THE OR SOCIETY

New Assignment Algorithms for the Multi-Depot Vehicle Routing Problem
Author(s): I. D. Giosa, I. L. Tansini and I. O. Viera
Source: *The Journal of the Operational Research Society*, Vol. 53, No. 9 (Sep., 2002), pp. 977-984
Published by: Palgrave Macmillan Journals on behalf of the Operational Research Society
Stable URL: http://www.jstor.org/stable/822842
Accessed: 21/08/2013 16:34

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at
http://www.jstor.org/page/info/about/policies/terms.jsp

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Palgrave Macmillan Journals and *Operational Research Society* are collaborating with JSTOR to digitize, preserve and extend access to *The Journal of the Operational Research Society.*

http://www.jstor.org

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

Journal of the Operational Research Society (2002) 53, 977–984 ©2002 Operational Research Society Ltd. All rights reserved. 0160-5682/02 $15.00

www.palgrave-journals.com/jors

New assignment algorithms for the multi-depot vehicle routing problem

ID Giosa[1], IL Tansini[1] and IO Viera[1]*

[1]*Universidad de la República Oriental del Uruguay, Montevideo, Uruguay*

This paper considers the design and analysis of algorithms for 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 offer the most promise for practical size problems. After describing six heuristics for the cluster part (assignment of customers to depots) an initial computational study of their performance is conducted. Finding, as expected, that the heuristics with the best results (in terms of the routing results) are those with the largest computational efforts.

Journal of the Operational Research Society (2002) **53,** 977–984. doi:10.1057/palgrave.jors.2601426

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

Introduction

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,[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. 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 and design and analyse six heuristics using a graphic tool developed under the Arcview 3.0 Geographical Information System platform.

To compare the assignment heuristics it is necessary to produce a final set of routes, in order to do this the same VRP heuristic[3] is used for each depot, 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, 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.

As expected, the comparisons of the assignment algorithms show that none of them gives good results in short execution time.

A comprehensive survey on VRP can bee found in Bodin *et al*,[1] for specific heuristics for VRP see Clark and Wright[3] and Mole and Jameson.[4] For a discussion on complexity of vehicle routing and scheduling problems see Lenstra and Rinnooy Kan.[2] For further reading on formulations of VRPTW see Bodin *et al*[1] Bramel and Simchi-Levi,[5] and on algorithms for VRPTW see Potvin and Rousseau,[6] Solomon,[7] Salhi and Nagy[8] and Ioannou *et al.*[9] Formulations and algorithms for solving MDVRP and MDVRPTW can be found in Bodin *et al*[1] and Salhi and Nagy.[8] 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.[10] A problem from the dairy industry motivated this paper: the daily transportation of milk from farms to processing plants, for background on the

*Correspondence: *IO Viera, Dpto. Investigación Operativa, Instituto de Computación, Facultad de Ingeniería, Universidad de la República Oriental del Uruguay, Montevideo, Uruguay.*
E-mail: viera@fing.edu.uy

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

project that motivated this paper see Giosa *et al*[11] and Urquhart *et al.*[12] Tools, real life problems, applications related to this paper and interfaces using geographic information system (GIS)[13,14] are in Urquhart *et al.*[12] For more information on the test cases and the software tool used to obtain the results shown in this paper see Giosa *et al.*[11]

This paper is organised as follows: in the following section a short problem definition is given. The assignment algorithms are succinctly presented in the third section. The fourth section contains a brief description of the routing algorithm used to calculate costs of the different assignments. In the penultimate section, computational results are discussed and analysed, and some ideas for future research are presented in the final section.

Problem definition

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

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.[9,11] 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 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. The time windows constraints are 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.

Finally, and due to the lack of documentation about solutions for the assignment problem related to the MDVRPTW, some of the methods we are about to describe are in some sense adaptations of more or less well-known heuristic solutions for the VRP and/or VRPTW.[7]

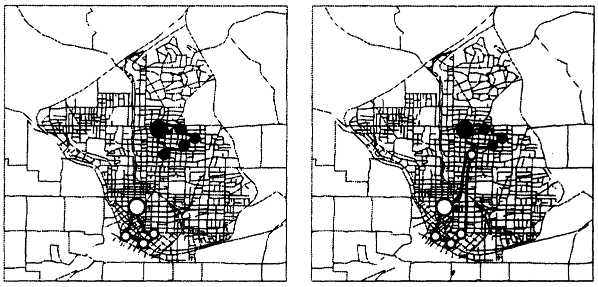

Figure 1 Comparing two routing costs: 6.4 km for one assignment and 9.1 km for a 'worse' assignment.

A general pseudo-code 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 next sections use different measures for the assignment of a customer to a depot: assignment through urgencies, cyclic assignment 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 *parallel* and *simplified* algorithms are adaptations of the algorithm for VRPTW presented by Potvin and Rousseau.[6]

The algorithms in this class vary only in the way the urgencies are calculated.

Parallel assignment. The name *parallel* is due to the fact that the urgency for each customer is calculated considering all depots at the same time. The urgency is calculated as follows:

$$\mu_c = \left(\sum_{\text{dep} \in D} d(c, \text{dep}) \right) - d(c, \text{dep}')$$

where $d(c, \text{dep})$ is the real distance between customer c and depot dep, D is the set of all depots and $d(c, \text{dep}')$ is the real distance between customer c and its closest depot dep$'$. Figure 2 shows how the urgency is calculated in this algorithm. The customer with the greatest value of μ is assigned to its closest depot.

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

This heuristic compares the cost of assigning a customer to its closest depot with the cost of assigning the customer to any other depot. The most urgent customer is the one for which μ is maximum.

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.

Simplified assignment. This algorithm belongs to the same class as the first one, the difference lies in the way in which the urgency is calculated. In this case, only two depots are involved in the evaluation of the urgency:

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

where $d(c, \text{dep}'')$ is the distance between customer c and its second closest depot and $d(c, \text{dep}')$ is the distance between customer c and its closest depot. Figure 3 shows how the urgency is calculated in this algorithm.

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

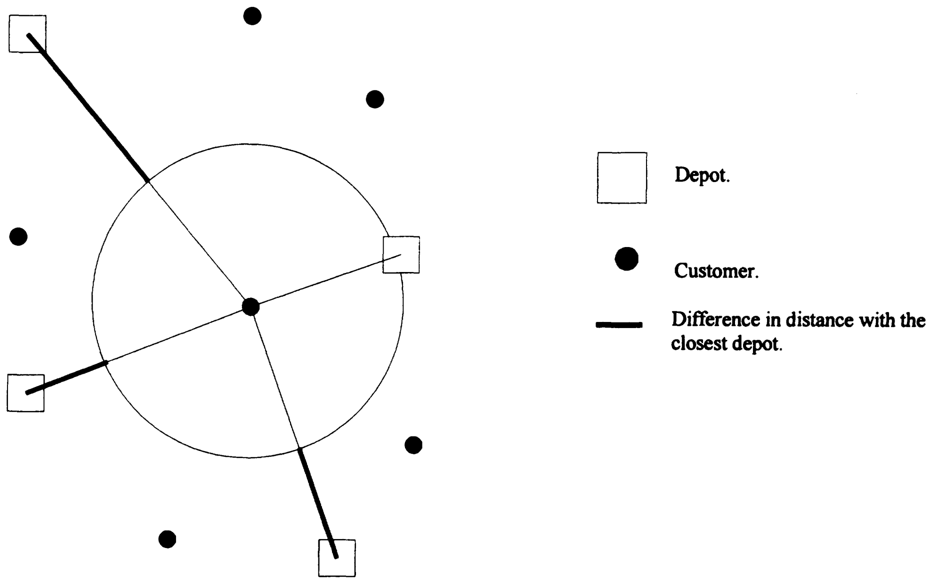

Figure 2 How the urgency is calculated for one client in the *parallel* assignment algorithm.

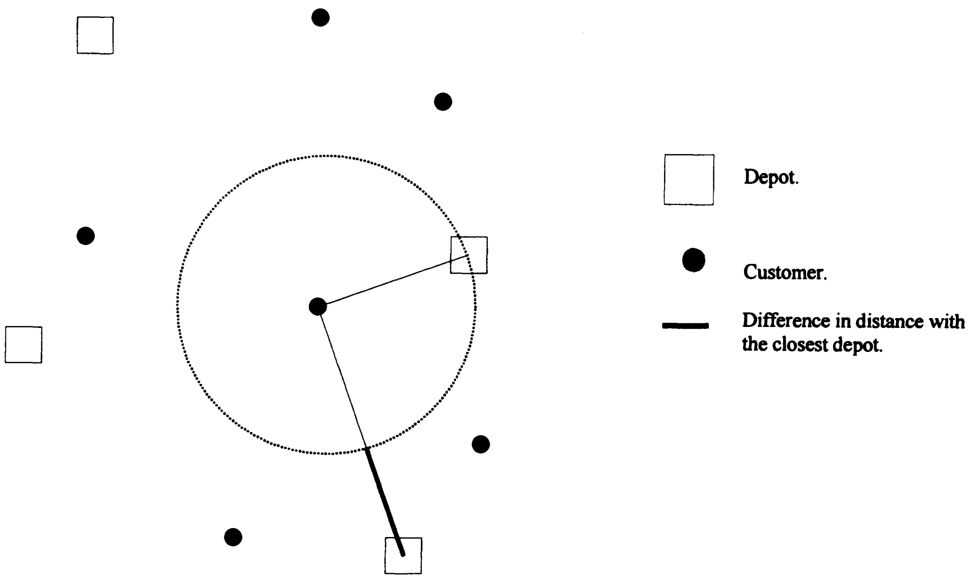

Figure 3 How the urgency is calculated for one client in the *simplified* assignment algorithm.

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

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.[1]

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

Sweep assignment. This heuristic was developed specifically for the problem of daily transportation of milk from farms to processing plants.[11,12] In this heuristic, the customers are attracted (swept) in the direction of the depot with the highest unsatisfied demand. First, it is necessary to determine a depot dep* with the highest unsatisfied demand. The urgency is measured as the difference between assigning a customer to its closest depot and dep*. In this case the evaluation of the urgency is:

$$\mu_c = d(c, \text{dep}^*) - d(c, \text{dep}')$$

A big value for the urgency means that it is more convenient to assign the customer to its closest depot dep′ than to assign it to dep*.

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

The evaluation of the urgency is $O(D^2 + DC + C)$ where $D^2 + DC$ corresponds to the evaluation of dep*. Figure 4 shows how to find dep* in this algorithm.

Cyclic assignment

The procedure consists of assigning in a cyclic way, one customer at a time. The heuristic assigns the closest customer to the last assigned one, to the same depot as this last one.

First, the algorithm assigns to each depot the closest customer. Then it assigns to each depot, the closest customer to the last assigned customer to that depot. In general, the assignment is very poor for last assigned customers. Figure 5 shows two depots (a black and a shaded one), each with three customers assigned. The numbers indicate the order in which the customers were assigned, the next customers that will be assigned to the depots are the ones closest to the last assigned customers of each depot (customer number 3). The complexity of the whole algorithm is (in the worst case) $O(CD + C^2)$.

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. When a customer is assigned to a cluster it means that this customer is assigned to that cluster's depot.

Coefficient propagation. The way in which customers are incorporated in a cluster is defined by associating attraction coefficients to depots and already assigned customers. These coefficients scale the distances with the unassigned customers. The next customer to be assigned in each step, is the one that minimises the scaled distance to a depot or an assigned customer.

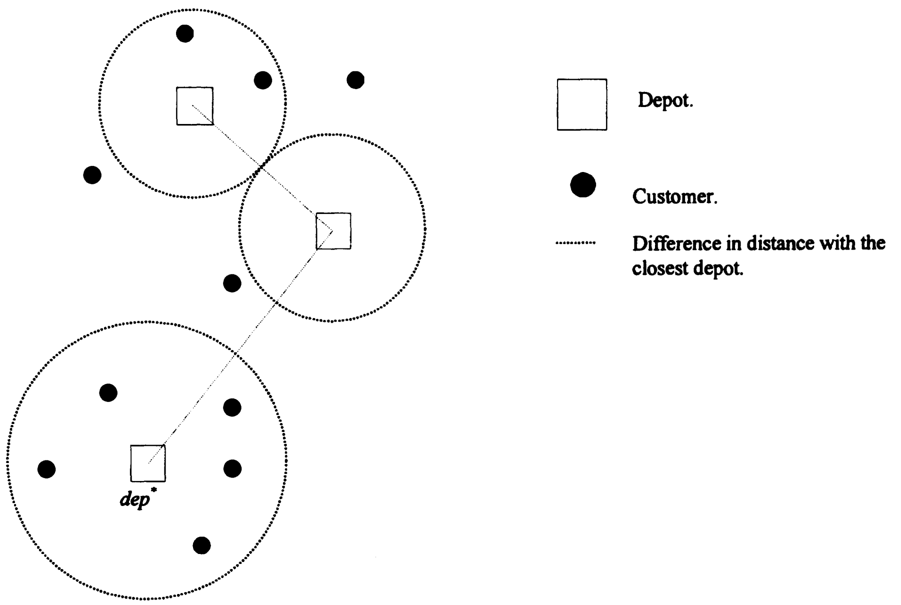

Figure 4 How the depot dep* is determined in the *sweep* assignment algorithm.

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

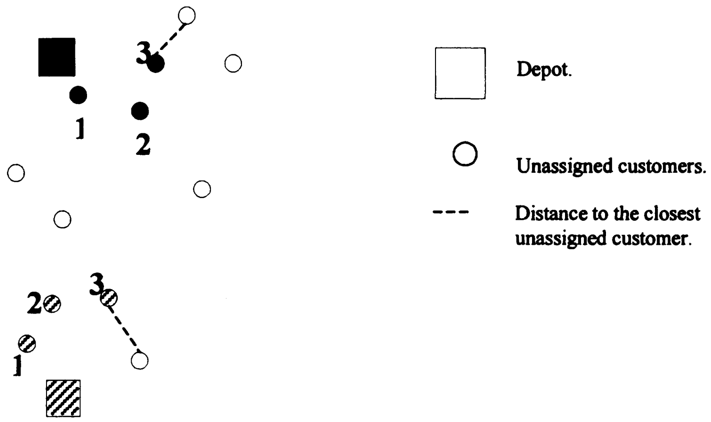

Figure 5 Cyclic assignment, with two depots and three assigned customers to each depot. The customers that will be assigned are the ones closest to the last assigned customers of each depot.

$$d_{\text{scaled}}(c, c') = d(c, c')^* \text{coef}_{c'}$$

If a customer or depot has an attraction coefficient less than one it shortens the distances to other customers (attract them). On the other hand, if the coefficient is greater than one, the distances are larger (reject them). When the coefficient is one the distances remain unchanged.

Since unassigned customers do not have an attraction coefficient, when a customer is assigned it obtains its own attraction coefficient through a function that can be changed. In this case, to calculate the attraction coefficient of c that was assigned because of its closeness to c', a degradation coefficient $\text{deg}_{c'}$ is used. The following function was employed:

$$\text{coef}_c = \min(1, \text{coef}_{c'} + (\text{coef}_{c'} \times \text{deg}_{c'}))$$

The degradation coefficient was arbitrarily set to 0.5 for all depots and customers. Newly assigned customers have higher coefficients, and thus attract less.

This algorithm is highly interactive, because of the selection of the initial coefficients, which must be made at least for the depots. We have not studied an 'optimal' selection of coefficients or done a sensitivity analysis. Figure 6 shows the scaling of distances made by the attraction coefficient. The complexity of the whole algorithm is (in the worst case) $O(C^3 + C^2D)$.

Three criteria clustering. This algorithm is an adaptation of an algorithm that assigned customers to days of the week for garbage collection. 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[10]).

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),

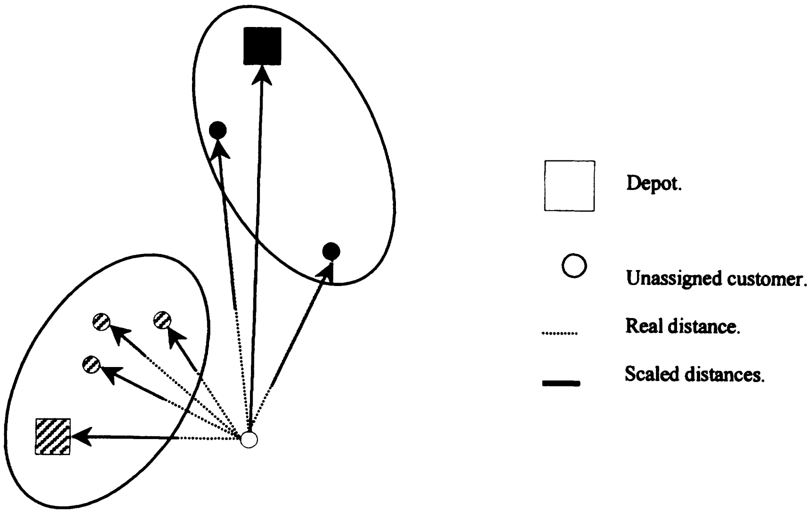

Figure 6 Distances in the coefficient propagation assignment algorithm.

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

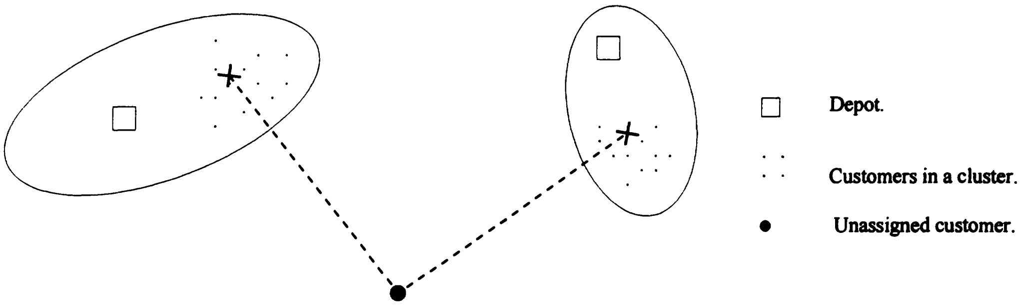

Figure 7 Determining the average distance to the clusters.

then it can be assigned; the one that maximises 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 clusters (40%) then it can be assigned, again the one that maximises 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, and now the customer that minimises this distance is assigned.

Figure 7 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 7 as a cross.

Figure 8 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 customers are closer to each other than in the left cluster.

Figure 9 shows the closest customer in each cluster to a non-assigned customer.

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

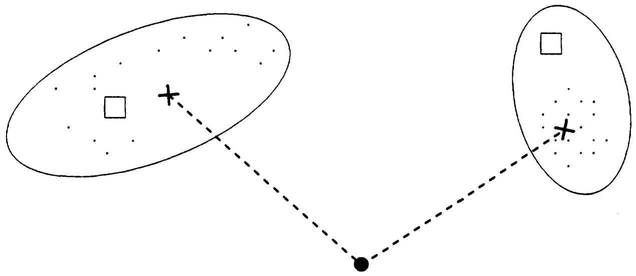

Figure 8 Clusters with different variance.

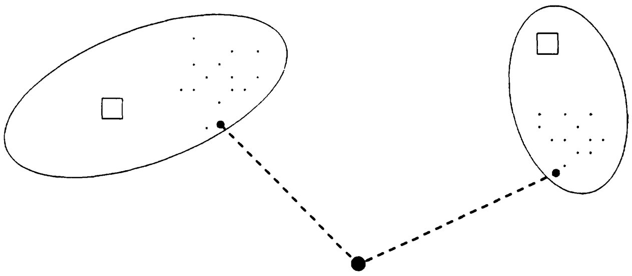

Figure 9 Closest customer in each cluster.

Routing algorithm

To compare the heuristics in terms of routing results, it is necessary to solve the MDVRP, 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 results, or lowest cost (see Figure 1).

Since it is necessary to produce a final set of routes in order to compare the assignment heuristics, the same VRP heuristic (a version of the Clark and Wright heuristic[3]) was 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 or time constraints.

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:

$$\mathrm{SAV}_c(i,j) = 2 \times d(\mathrm{dep},\ c) - (d(c,j) + d(i,\ c) - d(i,j))$$

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

A pseudo-code for this the algorithms is as follows:

While there are incomplete routes there is a customer c such that $\exists \mathrm{SAV}_c(i,j) \geqslant 0$
  determine the next customer c^* on a route of its own which minimises $\mathrm{SAV}_{c^*}(i,j)$, taking into account the demand of the customers, the capacity of the vehicles and the time windows for customers and depot, include c^* in a route between i and j.

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

Table 1 Primary computational results

Algorithm	*Average gain*	*Best results*	*1000 customers 30 depots (s)*	*400 customers 30 depots (s)*	*100 customers 6 depots (s)*
Three criteria	6.48	26	303	57	<1
Coefficient propagation	4.74	16	18	3	<1
Sweep	0.30	17	3	1	<1
Parallel	0.00	16	2	<1	<1
Simplified	−0.20	17	2	<1	<1
Cyclic	−29.80	6	1	<1	<1

Computational results

Primary results

As mentioned earlier, in order to compare the assignment heuristics, it is necessary to solve the MDVRP, this means to run the same routing heuristic for each assignment algorithm. 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). In order to compare the assignment algorithms we defined and used the following measure:

$$\text{gain}_i = \frac{(k_{\text{base}} - k_i)}{k_{\text{base}}} \times 100$$

Initially, a base algorithm is arbitrarily chosen, against which the rest of the algorithms are compared. After the routing has been done the total cost of the base algorithm k_{base} and the costs of the other algorithms k_i are compared and normalised. If the result is positive then the algorithm i has a better performance than the base algorithm, otherwise it has a worse performance.

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

$$\text{average gain}_i = \frac{\sum \text{gain}_i}{\text{number of tests}}$$

The average gain turns out to be a global measure of the behaviour of each algorithm compared with the base algorithm, thus providing a way to compare them.

For our results the *parallel* assignment algorithm was used as the base algorithm.

The results shown in Table 1 were obtained over the test cases using real distances calculated on the map. The second column of the table presents a resume of the average gain, the results are ordered by decreasing average gain compared with the *parallel* algorithm as base algorithm. The 'Best results' column shows how many times the algorithm obtained the best result.

Table 1 also shows the execution times in seconds for the algorithms in a Pentium 2 of 266 MHz with 64 Mb RAM memory and operating system Windows 98.

As expected, the comparisons of the assignment algorithms show that none of them gives good results in short execution time. It is worth investigating the *cyclic* algorithm in order to get better results, and in the *three criteria* and *parallel* algorithms in speeding them up.

Routing times are similar for all assignment algorithms, depending only on the size of the problem. Table 2 shows routing times when using real distances calculated on the map.

Results for known instances of the problem

It is interesting to consider the behaviour of our assignment algorithms on known instances of the problem. In this section the data and solutions for the MDVRP are taken from technical report CRT-95-76, these test cases are the ones corresponding to the MDVRP-old. In these instances Euclidean distances are used, as opposed to real distances measured on streets of a city as in the previous test cases.

Table 3 presents the results obtained over 23 test cases of different sizes, ranging from 50 customers and four depots to 360 customers and nine depots. The table is ordered by decreasing average gain compared with the base algorithm (*parallel* assignment again).

For these test cases it is not as relevant to show all results (which would be six for each test case, all together 138 results) as it is to show a resume of them by giving the average gain. As seen in the results table, the relationship between average gain for the different assignment algorithms is not preserved. Instead in these cases the best results are obtained by the algorithms that assign through urgencies and the algorithms that assign by clusters obtain results almost as good. This is probably due to the fact that customers and depots are located close to each other and that they are very uniformly distributed, also that the test cases are relatively small. We conclude that in these cases it is less important which algorithm we choose, although the best suited are the algorithms that assign through urgencies. Finally the cyclic assignment algorithm is still the one with the worst result.

Table 2 Routing times for different problem sizes

1000 customers 30 depots (s)	*400 customers 30 depots (s)*	*100 customers 6 depots (s)*
950	60	2

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions

Table 3 Results of average gain and time for known instances of the problem

Algorithm	*Average gain*	*360 customers 9 depots (s)*	*50 customers 4 depots (s)*
Parallel	0.00	<1	<1
Simplified	0.00	<1	<1
Sweep	0.00	1	<1
Three criteria	−2.29	50	<1
Coefficient propagation	−7.71	2	<1
Cyclic	−58.10	<1	<1

Further research

Besides the conclusions in the latter section, there are other interesting research directions. The time windows constraints in the MDVRPTW are enemies of the 'cluster first, route second' approach. The described algorithms take the customer's (and depot's) time windows into account only for checking compatibility. In this way it is possible to get (only) geographically compact clusters of customers for the depots but due to the time windows constraints, these clusters may contain customers with very different service times.

If the aim is to obtain routes with short waiting time between customers, the customers should have similar time windows. In this paper, customers were clustered (assigned to depots, forming clusters) to obtain compact geographical clusters, not with similar time windows. From the time windows point of view customers in one geographic cluster may have a similar time window as customers in another cluster, implying longer waiting times in the routes or the need for extra routes to service them in the original cluster, see Figure 10.

We believe that further research is necessary to avoid this type of problem generated by the time window constraints.

References

1 Bodin L, Golden B, Assad A and Ball M (1983). Routing and scheduling of vehicles and crews: the state of the art. *Comput Opns Res* **10**: 104–106.

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

3 Clark G and Wright J (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Opns Res* **12**: 568–581.

4 Mole R and Jameson S. A sequential route-building algorithm employing a generalised savings criterion. *Opl Res Q* **27**: 503–511.

5 Bramel J and Simchi-Levi D (1997). On the effectiveness of set covering formulations for the vehicle routing problem with time windows. *Opns Res* **45**: 295–301.

6 Potvin J and Rousseau J (1993). A parallel route building algorithm for the VRPTW. *Eur J Opl Res* **66**: 331–340.

7 Solomon M (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Opns Res* **35**: 254–264.

8 Salhi S and Nagy G (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with back-hauling. *J Opl Res Soc* **50**: 1034–1042.

9 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.

10 Russell R and Igo W (1979). An assignment routing problem. *Networks* **9**: 1–17.

11 Giosa D, Tansini L and Viera O (1999). Assignment algorithms for the multi-depot vehicle routing problem. *Proceedings of the 28th Conference of the Argentinian Operations Research Society* (28 Jornadas Argentinas de Informática e Investigación Operativa—SIO'99). SADIO: Buenos Aires, Argentina, pp 41–47.

12 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.

13 ESRI: *http://www.esri.com/* (last access 03/2001).

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

Received March 2001;
accepted April 2001 after two revisions

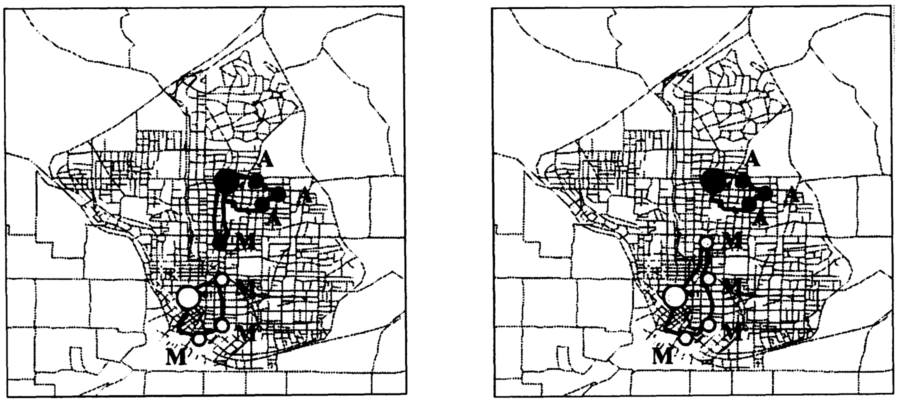

Figure 10 When considering time windows a better assignment takes into account location and time window similarities when assigning customers to depots. In the first map two routes were necessary to visit the morning (**M**) and afternoon customers (**A**) for one of the depots, giving a total cost of 9.5 km. In the second map the morning customer was assigned with the rest of the morning customers giving a lower routing cost of 8.3 km.

This content downloaded from 164.73.224.2 on Wed, 21 Aug 2013 16:34:56 PM
All use subject to JSTOR Terms and Conditions