PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 04-13

Adapted Clustering Algorithms for the Assignment Problem in the MDVRPTW

Libertad Tansini Omar Viera,

noviembre de 2004

Libertad Tansini, Omar Viera
Adapted Clustering Algorithms for the Assignment Problem in the MDVRPTW
ISSN 0797-6410
Reporte Técnico RT04-13
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay, noviembre de 2004

Adapted Clustering Algorithms for the Assignment Problem in the MDVRPTW

Ing. Libertad Tansini Ing. Omar Viera, MSc.

Dpto. Investigación Operativa, Instituto de Computación, Facultad de Ingeniería

Universidad de la República Oriental del Uruguay, Montevideo, Uruguay.

tel: (598-2) 711 42 44, fax: (598-2) 711 04 69, [libertad, viera] @fing.edu.uy

Correspondence should be addressed to:

Ing. Omar Viera, MSc.

Facultad de Ingeniería

Julio Herrera y Reissig 565 - 5o. piso

Montevideo - URUGUAY

Tel: (598)(2) 711.42.44 / 47 - Fax: (598)(2) 711.04.69

E-mail: viera@fing.edu.uy

Abstract

This paper proposes new applications of statistical and data mining techniques for the assignment problem in the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). Given the intrinsic difficulty of this problem class, approximation methods of the type "cluster first, route second" (two step approaches) seem to be the most promising for practical size problems. After describing five assignment algorithms designed specially for assignment of customers to depots (the cluster phase), the adapted clustering algorithms for the assignment problem are introduced and a preliminary computational study of their performance is presented. Concluding as expected, that the they can be adapted to solve this type problem and many times give very good results (in terms of the routing results), but are still far from some of the other algorithms when it comes to execution times.

Key words: multi-depot vehicle routing problem, clustering, assignment, time windows.

A key element of many distribution systems is the routing and scheduling of vehicles through a set of customers requiring service. The Vehicle Routing Problem (VRP) involves the design of a set of minimum-cost vehicle routes, originating and terminating at a central depot, for a fleet of vehicles that services a set of customers with known demands. Each customer is serviced exactly once and, furthermore, all customers must be assigned to vehicles without exceeding vehicle capacities. In the Vehicle Routing Problem with Time Windows constraints (VRPTW), the issues must be addressed under the added complexity of allowable delivery times, or time windows, stemming from the fact that some customers impose delivery deadlines and earliest-delivery-time constraints.

Whereas the VRP and VRPTW have been studied widely, the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) has attracted less attention.

In the MDVRPTW, customers must be serviced out from one of several depots. As with the VRP, each vehicle must leave and return to the same depot and the fleet size at each depot must range between a specified minimum and maximum. The MDVRPTW is NP-hard, see Bodin, Golden, Assad and Ball (1983) and Lenstra and Rinnooy Kan (1981), therefore, the development of heuristic algorithms for this problem class is of primary interest.

The MDVRPTW can be viewed as a clustering problem in the sense that the output is a set of vehicle schedules clustered by depot. This interpretation suggests a class of approach that clusters customers and then schedules the vehicles over each cluster.

This paper focuses on the assignment ("cluster" part) of customers to depots using known statistical and data mining techniques. First a brief description and analysis of five algorithms is offered, later a more in depth comparison of assignment algorithms designed specifically for the MDVRPTW and adapted clustering algorithms for the assignment is shown.

To compare the assignment heuristics it is necessary to produce a final set of routes, in order to do this the same VRPTW heuristic is used for each depot, Clark and Wright (1964), finally it is possible to compare the routing results for each one of the assignments.

Due to the operational nature of most of the MDVRPTW the computing time is an important aspect. Often in real life applications, the assignment of customers to depots and the construction of the routes for each cluster must be done, in the best case, from one day to another.

The study of the adapted clustering algorithms shows interesting results and possible paths to follow, but these algorithms are still not very competitive with some of the classical ones in terms of the trade off between execution time and final results measured in total distance.

A comprehensive survey on VRP can bee found in Bodin, Golden, Assad and Ball (1983), for specific heuristics for VRP see Clark and Wright (1964) and Mole and Jameson (1976). For a discussion on complexity of vehicle routing and scheduling problems see Lenstra and Rinnooy Kan (1981). For further reading on formulations of VRPTW see Bodin, Golden, Assad and Ball (1983), Bramel and Simchi-Levi (1997), and on algorithms for VRPTW see Potvin and Rousseau (1993), Solomon (1987), Salhi and Nagy (1999) and Ioannou, Kritikos and Prastacos (2001). Formulations and algorithms for solving MDVRP and MDVRPTW can be found in Bodin, Golden, Assad and Ball (1983) and Salhi and Nagy (1999). The idea for one of the algorithms presented in this paper comes from assigning customers to days of the week, which can be found in Russell and Igo (1979). A problem from the dairy industry motivated this paper: the daily transportation of milk from farms to processing plants, for background on the project that motivated this paper see Giosa, Tansini and Viera (1999). Tools, real life problems, applications related to this paper and interfaces using GIS, see ESRI and Geographic Information Systems, are in Urquhart, Viera, Gonzalez and Cancela (1997) and “A Vehicle Routing System Supporting Milk Collection” Urquhart and Viera (2002). For more information on the test cases and the software tool used to obtain the results shown in this paper see Giosa, Tansini and Viera (2002).

Tis paper is organised as follows: in the following section a short problem definition is given. The assignment algorithms are succinctly presented in the second section. The third section contains a description of the proposed adapted clustering algorithms. The fourth section contains the routing algorithm which is used to calculate costs of the different assignments. Computational results are presented and analysed in the fifth section. Finally in the last section conclusions and some ideas for future research are discussed.

1 Problem definition

The MDVRPTW consists of determining a set of vehicle routes in such a way that, Bodin, Golden, Assad and Ball (1983):

a) each route starts and ends at the same depot;

b) all customer requirements are met exactly once by a vehicle;

c) the time windows for both customers and the depots are respected;

d) the sum of all requirements satisfied by any vehicle does not exceed its capacity;

e) the total cost is minimised.

The MDVRPTW can be viewed as being solved in two stages: first, customers must be allocated (assigned) to depots; then routes must be built that link customers assigned to the same depot. Ideally, better results are obtained dealing with the two steps simultaneously, see Ioannou, Kritikos and Prastacos (2001) and Salhi and Nagy (1999). When faced with larger problems, say 1000 customers or more, however, this approach is no longer tractable computationally. A reasonable approach would be to divide the problem into as many sub problems as there are depots, and to solve each sub problem separately.

The algorithms described in the next section attempt to implement this strategy.

1 Assignment algorithms

First it is worth noting that the assignment problem and the routing problem in the "cluster first, route second" approach are not independent. A bad assignment solution will result in routes of higher total cost (distance) than one with a better assignment, as Figure 1 shows.

Initially in our case, Giosa, Tansini and Viera (2002), the time windows constraints were used to check for compatibility between customers and depots (to answer the question if it is possible to get from one customer to the depot in time) in the assignment step and for route feasibility in the routing step, later they were included in the proximity measures between costumers and/or depots. All the assignment algorithms described below assign

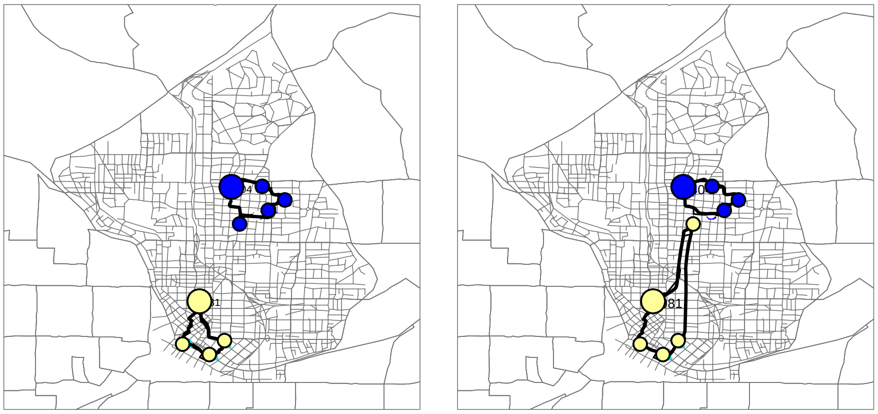

Figure 1 Comparing two routing costs: 6.4 Kms. for the assignment on the left and 9.1 Kms for a “worse” assignment on the right.

customers to depots so that the capacity of the depots is not exceeded.

Finally, the methods we are about to describe have been presented previously in Giosa, Tansini and Viera (2002) and, due to the lack of documentation about solutions for the assignment problem related to the MDVRPTW, some of them are in some sense adaptations of more or less well known heuristic solutions for the VRP and/or VRPTW, see Bodin, Golden, Assad and Ball (1983).

A general pseudocode for all the algorithms is as follows:

Until all customers have been assigned to the depots

determine the next best customer to be assigned and to which depot,

taking into account the demand of the customers, the capacity of the depots and the time windows for both customers and depots.

The algorithms presented in the this section use different criteria for the assignment of a customer to a depot: assignment through urgencies and assignment by clusters.

1.1 Assignment through urgencies

The urgency is a way to define a precedence relationship between customers; the urgency to be assigned could also be viewed as a priority. This precedence relationship determines the order in which customers are assigned to depots. The customers with the highest urgency are assigned first. The algorithms in this class vary only in the way the urgencies are calculated, see Giosa, Tansini and Viera (2002). The *Parallel* and *Simplified Parallel* algorithms are adaptations of the algorithm for VRPTW presented by Potvin and Rousseau (1993). The Sweep Assignment is the third heuristic in this class and the urgency is measured as the difference between assigning a customer to its closest depot and to depot $\boldsymbol{dep}^*$ with the highest unsatisfied demand. Only the Simplified parallel assignment will be described with more detail because proved to give better results and it will be used for further comparisons in this paper.

1.1.1 Simplified parallel assignment

In this case, two depots are involved in the evaluation of the urgency:

$$\mu_c = d(c, dep'') - d(c, dep')$$

Where ***d(c, d''(c))*** is the distance between customer ***c*** and its second closest depot and ***d(c, d'(c))*** is the distance between customer ***c*** and its closest depot.

Figure 2 gives an example of the urgency for one customer, represented by the solid black line, because in this algorithm the urgency is calculated as the difference between the distances to a customer's closest and second closest depot. The dotted circle shows the distance to the a customer's closest depot.

Once again, the customer with the greatest value of μ is assigned to its closest depot.

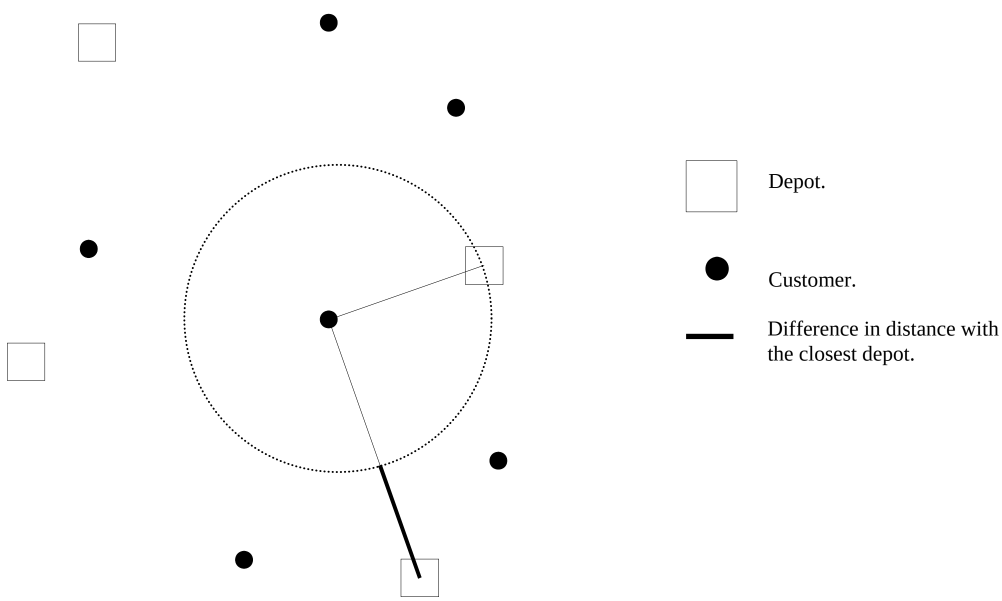

Figure 2 How the urgency is calculated for one client in the Simplified parallel assignment algorithm.

This heuristic compares the cost of assigning a customer to its closest depot with the cost of assigning it to its second closest depot. The most urgent customer is the one for which μ is maximum. This is a variant of the most common assignment algorithms found in the literature, Bodin, Golden, Assad and Ball (1983).

In order to incorporate time windows the following definition of urgency is used, Tansini (2001):

$$\mu_c = Closeness(c, dc'(c)) - Closeness(c, dc''(c)) \quad c \in C$$

Closeness is defined to take into account Affinity and distance:

$$Closeness(i,j) = d(i,j) / Affinity(i,j) \quad j \in D \quad i \in C$$

Where ***d(i,j)*** is the distance between ***i*** and ***j***.

The Affinity measures the degree of similarity in terms of the time windows of a customer with the group of customers already assigned to a depot:

$$Affinity(i,d) = \left\{ \frac{\sum_{j \in C(d) \cup \{d\}} e^{-(DTW(i,j) + TV_{ij})}}{|C|} \right\} \quad d \in D \quad i,j \in C$$

Where ***D*** is the set of depots and ***C*** the set of customers and ***C(d)*** is the set of already assigned customers to depot ***d***. TV_{ij} is the traveling time between ***i*** and ***j***.

DTW measures the distance in the time windows of a customer with another customer or with a depot:

$$DTW(i,j) = \begin{cases} e_j - l_i & si\ l_i < e_j \\ e_i - l_j & si\ l_j < e_i \\ 0 & otherwise \end{cases}$$

Where, ***l*** y ***e*** represent the beginning and end of the time window.

If considering only time windows, ideally a customer should be assigned to the depot whose already assigned customers are nearest in terms of time windows, that is the depot that maximizes the Affinity. On the other hand, considering distances, a customer should be assigned to the closest depot.

The complexity for the whole algorithm is (in the worst case) O($3CD+CD^2+C^2D$), where ***C*** is the number of customers, and ***D*** is the number of depots.

1.2 Assignment by clusters

A cluster is defined as the set of points consisting of a depot and the customers assigned to it. The algorithms in this class try to build compact clusters of customers for each depot, see Giosa, Tansini and Viera (2002). When a customer is assigned to a cluster it means that this customer is assigned to that cluster's depot. Two heuristics in this class were studied: Coefficient propagation and Three criteria clustering. Three criteria clustering proved to give better results and is therefore used in this paper for further comparisons and will be described in this section. In the Coefficient propagation algorithm, attraction coefficients are associated to depots and already assigned customers, these coefficients scale the distances with the unassigned customers.

1.2.1 Three criteria clustering

The criteria used by this algorithm to include a customer in a cluster are: average distance to the clusters, variance of the average distance to the customers in the clusters and distance to the closest customer in each cluster (the algorithm is an adaptation of another algorithm see Russell and Igo (1979)).

If there is a customer with an average distance to its closest cluster sufficiently smaller than the average distance to its second closest cluster (10% improvement or more), then it can be assigned; the one that maximizes de difference in average distances is assigned to its closest cluster. Otherwise the variance of the average distance is taken into account and if there is a customer with a small enough variance of the average distance to its closest cluster (40%) then it can be assigned, again the one that maximizes the difference is assigned. Finally if the first two measures fail, the decision is made based on the distance to the closest customer in its closest cluster, now the customer that minimizes this distance is assigned.

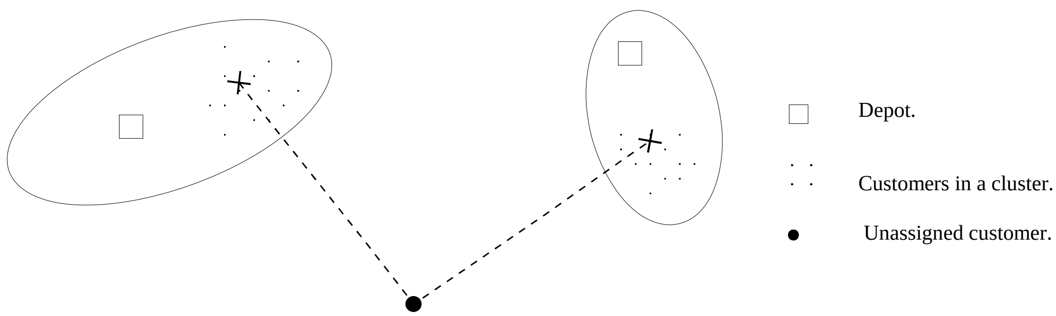

Figure 3 Determining the average distance to the clusters.

Figure 3 shows two clusters with a depot and a set of customers each. The average distance from the unassigned customer to the clusters is calculated as the average of the distances to all customers and the depot in the clusters, and can be viewed as a fictitious customer with the average coordinates of all customers and depot in the cluster, shown in Figure 3 as a cross.

Figure 4 shows two clusters with different variance of the average distance to the

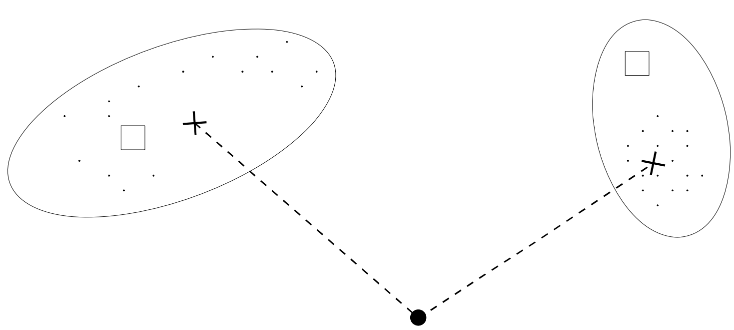

Figure 4 Clusters with different variance.

customers and depot in the clusters, the cluster on the right hand has a lower variance, since the customers are closer to each other than in the left cluster.

Figure 5 shows the closest customer in each cluster to a non-assigned customer.

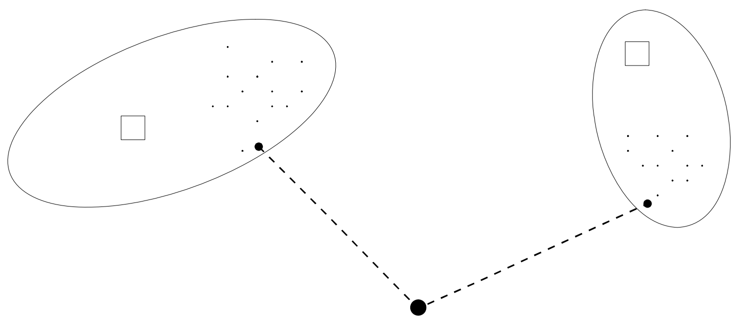

Figure 5 Closest customer in each cluster

The measure Angle, see Tansini (2001), is used to incorporate time windows, it derives from the comparison of vectors found when exploring clustering algorithms Linoff (1995). Originally the Angle measure is used to cluster data with categorical attributes, that are those belonging to non-ordered domains and for which no distance function can be defined naturally, for example colors.

The Angle is defined as following:

$$\boldsymbol{Angle(i,j) = Cos^{-1}\left(\frac{x_i \cdot x_j + y_i \cdot y_j + l_i \cdot l_j + e_i \cdot e_j}{(x_i^2 + y_i^2 + l_i^2 + e_i^2)^{1/2}(x_j^2 + y_j^2 + l_j^2 + e_j^2)^{1/2}}\right)}$$

Were $\boldsymbol{x}$ and $\boldsymbol{y}$ are the coordinates and $\boldsymbol{l}$ y $\boldsymbol{e}$ represent the beginning and end of the time window, $\boldsymbol{i}$ and $\boldsymbol{j}$ are customers or depots.

The complexity of the whole algorithm is (in the worst case) $O(3C^2D+3C^2D^2+CD^2)$.

An earlier study, see Giosa, Tansini and Viera (2002), shows that of the Assignment through Urgencies, the Simplified parallel assignment (SPA) is the algorithm with the best behaviour comparing results and execution times. Similarly the Three criteria assignment is the one that gives the best result among the Assignment by clusters algorithms. Therefore the adapted clustering algorithms will be compared against these two algorithms.

2 Adapted clustering algorithms

The time windows constraints in the MDVRPTW are enemies of the "cluster first, route second" approach. In this way it is possible to obtain geographically compact clusters of customers for each depot, but due to the time windows constraints, these clusters may contain customers with very different service times. For example if two nearby customers may have service times one in the morning and one in the afternoon, this will require two different routes one for the morning customer and one for the afternoon customer.

If the aim is to obtain routes with short waiting time between customers, the customers should not only be close to each other but also have similar time windows. When considering time windows, customers in one geographic cluster may have similar time window with customers in another cluster, implying longer waiting times in the routes or the need for extra routes to service them in the original cluster. This can be seen in Figure 6

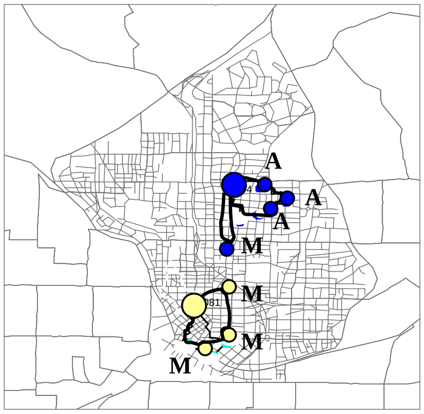

Figure 6 An assignment that doesn't take into account location and time window similarities when assigning customers to depots.

where two routes are necessary to visit the morning (**M**) and afternoon customers (**A**) for one of the depots, giving a total cost of 9.5 Km. In Figure 7 the morning customer is assigned with the rest of the morning customers giving a lower routing cost of 8.3 Km.

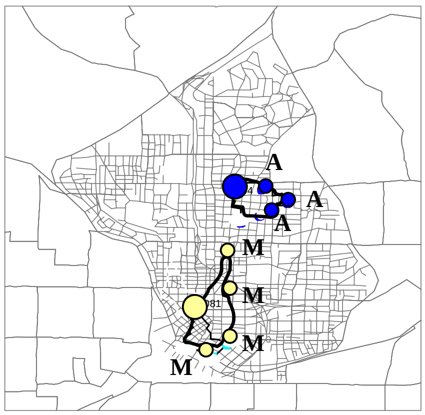

Figure 7 When considering time windows a better assignment takes into account location and time window similarities when assigning customers to depots.

In this paper clustering algorithms are adapted to the assignment of customers to depots. Clustering techniques belong to Statistics and are used to group similar data. They are used mainly in multivariate statistics, and though originated several years ago they have recently known greater development related to their application in the Data Mining field.

In this case the data are the customers and depots and the similarities looked for are in terms of geographical location and time windows. The fact that similarities have to be fund for several variables, in this case location and time windows, is what suggests considering the employment of clustering algorithms.

In order to adapt some classical clustering algorithms to the assignment of customers for the MDVRPTW it is necessary to add some restrictions, for example: each cluster must have one depot and the productions of the customers must not exceed the capacity of the depot in each cluster.

Two large classes of clustering algorithms, Linoff (1995), were chosen to be adapted to the assignment of customers for the MDVRPTW: Partitioning and Hierarchical. The selected Partitioning algorithms are: KNN, K-Means and PAM. The Hierarchical are: Agglomeration and Rock. These algorithms were chosen due to characteristics that made them more adaptable to the assignment problem.

In order to compare the results produced by the clustering algorithms it is necessary to compare the routing results, as in the previous cases.

2.1 Partitioning algorithms

2.1.1 KNN

The name KNN comes K-Nearest Neighbors because the K nearest neighbors to a new element help determine which cluster it will belong to. The new element will belong to the cluster with the highest number of already classified elements amongst the K nearest neighbors. It is important to consider that the order in which the elements are classified determines the final result. This influence could be removed by running the algorithm

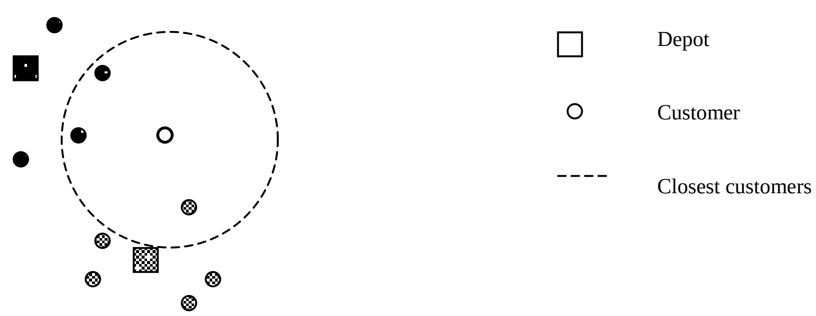

Figure 8 Finding the closest neighbours in the KNN algorithm, when $K = 3$. In this case the white customer has to be assigned to the dark or light coloured depots. Because it has two dark closest neighbours, it must be assigned to de dark depot

several times classifying the elements in different order and keeping the best result, which would of course increase the complexity of the algorithm, so that option is not considered in this paper.

The complexity of this algorithm is O(*DC*), were *C* y *D* are the number of customers and depots. The complexity in this case as well as the following are calculated based on the implementation.

2.1.2 K-Means

This algorithm is named K-Means because the means of the clusters are used to classify the element. Each element must belong to the cluster with closest mean. The distance to the mean takes into account both geographical coordinates and time window centres, this is explained further on in the section.

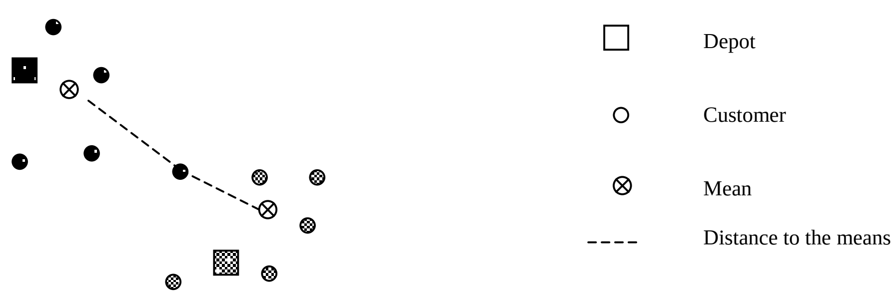

Figure 9 Finding the means for two clusters in the K-Means algorithm. The considered dark customer should be reassigned to the lighter depot, since its mean is closest. Then the means must be recalculated.

Initially K clusters are selected and their means are calculated. After that, each element is reassign to the cluster with closest mean. With the new clusters, the means are recalculated and the elements are reassigned with the same criterion until there are no reassignments.

The complexity of this algorithm is $O(DC^2)$, were C y D are the number of customers and depots.

2.1.3 PAM

PAM refers to Partitioning Around Medoids, because instead of using the means of the clusters as in K-Means, here Medioids are used. These are real elements closest to the means. Each element must belong to the cluster with closest Medioid.

Initially K elements are chosen as Medioids. Then in each step one of the Medioids is replaced in order to look for a better clustering. A clustering is considered better if it has a lower average distance of the elements to it's Medioid. This is done, until there are no more changes.

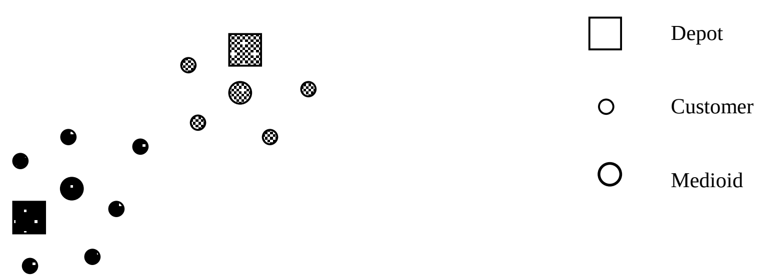

Figure 10 Medioids for two clusters in the PAM algorithm, they are shown as bigger customers.

Figure 10 shows an initial selection of Medioids and the clusters obtained. In Figure 11 a change of Medioid produces different clustering that could potentially turn out to be better, because customers are now closer to their clusters Medioids.

The depots can be chosen to be the first Medioids, then either depots or customers can be Medioids, as long as all restriction are taken into account.

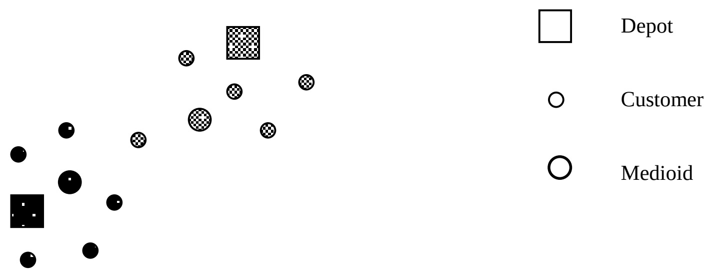

Figure 11 Change of Medioids for the lighter cluster. Due to this change a dark customer is reassigned to the light cluster.

The complexity of this algorithm is O(DC^3), were C y D are the number of customers and depots.

2.1.4 Distance Used in the Partitioning Algorithms

Time windows are incorporated to the usual distance. For two elements $e_i=(x_i, y_i, t_i)$ y $e_j=(x_j, y_j, t_j)$ the following distance was used:

$$WSum(i,j) = w_{xy}((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2} + w_t\left|t_i - t_j\right|$$

Were x and y are the coordinates and t is the time window centre calculated as $t=(e+l)/2$ were e and l are the beginning and end of the time window. w_{xy} and w_t are coefficients such that $w_{xy}>0$, $w_t>0$ and $w_{xy}+w_t=1$.

2.2.1 Agglomeration

This algorithm starts with a cluster for each element and in each step the closest clusters are joined, until the number of cluster does not exceed the number of depots and all restrictions are met. Variations of this algorithms are obtained with different definitions of inter-cluster distance, see Linoff (1995).

Some inter-cluster distances are:

- Single Linkage (SL): the distance between clusters is the minimum distance between pairs of elements belonging to different clusters. With this distance the elements are closer to elements in it’s own cluster than to those belonging to other clusters. Figure 12 shows the SL distance for two clusters.

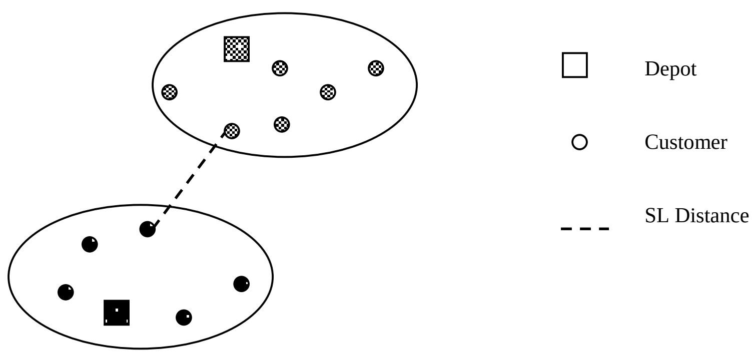

Figure 12 Single Linkage distance (SL) for the Agglomeration algorithm is the distance between the two closest customers.

- Complete Linkage (CL): the distance between clusters is the maximum distance between pairs of elements belonging to different clusters. Figure 13 shows the CL distance for two clusters.

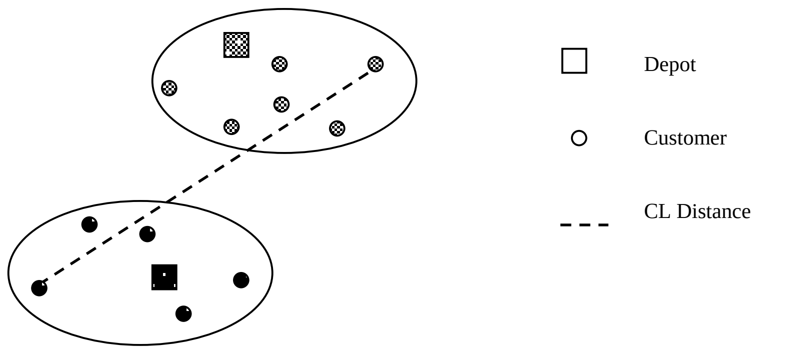

Figure 13 Complete Linkage distance (CL) is the distance between the farthest customers.

- Average Linkage (UPGMC, Unweighted Pair-Groups Method Centroid): this methods compute the centroid or mean of the items that join to from clusters. Figure 14 shows this distance.

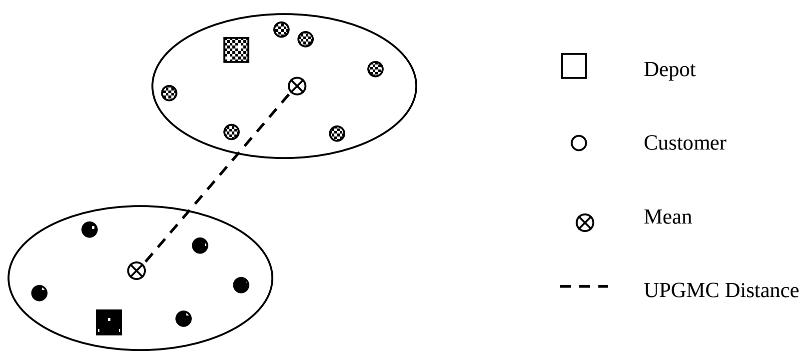

Figure 14 UPGMC distance is the distance between the means of the clusters.

Some other inter-cluster distances that will not be used in this study are:

- Weighted Average Linkage (WPGMC, Weighted Pair-Groups Method Centroid): the WPGMC differs from UPGMC by weighting the member most recently admitted to a cluster equal with all previous members in order to eliminate the influence of the different cluster sizes.

- Centroid (UPGMA, Unweighted Pair-Groups Method Average): the two clusters with the lowest average distance are joined together to form the new cluster.
- Median (WPGMA, Weighted Pair-Groups Method Average): the WPGMA differs from UPGMA by weighting the member most recently admitted to a cluster equal with all previous.
- Within Groups (WG): similar to UPGMA except clusters are fused so that within cluster variance is minimised. This tends to produce tighter clusters than the UPGMA method.
- Ward method (Ward): similar to WG but cluster membership is assessed by calculating the total sum of squared deviations from the mean of a cluster. The criterion for fusion is that it should produce the smallest possible increase in the error sum of squares.

SL and CL are very sensitive to “outliers”, these are elements that differ greatly from the majority.

UPGMC has difficulties in finding compact clusters when they are of different sizes or shapes, WPGMC has a better performance in this sense.

WPGMA gives better results than UPGMA when clusters are of different sizes.

The complexity of these algorithms is $O((D+C)^2+(D+C)^2 log(D+C))$ where D is the number of depots and C is the number of customers.

2.2.2 Rock

This clustering algorithm is based on the "link" between elements concept in order to determine which clusters to combine in each step, Linoff (1995). The link between a pair of elements is defined as the number of neighbours in common. The clusters with largest number of neighbours in common are combined.

In order to determine if a pair of elements are neighbours a similarity function (Sim) and a threshold (T) are used, two elements are considered neighbours if:

$$Sim(i,j) \leq T \quad \text{for elements } i \text{ and } j$$

In this case the function Sim is:

$$Sim(i,j) = \left| distance(i) - distance(j) \right| \quad \text{for elements } i \text{ and } j$$

The following proximity measure between clusters is defined for the application of the rock algorithm:

$$Prox(C_i, C_j) = \frac{Link(C_i, C_j)}{(n_i + n_j)^{1+2f(\theta)} - (n_i^{1+2f(\theta)} + n_j^{1+2f(\theta)})} \quad \text{for clusters } C_i \text{ and } C_j$$

The Link function is the number of links between clusters C_i and C_j:

$$Link(C_i, C_j) = \sum_{p_q \in C_i,\, p_r \in C_j} link(p_q, p_r) \quad \text{for clusters } C_i \text{ and } C_j$$

Were $Link(p_q,p_r)$ is the number of neighbours in common of p_q and p_r, and the function $f(\theta)$ is such that the expected number of neighbours of p_i is $n_i^{f(\theta)}$, with $n_i = |C_i|$ and $p_i \in C_i$. Finding the value of $f(\theta)$ is usually not an easy task, therefore many times an approximate value is used instead. An option to estimate this approximate value is to run a traditional agglomeration algorithm and calculate the average number of neighbours.

The complexity of the Rock algorithm is O($(D+C)^2+(D+C)\ ((D+C)-1)^{an}+(D+C)^2 log\ (D+C)$) were *an* average number of neighbours, D is the number of depots and C is the number of customers.

2.2.3 Distance Used in the Hierarchical Algorithms

The incorporation of time windows is done in the definition of the distances between elements. For two elements $e_i=(x_i, y_i, t_i)$ y $e_j=(x_j, y_j, t_j)$ the following distance was used:

$$Angle(i,j) = Cos^{-1}\left(\frac{x_i \cdot x_j + y_i \cdot y_j + t_i \cdot t_j}{(x_i^2 + y_i^2 + t_i^2)^{1/2}(x_j^2 + y_j^2 + t_j^2)^{1/2}}\right)$$

Were *x* and *y* are the coordinates and *t* is the time window centre calculated as $t=(e+l)/2$ were *e* and *l* are the beginning and end of the time window.

3 Routing algorithm

To compare the heuristics in terms of routing results, it is necessary to solve the MDVRPTW, this means to run the same routing heuristic for all assignments produced by the different assignment algorithms. Then, the best assignment algorithm is the one with the best routing results, or lowest cost (see Figure 1, page 5).

Since it is necessary to produce a final set of routes in order to compare the assignment heuristics, the same VRPTW heuristic (a version of the Clark and Wright heuristic, see Clark and Wright (1964)) is used for all assignment algorithms. The fleet of vehicles is considered unlimited.

A route is considered complete when it is not possible to include another customer due to vehicle load, time window constraints or maximum waiting time allowed in a route.

The concept of Saving (SAV) is used throughout the algorithm. The Saving obtained by including customer *c* in a route between *i* and *j* instead of a route of its own is defined as follows:

$$SAV_c(i,j) = 2 * d(dep,c) - (d(c,j) + d(i,c) - d(i,j))$$

Where *dep* is the depot, *i* and *j* can be either customers or the depot.

A pseudocode for this the algorithms is as follows:

While there are incomplete routes there is a customer c such that $\exists\, SAV_c\,(i, j) \geq 0$

Determine the next customer c on a route of its own which minimizes* $SAV_{c*}\,(i, j)$*, taking into account the demand of the customers, the capacity of the vehicles, the time windows for customers and depots, and the maximum waiting time allowed*

Include c in a route between i and, j*

4 Computational results

4.1 Comparisons

As mentioned earlier, in order to compare the assignment heuristics, it is necessary to solve the MDVRPTW. The best assignment algorithm is considered to be the one with the best routing result (or the lowest cost overall solution, considering the assignment and routing). A measure named Gain was defined and used in order to compare the assignments obtained with algorithm ***i*** and ***j*** the:

$$Gain_D(i,j) = \frac{\left(D_{Total}(\mathrm{j}) - D_{Total}(\mathrm{i})\right)}{D_{Total}(\mathrm{j})} * 100$$

If $Gain_D\,(i,j)$ is positive then algorithm ***i*** has a better performance than ***j***, otherwise it has a worse performance.

With several test cases the average gain is defined algorithm as:

$$\boldsymbol{AverageGain_D(i,j) = \frac{\sum Gain_D(i,j)}{number\ of\ test\ cases}}$$

The average gain turns out to be a global measure of the behaviour of a pair of algorithms compared to each other.

Likewise the total time of the routes obtained with two different algorithms can be compared:

$$\boldsymbol{Gain_T(i,j) = \frac{\left(T_{Total}(\mathrm{j}) - T_{Total}(\mathrm{i})\right)}{T_{Total}(\mathrm{j})} * 100}$$

$$\boldsymbol{AverageGain_T(i,j) = \frac{\sum Gain_T(i,j)}{number\ of\ test\ cases}}$$

4.2 Results

To provide test cases for the experimentation we chose the city map of Atlanta, Georgia, USA, provided by the Arcview 3.0 Geographical Information System, see ESRI. Fifty (50) test cases of different sizes, were generated and taken from the digitalized city map of a portion of Atlanta, which has 1502 street intersections, this gave an upper bound to the problem size (because customers and depots were located on the street intersections). 50 % of the test cases were generated randomly, the remaining 50 % were chosen to test different aspects of the algorithms. The test cases are divided in two groups: random and created test cases. The created cases have depots with short time windows or small capacities, others include depots with equal capacities or very different capacities. These were cases thought to have greater problems with the correct assignment of customers considering time windows and therefore a larger improvement in the results could be obtained, which was proven to be accurate.

The results in this paper were obtained over the test cases using Euclidean distances instead of the real distances calculated on the Atlanta map.

Initially a comparison of the partitioning algorithms was made in order to determine which one adapts better to solving the assignment of customers to depots in the MDVRPTW, Table 1 shows these results for 25 random test cases and 25 particular test cases.

PAM is used as the base algorithm, this means that the rest of the algorithms are compared against it. The different columns show Average $Gain_D$ (KNN with K=1, PAM), Average $Gain_D$ (KNN with K=4, PAM), Average $Gain_D$ (KNN with K=6, PAM) and Average $Gain_D$ (K-Means, PAM), this is how much better in terms of total distance these algorithms are compared to PAM.

Type of taste cases	*Average $Gain_D$ (KNN with K=1, PAM)*	*Average $Gain_D$ (KNN with K=4, PAM)*	*Average $Gain_D$ (KNN with K=6, PAM)*	*Average $Gain_D$ (K-Means, PAM)*
Created test cases	-26.12	-29.43	-33.27	-4.11
Random test cases	-24.70	-28.75	-30.84	-5.89

Table 1 *Average $Gain_D$* of KNN and K-Means against PAM.

As seen in Table 1 the Average $Gain_D$ is always negative, showing that PAM gives the best results.

Similarly the hierarchical algorithms were compared in order to determine which one adapts better to solving the assignment of customers to depots, Table 2 shows these results also for 25 random test cases and 25 particular test cases.

UPGMC is the base algorithm against which the other algorithms are compared. The different columns show Average $Gain_D$ (SL, UPGMC), Average $Gain_D$ (CL, UPGMC) y Average $Gain_D$ (Rock, UPGMC).

Type of taste cases	*Average $Gain_D$ (SL, UPGMC)*	*Average $Gain_D$ (CL, UPGMC)*	*Average $Gain_D$ (Rock, UPGMC)*
Created test cases	-6.73	-4.32	-27.67
Random test cases	-7.13	-5.94	-22.54

Table 2 *Average $Gain_D$* of SL, CL and Rock against UPGMC.

The negative Average $Gain_D$ indicates that the best results are provided by UPGMC.

Now a comparison between more classical assignment algorithms designed specifically for the MDVRPTW and the assignment algorithms derive from clustering is possible. PAM and UPGMC are compared to SPA with Affinity and Three Criteria with Angle Table 3 shows these results. SPA is the base algorithm.

Type of taste cases	*Average $Gain_D$ (Three Criteria, SPA)*	*Average $Gain_D$ (UPGMC , SPA)*	*Average $Gain_D$ (PAM, SPA)*
Created test cases	6.38	9.54	-9.47
Random test cases	5.71	8.77	-12.44

Table 3 *Average $Gain_D$* of Three Criteria, UPGMC and PAM against SPA.

This table indicates that the results provided by UPGMC are in general better than both SPA and Three Criteria because the Average $Gain_D$ with SPA is positive and higher than for Three Criteria. Pam gives worse results but not so far behind the other algorithms. In order to compare the algorithms more thoroughly it is necessary to also take into account the execution times.

Table 4 shows execution times for different sizes of the problem, on a Pentium 2 of 266 MHz with 64 Mb RAM memory and operating system Windows 98.

Algorithm	*1000 Customers 20 Depots*	*450 Customers 15 Depots*	*100 Customers 5 Depots*
PAM	312 sec	59 sec	< 1 sec
UPMG	453 sec	78 sec	4 sec
Three Criteria	297 sec	43 sec	< 1 sec
SPA usando Cercanía	16 sec	2 sec	< 1 sec

Table 4 Execution times on a Pentium 2 of 266 MHz with 64 Mb RAM memory and operating system Windows 98.

Routing times are similar for all assignment algorithms, depending only on the size of the problem, Table 5 shows routing times for other test cases than the ones used in the previous table.

1000 costumers 30 depots	**400 costumers 30 depots**	**100 costumers 6 depots**
950 sec	60 sec	2 sec

Table 5 Routing times for different problem sizes.

5 Conclusions and Further research

5.1 Conclusions

As seen before it is possible to adapt clustering algorithms to the assignment problem in the MDVRPTW. This is relevant because as far as we know there have been no other attempts to do these adaptations.

As proven by the computational results, the solutions given by some of the clustering algorithms are many times better than by the assignment algorithms. The problem with the clustering algorithms is that they have higher the execution times. If the size of the problem to be solved is small or medium this should not be an obstacle.

On the other hand, when solving larger scale problems SPA appears as the most attractive algorithm due to the beneficial trade off between execution time and Average Gain, giving good results in relatively short execution times.

5.2 Further research

One important direction of future research is to investigate classical clustering measures (inter-cluster and intra-cluster distances) used to determine if a clustering solution is acceptable or not in this context. Possibly these measures could substitute the final routing in order to decide whether an assignment is "good" or not and if necessary allow to choose between different solutions without the cost of routing.

Another interesting research direction is to propose pre-processing methods to group clients in super-clients, in order to reduce the problem size, that will then be assigned and routed.

In this paper only greedy assignment algorithms are considered, it would be interesting to explore the use of other heuristic approaches, for example GRASP, Genetic Algorithms and Simulated Annealing. In order to pursue this line of research, it would be of great use to have available inter-cluster and intra-cluster distances adequate to this particular assignment problem.

Finally it would also be interesting to propose post-optimization algorithms to "improve" a solution, that is, to reassign customers that may have been misplaced.

Bibliography

L. Bodin, B. Golden, A. Assad and M. Ball. (1983). “Routing and scheduling of vehicles and crews: The state of the art”, Computers and Operations Research Vol. 10.

G. Clark and J. Wright. (1964). “Scheduling of vehicles from a central depot to a number of delivery points”, Operations Research 12, 568-581.

R. Russell and W. Igo. (1979). “An assignment routing problem”, Networks Vol. 9, 1-17.

J. Lenstra and A. Rinnooy Kan. (1981). “Complexity of vehicle routing and scheduling problems”, Networks Vol. 11, 221-228.

R. Mole and S. Jameson. (1976). “A Sequential Route-building algorithm Employing a generalised Savings Criterion”, Operational Research Quarterly Vol. 27 (2), 503-511.

J. Bramel and D. Simchi-Levi. (1997). “On the Effectiveness of Set Covering Formulations for the Vehicle Routing Problem with Time Windows”, Operations Research Vol. 45, 295-301

J. Potvin and J. Rousseau. (1993). “A parallel route building algorithm for the VRPTW”, EJOR 66, 331-340, North-Holland.

M. Solomon. (1987). “Algorithms for the vehicle routing and scheduling problems with time window constraints”, Operations Research Vol. 35 (2), 254-264.

S. Salhi and G. Nagy. (1999). “A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling”, Journal of the Operational Research Society, 50, 1034-1042.

G. Ioannou, M. Kritikos and G. Prastacos. (2001). “A greedy look-ahead heuristic for the vehicle routing problem with time windows”, Journal of the Operational Research Society, 52, 523-537.

D. Giosa, L. Tansini and O. Viera. (1999). “Assignment algorithms for the Multi-Depot Vehicle Routing Problem”, Proceedings of the 28th Conference of the Argentinian Operations Research Society (28 Jornadas de Informática e Investigacion Operativa – SIO’99), SADIO, Buenos Aires, Argentina, 41-47.

M. Urquhart, O. Viera, M. Gonzalez and H. Cancela. (1997). “Vehicle routing techniques applied to a milk collection problem”, INFORMS Fall meeting, Dallas, TX, USA.

ESRI: http://www.esri.com/ (Last access 03/2001)

Geographic Information Systems: http://info.er.usgs.gov/research/gis/title.html (Last access 03/2001)

D. Giosa, L. Tansini and O. Viera. (2002). “New Assignment Algorithms for the Multi-Depot Vehicle Routing Problem”, Journal of the Operational Research Society, 53 (9), 977-984.

B. Linoff. (1995). “Data mining techniques: for marketing, sales and customer support”, Editorial John Wiley & Sons. ISBN 0-47-117980-9.

M. Urquhart and O. Viera. (2002). “A Vehicle Routing System Supporting Milk Collection”, Operational Research Society of India, 39 (1), 46-54.

L. Tansini. (2001). “Algoritmos de Asignación para MDVRPTW” Master in Computer Science - PEDECIBA Thesis at the Operations Research Department, Computer Science Institute, School of Engineering of the Universidad de la República (Montevideo, Uruguay) (Tesis para el Título de Magíster en Informática -

PEDECIBA, Departamento de Investigación Operativa, Instituto de Computación, Facultad de Ingeniería, UDELAR).