$$

$$\sum_{\{i \in V | (0,i) \in A\}} x_{0i}^k - \sum_{u \in U} \sigma_k^u = 0, k \in K \tag{12}$$

$$\sigma_k^u + \sigma_l^u - y_{kl}^u - y_{lk}^u \leq 1, k, l \in K, k \neq l, u \in U \tag{13}$$

$$1 - y_{kl}^u - y_{lk}^u \geq 0, k, l \in K, k \neq l, u \in U \tag{14}$$

$$d_k^{back} - d_l^{start} + My_{kl}^u \leq M, k, l \in K, u \in U \tag{15}$$

$$a_i \delta_i^k \leq S_i^k \leq b_i \delta_i^k, i \in V, k \in K \tag{16}$$

$$a_0 \leq d_k^{start} \leq b_0, k \in K \tag{17}$$

$$a_0 \leq d_k^{back} \leq b_0, k \in K \tag{18}$$

$$0 \leq d_k^{start} \leq T, k \in K \tag{19}$$

$$0 \leq d_k^{back} \leq T, k \in K \tag{20}$$

$$x_{ij}^k \in \{0,1\}, (i,j) \in A, k \in K \tag{21}$$

$$y_{kl}^u \in \{0,1\}, k, l \in K, u \in U \tag{22}$$

$$\sigma_k^u \in \{0,1\}, k \in K, u \in U \tag{23}$$

$$\delta_i^k \in \{0,1\}, i \in V \setminus \{0\}, k \in K \tag{24}$$

$$\alpha^k \geq 0, k \in K \tag{25}$$

where x_{ij}^k, σ_k^u, y_{kl}^u, δ_i^k, S_i^k, α^k, d_k^{start} and d_k^{back} are the decision variables. x_{ij}^k indicates if the arc (i,j) is in trip r_k or not, σ_k^u indicates if trip r_k is traveled by vehicle u, y_{kl}^u indicates whether trip r_l is traveled after trip r_k by vehicle u or not, δ_i^k indicates if customer i is visited by trip r_k. For a customer i visited by a trip r_k, S_i^k is the starting time of service. α^k, d_k^{start} and d_k^{back} are the loading time, the starting time of service and the arrival time of trip r_k to the depot, respectively.

K is the maximal number of trips needed to guarantee the coverage of all client nodes in the optimal solution. In this case, K is the number of customers.

Constraints (2) and (3) enforce the visit of every customer. Customers are allowed to be visited more than once. This relaxation is valid since, due to triangle inequality of distances, it is not optimal to visit a customer more than once. Constraints (4)-(5) define the trip structure and constraints (6) concern vehicle capacity. Constraints (7) define the vehicle loading time as the sum of the service times of all customers in a trip, multiplied by a given coefficient β. Constraints (8)-(10) and (16)-(18) concern the compliance of trips to time windows constraints. Note that in the solution, subtours are forbidden by previous inequalities. Constraints (19)-(20) concern the respect of planning horizon. Constraints (11) correspond to the deadline constraint for serving a customer. Note that constraints (16) ensure that S_i^k is set to 0 when customer i is not in trip r_k, and, consequently, constraint (11) is automatically satisfied in this case. Constraints (12)-(15) order the routes on available vehicles.

4 A new exact method for the MTVRPTW-LD

4.1 Introduction, formulation and definitions

In practice this formulation is not tractable for any instances of reasonable size and its linear relaxation is very weak. Thus we propose a two-phase algorithm to solve it, where, as in Azi et al. (2010), the first phase enumerates possible routes, and the second phase uses a Branch and Price scheme to choose best covering set of routes. We will require following definitions.

Definition 4.1 ***Structure***
A structure is an ordered list of customers than can be visited during a trip while satisfying their time constraints. A structure has a cost c_k*, a travel distance* D_k *and a minimal complete trip duration* d_k^{min} *needed to visit these customers and come back to depot, in this order.*

Definition 4.2 ***Trip timed structure***
A trip timed structure is a structure with a time window $[\mathcal{A}_k, \mathcal{B}_k]$ *that can be calculated such that* $\mathcal{A}_k$ *(*$\mathcal{B}_k$*, respectively) is the earliest departure time (latest arrival time, respectively) permitting to visit all customers of* s_k *and back to the depot with exactly the duration* d_k^{min}*. It will be hereafter simply denominated "structure".*

Definition 4.3 ***Trip***
A trip r_k *is defined as a structure* s_k *associated to a fixed starting time* $d_k^{start} \geq \mathcal{A}_k$*. Because* d_k^{min} *is an attribute of structures, the arrival time of a trip* $d_k^{start} + d_k^{min}$ *is also known. We will say that a trip has a fixed time position.*

4.2 Enumeration phase

As long as the duration limit is relatively short, it is possible to generate all the non-dominated structures (see Azi et al. (2010)). This problem is addressed via an approach that exploits an algorithm solving the elementary shortes path problem with resource constraints given in Feillet et al. (2004).

This algorithm consists in extending labels from one node to another through the graph G' defined in section 2.3. Each label represents a partial feasible path from the depot to one customer. To initialize the labelling process, one label is created on node o. This label is then extended to all successors of node o. Nodes are iteratively treated until no new labels are created. When a node is treated, all its new labels are extended towards every possible successor node. Once a label has been extended, its resource intervals are verified and the label is rejected if infeasible.

This basic method generates many labels. In order to decrease the number of generated labels, a label dominance relation is applied during the solution process on the generated labels associated with the same node.

For the classic VRPTW, a path k from node o to node j is labeled with L_k. L_k is defined by $|N|+4$ parameters represented by the vector $L_k = \{c_k, j, T_k^t, T_k^q, V_k^1, \cdots, V_k^{|N|}\}$, where c_k is cost of this partial path, j is the node to which the label is attached, T_k^t and T_k^q are the accumulated values of time and load, respectively, and $V_k^i = 1$ if node i is unreachable, 0 otherwise.

The dominance relation for VRPTW is as follows: If k and k' are two different paths from node o to node j with labels L_k and $L_{k'}$, respectively, then path k dominates k' if and only if $c_k \leq c_{k'}$, $T_k^t \leq T_{k'}^t$, $T_k^q \leq T_{k'}^q$ and $V_k^i \leq V_{k'}^i$, $\forall i$.

That is, path k dominates k' if its cost c_k is not greater, does not consume more resource for every resource considered, and every unreachable node is also unreachable for k'. As stated in Feillet et al. (2004), it is guaranteed that no potential optimal solution can be eliminated by this dominance relation.

In our case, the problem is not to find an elementary shortest path within the whole graph G' as it is for the VRPTW case. We need to find an elementary shortest path for each subset of customers that can be visited without violating constraints. This is why we adapted this algorithm, mainly by modifying resources and dominance rules, as explained hereunder.

Resources and dominance rule:

First, like in Azi et al. (2007), the cost c_{ij} on each arc is replaced by $c_{ij} - (max_{(i,j)\in A} c_{ij} + 1)$. The aim is to generate all feasible non dominated routes. In order to do this, we define the labels as follows:

Definition 4.4 ***Label.***
A path p from the origin 0 to node j is labeled with $L_p = \{c_p, j, h_p, q_p, d_p^{min}, \mathcal{A}_p, \mathcal{B}_p, W_p^1, \cdots, W_p^n\}$, where c_p is the reduced cost of this path, h_p and q_p are the values of time and load resources, respectively, accumulated along this path; d_p^{min} is the minimal trip duration of the path represented by L_p; $\mathcal{A}_p$ and $\mathcal{B}_p$ are the start and end of the label time window as specified in definition 4.2; and $W_p^i = 1$ if node i is visited by L_p, 0 otherwise.

During the extension of label L_p from a node i to j, to obtain $L_{p'}$, the label ressources are updated as follows:

- $c_{p'} = c_p + c_{ij}$ where c_{ij} is the cost of arc (i, j)
- $h_{p'}$ is calculated by adding all loading, service and travel times along the path from 0 to j. If $h_{p'} < a_j$, then $h_{p'}$ is set to a_j (waiting is allowed).
- $W_{p'}^j = 1$ and $W_{p'}^g = W_p^g$, $\forall g \in V' \setminus j$
- To compute the minimal trip duration $d_{p'}^{min}$ of $L_{p'}$, the waiting time is reduced as much as possible by delaying the departure time from depot to the latest possible date.
- To compute the time window of label $L_{p'}$, the maximum advancement and the maximum retardation of the label is computed, such that none of time constraints at customers is violated. The updated start and end of label's time window are thus obtained.

As for dominance, we use the following relation:

Definition 4.5 ***Dominance relation.***
If p and p' are two different paths from origin 0 to node j with labels L_p and $L_{p'}$, respectively, then p dominates p' if and only if the nodes visited by p and by p' are the same ($W_p^i = W_{p'}^i$ for every customer i), the time window of L_p includes the time window of $L_{p'}$ ($\mathcal{A}_p \leq \mathcal{A}_{p'}$ and $\mathcal{B}_p \geq \mathcal{B}_{p'}$), and $c_p \leq c_{p'}$, $h_p \leq h_{p'}$, $q_p \leq q_{p'}$, $d_p^{min} \leq d_{p'}^{min}$.

That is a path p dominates a path p' if (i) its cost c_p is not greater, (ii) it does not consume more resource for every resource considered, (iii) it visits the same customers and (iv) it has at least the same temporal positions.

Lemma 4.1 *If label L_1 dominates label L_2 then for all labels L_4 extended from L_2 there is a label L_3 which dominates label L_4.*

Proof:
Let L_1 dominates L_2 at node j. Then, we know that these two labels visit the same customers, the time window of L_1 includes the time window of L_2, $c_1 \leq c_2$, $h_1 \leq h_2$, $q_1 \leq q_2$ and $d_1^{min} \leq d_2^{min}$. For every feasible label L_4 arriving at node g at time h extended to L_2, there exists a feasible label L_3 arriving at node g at time h extended to L_1, such that the nodes visited by L_3 after the node j are the same, and are visited in the same order, than the nodes visited by L_4 after the node j. If there was no label L_3 with these properties, then either $h_2 < h_1$, the time window of L_1 would not include the time window of L_2 or $d_2^{min} < d_1^{min}$, and thus L_1 would not dominate L_2. We note by *path* the partial path between j and g. The resource consumptions on this partial path are the same. Thus $q_3 \leq q_4$ and L_3 and L_4 visit the same customers.

The reduced cost c_{path} and the minimal trip duration d_{path}^{min} along the path from j to g are the same for L_3 and L_4 because the nodes are visited in same order and at the same times. Thus, we can consider that the reduced cost of L_3 is equal to $c_1 + c_{path}$ and that the reduced cost of L_4 is equal to $c_2 + c_{path}$. Consequently we have $c_3 \leq c_4$.

We can also consider that the minimal trip duration of L_3 is equal to $d_1^{min} + d_{path}^{min} + d_{wait_3}^{min}$ and that the minimal trip duration of L_4 is equal to $d_2^{min} + d_{path}^{min} + d_{wait_4}^{min}$ where $d_{wait_3}^{min}$ (resp. $d_{wait_4}^{min}$) is the minimal waiting time necessary to connect the path represented by L_1 (resp. L_2) and the path *path* for L_3 (resp. L_4).

We know that, due to the dominance, $d_1^{min} + d_{path}^{min} \leq d_2^{min} + d_{path}^{min}$. The question is to know if $d_{wait_3}^{min} \leq d_{wait_4}^{min}$. We know that the time window of L_1 includes the time window of L_2 and thus the arrival time to node j of L_1 can be delayed until the arrival time to node j of L_2. Consequently the arrival time to node j of L_3 can be delayed until the arrival time to node j of L_4 and it follows that $d_{wait_3}^{min} \leq d_{wait_4}^{min}$ and $d_1^{min} + d_{path}^{min} + d_{wait_3}^{min} \leq d_2^{min} + d_{path}^{min} + d_{wait_4}^{min}$.

Once cost and route duration have been checked, the last concern to obtain proof of our dominance relation lemma is time windows. We know that $d_1^{min} \leq d_2^{min}$ and $[\mathcal{A}_2, \mathcal{B}_2] \subseteq [\mathcal{A}_1, \mathcal{B}_1]$ so if we can consider that $[\mathcal{A}'_1, \mathcal{B}'_1] = [\mathcal{A}_2, \mathcal{B}_2]$ as the time window for L_1 then if we add *path* to L_2 and *path* to L_1 with this time window, we obtain $[\mathcal{A}'_3, \mathcal{B}'_3] = [\mathcal{A}_4, \mathcal{B}_4]$ as time windows for L_3 and L_4. Since, we have $[\mathcal{A}_2, \mathcal{B}_2] = [\mathcal{A}'_1, \mathcal{B}'_1] \subseteq [\mathcal{A}_1, \mathcal{B}_1]$ thus the time windows of L_3 will include $[\mathcal{A}'_3, \mathcal{B}'_3]$ and we will have $[\mathcal{A}_2, \mathcal{B}_4] \subseteq [\mathcal{A}_3, \mathcal{B}_3]$.

Thus we have $c_3 \leq c_4$, $h_3 \leq h_4$, $q_3 \leq q_4$ and $d_3^{min} \leq d_4^{min}$ and the time window of L_3 includes the time window of L_4 then label L_3 dominates L_4.

◇◇◇

The labeling and dominance process can be illustrated with the example given in Figure 1. In this exemple, the node D represents the depot and nodes 1,2,3,4 the customers. α is equal to 3 and the costs marked on arcs take this value into account. The demand and the service time at each customer are set to 0.

Let us compare two labels L_1 and L_2, with the following values (see definition 4.4): $L_1 = \{-5,3,6,0,4,2,8,1,1,1,0\}$, $L_2 = \{-5,3,6,0,4,2,6,1,1,1,0\}$. L_1 and L_2 visit respectivly nodes 2,1,3 and nodes 1,2,3, in these orders. Thanks to our dominance relation, L_1 dominates L_2 at node 3. In this example, if we extend L_2 to node 4, we obtain a label L_4 such that $L_4 = \{-7,4,7,0,5,2,7,1,1,1,1\}$ and L_4 visits nodes 1,2,3,4 in this order. If we extend L_1 to node 4, we obtain a label L_3 such that $L_3 = \{-7,4,7,0,5,2,9,1,1,1,1\}$ and which visits nodes 2,1,3,4 in

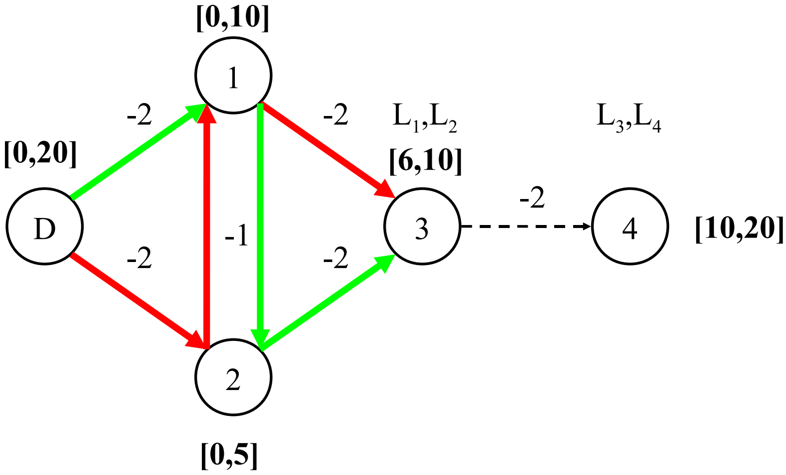

Figure 1: Illustration of dominance relation

this order. L_3 visits the same nodes as L_4, it has the same cost, time, load and minimal trip duration as L_4 but its time window includes the time window of L_4. Thus L_3 dominates L_4.

This lemma ensures that the dominance relation, defined in Definition 4.5, does not delete labels that could potentially contribute to the optimal solution.

Please note that the atteignability of clients is implicitly included in our definitions, thanks to the combination of explicit resources.

4.3 Column Generation

The second phase of the algorithm is based on Column Generation and Branch and Price. We propose a set covering formulation where columns (variables) represent trips (see Definition 4.3).

Master problem formulation:

The itinerary to planify for a vehicle consists in a set of successive trips. Two trips of a given vehicle cannot overlap in time. We partition the planning horizon in a set of time intervals Δ_t. During each Δ_t, at most U vehicles should be used. Δ_t is defined by $\Delta_t = [l_{min} * t, l_{min} * (t+1)[$ where l_{min} is a small value guaranteeing that the duration of any trip will be greater than l_{min}, and $t \in \{0, \cdots, \lfloor \frac{T}{l_{min}} \rfloor\}$. The set of variables is denoted Ω. With these assumptions we can formulate the master problem as follows:

$$z(\Omega) = minimize \sum_{r_k \in \Omega} c_k \theta_k \tag{26}$$

subject to

$$\sum_{r_k \in \Omega} a_{ik} \theta_k \geq 1 \qquad (v_i \in V \setminus \{v_0\}), \tag{27}$$

$$\sum_{r_k \in \Omega} b_{tk} \theta_k \leq U \qquad (\forall \Delta_t), \tag{28}$$

$$\theta_k \geq 0 \qquad (r_k \in \Omega), \tag{29}$$

where c_k is the cost of trip r_k, a_{ik} indicates whether customer i is visited by trip r_k or not, $b_{tk} \in [0, 1]$ is the fraction of the time interval Δ_t occupied by trip r_k and θ_k are decision variables. Constraints (27) enforce that every customer is visited at least once. For each trip r_k, $\sum_t b_{tk} > 1$. This is true because the length of any interval is inferior to the duration of any trip. Thus, constraints (28) enforce that at most U vehicles are used during any time interval.

In practice, this model contains a huge number of variables. We propose to solve it using a Branch and Price algorithm. Branch and Price is a special case of Branch and Bound where bounds are computed using a Column Generation technique. The Column Generation principle, explained in Feillet (2007) (see also Desrochers et al. (1992)), consists in decomposing the whole problem into a master problem and a subproblem. This technique is an iterative process that alternately solves a restricted master problem and a subproblem.

For each iteration w, the restricted master problem corresponds to the linear relaxation of the model restricted to a subset Ω_w of its variables Ω. It is solved by a simplex algorithm to provide primal and dual variable values.

The subproblem consists in finding the negative reduced cost variables in the master problem variable set. These variables are added to the variable set of the restricted master problem before beginning another iteration. Only trips r_k with a negative reduced cost can possibly decrease the cost of the current solution. The process stops when the subproblem cannot generate any negative reduced cost variable.

The dual of the master problem is given hereafter:

$$y(\Omega) = maximize \sum_{i \in V \setminus \{0\}} \lambda_i - U \sum_t \mu_t \tag{30}$$

subject to:

$$\sum_{i \in V \setminus \{0\}} a_{ik}\lambda_i - \sum_{\Delta_t} b_{tk}\mu_t \geq c_k (r_k \in \Omega) \tag{31}$$

$$\lambda_i \geq 0 \tag{32}$$

$$\mu_t \geq 0 \tag{33}$$

The subproblem:

The subproblem consists in finding trips (structure fixed in time) with a negative reduced cost $c_k - \sum_{i \in V \setminus \{0\}} a_{ik}\lambda_i + \sum_{\Delta_t} b_{tk}\mu_t$, where λ_i and μ_t are dual variables respectively corresponding to primal constraints (27 and 28). For the set of trips corresponding to a common given structure, only time position varies and affects the reduced cost. Every non-dominated structure has been previously enumerated (set S). Thus, the subproblem consists in finding, for every structure s_k, new trips, which are generated by selecting a time position in the time window $[\mathcal{A}_k, \mathcal{B}_k]$ and kept as new columns only if their reduced cost is negative. In fact, for a given structure, only the trip with the lowest negative reduced cost is kept as new column.

In order to find this time position, we have created a scheduling sub-algorithm. For each structure s_k in S, our algorithm translates s_k in its time window by unit time steps and computes the reduced cost of associated trip. Please note that the length of unit time step corresponds to the time granularity of the instance and is not to be confused with the length of a time interval. As soon as a negative reduced cost is obtained, translation stops. Otherwise translation stops when all possible temporal positions within the time windows of s_k have been tried. This algorithm has a polynomial-time complexity.

When no such columns can be found, the Column Generation process stops.

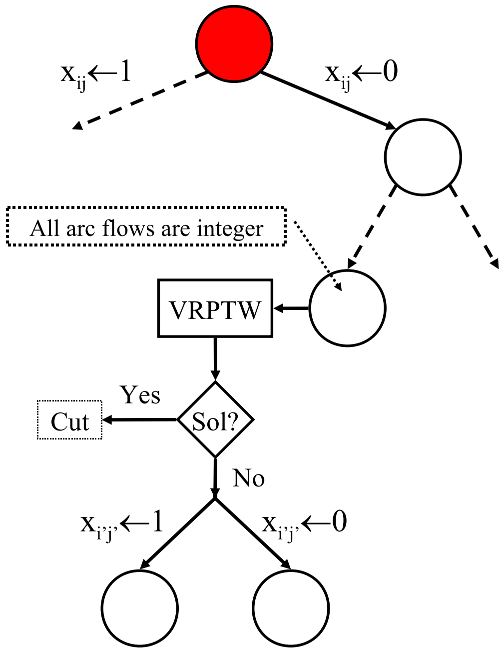

Figure 2: Branching strategy

Initialization process:

An initial solution is required in order to start the Column Generation process. Finding an initial solution might be not trivial or even impossible, as the fleet is limited. We thus create an initial "Super-trip" that visits all customers and associate a great cost to this trip. The Super-trip is generally not feasible for the real problem but it satisfies the master problem.

Definition 4.6 ***Super-trip*** *Let r^* be the trip with cost $c^* = 2\sum_{r_k \in \Omega} c_k$. r^* contains all the customers, its time windows is equal to the depot time windows and all b_{tk} are equal to the number of vehicles allowed.*

4.4 Branch and Price scheme

Column Generation is applied at each node of a search tree generated by the Branch and Price algorithm to compute a lower bound. Classically, when solving VRPTW, the branching is made on arcs' flows. This scheme is used because the alternative to branch on selection of routes for the clients' set covering would often cause the branching tree to be badly equilibrated which would result in poor performance. Our algorithm also involves branching on arcs. Nervertheless, in our case and unlike for VRPTW, because of temporal constraints, having all arcs with an integer flow does not imply that the solution (set of θ_ks) is integer. Then, once all arc flows are integer on a given node, we apply a repair strategy, which provides a simple rescheduling functionality without changing solution cost and which is explained below. If the repair algorithm does not succeed, we branch on arcs that have not been directly forced by the branching till then but were made integer as a consequence of the optimization process. Figure 2 illustrates this principle.

Branching scheme on arcs:

The branching is applied on an arc (i, j) if its flow is fractional. Two branches are created: one branch with $x_{ij} = 1$, where customer j must be visited immediately after customer i, and one branch with $x_{ij} = 0$ otherwise. In the first case, all arcs (k, j) and (i, k), with $k \neq j \neq i$, are forbidden and all corresponding x_{kj} and x_{ik} are fixed to 0. To update the set of variables of the restricted master problem, all θ_k representing the route r_k that use an arc (i, j) with the associated $x_{ij} = 0$, are set to zero.

Repair strategy:

When all arc flows are integer, the set of arcs containing units of flow is the same as the set of arcs which compose the structures (see Definition 4.2) of trips selected in current candidate solution. In this case and when the candidate solution is fractional, at least one of these structures appears at least twice in the set of selected trips, at two different temporal positions. The repairment consists in rescheduling in order to obtain an integer solution for this node if there is one.

From the scheduling theory standpoint, we may define the problem as a set of N tasks $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$ such each $\mathcal{T}_i$ (representing the structure s_i) admits a release date r_i ($\mathcal{A}_i$, the beginning of time window of the structure s_i) a deadline $\tilde{d}_i$ ($\mathcal{B}_i$, the end of time window of structure s_i) and a duration p_i (d_i^{min}, the minimal trip duration for structure s_i). So, the starting time of $\mathcal{T}_i$ must be after the release date and the completion time must be before $\tilde{d}_i$. For this scheduling analysis, let us restrict the problem to a single vehicle case, or equivalently for scheduling, to a single processor. Let $\mathcal{T}_i$ and $\mathcal{T}_j$ be two tasks then $[t_i, t_i + p_i] \cap [t_j, t_j + p_j] = \varnothing$ with t_i designs the starting time of task $\mathcal{T}_i$. The aim is to find a feasible schedule. We will suppose that each task admits a height of size one. We denote this problem as Π.

The integer constraints are relaxed: at time t, $x\%$ of a route k may be consumed by a vehicle u, and $y\%$ by another vehicle u'. Then, we define the fractional problem of Π as follows:

Definition 4.7 *A feasible fractional solution of the problem Π is a relaxation on the height integer value: $\exists$ $\mathcal{T}_i$ such that*

- $t_{i_1} \leq t_{i_2} \leq \ldots \leq t_{i_k}$ *such that* $r_i \leq t_{i_1}$ *and* $t_{i_k} + p_i \leq \tilde{d}_i$;
- *Let* $h_{i_1}, h_{i_2}, \ldots, h_{i_k}$ *be the fractional height of the tasks* $\mathcal{T}_i$ *with* $\sum_{j=1}^{k} h_{i_j} = 1$.

Notice that with this definition several tasks may be executed at time t. Nevertheless, the sum of height-tasks processed at time t cannot be greater than one.

Remark: A feasible fractional solution may not imply a feasible integer solution. To demonstrate this, let us consider the following instance:

- $\mathcal{T}_1$ with $p_1 = 2, r_1 = 6$ and $\tilde{d}_1 = 10$;
- $\mathcal{T}_2$ with $p_2 = 2, r_1 = 1$ and $\tilde{d}_2 = 5$;
- $\mathcal{T}_3$ with $p_1 = 6, r_1 = 0$ and $\tilde{d}_3 = 11$;

Consider the feasible fractional solution given by the Figure 3.

- $t_{1_1} = 6, t_{1_2} = 8$
- $t_{2_1} = 1, t_{2_2} = 3$
- $t_{3_1} = 0, t_{3_2} = 5$
- and $\forall i, j, h_{i_j} = 1/2$

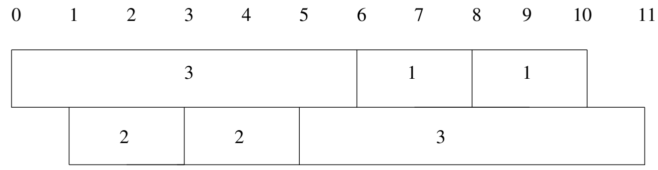

Figure 3: A counter-example for the non-existence of a feasible integer solution from a feasible fractional solution

There is no feasible integer solution:

- If the first scheduled task is $\mathcal{T}_3$ then the constraints cannot be respected for task $\mathcal{T}_2$.
- If the first scheduled task is $\mathcal{T}_2$ then the lower bound for the starting time for task $\mathcal{T}_1$ is 3. So the completion time of the task $\mathcal{T}_3$ is at least 9. The deadline for $\mathcal{T}_1$ is not respected.

The complexity of assessing the existence of a feasible integer solution is given by Theorem 4.1.

Theorem 4.1 *The problem of assessing the existence of an integer feasible solution for the problem* Π *is* $\mathcal{NP}-$*complete.*

Proof:
The problem of existence of a feasible integer solution is $\mathcal{NP}-$complete. Indeed, the problem Π is exactly the same as the scheduling problem denoted by $1|p_i, r_i, \tilde{d}_i|C_{max}$[1] for the minimization of $C_{max} = \max_i\{t_i + p_i\}$ which is proved $\mathcal{NP}-$complete, see Lenstra et al. (1977).

◇◇◇

Note that there exists a Branch and Bound algorithm for solving $1|p_i, r_i, \tilde{d}_i|C_{max}$ problem, see Bratley et al. (1971) and Blazewicz et al. (2007). We do not hold this method back, since the aim is to develop a method for a constant limited number of vehicles not only for a single vehicle.

In fact, this scheduling problem is similar to the VRPTW. At this node of the Branch and Price tree, there is a set of structures with their time windows and durations. Note that, unlike the VRPTW customers, the structure must be processed before the end of its time window $[\mathcal{A}_k, \mathcal{B}_k]$ thus the end is replaced by $\mathcal{B}_k - d_k^{min}$. With this simple modification, we have a VRPTW instance in which customers correspond to timed structures, customer demands are equal to 0 and distances (resp. costs) between two customers i and j are equal to the travel distance of structure s_j (resp. the cost of structure s_j). In order to solve this VRPTW, we use the Branch and Price algorithm described in Desrochers et al. (1992). Note that the Branch and Price algorithm is very fast. This is due to the fact that the cost of the optimal solution at root node is the same as the cost of the optimation solution for the node where an integer solution is found.

[1]In scheduling theory, the problems are characterized by the *three fields* notation scheme $\alpha|\beta|\gamma$, proposed by Graham et al. (1979), where α designates the environment processors, β the characteristics of the job and γ the criteria.

Remark about pruning:

The way we have defined the initial solution of master problem (Super-trip) allows to prune some branches during the Branch and Price process thanks to the following lemma:

Lemma 4.2 *If the Super-trip r^* is present in the optimal Master Problem solution at the current node, then there is no integer solution that does not contain the trip r^* in the sub-trees of this node.*

The proof of this lemma is out of the scope of this paper and is given in Hernandez (2010).

From this Lemma, it is possible to prune the branching tree when the subtree of the current node does not contain an integer solution other than the Super-trip. Our algorithm implements this pruning condition which has proved effective during the tests on Solmon's instances.

5 Results

5.1 Presentation

In this section, many results obtained with our exact algorithm are reported. First, the test instances of Solomon are presented, and we give the results for these instances, with 25 and 50 customers. Then, we analyse the impact of phase 1 dominance rule on Solomon's instances with 25 customers. The impact of limit duration and time windows is also investigated. The computing platform is a Pentium 4 2.0 GHz with 2 GB of RAM using GLPK to solve the master problem. The comparison with results in Azi (2010) is given in the following section.

5.2 Test instances

Our tests were performed using the well-known VRPTW benchmark instances created by Solomon (1987). These instances are divided into six classes that are a combination of two criteria. The first criterion concerns the spatial position of customers. There are three different spatial layouts: customers in clusters (”C” type), customers randomly located (”R” type), and intermediate case with part of customers clustered and the rest randomly located (”RC” type). The second criterion concerns the tightness of time constraints at customers. There are two types: tight time windows and small planning time (type ”1”) and large time windows and planning time (type ”2”). Combining these two criteria, there are six basic classes which are denominated: ”C1”, ”C2”, ”R1”, ”R2”, ”RC1” and ”RC2”. In total, there are 56 instances with 100 customers. Please note that for a given class, there are several instances with different customer time windows, but the spatial layout of customers is the same for the whole class. Instances are denoted as in following example: C201-25 correspond to the fist instances of class ”C2” where only the first 25 customers are considered. In this study, instances with tight planning time horizon were discarded. In fact, the short horizon does not allow a significant number of routes to be affected to the same vehicle. For these results, due to the difference between the service time of the instances of class C2 and the service time of instances of classes R2 and RC2, two different t_{max} values were tested. For instances of classes R2 and RC2, t_{max} were set to 75 and 220 for instances of class C2. Finally, for all tests, the parameter β for the trip loading time was set to 0.2 and the limit computation time was fixed to 30 hours.

5.3 Results on Solomon's benchmark

We close 25 of the 27 Solomon's instances with 25 customers and large time horizon and 2 vehicles and 22 of the 27 with 50 customers with large time horizon and 4 vehicles. The last 7

Instance	Root Solution	Solution	Total Time	Root time	Phase1 time	iter	column
c201-25	646.51	659.02	1.561	0.046	0	64	124
c202-25	634.772	653.37	45.819	0.499	0.031	564	555
c203-25	626.017	646.4	247.189	1.795	0.031	1133	854
c204-25	592.06	602.46	252.825	5.09	0.047	673	1268
c205-25	607.913	636.39	38.325	0.141	0	1209	304
c206-25	603.333	636.39	637.612	0.296	0.015	17034	576
c207-25	588.783	603.22	98.273	0.718	0.015	1159	725
c208-25	597.348	613.2	38.154	0.39	0.015	580	484
r201-25	757.79	762.43	0.234	0.046	0	8	174
r202-25	645.78	645.78	0.796	0.453	0.031	7	635
r203-25	620.177	621.97	2.216	0.89	0.078	12	859
r204-25	575.655	579.68	5.026	1.561	0.062	20	1005
r205-25	626.48	634.09	0.827	0.202	0.015	21	375
r206-25	596.74	596.74	0.686	0.515	0.031	4	713
r207-25	583.658	585.74	3.496	0.577	0.046	18	783
r208-25	575.616	579.68	6.681	1.53	0.047	23	1181
r209-25	598.107	602.39	1.67	0.281	0.016	22	562
r210-25	620.293	636.15	7.914	0.359	0.031	73	870
r211-25	568.54	575.91	25.805	1.295	0.046	198	1538
rc201-25	984.438	988.05	1.373	0.11	0	72	342
rc202-25	837.557	881.49	25.664	0.483	0.031	894	880
rc203-25	705.217	749.15	64.271	0.327	0.031	814	1088
rc204-25	-	-	-	-	-	-	-
rc205-25	808.579	840.35	3.746	0.249	0.015	137	598
rc206-25	726.097	761.03	35.703	0.281	0	2006	767
rc207-25	646.457	738.87	71807.37	0.859	0.031	452858	8212
rc208-25	-	-	-	-	-	-	-

Table 1: Results on the Solomon's benchmark (25 customers) with t_{max} value set (75; 220)

instances have not been solved within a limit on computing time set to 30 hours.

Tables 1 and 2 present these results. For each instance, we have the root solution cost and the solution cost of the Branch and Price scheme used in Phase 2, the computation time for both phases (Total time), the computation time of Branch and Price root in Phase 2 (Root time), the computation time of Phase 1 (Phase1 time), the number of iteration (iter) and the number of generated columns (column).

We can note, as for the VRPTW, there is great variation for time resolution between instances of the same class. We can also note a significant increase of total computation time between the instances with 25 and 50 customers.

5.4 Impact of the dominance rule in phase 1

In this section, we evaluate the impact of the dominance rule when feasible trips are generated with the elementary shortest path algorithm with ressource constraints in Phase 1. The number of available vehicles (U) was set to 2 for these tests.

In Table 3, for each instance, we compare the following criteria : the number of generated structures in Phase 1 (# trips), the computation time for both phases (Total time) and the computation time of Phase 1 (Phase1 time), with (Dom) and without (No Dom) dominance

Instance	Root Solution	Solution	Total Time	Root time	Phase1 time	iter	column
c201-50	1309.63	1324.32	1.912	0.17	0.016	14	343
c202-50	1280.44	1310.79	6067.16	1.882	0.25	12085	1635
c203-50	1236.3	1247.77	386.395	9.503	0.5	176	2681
c204-50	1181.61	1195.51	3351.04	63.501	0.796	620	6824
c205-50	1245.19	1265.61	771.309	0.4	0.031	3310	683
c206-50	1241.5	1262.47	6121.5	0.781	0.063	14776	1025
c207-50	1203.8	1216.24	1675.43	3.434	0.187	2038	1624
c208-50	1231.31	1249	4781.94	1.201	0.109	8830	1284
r201-50	1397.07	1405.52	6.529	0.19	0.109	69	699
r202-50	1221.82	1229.91	86.394	9.583	0.766	101	3791
r203-50	1101.63	1104.51	67.246	18.646	2.156	30	5314
r204-50	1010.65	1031.72	22044.9	45.245	5.266	9733	7452
r205-50	1219.64	1230.26	63.471	2.032	0.297	228	1930
r206-50	1150.62	1154.53	34.229	16.453	1.625	15	4887
r207-50	1086.15	1094.83	830.624	26.448	3.016	481	6056
r208-50	1010.65	1031.72	28145.3	57.142	5.375	11521	9702
r209-50	1126.47	1143.91	1619.44	8.342	0.765	2552	3629
r210-50	1152.64	1162.14	273.593	17.915	1.64	268	4871
r211-50	-	-	-	-	-	-	-
rc201-50	1814.12	1876.06	16.153	0.28	0.015	432	463
rc202-50	1678.02	1763.48	3538.58	0.991	0.078	30361	1098
rc203-50	-	-	-	-	-	-	-
rc204-50	1406.73	1457.3	33563.8	7.791	0.281	68145	3699
rc205-50	1698.02	1780.1	4160.9	0.59	0.047	39122	1880
rc206-50	-	-	-	-	-	-	-
rc207-50	-	-	-	-	-	-	-
rc208-50	-	-	-	-	-	-	-

Table 2: Results on the Solomon's benchmark (50 customers) with t_{max} value set (75;220)

Instance	Solution		Total Time (sec)			Phase1 time (sec)			# trips		
	Dom	No Dom	Dom	No Dom	Ratio	Dom	No Dom	Ratio	Dom	No Dom	Ratio
c201-25	659.02	659.02	1.561	1.953	25.11 %	0	0	0.00 %	100	102	2.00 %
c202-25	653.37	653.37	45.819	64.656	41.11 %	0.031	0.016	-48.39 %	439	453	3.19 %
c203-25	646.4	646.4	247.189	356.14	44.08 %	0.031	0.032	3.23 %	639	648	1.41 %
c204-25	602.46	602.46	252.825	371.343	46.88 %	0.047	0.078	65.96 %	801	824	2.87 %
c205-25	636.39	636.39	38.325	128.296	234.76 %	0	0	0.00 %	170	202	18.82 %
c206-25	636.39	636.39	637.612	1431.06	124.44 %	0.015	0.016	6.67 %	233	274	17.60 %
c207-25	603.22	603.22	98.273	135.352	37.73 %	0.015	0.016	6.67 %	409	454	11.00 %
c208-25	613.2	613.2	38.154	141.363	270.51 %	0.015	0.016	6.67 %	297	352	18.52 %
r201-25	762.43	762.43	0.234	0.375	60.26 %	0	0	0.00 %	145	162	11.72 %
r202-25	645.78	645.78	0.796	0.859	7.91 %	0.031	0.016	-48.39 %	515	535	3.88 %
r203-25	621.97	621.97	2.216	3.328	50.18 %	0.078	0.047	-39.74 %	692	710	2.60 %
r204-25	579.68	579.68	5.026	5.234	4.14 %	0.062	0.062	0.00 %	811	849	4.69 %
r205-25	634.09	634.09	0.827	1.578	90.81 %	0.015	0.015	0.00 %	314	368	17.20 %
r206-25	596.74	596.74	0.686	0.843	22.89 %	0.031	0.031	0.00 %	643	687	6.84 %
r207-25	585.74	585.74	3.496	3.015	-13.76 %	0.046	0.047	2.17 %	738	763	3.39 %
r208-25	579.68	579.68	6.681	7.875	17.87 %	0.047	0.062	31.91 %	818	859	5.01 %
r209-25	602.39	602.39	1.67	2.453	46.89 %	0.016	0.046	187.50 %	520	603	15.96 %
r210-25	636.15	636.15	7.914	9.093	14.90 %	0.031	0.047	51.61 %	608	661	8.72 %
r211-25	575.91	575.91	25.805	41.343	60.21 %	0.046	0.093	102.17 %	858	1042	21.45 %
rc201-25	988.05	988.05	1.373	2.531	84.34 %	0	0	0.00 %	96	112	16.67 %
rc202-25	881.49	881.49	25.664	29.813	16.17 %	0.031	0.015	-51.61 %	318	340	6.92 %
rc203-25	749.15	749.15	64.271	75.642	17.69 %	0.031	0.031	0.00 %	520	546	5.00 %
rc204-25	-	-	-	-	-	0.046	0.047	2.17 %	640	685	7.03 %
rc205-25	840.35	840.35	3.746	6.265	67.25 %	0.015	0.015	0.00 %	284	341	20.07 %
rc206-25	761.03	761.03	35.703	81.533	128.36 %	0	0.015	-	206	287	39.32 %
rc207-25	738.87	-	71807.37	-	-	0.031	0.047	51.61 %	455	640	40.66 %
rc208-25	-	-	-	-	-	0.047	0.078	65.95 %	658	990	50.46 %
Total	-	-	1545.866*	2901.943	87.72%	0.665	0.716	7.67 %	12927	14489	12.08 %

Table 3: Impact of dominance rule with t_{max} value set (75;220)

rule. For each criterion, we present the ratio (Ratio) between the case with dominance rule and the case without dominance rule. A "−" in the table indicates that the corresponding instance could not be solved.

We can check that the solution costs are not affected by the dominance relation. The rc207-25 instance is closed with dominance rule and not without. The bottom line of this table contains the sum of computation time for both phases, computation times of Phase 1 and number of generated structures in Phase 1. Note that, due to rc207-25 not being closed without dominance rule, the sum of computation time for both phases excludes this instance. As we can see, if we do not apply the dominance rule, the number of generated trips are increased by approxmately 12%, the computation time of phase 1 is increased by 7% and the total computation time is increased by 87%.

Table 4 presents the same comparison with a larger t_{max} value set. This value is set to 100 for instances of classes R2 and RC2 and to 250 for instances of class C2. Without applying the dominance rule, the number of generated structures is increased by 37%, the Phase 1 computation time is three times longer and the total computation time is increased by 30%. A "−" in the table indicates that the corresponding instance could not be solved. We can note that the impact of dominance rule is higher for a larger t_{max}, due to the increase of the number of feasible structures.

5.5 Impact of duration limit

In this section, we evaluate the impact of duration limit. U is set to 2. In Table 5, we compare the following criteria for each instance: the solution cost, the total computation time, including both phases (Total time), the computation time of Phase 1 (Phase1 time) and the number of generated structures in Phase 1 (# trips) for two couples of t_{max} values (75;220) and (100;250).

Instance	Solution		Total Time (sec)			Phase1 time (sec)			# trips		
	Dom	No Dom	Dom	No Dom	Ratio	Dom	No Dom	Ratio	Dom	No Dom	Ratio
c201-25	540.9	540.9	0.234	0.125	-46.58 %	0	0.015	-	151	153	1.32 %
c202-25	533.43	533.43	51.548	57.125	10.82 %	0.203	0.171	-15.76 %	1503	1576	4.86 %
c203-25	532.77	532.77	352.534	462.031	31.06 %	0.577	0.593	2.77 %	3356	3529	5.15 %
c204-25	525.46	525.46	5102.61	6037.55	18.32 %	1.185	1.389	17.21 %	4551	4864	6.88 %
c205-25	529.94	529.94	1.561	1.937	24.09 %	0	0.015	-	346	473	36.71 %
c206-25	527.84	527.84	118.271	360.656	204.94 %	0.031	0.031	0.00 %	616	832	35.06 %
c207-25	525.46	525.46	28.865	150.333	420.81 %	0.235	0.234	-0.43 %	1612	2169	34.55 %
c208-25	525.46	525.46	4.745	15.876	234.58 %	0.047	0.063	34.04 %	881	1243	41.09 %
r201-25	698.18	698.18	0.749	1.937	158.61 %	0.015	0.015	0.00 %	378	457	20.90 %
r202-25	617.53	617.53	3.887	5.531	42.29 %	0.343	0.437	27.41 %	2596	2900	11.71 %
r203-25	577.74	577.74	11.068	26.172	136.47 %	0.859	1.093	27.24 %	4224	4635	9.73 %
r204-25	483.3	483.3	30.925	45.547	47.28 %	1.873	3.187	70.15 %	5721	6602	15.40 %
r205-25	559.14	559.14	3.278	4.265	30.11 %	0.094	0.156	65.96 %	1293	1724	33.33 %
r206-25	523.64	523.64	6.484	9.171	41.44 %	0.686	0.921	34.26 %	3575	4397	22.99 %
r207-25	512	512	359.762	707.674	96.71 %	1.171	1.578	34.76 %	4780	5440	13.81 %
r208-25	483.3	483.3	85.987	134.953	56.95 %	2.139	3.484	62.88 %	5841	6817	16.71 %
r209-25	517.69	517.69	12.41	15.578	25.53 %	0.39	0.593	52.05 %	2653	3511	32.34 %
r210-25	547.23	547.23	2.513	2.953	17.51 %	0.656	0.812	23.78 %	3407	4103	20.43 %
r211-25	474.49	474.49	63.506	89.5	40.93 %	2.091	6.707	220.75 %	5461	8170	49.61 %
rc201-25	849.33	849.33	2.778	6.813	145.25 %	0.016	0	-	238	302	26.89 %
rc202-25	679.86	679.86	4.277	5.219	22.02 %	0.156	0.218	39.74 %	1626	2159	32.78 %
rc203-25	593.56	593.56	11.661	24.58	110.79 %	0.453	0.734	62.03 %	2927	3917	33.82 %
rc204-25	-	-	-	-	-	1.404	3.696	163.24 %	4670	6934	48.48 %
rc205-25	702.49	702.49	2.154	4.015	86.40 %	0.125	0.171	36.80 %	1137	1865	64.03 %
rc206-25	604.12	604.12	2.232	5.109	128.90 %	0.046	0.093	102.17 %	734	1268	72.75 %
rc207-25	514.81	514.81	46.062	77.373	67.98 %	0.421	1.265	200.48 %	2385	5139	115.47 %
rc208-25	-	-	-	-	-	2.355	24.751	950.99 %	4779	12790	167.63 %
Total	-	-	6310.101	8252.023	30.77 %	17.571	52.422	198.34 %	71441	97969	37.13 %

Table 4: Impact of dominance rule with t_{max} value set (100;250)

We give the ratio between each criterion for t_{max} (75;220) and same criterion for t_{max} (100;250). A ”−” in the table indicates that the corresponding instance could not be solved. Note that we have chosen these values for the purpose of comparing our results with the results presented in Azi (2010).

As expected, the solution cost decreases when duration limit increases. It can be observed that increasing the duration limit does not necessarily increase the total computation time. Indeed, the main time comsumption is generated by the second phase and the difficulty of the covering problem is not necessarily correlated with the trip length.

We note an important decrease of the total computation time for the instance rc207-25 and an important increase for the instance r207-25. Because of these pecularіaties, we distinguish two total sums, one that includes these special instances denoted Total and an other without them denoted Total*. We have an increase of 300 % of the total computation time for Total* and a decrease of 90 % for Total.

Also, for both Total and Total*, we can see an increase of 2200 % in computation time of Phase 1 and an increase of 450 % of Phase 1 generated trips. Indeed the maximal number of visited customers per feasible trip has an important impact on the dynamic programming and the number of generated labels. However, this does not affect much the total computation time, as the Phase 1 computation time is rather small.

5.6 Impact of time windows

In this section, we evaluate the impact of the length of time windows. U is set to 2. For this purpose, we have reduced the length of customer time windows by half. Four means of reducing this length have been used. In the first case (denoted Type 1), we compute the center c_i of each customer time window as follows $c_i = \frac{a_i+b_i}{2}$ and we set its new start a'_i (resp. new end b'_i) to

Instance	Solution			Total Time			Phase1 time			# trips		
R2/RC2	75	100	Ratio	75	100	Ratio	75	100	Ratio	75	100	Ratio
C2	220	250		220	250		220	250		220	250	
c201-25	659.02	540.9	-17.92 %	1.561	0.234	-85.01 %	0	0	0.00 %	100	151	51.00 %
c202-25	653.37	533.43	-18.36 %	45.819	51.548	12.50 %	0.031	0.203	554.84 %	439	1503	242.37 %
c203-25	646.4	532.77	-17.58 %	247.189	352.534	42.62 %	0.031	0.577	1761.29 %	639	3356	425.20 %
c204-25	602.46	525.46	-12.78 %	252.825	5102.61	1918.24 %	0.047	1.185	2421.28 %	801	4551	468.16 %
c205-25	636.39	529.94	-16.73 %	38.325	1.561	-95.93 %	0	0	0.00 %	170	346	103.53 %
c206-25	636.39	527.84	-17.06 %	637.612	118.271	-81.45 %	0.015	0.031	106.67 %	233	616	164.38 %
c207-25	603.22	525.46	-12.89 %	98.273	28.865	-70.63 %	0.015	0.235	1466.67 %	409	1612	294.13 %
c208-25	613.2	525.46	-14.31 %	38.154	4.745	-87.56 %	0.015	0.047	213.33 %	297	881	196.63 %
r201-25	762.43	698.18	-8.43 %	0.234	0.749	220.09 %	0	0.015	0.00 %	145	378	160.69 %
r202-25	645.78	617.53	-4.37 %	0.796	3.887	388.32 %	0.031	0.343	1006.45 %	515	2596	404.08 %
r203-25	621.97	577.74	-7.11 %	2.216	11.068	399.46 %	0.078	0.859	1001.28 %	692	4224	510.40 %
r204-25	579.68	483.3	-16.63 %	5.026	30.925	515.30 %	0.062	1.873	2920.97 %	811	5721	605.43 %
r205-25	634.09	559.14	-11.82 %	0.827	3.278	296.37 %	0.015	0.094	526.67 %	314	1293	311.78 %
r206-25	596.74	523.64	-12.25 %	0.686	5.104	644.02 %	0.031	0.686	2112.90 %	643	3575	455.99 %
r207-25	585.74	512	-12.59 %	3.496	359.762	10190.68 %	0.046	1.171	2445.65 %	738	4780	547.70 %
r208-25	579.68	483.3	-16.63 %	6.681	85.987	1187.04 %	0.047	2.139	4451.06 %	818	5841	614.06 %
r209-25	602.39	517.69	-14.06 %	1.67	12.41	643.11 %	0.016	0.39	2337.50 %	520	2653	410.19 %
r210-25	636.15	547.23	-13.98 %	7.914	2.513	-68.25 %	0.031	0.656	2016.13 %	608	3407	460.36 %
r211-25	575.91	474.49	-17.61 %	25.805	63.506	146.10 %	0.046	2.091	4445.65 %	858	5461	536.48 %
rc201-25	988.05	849.33	-14.04 %	1.373	2.778	102.33 %	0	0.016	0.00 %	96	238	147.92 %
rc202-25	881.49	679.86	-22.87 %	25.664	4.277	-83.33 %	0.031	0.156	403.23 %	318	1626	411.32 %
rc203-25	749.15	593.56	-20.77 %	64.271	11.661	-81.86 %	0.031	0.453	1361.29 %	520	2927	462.88 %
rc204-25	-	-	-	-	-	-	0.046	1.404	2952.17 %	640	4670	629.69 %
rc205-25	840.35	702.49	-16.41 %	3.746	2.154	-42.50 %	0.015	0.125	733.33 %	284	1137	300.35 %
rc206-25	761.03	604.12	-20.62 %	35.703	2.232	-93.75 %	0	0.046	-	206	734	256.31 %
rc207-25	738.87	514.81	-30.32 %	71807.37	46.062	-99.94 %	0.031	0.421	1258.06 %	455	2385	424.18 %
rc208-25	-	-	-	-	-	-	0.047	2.355	4910.64 %	658	4779	626.29 %
Total				73353.236	6308.721	-91.40 %	0.758	17.571	2218.07 %	12927	71441	452.65 %
Total*				1545.866	6262.659	305.12 %	0.727	17.15	2259.01 %	12472	69056	453.69 %

Table 5: Impact of t_{max} value set on the Solomon's benchmark (25 customers)

the value $\frac{a_i+c_i}{2}$ (resp. $\frac{b_i+c_i}{2}$). For the second case (denoted Type 2), we compute $c_i = \left\lfloor \frac{a_i+b_i}{2} \right\rfloor$ and we set a'_i (resp. new end b'_i) to the value $\frac{a_i+c_i}{2}$ (resp. $\frac{b_i+c_i}{2}$). For the third case (denoted Type 3), we compute $c_i = \frac{a_i+b_i}{2}$ and we set a'_i (resp. new end b'_i) to the value $\lfloor \frac{a_i+c_i}{2} \rfloor$ (resp. $\left\lfloor \frac{b_i+c_i}{2} \right\rfloor + 1$). For the last case (denoted Type 4), we compute $c_i = \left\lfloor \frac{a_i+b_i}{2} \right\rfloor$ and we set a'_i (resp. new end b'_i) to the value $\lfloor \frac{a_i+c_i}{2} \rfloor$ (resp. $\left\lfloor \frac{b_i+c_i}{2} \right\rfloor + 1$). As a last experiment (denoted Without), we have relaxed all customer time windows, that is, we have assigned the time window of the depot to each customer. In this case, only three instances were considered, one for each class because there is only one spatial layout for each class.

In Table 6, for each instance, we compare the solution cost, the total computation time of the two phases (Total time) and the computation time of Phase 1 (Phase1 time), for the basic case and the reduced time windows of Type 1. A "no sol" in the table indicates that the corresponding instance does not have a solution for this setting.

All instances with 25 customers are closed. As expected, the solution cost increases with reduced time windows, and 8 instances have no solution in this setting. The solution time of these instances are very small thanks to the application of the Lemma 4.2. We can also see, unlike the effect observed in Azi (2010), that this reduction does not necessarily decreases the total computation time.

In Table 7, for each instance, we present the solution cost and the total computation time for the different reduction types. As a first finding, c201-25 gets a solution for reduction Types 3 and 4 and not for Types 1 and 2. We can also note for types 3 and 4 that the solution cost can decrease with these little modifications of time windows. By using two different types of time windows reduction, a maximum variation of 2 time units is obtained for a given client. This

Instance	Solution		Total time			Phase 1 time		
	Type 1	Base	Type 1	Base	Ratio	Type 1	Base	Ratio
c201-25	nosol	659.02	0.062	1.561	25.18	0	0	1.00
c202-25	672.71	653.37	7.484	45.819	6.12	0.015	0.031	2.07
c203-25	659.96	646.4	36.314	247.189	6.81	0.016	0.031	1.94
c204-25	629.15	602.46	267.59	252.825	0.94	0.031	0.047	1.52
c205-25	702.34	636.39	12.781	38.325	3.00	0	0	1.00
c206-25	656.53	636.39	12.985	637.612	49.10	0	0.015	-
c207-25	636.39	603.22	762.351	98.273	0.13	0	0.015	-
c208-25	636.39	613.2	70.784	38.154	0.54	0	0.015	-
r201-25	nosol	762.43	0.063	0.234	3.71	0	0	1.00
r202-25	nosol	645.78	0.312	0.796	2.55	0.015	0.031	2.07
r203-25	nosol	621.97	0.578	2.216	3.83	0.016	0.078	4.88
r204-25	655.35	579.68	**76.344**	**5.026**	**0.07**	0.031	0.062	2.00
r205-25	762.43	634.09	1.218	0.827	0.68	0.015	0.015	1.00
r206-25	684.89	596.74	4.093	0.686	0.17	0.016	0.031	1.94
r207-25	654.52	585.74	4.484	3.496	0.78	0.031	0.046	1.48
r208-25	609.81	579.68	191.706	6.681	0.03	0.047	0.047	1.00
r209-25	688.99	602.39	3.672	1.67	0.45	0	0.016	-
r210-25	703.62	636.15	1.437	7.914	5.51	0.015	0.031	2.07
r211-25	622.69	575.91	0.609	25.805	42.37	0.015	0.046	3.07
rc201-25	nosol	988.05	0.031	1.373	44.29	0	0	1.00
rc202-25	nosol	881.49	0.296	25.664	86.70	0	0.031	-
rc203-25	nosol	749.15	0.421	64.271	152.66	0.015	0.031	2.07
rc204-25	753.38	-	45.173	-	-	0.031	0.046	1.48
rc205-25	nosol	840.35	0.109	3.746	34.37	0.015	0.015	1.00
rc206-25	978.13	761.03	1.203	35.703	29.68	0	0	-
rc207-25	949.19	738.87	**231.987**	**71807.37**	**309.53**	0	0.031	-
rc208-25	688.166	-	3852.62	-	-	0.062	0.047	0.76

Table 6: Comparison of results obtained with reduction of type 1 and no reduction on the Solomon's benchmark (25 customers) with t_{max} value set (75;220)

Instance	Solution				Total Time			
Type	type 1	type 2	type 3	type 4	type 1	type 2	type 3	type 4
c201-25	no sol	no sol	**705.03**	**705.03**	0.062	0.078	0.312	0.328
c202-25	672.71	672.71	672.71	672.71	7.484	7.482	11.594	13.281
c203-25	659.96	659.96	659.96	659.96	36.314	24.383	25.126	29.749
c204-25	629.15	629.15	629.15	629.15	267.59	297.046	313.924	358.777
c205-25	702.34	702.34	**659.02**	**659.02**	12.781	12.359	1.484	1.453
c206-25	656.53	656.53	656.53	656.53	12.985	12.092	12.906	15.453
c207-25	636.39	636.39	636.39	636.39	762.351	738.897	778.247	784.057
c208-25	636.39	636.39	636.39	636.39	70.784	71.049	68.691	68.675
r201-25	no sol	no sol	no sol	no sol	0.063	0.047	0.094	0.141
r202-25	no sol	no sol	no sol	no sol	0.312	0.328	0.328	0.312
r203-25	no sol	no sol	no sol	no sol	0.578	0.86	0.703	0.734
r204-25	655.35	655.35	**654.65**	**654.65**	76.344	75.501	75.016	70.968
r205-25	762.43	762.43	**757.2**	**757.2**	1.218	1.187	0.203	0.281
r206-25	684.89	684.89	684.89	684.89	4.093	3.562	3.312	3.624
r207-25	654.52	654.52	**646**	**646**	4.484	4.156	8.234	8.281
r208-25	609.81	609.81	609.81	609.81	191.706	194.58	190.08	189.26
r209-25	688.99	688.99	688.99	688.99	3.672	3.968	3.812	4.656
r210-25	703.62	703.62	703.62	703.62	1.437	0.859	1.484	1.406
r211-25	622.69	622.69	622.69	622.69	0.609	0.578	0.562	0.671
rc201-25	no sol	no sol	no sol	no sol	0.031	0.031	0.031	0.046
rc202-25	no sol	no sol	no sol	no sol	0.296	0.265	0.312	0.453
rc203-25	no sol	no sol	no sol	no sol	0.421	0.421	0.562	0.687
rc204-25	753.38	753.38	753.38	753.38	45.173	44.171	52.268	61.78
rc205-25	no sol	no sol	no sol	no sol	0.109	0.109	0.078	0.249
rc206-25	978.13	978.13	978.13	978.13	1.203	1.187	2.89	3.375
rc207-25	949.19	949.19	949.19	949.19	231.987	233.909	196.956	196.83
Total					1734.087	1729.105	1749.209	1815.527

Table 7: Comparision results on the Solomon's benchmark (25 customers) with different types of reduced time

corresponds to about 1% of time horizon for instances of classes and r2 rc2 and about 0.3% for instances of the class c2. Thus, we can note that this problem is indeed very sensitive to time windows.

Finally, in Table 8, we present the results obtained when we relax the time windows of customers. We note that only three instances are considered, one for each class. For each class, as for the case without time windows (Without), we present the solution cost, the total resolution time and Phase 1 resolution time. As for the basic case, we give, for each class, the cost for the instance with cheapest solution, the total resolution time for the instance with highest total resolution time, and the Phase 1 resolution time for the instance with highest Phase 1 resolution time. We also give, for each class, and for total resolution time and Phase 1 resolution time, the ratio of the value obtained in the basic case on the value obtained in the no time window case.

Firstly for Table 8, the instance rc2XX-25 is not solved. Secondly, the solution cost of the no time window case is a lower bound of the solution cost of all this class' instances. The total resolution time is lower than the highest resolution time of each class unlike the Phase 1 resolution time, which is always higher.

Instance	Solution		Temps total			Temps phase 1		
	Without	Basic	Without	Basic	Ratio	Without	Basic	Ratio
c2XX-25	584.44	602.46	188.953	637.612	3.37	0.156	0.047	0.30
r2XX-25	574.23	575.91	25.031	25.805	1.03	0.141	0.078	0.55
rc2XX-25	-	-	-	-	-	-	-	-

Table 8: Comparison of results obtained with no window of time and without reduction on the Solomon's benchmark with t_{max} value set (75;220) and no time windows

The above results show that the time windows, their length and their time positions, have much impact on the solution existence and the solution cost. We can also note that, still, the resolution times of Phase 1 increase when the size of time windows increases. This is due to the dynamic programming used during this phase and the fact that the structure search space is larger.

6 Discussion

6.1 Comparison to previous results on this problem

Table 9 gives, for each instance, the cost of the optimal solution found, the total solving time with t_{max} set to $(75; 220)$. The comparison between our results and those obtained in Azi (2010) is given for all these data. Recall that results in Azi et al. (2010) were corrected in Azi (2010), thus explaining that we only compare with the latter. For the total solving time, the following ratio is also provided: processing time in Azi (2010) divided by processing time of our method.

We can distinguish two categories among instances: (i) those for which we find the same optimal cost and (ii) the ones which we close and which were not closed in Azi (2010).

17 of the 27 Solomon instances with 25 clients and a large time horizon fall in the first category. In these cases, we get significantly faster results, up to 25000 times faster, and 1000 times faster in average.

As for the second category, and with an available fleet size of 2, we provide an optimal solution for the instances c202-25, c203-25, c204-25, c206-25, c207-25 and c208-25 with $t_{max} = 220$ and for the instances rc202-25, rc203-25 and rc207-25 with $t_{max} = 75$.

Table 10 gives, for each instance, the cost of the optimal solution and the total solving time for $t_{max} = (100; 250)$. Like in Table 9, the comparison between our results and those obtained in Azi (2010) is given. The numerator of the ratio is the solving time in Azi (2010) and its denominator is the solving time provided by our method.

We can use the same 2 categories as above for the shorter t_{max}.

In the first category, we find 17 of the 27 Solomon instances with 25 clients and a large time horizon. Like for the smaller t_{max} case, our method is 1000 times faster on average for these instances than the method in Azi (2010), and the ratio varies between 5 times faster to 10000 times faster.

As for the second category, and with an available fleet size of 2, we provide an optimal solution for the instances c202-25, c203-25, c207-25 and c208-25 with $t_{max} = 250$ et and for the instances r204-25, r207-25, r208-25, r211-25, rc203-25 et rc207-25 with $t_{max} = 100$.

Instance	Solution		Total Time		
	Azi (2010)		Azi (2010)		Ratio
c201-25	659.02	659.02	40361.2	1.561	**25855.99**
r201-25	762.43	762.43	68.3	0.234	291.88
r202-25	645.78	645.78	205.2	0.796	257.79
r203-25	621.97	621.97	1333.2	2.216	601.62
r204-25	579.68	579.68	30983.3	5.026	6164.60
r205-25	634.09	634.09	354.1	0.827	428.17
r206-25	596.74	596.74	318.4	0.686	464.14
r207-25	585.74	585.74	2853.5	3.496	816.22
r208-25	579.68	579.68	9270.3	6.681	1387.56
r209-25	602.39	602.39	262.6	1.67	157.25
r210-25	636.15	636.15	5094.1	7.914	643.68
r211-25	575.91	575.91	5648.6	25.805	218.90
rc201-25	988.05	988.05	3.1	1.373	**2.26**
rc204-25	-	-	-	-	-
rc205-25	840.35	840.35	28.8	3.746	7.69
rc206-25	761.03	761.03	7156.8	35.703	200.45
rc208-25	-	-	-	-	-
Total			103941.5	97.734	1063.51
Newly closed instances					
c202-25	-	653.37	-	45.819	-
c203-25	-	646.4	-	247.189	-
c204-25	-	602.46	-	252.825	-
c205-25	-	636.39	-	38.325	-
c206-25	-	636.39	-	637.612	-
c207-25	-	603.22	-	98.273	-
c208-25	-	613.2	-	38.154	-
rc202-25	-	881.49	-	25.664	-
rc203-25	-	749.15	-	64.271	-
rc207-25	-	738.87	-	71807.37	-

Table 9: Comparision results with *Azi (2010) on the Solomon's benchmark (25 customers) with t_{max} value set (75;220)

Instance	Solution		Total Time		
	Azi (2010)		Azi (2010)		Ratio
c201-25	540.9	540.9	1.3	0.234	**5.56**
c204-25	-	-	-	-	-
c205-25	529.94	529.94	116.6	1.561	74.70
c206-25	527.84	527.84	1987.2	118.271	16.80
r201-25	698.18	698.18	43.6	0.749	58.21
r202-25	617.53	617.53	25249.9	3.887	6495.99
r203-25	577.74	577.74	75729.3	11.068	6842.18
r205-25	559.14	559.14	1202.3	3.278	366.78
r206-25	523.64	523.64	28498.1	5.104	5583.48
r209-25	517.69	517.69	11173.9	12.41	900.39
r210-25	547.23	547.23	26690	2.513	**10620.77**
rc201-25	849.33	849.33	16.06	2.778	5.78
rc202-25	679.86	679.86	1096.3	4.277	256.32
rc204-25	-	-	-	-	-
rc205-25	702.49	702.49	262.8	2.154	122.01
rc206-25	604.12	604.12	222.7	2.232	99.78
rc208-25	-	-	-	-	-
Total			172290.06	170.516	1010.40
Newly closed instances					
c202-25	-	533.43	-	51.548	-
c203-25	-	532.77	-	352.534	-
c207-25	-	525.46	-	28.865	-
c208-25	-	525.46	-	4.745	-
r204-25	-	483.3	-	30.925	-
r207-25	-	512	-	359.762	-
r208-25	-	483.3	-	85.987	-
r211-25	-	474.49	-	63.506	-
rc203-25	-	593.56	-	11.661	-
rc207-25	-	514.81	-	46.062	-

Table 10: Comparision results with Azi (2010) on the Solomon's benchmark (25 customers) with t_{max} value set (100;250)

6.2 Analysis of performance in regards to master problem formulation

We obtain a clear enhancement in solving time for the instances solved. To analyse this ameliorated performance, we should compare the formulation of the master problem in the column generation used in Azi et al. (2010) and Azi (2010) compared to ours. In these works, the variables of the master problem are workdays which correspond to a succession of trips. The computation of a workday involves to select a number of trip timed structures and to associate to each trip a temporal position, as well as a successor and a predecessor. The variables of our master problem are trips, with time position but no need to associate successor and predecessor. The size of the variables' set of our master problem is then much smaller than in Azi et al. (2010) and Azi (2010). Our subproblem needs then to search a smaller space to find a variable with negative reduced cost.

Furthermore, finding a workday with negative reduced cost implies to solve an ESPPRC as the subproblem. With our own variable definition, we can avoid the ordering part of this problem, and only need to find the best temporal position for each timed structure. Thus, we could provide a pseudo-polynomial complexity to our subproblem. The task of ordering trips is handled by the constraint 28 in the master problem.

The combination of these two differences, smaller variables' set and subproblem with reduced complexity, explains the clear reduction observed in the solving times.

7 Conclusion and future research

In this paper, we addressed the exact solution of the MTVRPTW-LD, previously introduced and investigated in Azi et al. (2010). We proposed an efficient branch and price scheme, achieving a fast improvement compared to Azi (2010) with regard to computing times. A main advantage of our approach lies in the efficiency of the Column Generation subproblem, solved with a fast pseudo-polynomial algorithm. The future research is to extend the previous scheme to the general problem without limited duration.

Acknowledgment

We acknowledge Cemagref and Région Languedoc-Roussillon in France for providing the basic funding of this research, as well as contributions from LIRMM, PNRB MOBIPE project supported by french ANR, and A2PV SyDeReT project. We thank Pr J.C. König for finding the counter exemple given in section 4.4.

References

Alonso, F., M. J. Alvarez, J. E. Beasley. 2008. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society* **59**(7) 963–976.

Azi, N. 2010. Méthodes exactes et heuristiques pour le problème de tournées avec fenêtres de temps et réutilisation de véhicules. Ph.D. thesis, Université de Montréal.

Azi, N., M. Gendreau, J. Y. Potvin. 2007. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research* **178**(3) 755–766.

Azi, N, M Gendreau, J. Y. Potvin. 2010. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research* **202**(3) 756–763.

Battarra, M., M. Monaci, D. Vigo. 2009. An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Computers & Operations Research* **36**(11) 3041–3050.

Blazewicz, J., K. Ecker, , E. Pesch, G. Schmidt, J. Węglarz. 2007. *Handbook on Scheduling*. Springer.

Brandão, J. C. S., A. Mercer. 1998. The multi-trip vehicle routing problem. *Journal of the Operational Research Society* **49**(8) 799–805.

Bratley, P., M. Florian, P. Robillard. 1971. Scheduling with earliest start and due date constraints. *Naval Research Logistics Quaterly* **18**(4) 511–517.

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

Desrochers, M., J. Desrosiers, M. Solomon. 1992. A new optimization algorithm for the vehicle-routing problem with time windows. *Operations Research* **40**(2) 342–354.

Dror, M. 1994. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research* **42**(5) 277–278.

Feillet, D. 2007. Solution of vehicle routing problem with branch and price. Thesis for accreditation to supervise research (hdr), Université d'Avignon et des Pays de Vaucluse.

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

Fleischmann, B. 1990. The vehicle routing problem with multiple use of vehicles. Working paper, Fachbereich Wirtshaftswissenschaften, Universität Hamburg.

Graham, R.L., E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan. 1979. Optimization and approximation in deterministic sequencing and scheduling theory: a survey. *Annals of Discrete Mathematics* **5** 287–326.

Hernandez, F. 2010. Méthodes de résolution exactes pour le problème de routage de véhicules avec fenêtres de temps et routes multiples. Ph.D. thesis, Université Montpellier II.

Hernandez, F., D. Feillet, R. Giroudeau, O. Naud. 2009. Multi-trip vehicle routing problem with time windows for agricultural tasks. Odysseus, Fourth International Workshop on Freight Transportation and Logistics, Izmir, Turkey.

Lenstra, J.K., A.H.G. Rinnooy Kan, P. Brucker. 1977. Complexity of scheduling machine problems. *Annales of Discrete Mathematics* **1** 343–362.

Olivera, A., O. Viera. 2007. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research* **34**(1) 28–47.

Petch, R. J., S. Salhi. 2003. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics* **133**(1-3) 69–92.

Salhi, S., R. J. Petch. 2007. A ga based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modelling and Algorithms* **6**(4) 591–613.

Sen, A., K. Bülbül. 2008. A survey on multi trip vehicle routing problem.

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

Taillard, E. D., G. Laporte, M. Gendreau. 1996. Vehicle routing with multiple use of vehicles. *Journal of the Operational Research Society* **47**(8) 1065–1070.