Comparing assignment algorithms for the Multi-Depot VRP

Libertad Tansini, Maria Urquhart, Omar Viera

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

Julio Herrera y Reissig 565, 5to piso, 11300 Montevideo, URUGUAY

[libertad, urquhart, viera] @fing.edu.uy

Abstract

This paper considers the Multi-Depot Vehicle Routing Problem (MDVRP). Given the intrinsic difficulty of this problem class, approximation methods of the type "cluster first, route second" (two step approaches) seem to be promising for practical size problems. The first step, clustering is usually solved by assignments algorithms. The total cost of the solution for a MDVRP problem depends strongly on the assignment algorithm used in the first step, and these algorithms depend on the geographic topology of the instance of the problem to solve. We compare the results obtained by six heuristic algorithms for the assignment of customers to depots, with assignments obtained from solving the Transport Problem (TP). To compare the assignment algorithms we run the same routing heuristic, namely a modified version of the Clark and Wright heuristic [5], and compare the routing results for each one of them using STAAR [9] developed under the Arcview 3.0 Geographical Information System platform. In earlier work [10] we confirmed that the heuristics with best results were those with the largest computational efforts. We now find that the solutions obtained solving the TP give good results for our test cases and are worth using in real-life problems.

Key words: Multi-Depot Vehicle Routing Problem, Transport Problem, clustering, assignment.

1. 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 [1].

Whereas the VRP has been studied widely, the MDVRP has attracted less attention. In the MDVRP, customers must be serviced by 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 MDVRP is NP-hard [1], [4], therefore, the development of heuristic algorithms for this problem class is of primary interest.

The MDVRP 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. The MDVRP can be 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, it is more efficient to deal with the two steps simultaneously. When faced with larger problems, 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.

This work focuses on the assignment ("cluster" part) of customers to depots and compares the results obtained by six heuristics with assignments obtained from solving a Transport Problem [13] for the same cases. Given the supplies at several depots and the demands of several customers, to solve the Transport Problem (TP) implies to decide how much shipment there should be between each depot and

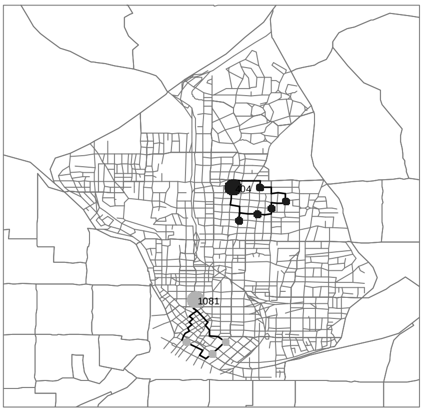

Fig.1A. Cost: 6.4 Kms.

Fig.1B. Cost: 9.1 Kms.

customer. Each customer is serviced at least by one depot, this gives an assignment of customers to depots, that we then use to solve the MDVRP.

To compare the assignment algorithms we run the same routing heuristic, a modified version of the Clark and Wright heuristic [5], and compare the routing results for each one of them. The comparisons of the first six assignment heuristics with the solutions obtained with TP, show that TP is an attractive algorithm since it gives good results in short execution time.

This paper is organised as follows: in Sections 2, 3 and 4 the assignment algorithms are presented; in Section 6 the Transport Problem is discussed; computational results are shown in Section 7 and in Section 8 conclusions and future research are exposed.

2. Assignment algorithms

It is worth to note 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 with a better assignment, as shown in Figure 1.

All the assignment algorithms described in the following sections assign customers to depots so that the capacity of the depots is not exceeded. Due to the lack of documentation about solutions for the assignment problem, the methods we are about to describe are all, in some sense, adaptations of more or less well known heuristic solutions for the VRP and. These methods are: a) assignment through urgencies, b) cyclic assignment, c) assignment by clusters [10] and d) assignment using the Transport Problem [13]. The algorithms use different measures for the assignment of a customer to a depot. A general description of all the assignment algorithms is the following:

Until all customers have been assigned to the depots
determine the next customer to be assigned to a depot
taking into account restrictions like
the demand of the customers and the capacity of the depot.

3. Assignment through urgencies.

This class of algorithms assigns the customers with highest urgency first. The urgency is a way to define a precedence relationship between customers. This precedence relationship determines the order in which customers are assigned to depots. The heuristics that belong to this class are: the *Parallel* assignment algorithm [10], the *Simplified* assignment algorithm [9] and the *Sweep* assignment algorithm [11]. They vary only in the way the urgencies are calculated. Expressed as pseudocode, a general procedure for the assignment algorithms through urgencies is presented below (Figure 2). Each costumer belongs to only one of the following sets: 1) *A* if the costumer has been assigned to a depot, 2) *NA* if the costumer has not yet been assigned to any depot. Each depot belongs to only one of the following sets: 1) *DS* if the demand of the depot has been satisfied, 2) *DNS* if the demand of the depot has not been satisfied. An assignment of a costumer to a depot is feasible if: a) the depot belongs to *DNS*, b) the costumer belongs to *NA*. The urgency of customer c_j is calculated by the function μ_j, different for each type of assignment.

```
procedure GeneralUrgency( )
    while < NA ≠ ∅ >
        for < all  cj ∈ NA >
            <calculate μj for cj>;
        end for
        while < NA ≠ ∅ > and < ¬ ∃ d ∈ DNS  with satisfied demand >
            < determine c* ∈ NA / maximizes μj >;
            < assign c* to its closest, feasible depot >;
            < NA := NA \ {c*}  >;
            < A := A ∪ {c*} >;
        end while
        if < NA ≠ ∅ > and < ∃ d ∈ DNS  /  has now satisfied its demand >
            < DNC := DNC \ {d} >;
            < DC := DC ∪ {d} >;
        end if
    end while
end procedure
```

Fig.2 General procedure for the assignment algorithms through urgencies.

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. 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 customer with the greatest value of μ is assigned to its closest depot. The urgency μ to each customer is calculated as in (1), it is graphically represented in Figure 3.

$$\mu_c = \sum_{c,D} d((c,D) - d^*(c,D^*)) \qquad (1)$$

d(c, D) is the real distance between customer c and depot D;

$d^*(c, D^*)$ is the real distance between customer c and its closest depot D^*.

The complexity for the whole algorithm is (in the worst case) O($3cd + cd^2 + c^2d$), where c is the amount of customers, and d is the amount of depots.

Simplified assignment

This heuristic is a variant of the most common assignment algorithms found in the literature [1]. As in the others urgency algorithms, the most urgent customer is the one for which μ is maximum. In this case, only two depots are involved in the evaluation of the urgency; the comparison is between the cost of assigning a customer to its closest depot with the cost of assigning it to its second closest depot. The urgency μ to each customer is calculated as in (2), it is graphically represented in Figure 3.

$$\mu_c = d(c, D'') - d(c, D^*) \qquad (2)$$

$d(c, D'')$ is the distance between customer c and its second closest depot;

$d(c, D^*)$ is the distance between customer c and its closest depot.

The complexity for the whole algorithm is (in the worst case) O($3cd + cd^2 + c^2d$), where c is the amount of customers and d is the amount of depots.

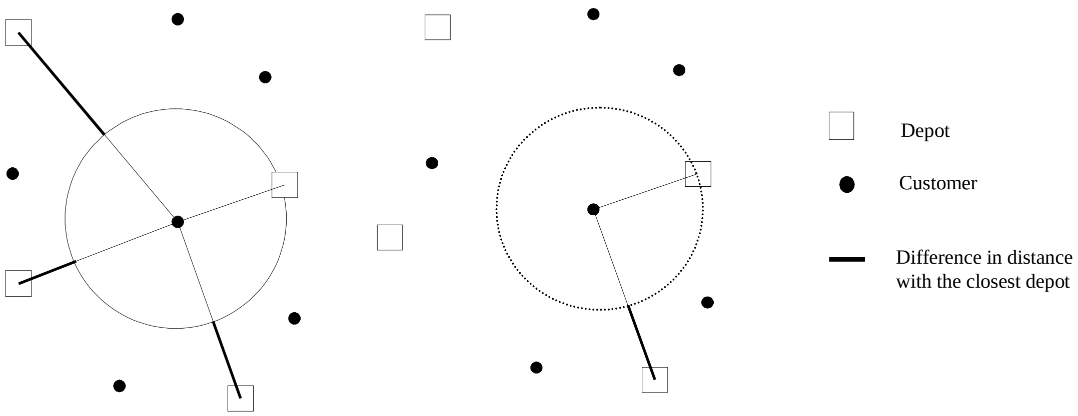

Fig.3 Urgency for one client in the Parallel and in the Simplified assignment

Sweep Assignment

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 D^* with the highest unsatisfied demand. The urgency is measured as the difference between assigning a customer to its closest depot (D_c) and D^*. In this case the urgency is calculated as in (3).

$$\boldsymbol{\mu_c = d(c, D^*) - d(c, D_c)} \qquad (3)$$

A big value for the urgency means that it is more convenient to assign the customer to its closest depot than to assign it to D^*, Figure 4 represent how the algorithm determines D^*.

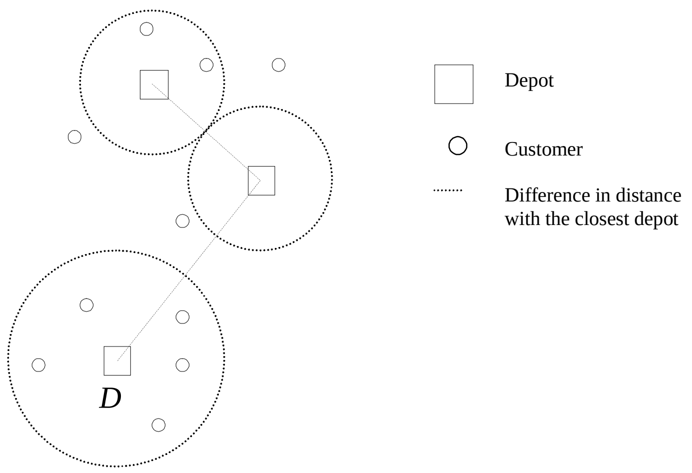

Fig.4. D^* in Sweep urgency-assignment

The complexity of the whole algorithm is O($3dc + c^2 d + d\left(d^2 + dc + c\right)$) (in the worst case). The evaluation of the urgency is O($d^2 + dc + c$) where $d^2 + dc$ corresponds to the evaluation of D^*, c is the amount of customers and d is the amount of depots.

4. Cyclic assignment.

The procedure for this class of algorithms consists in assigning in a cyclic way, one customer at the time to each depot. First, the algorithm assigns to each depot the closest customer. Then assigns to each depot, the closest customer to the last assigned customer to the depot. In general, the assignment is very inconvenient for last assigned customers.

The complexity of the whole algorithm is (in the worst case) O($dc + c^2$), c is the amount of customers, and d is the amount of depots.

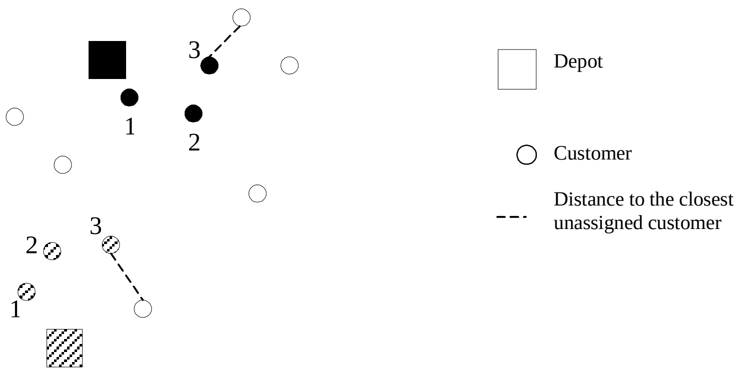

Fig.5. Cyclic assignment.

5. Assignment by clusters.

This class of algorithms build compact clusters of customers for each depot. We define a cluster as the set built by a depot and the customers assigned to it. When a customer is assigned to a cluster it means that this customer is assigned to that clusters depot. In this paper we consider the Coefficient Propagation [9] and Three Criteria Clusterization algorithms [2].

Coefficient propagation

The way in which customers are incorporated to a cluster is defined by associating attraction coefficients to already assigned customers and depots. These coefficients scale the distances. If a customer or depot has an attraction coefficient less than one, it shortens the distances to other customers (attracts them). On the other hand, if the coefficient is greater than one, the distances are larger (rejects them). In the case that the coefficient is one, the distances remain unchanged.

This algorithm is highly interactive, because of the selection of the initial coefficients. Since the selection

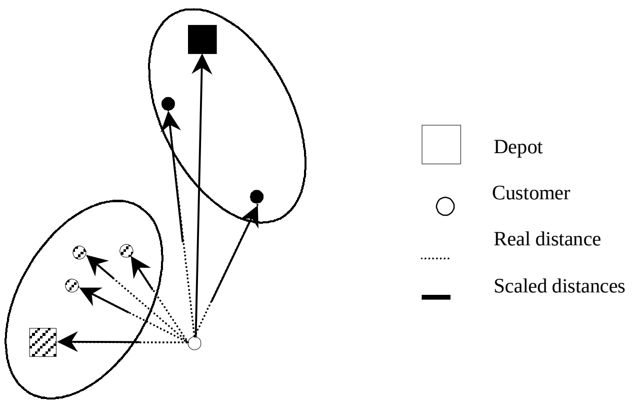

Fig.6. Distances in the Coefficient propagation assignment.

of coefficients depends strongly on the actual problem to solve, we leave the "optimal" selection of coefficients to future research.

The complexity of the whole algorithm is (in the worst case) $O(c^3 + c^2d)$, where c is the amount of customers and d is the amount of depots.

Three criteria clusterization

This algorithm is an adaptation of the procedure proposed in [2] to assign costumers to days of the week. The criteria used in this approach 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 complexity of the whole algorithm is (in the worst case) O($3dc^2 + 3c^2d^2 + cd^2$),where c is the amount of customers and d is the amount of depots.

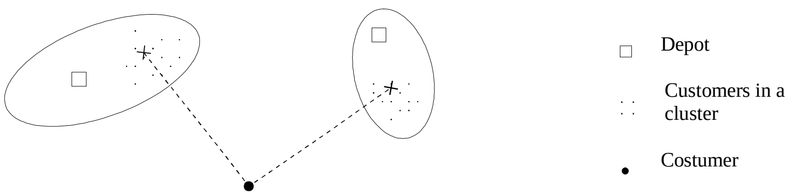

Fig. 7 Average distance to the clusters.

6. Transport Problem (TP)

The transport problem implies to decide how much shipment there should be between depots and customers, given the supplies at the depots and the demands of several customers so as to minimize total transport cost [13]. Each customer is serviced at least by one depot, this gives an assignment of customers to depots.

The well-known algebraic model of TP is:

$$Min \quad \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot c_{ij}$$

$$/ \qquad \sum_{i=1}^{n} x_{ij} \quad \geq \quad b_j \quad \forall j$$

$$\sum_{j=1}^{m} x_{ij} \quad \leq \quad a_i \quad \forall i$$

Where m is the number of depots and n the number of customers. a_i are the supplies at depot i, b_j is the demand of customer j and c_{ij} is the cost per unit shipment between depot i and customer j. The decision variable x_{ij} is the amount of supplies to ship from depot i to customer j.

7. Computational results

The purpose of this paper is to compare different assignment algorithms in terms of a) execution time and b) routing results for the original MDVRP problem.

To compare the heuristics in terms of routing results, is necessary to solve the MDVRP, i. e., to run the same routing heuristic for each assignment algorithm. Then, the best assignment algorithm is the one with the best results, or lowest cost (see Figures 1A and 1B). In order to compare the assignment algorithms we defined and used the following measures:

$$gain_i = \frac{(k_{base} - k_i)}{k_{base}} * 100$$

Initially we choose a base algorithm, 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 normalized. 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 it is possible to define the average gain for each algorithm:

$$average\ gain_i = \frac{\sum gain_i}{number\ of\ tests}$$

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.

To provide test cases for the experimentation we choose the city of Atlanta, since Arc View provides part of its map as a demonstration. Forty (40) 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. 25 % of the test cases were generated randomly, the remaining 75 % were chosen by a practitioner.

The results shown in Table 1 were obtained over the test cases using real distances calculated on the map. The second column of Table 1 presents a resume of the average gain; the results are ordered by decreasing average gain compared with the Parallel algorithm as base algorithm.

Algorithm	**Average gain (%)**	**400 costumers 30 depots**	**100 costumers 6 depots**
Three Criteria	6.48	57sec	< 1 sec
Coef. Propagation	4.74	3 sec	< 1 sec
TP	**2.55**	**< 1 sec**	**< 1 sec**

Sweep	0.30	1 sec	< 1 sec
Parallel	**0.00**	< 1 sec	< 1 sec
Simplified	-0.20	< 1 sec	< 1 sec
Cyclic	-29.80	< 1sec	< 1 sec

Table 1

The two last columns of the Table 1 show the execution time in seconds for the algorithms in a Pentium 2 of 266 MHz with 64 Mb RAM memory and Windows 98 as operating system [9].

The results for the six heuristics were obtained using STAAR [9], a graphic tool developed under the Arcview 3.0, a Geographical Information System platform. The assignments with TP were obtained using GAMS [13] and STAAR was used, as in the other cases, for the final routing (see Figure 8).

STAAR

STAAR is used for the development and testing of assignments algorithms and for solving MDVRP. The different algorithms (assignment, routing and improvements) are implemented in the C++ programming language. The problem data (customers, depots location, time windows, routes, etc. are stored in data bases managed by Arc View. Some of the features provided are: insertion of customers and depots by clicking on the map, random generation of customers and depots, specify vehicle capacity, apply any of the assignment algorithms and apply the routing algorithm. It is also possible to apply two improvement algorithms, to save and open test cases of customers and depots, so as to be able to repeat any test. In order to analyse the results, it is possible to store results and to obtain gains an average gains over several test cases. STAAR provides facilities for two kind of different experimentation analysis, accumulation of results and report generation in HTML. Figure 8 shows an assignment of customers to depots, using STAAR. The depots are the biggest points, and the customers are the smaller points and have the same colour as their assigned depot.

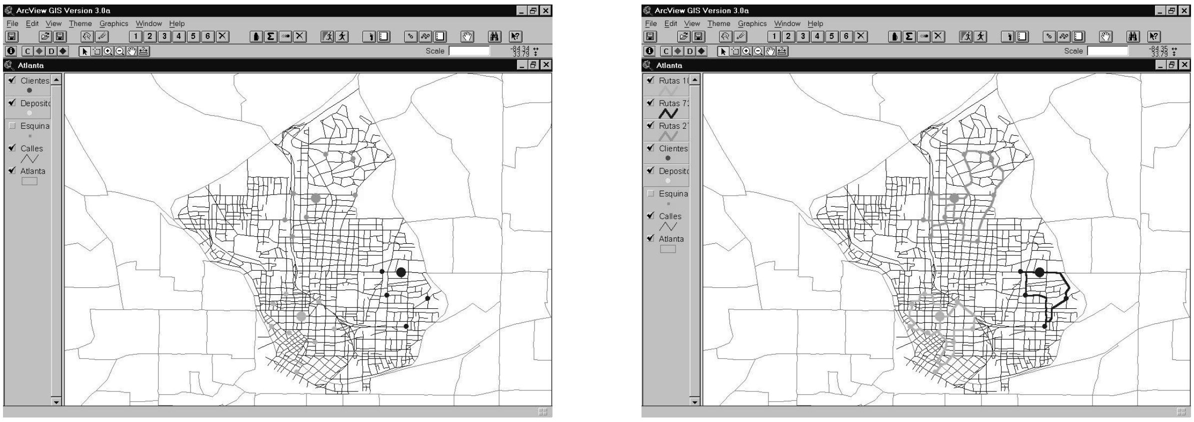

Fig.8 Assignment and Routing using STAAR.

8. Conclusions

It is worth to note 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 in this article). Good results depend not only on the algorithms but also on the instance of the problem to be solved.

The comparison of the heuristic algorithms shows three classes of algorithms: those that give good results (with high execution time), low execution time (with bad results) or medium execution time and results that are not to bad. In this last class we have the urgency algorithms, and therefore recommendable for big real life problems.

The use of TP as an assignment procedure, has proven to be better than the urgency algorithms, since it gives good results in short execution time. This makes it an attractive algorithm, as expected. On the other hand the solutions obtained with TP are attractive because the TP has been studied widely and there are many software tools to solve this problem [13]. The advantages of this approach are that TP is solved in an exact way, now a day with computers with increasing speed and computing power, the solutions obtained this way can provide upper limits or initial solutions to big MDVRP problems. This work has to continue testing and comparing other heuristics, using a bigger geographic map-zone and test cases. Other topics for future work are to include time windows and to consider solving the cases in which TP gives "split delivery" of supplies from several depots to the same customer.

9. Bibliography

[1] - L. BODIN, B. GOLDEN, A. ASSAD and M. BALL, *"Routing and scheduling of vehicles and crews: The state of the art"*. Computers and Operations Research Vol. 10, 1983.

[2] - R. RUSSELL, W. IGO, *"An assignment routing problem"*. Networks Vol. 9, 1-17, 1979.

[3] - J. FERLAND, A. HERTZ, A. LAVOIE, *"An object-oriented methodology for solving assignment-type problems with neighbourhood search techniques"*. Operations Research Vol. 44, 347-359, 1996.

[4] - J. LENSTRA, A.RINNOOY KAN, *"Complexity of vehicle routing and scheduling problems"*. Networks Vol. 11, 221-228, 1981.

[5] - G. CLARK and J. WRIGHT, *"Scheduling of vehicles from a central depot to a number of delivery points"*. Operations Research 12, 568-581, 1964.

[6] - R. MOLE S. JAMESON. "*A Sequential Route-building algorithm Employing a generalised Savings Criterion"*. Operational Research Quarterly Vol. 27 No. 2, 503-511.

[7] - S. MARTELLO, P. TOTH. *"An algorithm for the generalized assignment problem"*. Operational Research, 589-603, 1981.

[8] - J. POTVIN, J. ROUSSEAU. *"A parallell route buildning algorithm for the VRPTW*". EJOR 66 331-340, North-Holland, 1993.

[9] - D. GIOSA, L. TANSINI, O. VIERA. *"Desarrollo y Testeo de Algoritmos de Asignación para el Problema de Ruteo de Vehículos"*. Taller V. Instituto de Computación, Facultad de Ingeniería, 1998.

[10] - D. GIOSA, L. TANSINI, O. VIERA. *"Assignment algorithms for the Multi-Depot Vehicle Routing Problem"*. 28 Jornadas de Informática e Investigación Operativa, Buenos Aires, Argentina, September 1999.

[11] - MARIA URQUHART, OMAR VIERA. *"Geographical Information System and Vehicle Routing techniques applied to a specific collection problem"*. 25 Jornadas Argentinas de Informatica e Investigacion Operativa., Setiembre 1996.

[12] - MARIA URQUHART, LUIS PETINGI, OMAR VIERA. *"Vehicle Routing techniques applied to a specific distribution management problem"*. Conferencia sobre Logistica Industrial, ICIL 95, 10-14/12-95.

[13] - A. BROOKE, D. KENDRIK, A. MEERAUS. *"GAMS Release 2.25, A User's Guide"*. GAMS Development Corporation, Washington, USA, 1996.

[14] - PELLEGRIN, CANOVAS. *"A new assignment rule to improve seed points algorithms for the continuous k-centre problem"*. European Journal of Operational Research 104, 366 374 1998.

[15] - F.S.HILLIER AND G.J.LIEBERMAN. "*Introducción a la Investigación de Operaciones*". McGraw-Hill, 5ta ed., 1991.????.