GENERALIZED DISAGGREGATION ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND MULTIPLE ROUTES

Rita Macedo[1], Saïd Hanafi[1], François Clautiaux[2], Cláudio Alves[3] and J. M. Valério de Carvalho[3]

[1] *LAMIH-SIADE, UMR 8530, Université de Valenciennes et du Hainaut-Cambrésis Le Mont Houy, 59313 Valenciennes Cedex 9, France*

[2] *Université des Sciences et Technologies de Lille, LIFL UMR CNRS 8022, INRIA Bâtiment INRIA Parc de la Haute Borne, 59655 Villeneuve dAscq, France*

[3] *Centro de Investigação Algoritmi da Universidade do Minho, Escola de Engenharia, Universidade do Minho 4710-057 Braga, Portugal*

Keywords: Integer programming, Combinatorial optimization, Vehicle Routing, Network flow model.

Abstract: In this paper, we address the VRP with multiple routes and time windows. For this variant of the VRP, there is a time interval within which every customer must be visited. In addition, every vehicle is allowed to perform more than one route within the same planning period. Here, we propose a general disaggregation algorithm that improves the exact approach described in (Macedo et al., 2011). We describe a novel rounding rule and a new node disaggregation scheme based on different discretization units. We describe a new integer programming model for the problem, which can be used, with a slight modification, to exactly assess whether a given solution is feasible or not. Finally, we report some computational experiments performed on a set of instances from the literature.

1 INTRODUCTION

The vehicle routing problem (VRP) is a combinatorial optimization problem that was first addressed in (Dantzig and Ramser, 1959), as a generalization of the traveling salesman problem. It generally consists of scheduling vehicles to visit and deliver goods to a set of customers. Its application area is very wide, and there are many variants of this problem in the literature. General surveys on the VRP and its variants are provided in (Toth and Vigo, 2002; Cordeau et al., ; Laporte, 2009). According to (Baldacci et al., 2010), the most effective exact approaches currently available for the classical version of the VRP are the ones proposed in (Lysgaard et al., 2004; Fukasawa et al., 2006; Baldacci et al., 2008).

We address the VRP with multiple routes and time windows (MVRPTW). For this variant of the VRP, there is a time interval within which every customer must be visited. In addition, every vehicle is allowed to perform more than one route within the same planning period. There are two different exact approaches for this variant described in the literature. In (Azi et al., 2010), the authors propose a branch-and-price algorithm for this problem and in (Macedo et al., 2011), a network flow model embedded in an iterative algorithm is described.

In this paper, we improve the algorithm proposed in (Macedo et al., 2011). We consider a different rounding rule and a different disaggregation of nodes. We also generalize the previous approach and consider different discretization units. Finally, we describe a new integer programming model for this problem, which can be used, with a slight modification, to exactly assess whether a given solution is feasible or not.

1.1 Problem Definition

The problem can be defined in a complete graph $G = (V,A)$, being V its set of nodes and $A = \{(i, j) : i, j \in V\}$ its set of arcs. We consider that there is an homogenous fleet of k capacitated vehicles that begin and end their routes in a single depot o. Each vehicle $i \in K = \{1, \ldots, k\}$ has a capacity of Q units and is allowed to perform more than one single route dur-

Macedo R., Hanafi S., Clautiaux F., Alves C. and M. Valério de Carvalho J..
GENERALIZED DISAGGREGATION ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND MULTIPLE ROUTES.
DOI: 10.5220/0003755003050312
In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems* (ICORES-2012), pages 305-312
ISBN: 978-989-8425-97-3

ing the planning period. This means that a vehicle can perform a route, reload at the depot and perform another route, until the end of the planning period, which has a duration equal to W. The sequence of routes and waiting times allocated to one vehicle is defined as its workday. The duration of one workday is, therefore, also equal to W, and the interval $[0,W]$ can be seen as the depot's opening time. This means that vehicles can start loading from instant 0 of their workday, and that they can not arrive at the depot after instant W. Furthermore, the maximum duration between the beginning of a route and the moment that the vehicle starts to service the last customer of that route can not be greater than t_{max}. This variant specially applies to situations where vehicles perform short trips and are, therefore, available to perform more than one route per planning period. This is, for example, the case of the delivery or collection of perishable goods, which can not be subjected to long trips.

For every customer $i \in N = \{1,\dots,n\}$, we define q_i, g_i, s_i and $[a_i,b_i]$ as its demand, revenue, service time and time window, respectively. For every arc $(i,j) \in A$, with $i,j \in V = N \cup \{o\}$, there is an associated distance d_{ij} and a traveling time t_{ij}. We define $N_r \subseteq N$ as the subset of customers visited in route $r \in R$, R being the set of all feasible routes. Route r is defined by a sequence of all customers in N_r. It is considered to be feasible if $\sum_{i\in N_r} q_i \leq Q$ and if all customers are visited within their time windows. For every route $r \in R$, there is a setup time defined by $\beta \sum_{i\in N_r} s_i$, with $\beta \in \mathbb{R}^+$. It represents the amount of time to be spent by the vehicle, at the depot, before it begins the route. It is not mandatory to visit all customers, but the number of visited customers must be maximized.

1.2 Integer Programming Models

In what concerns integer programming models for the MVPTW, we briefly describe three models from the literature, and propose a new one.

1.2.1 A Compact Model

In (Azi et al., 2010), the authors describe a compact model for the MRPTW. Let x_{ij}^r and y_i^r be the binary variables that determine if arc (i,j) and customer i belong to route $r \in R$, respectively. Variable t_i^r represents the instant at which customer i begins to be serviced, in route r. The beginning and ending instants of r are represented by t_0^r and $t_o'^r$, respectively. The constant α is a parameter that must be large enough in order to ensure that the number of visited customers is maximized. Let M be a sufficiently large number. Finally, binary variable z_{rs} determine whether routes r and s are consecutively performed by a same vehicle $(r < s)$.

$$\min \sum_{r\in R}\sum_{(i,j)\in A} d_{ij}x_{ij}^r - \alpha \sum_{r\in R}\sum_{i\in N} g_i y_i^r \tag{1}$$

$$\text{s.t.} \sum_{j\in V} x_{ij}^r = y_i^r, \quad \forall i \in N, \quad \forall r \in R, \tag{2}$$

$$\sum_{r\in R} y_i^r \leq 1, \quad \forall i \in N, \tag{3}$$

$$\sum_{i\in V} x_{ih}^r - \sum_{j\in V} x_{hj}^r = 0, \quad \forall h \in N, \forall r \in R, \tag{4}$$

$$\sum_{i\in N} x_{oi}^r = 1, \quad \forall r \in R, \tag{5}$$

$$\sum_{i\in N} x_{io}^r = 1, \quad \forall r \in R, \tag{6}$$

$$\sum_{i\in N} q_i y_i^r \leq Q, \quad \forall r \in R, \tag{7}$$

$$t_i^r + s_i + t_{ij} - M(1 - x_{ij}^r) \leq t_j^r, \quad \forall (i,j) \in A, \quad \forall r \in R, \tag{8}$$

$$a_i y_i^r \leq t_i^r \leq b_i y_i^r, \quad \forall i \in N, \quad \forall r \in R, \tag{9}$$

$$t_o^r \geq \beta \sum_{i\in N} s_i y_i^r, \quad \forall r \in R, \tag{10}$$

$$t_i^r \leq t_o^r + t_{max}, \quad \forall i \in N, \quad \forall r \in R, \tag{11}$$

$$t_o^s + M(1 - z_{rs}) \geq t_o'^r + \beta \sum_{i\in N} s_i y_i^s, \quad \forall r,s \in R, r < s, \tag{12}$$

$$\sum_{r\in R}\sum_{s\in R|r<s} z_{rs} \geq |R| - k, \tag{13}$$

$$x_{ij}^r \in \{0,1\}, \quad \forall (i,j) \in A, \quad \forall r \in R, \tag{14}$$

$$y_i^r \in \{0,1\}, \quad \forall i \in N, \quad \forall r \in R, \tag{15}$$

$$z_{rs} \in \{0,1\}, \quad \forall r,s \in R, r < s, \tag{16}$$

$$t_i^r \geq 0, \quad \forall i \in N, r \in R, \tag{17}$$

$$t_o^r, t_o'^r \geq 0, \quad \forall r \in R. \tag{18}$$

The objective function (1) translates the objective of maximizing the customers' revenues, whilst minimizing the traveled distances. Customer i is visited at most once (3) by one vehicle (2). Constraints (4) are flow conservation constraints. Every route r begins and ends at the depot (5)-(6) and the sum of the demands of all the customers in N_r can not exceed the vehicle's capacity Q (7). All time constraints, including traveling times (8), time windows (9), setup times (10), (12) and maximum route durations (11), must be respected. Finally, constraint (13) ensures that the number of workdays does not exceed the number of available vehicles k.

1.2.2 A New MIP Model

We define a new MIP model for this problem. The set of all feasible routes R is assumed to be known. The duration of route r is represented by σ_r. Let μ_r^k, $\forall k \in K, \forall r \in R$, be the binary variable that defines whether route r is performed by vehicle k or not, and t_r be the beginning instant of route r. Variable $\varepsilon_{rr'}$ defines if route r' is performed after route r. Let M be a sufficiently large number. For each feasible route $r \in R$, there is a time interval $T_r^{beg} = [T_r^-, T_r^+]$ that represents the instants from which route r can begin, in order to be feasible and to have the minimum possible duration (see section 2.2). The model states as follows.

$$\min \sum_{r\in R}\sum_{k\in K}\left(d_r - \alpha \sum_{i\in N_r} g_i\right)\mu_r^k \qquad (19)$$

$$\text{s.t.} \sum_{k=1}^{m} \mu_r^k \leq 1, \forall r \in R \qquad (20)$$

$$t_r + \sigma_r \leq t_{r'} + M(2-\mu_{r'}^k - \mu_r^k) + M(1-\varepsilon_{rr'}), \quad \forall r, r' \in R, \qquad (21)$$

$$t_{r'} + \sigma_{r'} \leq t_r + M(2-\mu_{r'}^k - \mu_r^k) + M(1-\varepsilon_{r'r}), \quad \forall r, r' \in R, \qquad (22)$$

$$\varepsilon_{rr'} + \varepsilon_{r'r} = 1, \quad \forall r, r' \in R, \qquad (23)$$

$$\mu_r^k \in \{0,1\}, \quad \forall r \in R, \forall k \in K, \qquad (24)$$

$$\varepsilon_{rr'} \in \{0,1\}, \quad \forall r, r' \in R, \qquad (25)$$

$$t_r \in [T_r^-, T_r^+]. \qquad (26)$$

The objective function (19) is equivalent to the one defined in (1)-(18). Constraints (20) ensure that any route is allocated to at most one vehicle. In order to have a feasible solution, for every route r performed by one vehicle, the interval $]t_r, t_r + \sigma_r[$ must be disjoint of any other interval $]t_{r'}, t_{r'} + \sigma_{r'}[$ related to any other route r' performed by that same vehicle. Hence, constraints (21)-(22) ensure that $]t_r, t_r + \sigma_r[\cap]t_{r'}, t_{r'} + \sigma_{r'}[= \emptyset$.

1.2.3 A Column Generation Model

In (Azi et al., 2010), the authors propose a branch-and-price algorithm to solve exactly the MVRPTW. Each column of the master problem represents one possible workday. The set of all feasible workdays is represented by Ω. For every workday $w \in \Omega$, d_w and g_w represent its cost and revenue, respectively. Binary variable $x_w, \forall w \in \Omega$ define whether or not workday w belongs to the solution and constant a_{iw} is equal to 1 if customer i is visited in workday w. The master problem states as follows.

$$\min \sum_{w\in\Omega} (d_w - \alpha g_w) x_w \qquad (27)$$

$$\text{s.t.} \sum_{w\in\Omega} a_{iw} x_w \leq 1, \quad i \in N, \qquad (28)$$

$$\sum_{w\in\Omega} x_w \leq k, \qquad (29)$$

$$x_w \in \{0,1\}, \quad w \in \Omega. \qquad (30)$$

The pricing subproblems are elementary shortest path problems with resource constraints, which can be defined in a graph whose nodes correspond to feasible routes and whose arcs define the sequencing of those routes. They use the algorithm described in (Feillet et al., 2004) to solve it.

1.2.4 Network Flow Model

In (Macedo et al., 2011), the authors propose a pseudo-polynomial network flow model for the MVRPTW, which can be described as follows. Let U be the unit discretization of the continuous interval $[0,W]$ (see section 2.2). We define a directed graph $\Pi = (\Delta, \Psi)$, with $\Delta = \{0, U, 2U, \ldots, W\}$ as its set of nodes and $\Psi = \{(u,v)^r : 0 \leq u < v \leq W, u \in T_r^{beg} \cap \Delta, r \in R\} \bigcup \{(u,v)^0 : u$ and v are consecutive nodes of $\Delta\} \bigcup \{(W,0)\}$ as its set of arcs. Each path of this graph corresponds to a workday to be schedule to one of the available vehicles. Each node of this graph corresponds to one time instant of the planning period. In the flow model, variables λ_{uv}^r represent the flow that goes through arc $(u,v)^r \in \Psi$, and d_r stands for the the total cost of route $r \in R$. The total flow going through $\prod$ equals z and represents the exact number of assigned vehicles, or equivalently, the number of defined workdays in the solution. The model states as follows.

$$\min \sum_{(u,v)^r\in\Psi}\left(d_r - \alpha \sum_{i\in N_r} g_i\right)\lambda_{uv}^r \qquad (31)$$

$$\text{s.t.} \sum_{(u,v)^r\in\Psi | i\in N_r} \lambda_{uv}^r \leq 1, \quad \forall i \in N, \qquad (32)$$

$$\sum_{(u,v)^r\in\Psi} \lambda_{uv}^r - \sum_{(v,y)^s\in\Psi} \lambda_{vy}^s = \begin{cases} -z & \text{, if } v = 0 \\ 0 & \text{, if } v = U, \ldots, W-U, \\ z & \text{, if } v = W \end{cases} \qquad (33)$$

$$z \leq k, \qquad (34)$$

$$\lambda_{uv}^r \geq 0 \text{ and integer}, \quad \forall (u,v)^r \in \Psi, \qquad (35)$$

$$z \geq 0 \text{ and integer}. \qquad (36)$$

The objective function (31) is equivalent to the ones of the previous models. Constraints (32) ensure that any customer is visited at most once and constraints (33) are flow conservation constraints. Finally, constraint (34) ensures that the total flow going trough the graph does not exceed k, which means that no more than k workdays are defined and, therefore no more than k vehicles are used.

2 IMPROVED ITERATIVE ALGORITHM

An iterative algorithm based on model (31)-(36) has been proposed in (Macedo et al., 2011). In this paper, we investigate several improvements of this convergent iterative algorithm. In (Macedo et al., 2011) the discretization unit is constant (i.e. $U = 1$). We propose a generalization by considering different discretization units for the set of nodes Δ. We also consider a different rounding rule and a different disaggregation of nodes, and propose a new integer model to assess exactly whether a given solution is feasible or not. In this section, we describe these different steps of the global algorithm.

2.1 Rounding Procedure and Possible Infeasibilities

In (Macedo et al., 2011), the rounding strategy to transform values u and v of every arc $(u,v)^r \in \Psi$ into the discrete values belonging to the set of vertices $\Delta = \{0,\dots,W\}$ of graph Π consisted of considering $u = \lceil u \rceil$ and $v = \lfloor v \rfloor$. This rounding strategy leads to a relaxation of the problem and, therefore, its solution represents a lower bound, which means that it may be infeasible. In this case, infeasibilities may occur whenever there are two routes r_1 and r_2 in the solution, represented by arcs $(u_1,v_1)^{r_1}$ and $(u_2,v_2)^{r_2}$, such that (considering $v_1 \leq u_2$, without loss of generality) $v_1 = u_2$ or $v_1 = u_2 - 1$.

For our new algorithm, we use a different rounding strategy. We consider every value u as $u = \left\lfloor \frac{u}{U} \right\rfloor \times U$. This rounding procedure also leads to a relaxation of the problem, but it only eventually originates one of the previous infeasibilities, for the cases where $v_1 = u_2$.

For every route $r \in R$, the arcs to consider are arcs $(u,v)^r$ such that $u \in (T_r^{beg} \bigcap \Delta) \bigcup \left\{ \left\lfloor \frac{T_r^-}{U} \right\rfloor \times U \right\}$ and $v = \lfloor u + \sigma_r \rfloor$. Figure 1 illustrates the arcs to be considered for route r. In this case, given that $p_1 < T_r^- < p_2$ and $p_4 < T_r^+ < p_5$ (and considering that $\lfloor p_1 + \sigma_r \rfloor = p_6$), the arcs that represent route r would be $(p_1,p_6)^r, (p_2,p_7)^r, (p_3,p_8)^r$ and $(p_4,p_9)^r$.

2.2 Variable Discretization

For each feasible route $r \in R$, there is a time interval $T_r^{beg} = [T_r^-, T_r^+]$ that represents the instants at which route r can begin, in order to be feasible and to have the minimum possible duration. Values T_r^- and $T_r^+, \forall r \in R$, can be recursively calculated (Macedo et al., 2011). Route r beginning at instant t is represented by r_t. As explained in (Macedo et al., 2011), route r_t such that $t \notin T_r^{beg}$ is either infeasible or it is dominated by route $r_{T_r^-}$. The number of considered routes is therefore equal to $\sum_{r\in R} \frac{(T_r^+ - (T_r^+ \bmod U)) - (T_r^- - (T_r^- \bmod U))}{U} + 1$. In (Macedo et al., 2011), $U = 1$. We now generalize this concept and consider that U can take different values. It is clear that considering $U = u_1$ rather than $U = u_2$, with $u_1 > u_2$, implies having a smaller model, with less variables and constraints. On the other hand, given that with $U = u_1$ we have a coarser rounding, the number of iterations of the algorithm may be larger, as there is a higher probability of obtaining an infeasible solution. In section 3, we test, for each instance, different values of U.

2.3 New Disaggregation Method

When the model finds a solution that is infeasible, all nodes that are simultaneously the beginning and the end of at least two arcs in the solution, are disaggregated. To disaggregate a node p that belongs to the original set of nodes Δ_0 and has not yet been disaggregated means to consider additional nodes between p and $p+U$. The number of new nodes to add depends on the chosen unit ε. Disaggregating point p implies adding the $(\varepsilon - 1)$ equidistant nodes $\left\{p + \left(\frac{U}{\varepsilon}\right), \dots, p + U - \left(\frac{U}{\varepsilon}\right)\right\}$ to the graph. Whenever there is such a node, p^*, that does not belong to Δ_0 or has already been disaggregated in a previous iteration, the node to be disaggregated is the highest node $p' \in \Delta_0$, such that $p' \leq p^*$. Let ε^* be the distance between any two consecutive nodes in $[p', p'+U]$. The set of $(a\varepsilon^* - 1)$ equidistant nodes to be added to the original graph is $\left\{p' + \left(\frac{U}{a\varepsilon^*}\right), \dots, p' + U - \left(\frac{U}{a\varepsilon^*}\right)\right\}$.

When nodes are added to the graph, there are additional arcs to be considered. Figure 2 illustrates the disaggregation of node p, and the modifications that occur in what concerns arcs representing routes r_1 and r_2.

2.4 Exact Feasibility Model

A solution given by model (31)-(36) is represented by a set of arcs. Given a set of arcs, it may be possible to find different solutions, as illustrated in figure (3).

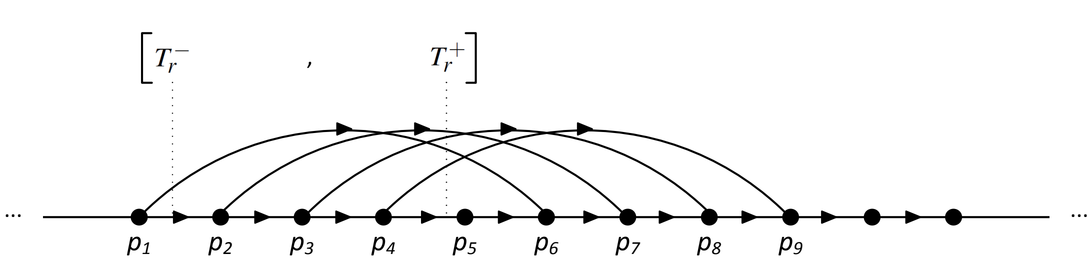

Figure 1: Arcs representing route r.

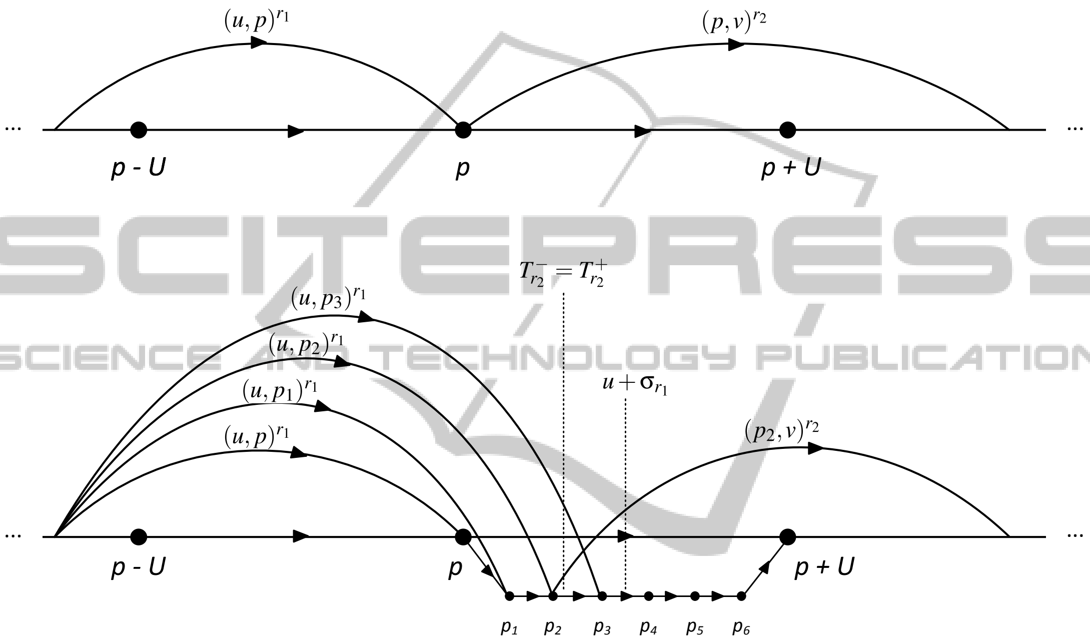

Figure 2: Disaggregation of node p, with $\varepsilon = 7$.

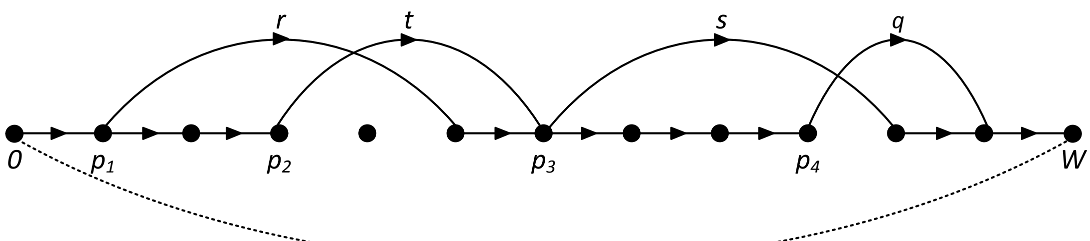

Figure 3: Arcs in a solution given by model (31)-(36).

For this solution, there are four routes to be performed by two vehicles. But we can have a vehicle k_1 performing route r_{p_1} and then route s_{p_3} and a vehicle v_2 performing route t_{p_2} followed by route q_{p_4}, or we can allocate routes r_{p_1} and q_{p_4} to vehicle v_1 and routes t_{p_2} and s_{p_3} to vehicle v_2. Although the first solution is guaranteed to be feasible, the second solution may not be, as node p_3 is simultaneously the beginning and ending node of two routes performed by the same vehicle.

We use model (19)-(26), with a slight modification, to assesses exactly whether it is possible or not to build a feasible solution with the arcs of the solution, instead of only checking if one possible solution is feasible or not.

Let $R' \subseteq R$ be the set of routes associated to the arcs that compose the optimal solution given by model (31)-(36). The model states as follows.

$$\min C \tag{37}$$

$$\text{s.t. } \sum_{k=1}^{m} \mu_r^k = 1, \forall r \in R', \tag{38}$$

$$t_r + \sigma_r \leq t_{r'} + M(2 - \mu_{r'}^k - \mu_r^k) + M(1 - \varepsilon_{rr'}),$$
$$\forall r, r' \in R', \tag{39}$$

$$t_{r'} + \sigma_{r'} \leq t_r + M(2 - \mu_{r'}^k - \mu_r^k) + M(1 - \varepsilon_{r'r}),$$
$$\forall r, r' \in R', \tag{40}$$

$$\varepsilon_{rr'} + \varepsilon_{r'r} = 1, \quad \forall r, r' \in R', \tag{41}$$

$$\mu_r^k \in \{0,1\}, \quad \forall r \in R', \forall k \in K, \tag{42}$$

$$\varepsilon_{rr'} \in \{0,1\}, \quad \forall r, r' \in R', \tag{43}$$

$$t_r \in [T_r^-, T_r^+], \forall r \in R'. \tag{44}$$

The differences between model (19)-(26) and (37)-(44) rely on constraints (38). As an input, there are only the routes that are present in the solution given by model (31)-(36). With this model, we want to assess whether it is possible or not to build a feasible solution with those chosen routes. That is the reason for the equality in the constraints, as opposed to the inequality in (19). Moreover, the objective function (37) is not relevant, and thus we can just set it to a constant value C. The solution found by model (31)-(36) is feasible if and only if model (37)-(44) finds a feasible solution for set R'.

2.5 Generalized Algorithm

We propose a generalization of the method described in (Macedo et al., 2011), with some further modifications. Algorithm 1) summarizes this method.

All feasible routes are implicitly generated. Graph $\prod = (\Delta, \Psi)$ is then built using the new rounding and discretization rules described in sections 2.2 and 2.1. We then apply all the arc reduction criteria defined in (Macedo et al., 2011), in order to increase the model's efficiency. Model (31)-(36) is solved and its solution's feasibility is checked by solving the exact feasibility model (37)-(44). Whenever there is an infeasibility, we proceed to some local node disaggregations, as described in section 2.3, and repeat the process.

Algorithm 1: Iterative Disaggregation Algorithm.

Input: Instance I of the MVRPTW, U, ε and a
Output: Optimal solution x^*
Build $\Pi = (\Delta, \Psi)$;
optimal=False;
while *optimal=False* **do**
  Apply arc reduction;
  Solve I with model (31)–(36), obtaining solution x';
  Check if x' is feasible by solving model (37)–(44);
  if x' *is feasible* **then**
    $x^* = x'$, optimal=True
  else
    Apply disaggregation obtaining $\Pi' = (\Delta', \Psi')$;
    $\Pi = \Pi'$;

3 COMPUTATIONAL RESULTS

To investigate the effects of considering different values for the discretization unit, we conducted a set of computational experiments on benchmark instances from the literature. The considered instances are the same ones described in (Azi et al., 2010; Macedo et al., 2011).

The algorithm was implemented in C++ and the network flow model was solved with ILOG CPLEX 12.2. The computational tests were run on a PC with Intel Core i7, CPU with 2.00GHz and 4GB of RAM.

The cost of a route is considered to be equal to its traveled distance. The used parameters, defined in the previous sections, were set to $k = 2$, $\alpha = 2max_{(i,j)\in A} d_{ij} + 1$, $\beta = 0.2$, $g_i = 1, \forall i \in N$, $\varepsilon = 4$ and $a = 2$.

Table 1 reports the computational results obtained for some of these instances, and for some different values of U within each of the instances. We consider values of U smaller than the minimum duration of all routes. Columns *Inst*, n and t_{max} refer, respectively, to the instance's name, the number of customers and the value of the maximum route duration. The number of different routes is represented by $|R|$, and the percentage of visited customers and optimal solution are represented by %Cust and z^*. All of the values previously described are equal for a same instance. We then report, for each value of U, the number of all routes, for all their beginning instances, $|R_{total}|$, the number of iterations of the algorithm, n_{it}, and the computational time, t, in seconds. The two last columns, $\%|R_{total}|$ and $\%t$, report the percentage of the values of $|R_{total}|$ and t when compared to the ones obtained with $U = 1$.

When the value of U increases, the total number of routes decreases in an approximate proportion. In what concerns the computational times, this decrease is even more expressive, except for some cases where the number of iterations increases considerably.

All instances are solved within the first two iterations when $U = 1$. For most of them, the problem is even solved in the first one. This had been already observed in (Macedo et al., 2011). When the value

Table 1: Computational results for different values of U.

Inst	n	t_{max}	$\|R\|$	%Cust	z^*	U	$\|R_{total}\|$	n_{it}	t (s)	$\%\|R_{total}\|$	$\%t$
RC201	25	75	92	100	988.20	1	6237	1	1.55	100.00%	100.00%
						20	401	1	0.14	6.43%	9.03%
						40	248	2	0.11	3.98%	7.10%
						60	190	2	0.14	3.05%	9.03%
R204	25	75	922	100	579.75	1	325871	1	310.45	100.00%	100.00%
						7	47163	1	4.64	14.47%	1.49%
						14	23933	1	1.30	7.34%	0.42%
						20	17045	1	1.16	5.23%	0.37%
C202	25	220	467	100	653.50	1	166767	1	754.30	100.00%	100.00%
						7	24193	1	2.57	14.51%	0.34%
						14	12300	1	0.84	7.38%	0.11%
						20	8742	1	2.54	5.24%	0.34%
RC202	40	75	425	92.5	1458.09	1	67063	1	15.28	100.00%	100.00%
						20	3734	5	10.72	5.57%	70.16%
						40	2069	8	50.95	3.09%	333.44%
						60	1498	8	42.28	2.23%	276.70%
R209	40	75	2330	100	935.95	1	284468	1	434.38	100.00%	100.00%
						4	72855	3	125.88	25.61%	28.98%
						7	42543	1	14.48	14.96%	3.33%
						10	30604	5	68.04	10.76%	15.66%
C208	40	220	772	100	1072.22	1	258795	1	84.66	100.00%	100.00%
						7	37556	1	4.22	14.51%	4.98%
						14	19108	1	2.42	7.38%	2.86%
						20	13609	1	1.18	3.38%	1.39%
RC207	25	100	4515	100	514.90	1	320345	1	65.30	100.00%	100.00%
						20	20192	6	14.46	6.30%	22.14%
						40	12326	6	13.31	3.85%	20.38%
						60	9536	5	10.33	2.98%	15.82%
R202	25	100	2909	100	617.60	1	286897	1	50.98	100.00%	100.00%
						7	43376	1	6.26	15.12%	12.28%
						14	23102	1	3.28	8.05%	6.43%
						20	17056	1	2.66	5.94%	5.22%
C207	25	250	1685	100	525.57	1	577576	1	79.39	100.00%	100.00%
						7	83904	1	4.38	14.53%	5.52%
						14	42847	1	2.54	7.42%	3.06%
						20	30479	1	2.03	5.28%	2.33%
RC201	40	100	457	85	1157.65	1	21966	1	3.33	100.00%	100.00%
						20	1542	4	1.61	7.02%	48.35%
						40	997	5	1.91	4.54%	57.36%
						60	818	6	4.01	3.72%	120.42%
RC205	40	100	2209	95	1195.51	1	134050	1	42.09	100.00%	100.00%
						20	8786	3	3.60	6.55%	8.55%
						40	5514	3	1.87	4.11%	4.44%
						60	4400	3	2.09	3.28%	4.97%
C205	40	250	1153	100	921.37	1	137358	2	619.48	100.00%	100.00%
						7	20518	2	4.42	14.94%	0.71%
						14	10789	1	2.43	7.85%	0.39%
						20	7952	1	1.85	5.79%	0.30%

of U increases, the rounding becomes coarser. This translates into a greater number of iterations for some of the instances, although it is not always the case.

4 CONCLUSIONS

In this paper, we address a variant of the vehicle routing problem with time windows and multiple use of vehicles. We propose an improvement of the algorithm described in (Macedo et al., 2011). It mainly consists of considering variable discretization units for the time interval represented by the set of nodes of the underlying graph. We also propose new rounding and disaggregation rules. This new algorithm was tested with the set of instances proposed in (Azi et al., 2010; Macedo et al., 2011). The results show considerable improvements of the computational times for most of the instances, when we increase the value of the discretization unit.

REFERENCES

Azi, N., Gendreau, M., and Potvin, J.-Y. (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.

Baldacci, R., Christofides, N., and Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385.

Baldacci, R., Toth, P., and Vigo, D. (2010). Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175(1):213–245.

Cordeau, J.-F., Laporte, G., Savelsbergh, M., and Vigo, D. Vehicle routing. In Barnhart, C. and Laporte, G., editors, *Transportation, Handbooks in Operations Research and Management Science*, volume 14. Elsevier, Amsterdam.

Dantzig, G. and Ramser, J. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

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

Fukasawa, R., Longo, H., Lysgaard, J., de Aragão, M. P., Reis, M., Uchoa, E., and Werneck, R. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.

Lysgaard, J., Letchford, A., and Eglese, R. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445.

Macedo, R., Alves, C., Valério de Carvalho, J., Clautiaux, F., and Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214(3):536 –545.

Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.