Journal of the Operational Research Society (2006) 57, 241–249
www.palgrave-journals.com/jors

New measures of proximity for the assignment algorithms in the MDVRPTW

L Tansini[1,2]* and O Viera[1]

[1]Universidad de la República Oriental del Uruguay, Montevideo, Uruguay; and [2]Chalmers University of Technology, Gothemburg, Sweden

This paper proposes new proximity measures for assignment algorithms for the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). Given the intrinsic difficulty of this problem class, two-step approximation methods of the type 'cluster first, route second' seem to be the most promising for practical size problems. The focus is on the clustering phase, the assignment of customers to depots. Our approach is to extend the existing metrics with time windows. New measures that include time windows and distances are added to two assignment heuristics, that previously only used distance to evaluate proximity between customers and depots. A computational study of their performance is presented, which shows that the inclusion of time windows in the measures of proximity gives better results, in terms of routing, than only using the distance.
Journal of the Operational Research Society (2006) **57,** 241–249. doi:10.1057/palgrave.jors.2601979
Published online 27 April 2005

Keywords: multi-depot vehicle routing problem; clustering; assignment; time windows

Introduction

A key element of many distribution systems is the routing and scheduling of vehicles between 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 the vehicle that services a customers may originate from any 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,[1,2] 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.[3] This interpretation suggests a class of approaches that clusters customers and then schedules the vehicles over each cluster.

This paper concentrates on the clustering phase, that is, the assignment of customers to depots. Two new measures are proposed that take into account not only distances but also time windows in order to evaluate proximity.

To compare the assignment heuristics that use the new measures it is necessary to produce a final set of routes. The analysis of results is carried out on the total distance covered by the routes that service all customers and meet the vehicle load, time window and maximum waiting time constraints. Other measures of interest could be: total time to cover the final set of routes, maximum route length or route time, number of routes, etc. The same VRPTW heuristic[4] is used for all depots, and adding up the individual solutions allows the comparison of the routing results for the assignments.

Owing to the operational nature of most of the MDVRPTW the computing time is an important aspect. Often, the assignment of customers to depots and the construction of the routes for each cluster must be done quickly, as routing information changes hourly, and usually at least daily.

As expected, the study of the assignment algorithms using the new proximity measures establishes that they give better results than using only distance, with respect to total distances. This is carried out with a slight or no increase in execution time.

A comprehensive survey of VRP can be found in Assad *et al*[1] and Laporte *et al.*[5] As mentioned earlier the VRP problem is a well-known integer programming problem that

*Correspondence: *L Tansini, Julio Herrera y Rissig 565, 5to piso, InCo, Fac. Ingenieria, UDELAR, Montevideo, Uruguay.*
E-mail: liber73@yahoo.com

falls in the category of NP-hard problems. For further reading see the discussion on complexity of vehicle routing and scheduling problems in Lenstra and Rinnooy Kan.[2] Owing to these results it is interesting to look for heuristics to find approximate solutions, some specific heuristics for VRP, for example, using Tabu Search, can be found in Caseau and Laburthe,[4] Toth and Vigo[6] and Cordeau *et al.*[7] Besides Tabu Search other meta-heuristics have been used to solve the VRP, MDVRP and VRPTW, some of them are: Ant Systems,[8] Constraint Programming,[9] Genetic Algorithms,[10] Simulated Annealing[11] and Multi-Objective Approaches.[12]

The formulation of VRPTW used here and by others can be seen in Assad *et al*[1] and Bramel and Simchi-Levi.[13] VRPTW is also an NP-hard problem and algorithms for finding approximate solutions are also of importance.[14–17]

Formulations and algorithms for solving MDVRP and MDVRPTW can be found in Assad *et al*,[1] Laporte *et al*[5] and Cordeau *et al.*[18] A more recent heuristic method for solving the MDVRPTW is the Unified Tabu Search heuristic presented in Cordeau *et al*,[7] which will be used later in a comparative study.

There are many variants of the VRP problem, two of them are the Periodic VRP and MDVRP,[7,16] where the planning is made for a period, possibly longer than 1 day. Other related topics are determining the vehicle fleet composition[19] and consideration of waiting times.[20] We will not deal with these topics here.

One of the assignment algorithms presented in this paper is based on an algorithm described by Russell and Igo,[21] intended for the assignment customers to days of the week for garbage collection.

A real-life 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 Urquhart *et al.*[22] A similar problem from the New Zealand dairy industry can be found in Foulds and Wilson.[23]

Tools, real-life problems, applications related to this paper and interfaces using GIS (see ESRI: http://www.esri.com/ last access 03/2001 and Geographic Information Systems: http://info.er.usgs.gov/research/gis/title.html last access 03/2001) are in Giosa *et al.*[24] For more information on the test cases and the software tool used to obtain the results shown in this paper see Tansini.[3]

Some of the ideas for the new measures presented in this paper are based on clustering algorithms, such as the Angle measure.[25]

This paper is organized as follows: in the next section a short problem definition is given. The assignment algorithms are succinctly presented in the subsequent section. The forthcoming section contains a description of the proposed new proximity measures, and the assignment algorithms that include them. Next the routing algorithm which is used to calculate costs of the different assignments are described. In the last section computational results are discussed and analysed, and the new heuristics are compared against the Unified Tabu Search heuristic.[7] Finally, some ideas for future research are presented.

Problem definition

The MDVRPTW[1] consists of determining a set of vehicle routes in such a way that:

(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 minimized.

The MDVRPTW can be viewed as being solved in two stages: first, customers must be allocated or 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.[17,24] 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 subproblems (VRPTW) as the number of depots, and to solve each subproblem separately. The algorithms presented in the next section attempt to implement this strategy.

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.

This section contains a brief description and analysis of two heuristics using a graphic tool developed under the

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.

Arcview 3.0 Geographical Information System platform (see ESRI: http://www.esri.com/ last access 03/2001 and Geographic Information Systems: http://info.er.usgs.gov/research/gis/title.html last access 03/2001). These heuristics are the ones modified to use the new measures.

Originally[26] time window constraints were only 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. All the assignment algorithms described below assign customers to depots so that the capacity of the depots is not exceeded.

The methods we are about to describe have been presented previously, for example, in Giosa *et al.*[26] Some of them are in some sense adaptations of more or less well known heuristic solutions for the VRP and/or VRPTW.[1]

In general, all the assignment algorithms reduce to this schematic pseudocode:

Until all customers have been assigned to the depots

- *Select the best assignment of a customer to a depot, taking into account the demand of the customers, the capacity of the depots and the time windows*
- *Assign the customer*

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

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 most urgency are assigned first. The algorithms in this class vary only in the way the urgencies are calculated. Two examples of this type are *Parallel* and *Simplified parallel* algorithms, which are adaptations of the algorithm for VRPTW presented in Potvin and Rousseau.[14] Only the Simplified parallel assignment will be described with more detail because it will be used further on in the study of new measures that include time windows.

Simplified parallel assignment. In this case, two depots are involved in the evaluation of the urgency:

$$\mu_c = d(c, d''(c)) - d(c, d'(c)) \quad (1)$$

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 customer's closest depot.

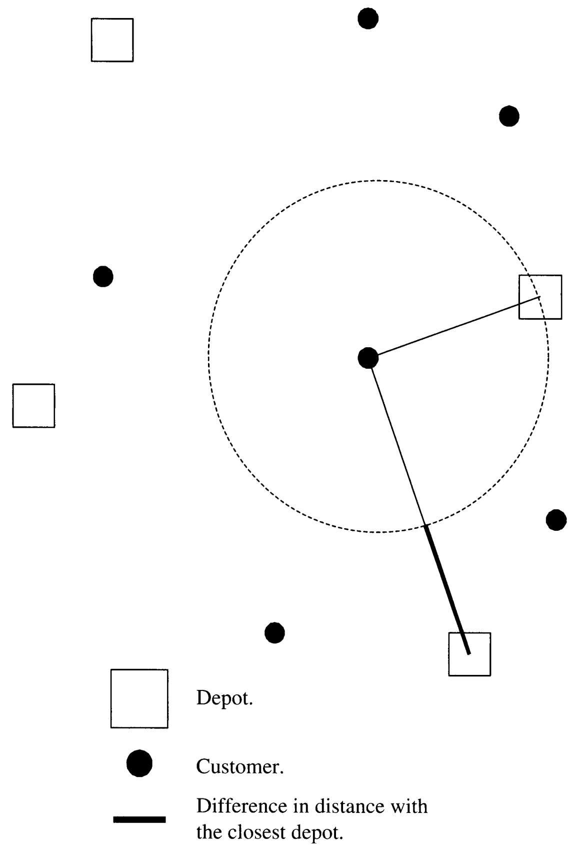

Figure 2 How the urgency is calculated for an unassigned client in the simplified parallel assignment algorithm. The unassigned client in the middle will be assigned to its closest depot, if it maximizes the difference in distance to its two closest depots, shown as a bold line.

As explained earlier, the customer with the greatest value of μ is assigned to its closest depot.

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, for example in Assad *et al.*[1]

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

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, so assigning a customer to a cluster means that this customer is assigned to that cluster's depot. Two examples of heuristics in this class are: Coefficient propagation and Three criteria clustering.[26]

The Coefficient propagation algorithm uses attraction coefficients that are associated to depots and already assigned customers. These coefficients scale the distances with the unassigned customers. Three criteria clustering, that will be described in this section, proved to give better results and is therefore used in the study of new measures that include time windows, so we do not further discuss Coefficient propagation.

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. This algorithm is an adaptation of another algorithm that assigns customers to days of the week.[21]

If there is a customer with a small enough (10%) average distance to its closest cluster (being the closest depot the one with shortest average distance) compared to its second closest cluster, then it can be assigned; the one that maximizes the difference of 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 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.

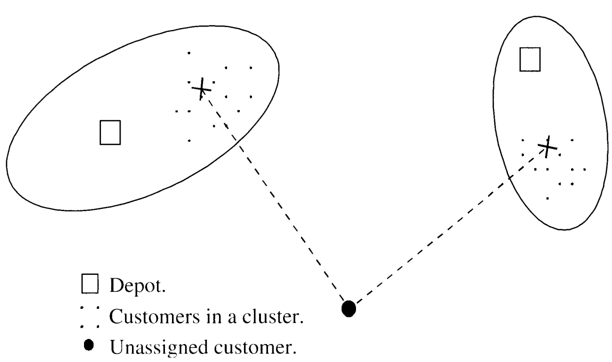

Figure 3 The average distance from an unassigned customer to the cluster is the sum of the distances to the members of the cluster (assigned customers and the depot) divided by the size of the cluster. It can also be seen as the distance to an 'average' point of the cluster marked with x.

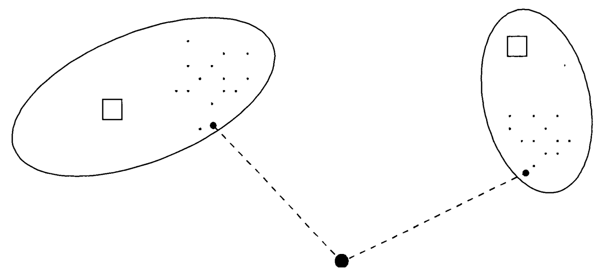

Figure 4 The cluster on the right hand has a lower variance, since the distance to the unassigned customer varies less than to the left cluster. The closest customer in each cluster is shown as a bold point.

Figure 4 shows two clusters with different variance of the average distance to the customers and depot in the clusters. The cluster on the right hand has a lower variance, since the distance to the unassigned customer varies less than in the left cluster.

Figure 4 also shows the closest customer in each cluster to the non-assigned customer.

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

New measures that include time windows

Time window constraints in the MDVRPTW are enemies of the 'cluster first, route second' approach. So far the described assignment algorithms take the customer's and depot's time windows into account only for checking compatibility and feasibility. In this way it is possible to obtain geographically compact clusters of customers for each depot; however, due to the time window constraints, these clusters may contain customers with very different service times. For example, two nearby customers may have service times one in the morning and one in the afternoon. This will require one route for the morning customer and one for the afternoon customer, or one route with long and maybe even unacceptable waiting time (Figure 5).

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 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 6 the morning customer is assigned with the rest of the morning customers giving a lower routing cost of 8.3 km.

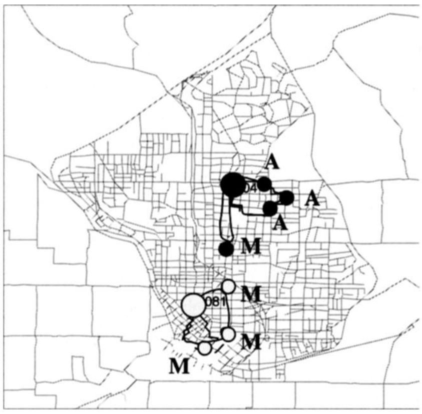

Figure 5 In this assignment time window similarities are not taken into account. Therefore morning customers (M) and afternoon customers (A) are assigned to the same blue dept, while the only morning customer would be better assigned to the yellow depot together with the rest of the morning customers.

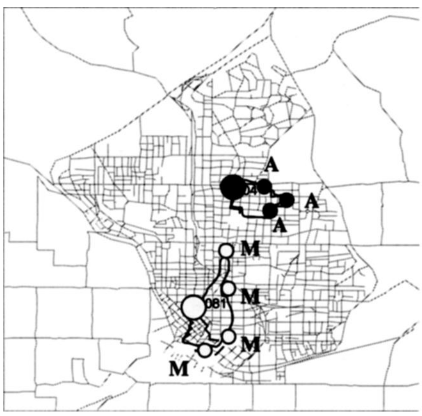

Figure 6 When considering time windows a better assignment takes into account location and time window similarities when assigning customers to depots.

The main contribution of this paper is to introduce new measures of proximity in order to avoid this type of problems generated by the time window constraints. These new measures take into account not only distance but also time windows.

Affinity and closeness

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

$$Affinity(i,d) = \left\{ \frac{\sum_{j \in C(d) \cup \{d\}} e^{-(DTW(i,j) + TV_{ij})}}{|C|} \right\} d \in D \quad i,j \in C \qquad (2)$$

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

$$DTW(i,j) = \begin{cases} e_j - l_i sil_i < e_j \\ e_i - l_j sil_j < e_i \\ 0 \quad \text{otherwise} \end{cases} \qquad (3)$$

where, l and 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.

Closeness is defined to take into account Affinity and distance:

$$Closeness(i,j) = d(i,j)/Affinity(i,j), \quad j \in D,\ i \in C \qquad (4)$$

where $d(i, j)$ is the distance between i and j.

The Closeness indicates the proximity of a customer to a depot and its group of already assigned customers in terms of both distance and time windows.

The worst case complexity for calculating Affinity is O(C), therefore the same for Closeness.

Angle

The measure Angle derives from the comparison of vectors found when exploring clustering algorithms.[25] 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 colours.

For the division of the MDVRPTW customers and depots are considered the data to cluster and the geographical coordinates and the beginning and end of the time windows are the attributes. In this manner clustering is made based on similarities in geographical location and time windows.

The Angle is defined as follows:

$$Angle(i,j) = \mathrm{Cos}^{-1}\left(\frac{x_i x_j + y_i y_j + l_i l_j + e_i e_j}{(x_i^2 + y_i^2 + l_i^2 + e_i^2)^{1/2}(x_j^2 + y_j^2 + l_j^2 + e_j^2)^{1/2}} \right) \qquad (5)$$

where x and y are the coordinates and l and e represent the beginning and end of the time window, i and j are customers or depots.

The worst case complexity for calculating Angle is O(1).

Assignment algorithms using the new measures

An earlier study,[26] 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 these two algorithms were chosen to incorporate the new proximity measures.

Simplified parallel assignment. As mentioned earlier only two depots are involved in the evaluation of the urgency:

$$\mu_c = d(c, d'(c)) - d(c, d''(c)), \quad c \in C \tag{6}$$

where $d'(c)$ is the closest depot to c and $d''(c)$ is the second closest depot to c.

The urgency is redefined using the *Closeness* as follows:

$$\mu_c = Closeness(c, dc'(c)) - Closeness(c, dc''(c)) \quad c \in C \tag{7}$$

where $dc'(c)$ is the closest depot to c and $dc''(c)$ is the second closest depot to c considering the measure *Closeness*.

The worst case complexity for the whole algorithm is $O(3CD + CD^2 + C^3D)$, because of the evaluation of *Closeness*.

Three criteria clustering. For this algorithm the usual distance, whether it is Euclidean distance on a grid or real distance on a map, is changed for the Angle distance. Then the three criteria used by this algorithm to include a customer in a cluster are: average *Angle* distance to the clusters, variance of the average *Angle* distance to the customers in the clusters and *Angle* distance to the closest customer in each cluster.

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

Routing algorithm

In order to compare the heuristics in terms of routing results, it is necessary to solve the MDVRPTW. The final results are obtained by running the same routing heuristic for all assignments produced by the different assignment algorithms. The best assignment algorithm is the one with the best routing results, or lowest cost (see Figure 1). Other alternative costs could also be considered, for example the objective might be to reduce the number of routes or vehicles used or the total waiting or travel time between customers.

Since it is necessary to produce a final set of routes in order to compare the assignment heuristics, the same VRPTW heuristic, Incremental Local Insertion for Time Windows[4] (ILO), is used for all assignment algorithms

The ILO heuristic is an insertion algorithm with simple local optimizations performed after each insertion. The insertion algorithm, starts with a set of empty routes, then the customers are considered one at a time to be inserted in a route for which the increase in length is minimal. The insertion cost of i between x and y is: $d(x, i) + d(y, i) - d(x, y)$. Then local optimizations are performed, 3-opt and 2-edge exchange among different routes. Feasibility checks include vehicle load, time window constraints or maximum waiting time allowed in a route.

The fleet of vehicles is considered homogeneous and unlimited.

Computational results

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 is defined and used in order to compare the assignments obtained with algorithm α and β:

$$Gain_D(\alpha, \beta) = \frac{(D_{\text{Total}}(\beta) - D_{\text{Total}}(\alpha))}{D_{\text{Total}}(\beta)} 100 \tag{9}$$

where $D_{\text{Total}}(\alpha)$ is the total distance of the solution or final routing for all depots, produced after using assignment algorithm α. If $Gain_D$ (α, β) is positive then algorithm α has a better performance than β, otherwise it has a worse performance.

With several test cases the average gain is defined as:

$$AverageGain_D(\alpha, \beta) = \frac{\sum Gain_D(\alpha, \beta)}{number\ of\ test\ cases} \tag{10}$$

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

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

$$Gain_T(\alpha, \beta) = \frac{(T_{\text{Total}}(\beta) - T_{\text{Total}}(\alpha))}{T_{\text{Total}}(\beta)} 100 \tag{11}$$

$$AverageGain_T(\alpha, \beta) = \frac{\sum Gain_T(\alpha, \beta)}{number\ of\ test\ cases} \tag{12}$$

where $T_{\text{Total}}(\alpha)$ is the total time of the solution (final routing for all depots) produced after using assignment algorithm α. If $Gain_D$ (α, β) is positive then algorithm α has a better performance than β, otherwise it has a worse performance.

Results

Test cases had to be generated since, as far as we know, there are not many known tests instances of large enough size to be considered realistic, for the MDVRPTW on a map and using real distances.

To provide test cases for the experiments we chose the city map of Atlanta, Georgia, USA, provided by the Arcview 3.0 Geographical Information System. In all, 60 test cases of different sizes were generated and taken from the digitized city map of a portion of Atlanta, which has 1502 street intersections. This gave an upper bound to the problem size as customers and depots were located on the street intersections. The cases range from eight customers and six depots to 1000 customers and 20 depots. Half of the test cases were generated randomly and the remainder were chosen to test different aspects of the algorithms. The test cases are divided into two groups:

- Randomly generated cases
- Created cases with different caracteristics, such as depots with short-time windows or small capacities, others include depots with equal capacities or very different capacities.

The Created cases were cases thought to make it more difficult for the original assignment algorithms to correctly assign customers if time windows were present. In these cases a larger improvement was expected, using the new measures.

The results shown in Table 1 were obtained over the test cases using real distances calculated on the Atlanta map. As mentioned before the vehicle fleet is considered homogeneous and unlimited.

Table 2 shows execution times for both assignment algorithms. All tests were run on the same computer, a Pentium 2 266 MHz with 64 Mb RAM memory and operating system Windows 98. Routing execution times are discussed later.

The conclusion extracted from Table 1 is that there is a better performance of SPA with Closeness both in total distance and in total time of the routes. A significant average saving of 6% in total distance suggests that for large size problems this reduction is worthwhile. There is also an average reduction in total time of over 7%, which is not surprising since now customers that have similar time windows are assigned to the same depots.

Table 3 shows the comparison of Three Criteria using Angle an Three Criteria using distance.

Table 4 shows execution times for both assignment algorithms.

As in the previous case there are reductions in total distance and total time. Here an average saving of 5% in total distance can be seen, which for large-scale problems may be significant. However, some pathological cases were detected: when one time window is included in another, but is very small and they have very different beginning and end, they are not recognized as similar. This is because the Angle measures the similarity of the beginning and end of the time windows only, for example, 1015–1230 is very different from 0430 to 1730, even though one is nested in the other. A good path for further research would be to improve the Angle measure to avoid this problem.

Again it is not surprising that there is a reduction in total time, in this case also an average of more than 7%.

Routing times are similar for all assignment algorithms, depending only on the size of the problem, Table 5 shows

Table 1 Average gain of SPA using closeness compared to SPA using distance

Type of test cases	*$AverageGain_D$ (SPA using Closeness, SPA using distance)*	*$AverageGain_T$ (SPA using Closeness, SPA using distance)*
Created test cases	10.05	11.65
Random test cases	2.33	3.53

Table 2 Execution times for the assignment phase of SPA using Closeness and SPA using distance

Algorithm	*1000 customers 20 depots*	*450 customers 15 depots*	*100 customers 5 depots*
SPA using Closeness (s)	16	2	<1
SPA using distance (s)	2	<1	<1

Table 3 Average gain of Three Criteria using Angle compared to Three Criteria using distance

Type of test cases	*$AverageGain_D$ (Three Criteria using Angle, Three Criteria using distance)*	*$AverageGain_T$ (Three Criteria using Angle, Three Criteria using distance)*
Created test cases	7.35	9.98
Random test cases	3.05	5.61

Table 4 Execution times for the assignment phase of Three Criteria using Angle and Three Criteria using distance

Algorithm	*1000 customers 20 depots*	*450 customers 15 depots*	*100 customers 5 depots*
Three Criteria using Angle (s)	297	43	<1
Three Criteria using distance (s)	294	41	<1

Table 5 Routing times for different problem sizes

1000 customers 20 depots	*400 customers 15 depots*	*100 customers 5 depots*
684 s	54 s	2 s

Table 6 Average Gain of SPA using Closeness, Three Criteria using Angle and UTSH compared to SPA using Closeness

AverageGain$_D$ (SPA using Closeness, SPA using Closeness)	*AverageGain$_D$ (Three Criteria using Angle, SPA using Closeness)*	*AverageGain$_D$ (UTSH, SPA using Closeness)*
0	5.93	16.80

average routing times of 10 cases of each size, when using real distances calculated on the map. We acknowledge that there are better routing algorithms, but the main concern of this work is the assignment phase, and the routing is merely necessary for the comparisons.

It is also interesting to compare SPA with Closeness and Three Criteria using Angle on test cases such as those in Cordeau *et al*[7] and to the heuristic method presented in that work, Unified Tabu Search Heuristic (UTSH). These test cases can be found at http://www.crt.umontreal.ca/˜cordeau/tabu/ last access 04/2002. Some of the information provided in the test data was not relevant for our experiments and was therefore ignored, such as number of possible visit combinations for a customer and the number of vehicles. Euclidean distances were used and the total length of the routes had to be calculated from the ordered sequence of customers composing them. There are 20 test cases ranging from 48 customers and four depots to 288 customers and six depots. Table 6 shows the comparisons in Average Gain compared to SPA using Closeness of SPA using Closeness, Three Criteria using Angle and UTSH. Observe that AverageGain$_D$ (SPA using Closeness, SPA using Closeness) is 0 because the algorithm is being compared to itself.

As seen in Table 6, the results show that UTSH gives better results, which is not surprising since both SPA and Three Criteria are approximation methods used to solve the MDVRPTW in a 'cluster first, route second' manner. Both SPA and Three Criteria give relatively good results compared to UTSH, which is one of the best heuristics presented so far. Owing to its short execution times SPA using Closeness is the most attractive heuristic method for solving medium- and large-scale problem instances.

Conclusions

This study has shown that incorporating time windows into measures of proximity in the assignment algorithms for the MDVRPTW gives better results than using them only to check compatibility. The two measures with the two different algorithms, SPA with Closeness and Three Criteria using Angle, proved to give average savings in total distance of 6 and 5%.

It was not surprising that for both algorithms there were average reductions in total time of more than 7%, since customers that have similar time windows are assigned to the same depots.

Even though the execution times of the SPA algorithms are slightly increased, it is still the most promising and the best choice for medium- and large-problem instances.

Further research

The Angle measure gave very promising results but needs to be improved in order to avoid the problem of nested time windows. This would be a very interesting topic for future research.

Another important direction of future research is to investigate pre-processing methods to group clients in super-clients, in order to reduce the size of the problem.

In this paper, only greedy assignment algorithms are considered, it would be appealing to explore the use of other heuristic approaches, for example, GRASP, Genetic Algorithms and Simulated Annealing.

Finally, it would also be interesting to propose post-optimization algorithms to 'improve' a solution, so that customers that may have been misplaced are reassigned.

References

1 Assad A, Ball M, Bodin L and Golden B (1983). Routing and scheduling of vehicles and crews: the state of the art. *Comput Opns Res* **10**: 63–211.

2 Lenstra J and Rinnooy Kan A (1981). Complexity of vehicle routing and scheduling problems. *Networks* **11**: 221–228.

3 Tansini L (2001). *Algoritmos de Asignación para MDVRPTW*. Master Thesis–PEDECIBA, 2001, Instituto de Computación, Facultad de Ingeniería, UDELAR.

4 Caseau Y and Laburthe F (1998). A fast heuristic for large routing problems. *Presented at IFORS 98*, Kaunas, Lithuania.

5 Laporte G, Gendreau M, Potvin JY and Semet F (2000). Classical and modern heuristics for the vehicle routing problem. *Int Trans Opl Res* **7**: 285–300.

6 Toth P and Vigo D (1998). The granular tabu search (and its application to the vehicle routing problem). *Working paper, DEIS, University of Bologna.*

7 Cordeau JF, Laporte G and Mercier A (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *J Opl Res Soc* **52**: 928–936.

8 Reimann M, Doerner K and Hartl RF (2003). Analyzing a unified ant system for the VRP and some of its variants. In: Günther *et al* (ed). *EvoWorkshops 2003, Lecture Notes in Computer Science*, Vol 2611, Springer-Verlag, Berlin, Heidelberg, pp 300–310.

9 Rousseau LM, Gendreau M, Pesant G and Focacci F (2004). Solving VRPTWs with constraint programming based column generation. *Ann Opl Res* **130**: 199–216.

10 Berger J, Barkaoui M and Bräysy O (2001). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Working paper, Defense Research Establishment Valcartier*, Canada.

11 Czech ZJ and Czarnas P (2002). Parallel simulated annealing for the vehicle routing problem with time windows. *Presented at the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain.*
12 Sa'adah P and Paechter B (2004). Improving vehicle routing using a customer waiting time colony. In: Goos G, Hartmanis J and van Leeuwen J (eds). *EvoCOP 2004, Lecture Notes in Computer Science*, Vol 3004, Springer-Verlag, Berlin, pp 188–198.
13 Bramel J and Simchi-Levi D (1997). On the effectiveness of set covering formulations for the vehicle routing problem with time windows. *Ops Res* **45**: 295–301.
14 Potvin J and Rousseau J (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur J Opl Res* **66**: 331–340.
15 Solomon M (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Opns Res* **35**: 254–264.
16 Salhi S and Nagy G (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *J Opl Res Soc* **50**: 1034–1042.
17 Ioannou G, Kritikos M and Prastacos G (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. *J Opl Res Soc* **52**: 523–537.
18 Cordeau JF, Gendreau M and Laporte G (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30**: 105–119.
19 Salhi S and Sari M (1997). A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *Eur J Opl Res* **103**: 95–112.
20 Desaulniers G, Lavigne J and Soumis F (1998). Multi-depot vehicle scheduling problems with time windows and waiting costs. *Eur J Opl Res* **111**: 479–494.
21 Russell R and Igo W (1979). An assignment routing problem. *Networks* **9**: 1–17.
22 Urquhart M, Viera O, Gonzalez M and Cancela H (1997). Vehicle routing techniques applied to a milk collection problem. *Presented at INFORMS Fall Meeting, Dallas, TX, USA.*
23 Foulds LR and Wilson JM (1997). A variation of the generalized assignment problem arising in the New Zealand Dairy Industry. *Ann Opns Res* **69**: 105–114.
24 Giosa D, Tansini L and Viera O (1999). Assignment algorithms for the multi-depot vehicle routing problem. *Presented at SADIO, Buenos Aires, Argentina.*
25 Berry M and Lindoff G 1995. *Data Mining Techniques: for Marketing, Sales and Customer Support*. John Wiley & Sons: Chichester.
26 Giosa D, Tansini L and Viera O (2002). New assignment algorithms for the multi-depot vehicle routing problem. *J Opl Res Soc* **53**: 977–984.

Received August 2003;
accepted November 2004 after two revisions