EVOLUTIVE AND ACO STRATEGIES FOR SOLVING THE MULTI-DEPOT VEHICLE ROUTING PROBLEM

H. I. Calvete[1], C. Galé[2] and M. J. Oliveros[3]

[1]*Dpto. de Métodos Estadísticos, IUMA, Universidad de Zaragoza, Pedro Cerbuna 12, 50009 Zaragoza, Spain*

[2]*Dpto. de Métodos Estadísticos, IUMA, Universidad de Zaragoza, María de Luna 3, 50018 Zaragoza, Spain*

[3]*Dpto. de Ingeniería de Diseño y Fabricación, Universidad de Zaragoza, María de Luna 3, 50018 Zaragoza, Spain*

Keywords: Multi-depot vehicle routing problem, Ant colony optimization, Genetic algorithm.

Abstract: This paper addresses the multi-depot vehicle routing problem. This problem involves designing a set of routes in order to deliver goods from several depots to a set of geographically dispersed customers. For solving this problem, we propose two different approaches. Both have in common the use of an Ant Colony Optimization algorithm to construct the routes from each depot. The approaches differ in the manner in which depots are dealt with in terms of how customers are assigned to depots. In the first method, called ACO-MDVRP, the customer assignment process is controlled by the ant colony by adding a super-depot which is connected with each depot by arcs with zero unit cost. The second method, called GA-MDVRP, is a hybrid algorithm in the sense that an Ant Colony Optimization algorithm is embedded in a genetic algorithm. In order to construct a feasible solution, the procedure uses a genetic algorithm to assign customers to depots. Then, under the given data on each depot, the corresponding vehicle routing problems are solved by using Ant Colony Optimization.

1 INTRODUCTION

The vehicle routing problem (VRP) consists of designing a set of routes for serving a number of geographically dispersed customers from a central depot. The objective is to minimize total distance or total travel time. In this problem customer demand is fixed and known in advance, vehicles are assumed to be identical and cannot be overloaded, all routes start and end at the depot and each customer is visited exactly once by a single vehicle. Due to its theoretical and practical importance, a lot of effort has been devoted to solve this problem (Laporte, 2009). A lot of variants of the VRP have been proposed in the literature to model more precisely real systems. The VRP and its variants are NP hard and, usually, only small instances can be solved to optimality within reasonable computational time. Although some exact methods have been proposed to solve the VRP, most techniques described in the literature are heuristics or metaheuristics which aim to provide cuasi-optimal solutions in acceptable computational times.

A well known variant of the VRP is the multi-depot vehicle routing problem (MDVRP). In this problem a company owns several depots from which it can serve the customers. There is a fleet of vehicles based at each depot. Each vehicle starts from one depot, services a set of the customers assigned to that depot and returns to the same depot. The objective of the problem is to service all customers while minimizing the total travel time. The MDVRP arises naturally in a large variety of contexts and has considerable economic importance. We refer the reader to (Renaud et al., 1996) and (Cordeau et al., 1997) for more in-depth discussion and details regarding the structure and formulation of the MDVRP. Recent algorithms for this problem or some variants have been proposed in (Crevier et al., 2007; Giosa et al., 2002; Pisinger and Ropke, 2007) and (Tansini and Viera, 2006).

When solving the MDVRP two decisions are made: 1) how customers are assigned to depots and 2) which delivery routes are constructed. Therefore, most methods developed in the literature to solve the MDVRP propose a two-phase approach. In the first phase, customers are assigned to depots, usually according to distance. In the second phase, routes are constructed. Moreover, in order to improve the solution, different strategies of either relocating customers in a route or transferring customers from a route to another can be applied.

In this paper we propose two different metaheuristic procedures for solving the MDVRP: ACO-

I. Calvete H., Galé C. and J. Oliveros M..
EVOLUTIVE AND ACO STRATEGIES FOR SOLVING THE MULTI-DEPOT VEHICLE ROUTING PROBLEM.
DOI: 10.5220/0003673400730079
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2011), pages 73-79
ISBN: 978-989-8425-83-6

MDVRP and GA-MDVRP. Both have in common the use of an Ant Colony Optimization (ACO) algorithm to construct the routes from each depot. ACO is one of the most powerful metaheuristics developed for solving complex and large combinatorial problems (Dorigo and Stützle, 2004; Dorigo and Stützle, 2010). It is inspired by real behavior of ants when looking for food. Ants are able to communicate information about food sources by laying a chemical pheromone trail on the ground which guides other ants. First ACO algorithms were proposed in the early 1990s for solving the traveling salesman problem. Since then, several variants and extensions have been developed and a variety of optimization models have been solved by using ACO algorithms. Concerning the MDVRP, (Calvete et al., 2011) study a complex hierarchical production-distribution planning problem which has embedded a MDVRP. They propose an ACO algorithm for solving the MDVRP in which a super-depot is added and connected with each depot by arcs with zero unit cost. (Yu et al., 2011) introduce also a virtual central depot and transform the MDVRP into a VRP with this virtual depot as the origin. Then they solve this VRP by using a parallel improved ACO.

The ACO-MDVRP method proposed in this paper follows previous line of thinking and introduces a super-depot node connected at zero cost with the depots. In this method, there is no initial assignment of customers to depots. Instead, every ant starts at the super-depot and selects the real depot to visit first. From this depot, a route is built which serves a set of customers. This set of customers is assigned to that depot. The GA-MDVRP method is based on the two-phase methodology. Hence, a genetic algorithm deals with the problem of assigning customers to depots. Having made this decision, as many VRP as depots are solved by using an ACO algorithm. The paper is organized as follows. Section 2 states the problem. In Section 3 the algorithms are developed. Section 4 goes on to analyze the computational performance of the procedures on a set of benchmark problems. Finally, conclusions are presented in Section 5.

2 THE MULTI-DEPOT VEHICLE ROUTING PROBLEM

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed network where $\mathcal{N} = \{d_1, \ldots, d_L, 1, 2, \ldots, n\}$ is the set of nodes and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}\}$ is the set of directed arcs. Node d_l represents the location of depot l, $l = 1, \ldots, L$. Nodes 1 to n represent customer locations. Each arc $(i, j) \in \mathcal{A}$ refers to a direct connection from i to j and has an associated non-negative travel time t_{ij}. We assume that no routes connect depots to each other, so there is no arcs (i, j) with $i, j \in \{d_1, \ldots, d_L\}$.

We assume that there is a fleet of identical vehicles with fixed capacity U. Each vehicle is located at the depot from which its route starts. The route of each vehicle starts and ends at the same depot. Vehicles are assumed not to be overloaded, hence total demand of all customers on one particular route must not exceed the capacity of the vehicle assigned to this route. Moreover, for every vehicle route, its total route length must not exceed a given bound usually due to working time restrictions.

We assume that a fixed quantity of goods q_i has to be delivered at customer i. This quantity is known when delivery routes are established. Delivery at customer i requires a service time s_i. This delivery cannot be divide up amongst vehicles, i.e. every customer is served by a single vehicle and it is visited exactly once. Therefore, it is assumed that $q_i \leqslant U$ for all $i = 1, \ldots, n$. Otherwise, the corresponding customer could not be served.

Let $\mathcal{R}$ be a feasible solution, i.e. $\mathcal{R}$ is a set of routes verifying above constraints. Let $R = \{d_l, i_1, i_2, i_3, \ldots, i_k, d_l\}$ be a route in $\mathcal{R}$ which starts at depot d_l and sequentially visits customers i_j, $j = 1, \ldots, k$.

In order to be a feasible route, working hours regulations have to be met:

$$t_{d_l i_1} + s_{i_1} + t_{i_1 i_2} + s_{i_2} + \cdots + s_{i_k} + t_{i_k d_l} \le t_w \qquad (1)$$

where t_w refers to the maximum duration of a route. Moreover, bearing in mind that vehicles cannot be overloaded:

$$q_{i_1} + q_{i_2} + \cdots + q_{i_k} \le U \qquad (2)$$

The MDVRP consists of determining a set of routes that minimizes the total travel time. A route involves information about the depot in which the vehicle is located, the customers which are visited and the order in which they are visited.

3 ALGORITHMS

Both algorithms developed in this paper to solve the MDVRP use artificial ants to build feasible solutions. As indicated above, a feasible solution is a set of routes which visit all customers and satisfy the constraints (1) and (2). Each route must start and end at one depot. Each ant constructs a feasible solution and, when moving, lays a pheromone trail. This trail and the heuristic information are used to guide next movements of ants based on a stochastic rule.

At the end of every iteration, a number of feasible solutions are available. The size of the ant colony in each iteration is one of the algorithm parameters and is fixed a priori. Then, total travel time of these solutions is compared with the best-so-far solution and the best one is selected. This new best-so-far solution is used to guide other ants by depositing a quantity of pheromone on its arcs. The customer selection process and the guide process are main characteristics of ACO algorithms. Following subsections explain in more detail these processes for each of the algorithms proposed.

3.1 The ACO-MDVRP Algorithm

The idea of this method is to add a super-depot node which is connected with each real depot by arcs with zero unit cost. As a consequence, the original MDVRP is transformed into a VRP with the super-depot acting as the central depot. Hence, the first 'customer' of every route is selected from the set of depots $\{d_1,\dots,d_L\}$. Afterwards, the route successively selects the following customer from the set of accessible customers not yet visited. Only when the route leaves the real depot, real customers can be added to the route. Summarizing, in this algorithm a depot is assigned to each route in the same way as customers. There is no a priori assignment of customers to depots.

3.1.1 Initialization

In order to construct an initial feasible solution, in the implementation of the algorithm we have used a variant of a nearest neighbor heuristic. First, the real depot which is reached from the super-depot is randomly selected. Second, the first customer of the route is randomly selected amongst the reachable customers from the depot. Then, while it is possible to add a customer to the route, the nearest customer to the incumbent customer is selected according to travel time. When it is not possible to add a new customer to the route, the artificial ant returns to the same depot from which the route started and then goes back to the super-depot. From the super-depot the ant starts a new route.

This process continues until every customer has been visited. It is worth mentioning that this variant of the nearest neighbor heuristic, which involves a random selection in the first steps, allows us to build different initial solutions. This feature will be used to improve the performance of the algorithm.

Once an initial solution has been computed, main parameters of the algorithm are initialized. Let $\widetilde{T}$ the total travel time of the initial solution and $\tau = 1/n\widetilde{T}$. In the implementation of the algorithm, the initial pheromone trail is set to $\tau_{ij} = \tau$, $(i,j) \in \mathcal{A}$; $\tau_{0l} = \tau$, $l = 1,\dots,L$ where index 0 denotes the super-depot. The heuristic information is set to $\eta_{ij} = 1/t_{ij}$, $(i,j) \in \mathcal{A}$; $\eta_{0l} = 1$, $l = 1,\dots,L$. Let M denote the size of the colony of ants in each iteration.

3.1.2 Main Iteration

In every iteration, each of M ants constructs a feasible solution of the MDVRP. For this purpose, the ant selects the next node (a real depot at the beginning, a real customer afterwards) to be visited by applying the pseudo-random-proportional rule. Let i be the incumbent node and $\mathcal{N}_i$ be the set of reachable customers from i not yet visited. Let Q be generated from a uniform random variable in the interval $(0,1)$. Then, if $Q \leqslant Q_0$, $j \in \mathcal{N}_i$ is selected as:

$$j = \underset{h\in\mathcal{N}_i}{\operatorname{argmax}}\{\tau_{ih}(\eta_{ih})^{\beta}\}$$

where β is a parameter which determine the relative influence of the heuristic information.

If $Q > Q_0$ the ant chooses to go to customer $j \in \mathcal{N}_i$ with probability:

$$p_{ij} = \frac{\tau_{ij}(\eta_{ij})^{\beta}}{\sum\limits_{h\in\mathcal{N}_i}\tau_{ih}(\eta_{ih})^{\beta}}$$

Pheromone trails are updated locally and globally in order to guide future ants to get improved solutions. Immediately after the ant has crossed the arc (i,j), the pheromone trail of this link is updated by using the expression

$$\tau_{ij} = (1-\xi)\tau_{ij} + \xi\tau, \quad 0 < \xi < 1$$

where the parameter ξ reflects the tendency towards exploration. The goal of this local pheromone trail updating is to make this arc less desirable for the following ants in order to encourage the exploration of arcs not yet visited.

Having built the M ants their feasible solution, the best-so-far solution is selected. At this moment, it is very useful to apply local search methods to improve the quality of the solution selected. We have implemented the 2-opt intra routes and the 2-opt inter route procedures (Bräysy and Gendreau, 2005). To end the iteration, a global updating of the pheromone trail is performed. Some pheromone is added to the arcs of the best-so-far solution by using the expression:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\widetilde{\tau}, \quad 0 < \rho < 1$$

where $\widetilde{\tau}$ is the inverse of the total travel time of the best-so-far solution. The parameter ρ reflects the speed of the pheromone evaporation.

In order to improve the convergence of the algorithm and avoid stalling problems, we also propose a reinitialization of the algorithm after a number of iterations without decreasing the total travel time. Taking into account that the initial solution will be usually different due to the variant of the nearest neighbor heuristic implemented, the reinitialization process improves the exploration of different feasible solutions.

3.2 The GA-MDVRP Algorithm

In contrast to the previous algorithm, in the GA-MDVRP algorithm customers are assigned to depots before the construction of routes starts. Moreover, this process is controlled by a genetic algorithm (GA). GA are stochastic search techniques inspired by natural biological evolution. They were introduced by (Holland, 1975). When applying GA to solving a problem, each solution is encoded as a string of symbols which is called chromosome. Each position in the string is a gene and its value is the allele value. Each chromosome has a fitness value associated which measures its quality, usually in terms of the objective function.

To start the algorithm, an initial population of chromosomes is generated and the fitness of each chromosome is computed. The size p of this population is a parameter of the algorithm. In each iteration, a new population is generated. New chromosomes are formed by combining chromosomes from the current population (parents) using a crossover operation or by modifying a chromosome using a mutation operation. After evaluating the fitness of the resulting chromosomes, some of the chromosomes of the old population and some new chromosomes (offspring) are selected to form the new population. Some variants of the selection procedure have been proposed in the literature. The most common is the elitist strategy in which the best p chromosomes in terms of fitness are kept from one generation to the next. The algorithm proceeds by building populations until the stopping condition is met.

To solve the MDVRP, we have developed a GA aiming to find good customer assignments to depots that lead to good solutions of the MDVRP in acceptable computational times. The idea underlying the proposed GA is to associate chromosomes with assignments. A gene corresponds to a customer and the allele value indicates the depot which serves him. The fitness of a chromosome is the value of the total travel time of the solution of the MDVRP obtained by putting together the best solutions provided by an ACO algorithm applied to each of the L VRPs arisen when considering each depot and its assigned set of customers.

3.2.1 Initialization

The initial population is formed by p chromosomes. Several routines for selecting the members of this initial population can be envisaged. For instance, a chromosome can be constructed by assigning each customer to its nearest depot. Another possibility is to assign the customer to a depot at random. This possibility could also limit the number of customers assigned to each depot or not. Initially, two routines were implemented in the algorithm: Nearest depot and completely random assignment. The first one usually gives a good chromosome from the fitness point of view. However, random assignment provides very bad chromosomes. Hence, in order to have more promising chromosomes in the initial population we implemented a routine in which every customer has the possibility of being assigned only to a set of depots which are close to him. In order to define the closeness of a depot, let $t_{max} = \max\{t_{id_l} : (i,d_l) \in \mathcal{A}; i = 1,\ldots,n; l = 1,\ldots,L\}$ and $t^i_{min} = \min\{t_{id_l} : (i,d_l) \in \mathcal{A}; l = 1,\ldots,L\}$, $i = 1,\ldots,n$. The depot d_l is close to the customer i if

$$t_{id_l} - t^i_{min} \leqslant \frac{t_{max}}{4}$$

Then, the customer i is randomly assigned to one of the depots close to him.

To evaluate the fitness of each chromosome, we compute a good solution of the MDVRP bearing in mind the customer assignment determined by the chromosome. For this purpose, we propose to solve the VRP associated to each depot by using an ACO algorithm. The description of this algorithm is similar to the one given in Section 3.1. Instead of a super-depot, L depots and n customers, there is one depot, say d_l, and n_l customers (those previously assigned to d_l). Let T_l be the total travel time of the best solution provided by the ACO algorithm after solving the VRP corresponding to the depot d_l. The fitness of a chromosome C is

$$f(C) = \sum_{l=1}^{L} T_l$$

3.2.2 Main Iteration

The GA proceeds by performing crossover, mutation, evaluation and selection, until the stopping condition is met. In each generation a population formed by p chromosomes is maintained. In the crossover process, chromosomes are selected from the source population and combined to form offspring which are potential members of the successor population. A chromosome of the current generation is selected for the crossover operation with probability p_c. The parents selected are taken in pairs to provide offspring. Let us assume that there are 4 depots and 10 customers. Let the two parents selected be:

Parent 1: $\begin{pmatrix} 1 & 2 & 1 & 1 & 3 & 4 & 1 & 2 & 4 & 3 \end{pmatrix}$

Parent 2: $\begin{pmatrix} 3 & 3 & 1 & 2 & 1 & 4 & 1 & 3 & 1 & 3 \end{pmatrix}$

Then a location is randomly selected (for example the seventh one) and the right-hand indices are exchanged in pairs:

Parent 1: $\begin{pmatrix} 1 & 2 & 1 & 1 & 3 & 4 & 1 & 3 & 1 & 3 \end{pmatrix}$

Parent 2: $\begin{pmatrix} 3 & 3 & 1 & 2 & 1 & 4 & 1 & 2 & 4 & 3 \end{pmatrix}$

After crossover, the mutation operation is carried out. A chromosome of the current population is selected for the mutation operation with probability p_m. Once it has been selected, several customers are randomly selected from the chromosome and a depot is randomly assigned to each of them.

After mutation, the fitness of the new chromosomes obtained from crossover and mutation is computed. The offspring are added to the current population and the whole set of chromosomes is passed on to the selection step. We use the elitist strategy which selects the best p chromosomes of the current set of chromosomes available (current population plus offspring resulting from the crossover and mutation operations). These chromosomes form the population of the next iteration.

The population size, the probability of crossover, the probability of mutation, the number of genes affected by the mutation operation and the stopping condition are control parameters of the algorithm which have to be set in the beginning.

4 COMPUTATIONAL EXPERIMENT

The performance of the algorithms has been tested on the set of MDVRP benchmark instances of (Cordeau et al., 1997), which can be found in http://neumann.hec.ca/chairedistributique/data/. The ten problems vary in size from 48 to 288 retailers and have 4 or 6 depots. Table 1 displays the characteristics of the problems and the total travel time of the best known solution http://neo.lcc.uma.es/radi-aeb/WebVRP.

The purpose of the experiment was twofold. On the one hand, to check the algorithm efficiency in terms of the closeness of the best solution provided by each algorithm to the best known solution. On the other hand, to compare both algorithms aiming to conclude which one seems to have a better performance. The experiments were performed on a PC Core Quad 6600 at 2.4 GHz having 4 GB of RAM under Windows 7. Both algorithms were coded in C++ using Borland Builder. For each problem five runs with each algorithm were done.

Table 1: Description of problems and total travel time of the best known solution.

Problem	# of customers	# of depots	f_{bk}
1	48	4	861.32
2	96	4	1288.37
3	144	4	1782.58
4	192	4	2072.52
5	240	4	2343.66
6	288	4	2675.16
7	72	6	1085.61
8	144	6	1666.60
9	216	6	2153.10
10	288	6	2811.49

Based on pilot testing, the general parameters of the ACO-MDVRP algorithm have been set to $\xi = \rho = 0.1$, $\beta = 2$ and $Q_0 = 0.5$. Taking into account that better solutions are obtained when combining ACO algorithms with local search algorithms, we have implemented the 2-opt intra routes and the 2-opt inter routes local search procedures. These methods are applied to the best-so-far solution at the end of each iteration. Reinitialization of the algorithm, as explained in 3.1.2 is carried out after two iterations without objective function improvement. The number M of ants in each iteration has been set to 50. The termination condition of this algorithm is established in terms of computing time. In order to compare both algorithms, the ACO-MDVRP algorithm is assigned for each problem a computing time similar to that used by the GA-MDVRP algorithm in the same problem.

Regarding the GA-MDVRP algorithm, each run consists of eight iterations, the population size is $p = 4$, the probability of crossover is $p_c = 0.3$, the probability of mutation is $p_m = 0.1$ and the mutation operation is carried out on 5 genes (customers) randomly selected. The general parameters of the ACO algorithm applied to solve the VRP associated to each depot are set to $\xi = \rho = 0.1$, $\beta = 2$ and $Q_0 = 0.5$ and local search procedures are applied as explained above. In this case, $M = 4$ ants are used in each iteration and the ACO algorithm ends after 2 minutes of computing time.

Tables 2 and 3 display the results of the experiment. For every problem, the first column indicates the number of the problem. Columns two and three refer to the objective function values, that is to say

Table 2: Experimental results of the ACO-MDVRP algorithm: Total travel time of the best solution f_b, average of the five runs $\bar{f}$ and percentage of deviation %Dev.

Problem	f_b	$\bar{f}$	%Dev
1	872.36	886.42	1.28
2	1366.34	1381.52	6.05
3	1926.05	1931.99	8.05
4	2208.90	2246.57	6.58
5	2543.51	2551.21	8.53
6	2934.56	2971.38	9.70
7	1126.78	1142.91	3.79
8	1760.57	1784.43	5.64
9	2334.86	2365.47	8.44
10	3088.71	3116.20	9.86

the total travel time, obtained in the five runs of each instance. Column two shows the total travel time of the best solution provided by the corresponding algorithm and column three displays the average of the five runs of the instance. In column four the best solution obtained is compared with the best known solution by using the expression

$$\%\text{Dev} = \frac{f_b - f_{bk}}{f_{bk}} \times 100$$

where f_b is the best objective function value provided by the algorithm and f_{bk} is the objective function value of the best known solution.

From the comparison of both tables, we can derive that GA-MDVRP shows a better performance than ACO-MDVRP. In fact, GA-MDVRP provides a better solution than ACO-MDVRP for all problems. Moreover, GA-MDVRP provides good solutions to all problems since only 3 out of 10 problems (which are the larger ones) show a deviation higher than 5.5%, but less than 9%, from the best known solution. Figure 1 shows the percentage of deviation for both algorithms. We can see that it is higher for ACO-MDVRP than GA-MDVRP. Moreover, %Dev increases in both algorithms with the size of the problem, specially with the number of customers.

Finally, it is worth making some comments on the computing time invested by both algorithms. In order to be able to compare both algorithms, we carried out the experiment with the GA-MDVRP algorithm, took the average time invested in the five runs and assigned the ACO-MDVRP algorithm a similar computing time for each run. However, we observed that the best solution of each run of the ACO-MDVRP algorithm was provided in much less time than assigned. This suggests that we could have made fewer iterations when applying this algorithm, thus saving the time invested, without, in general, losing accuracy.

Table 3: Experimental results of the GA-MDVRP algorithm: Total travel time of the best solution f_b, average of the five runs $\bar{f}$ and percentage of deviation %Dev.

Problem	f_b	$\bar{f}$	%Dev
1	861.32	880.82	0.00
2	1310.20	1339.54	1.69
3	1872.02	1884.23	5.02
4	2177.19	2203.61	5.05
5	2517.14	2534.88	7.40
6	2913.02	2945.42	8.89
7	1098.56	1110.16	1.19
8	1719.78	1745.84	3.19
9	2268.83	2280.82	5.37
10	3028.47	3064.58	7.72

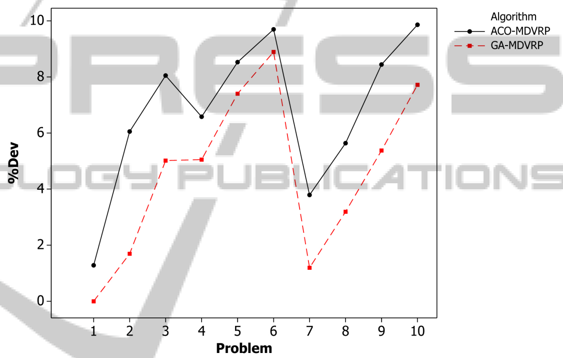

Figure 1: %Dev versus Problem.

5 CONCLUSIONS

The MDVRP is a generalization of the standard VRP in which more than one depot is available to deliver goods to a set of geographically dispersed customers. The objective of the problem is to design a set of routes for serving the customers while minimizing the total travel time. Each vehicle starts from one depot, services a set of customers and returns to the same depot. In this paper we have develop two algorithms for solving this problem: ACO-MDVRP and GA-MDVRP. They differ essentially in the way in which customers are assigned to depots.

The ACO-MDVRP algorithm does not execute an a priori assignment of customers to depots. By introducing a super-depot, the ACO-MDVRP algorithm uses an ACO approach in which the assignment of customers to depots is controlled by the colony of ants and is made as the route is being built.

On the contrary, the GA-MDVRP algorithm is a hybrid algorithm which uses GA to assign a priori customers to depots. A chromosome is an assignment of customers to depots. The fitness of a chromosome is defined as the total travel time of the routes which serve the customers assigned to each depot. For the purpose of computing the fitness, an ACO algorithm is applied to design the routes.

The results of the computational experiments carried out using a set of MDVRP benchmark instances indicate a better performance of the GA-MDVRP algorithm. Moreover, this algorithm provides good results in terms of accuracy of the solution.

ACKNOWLEDGEMENTS

This research work has been supported by the Spanish Ministry of Education and Science under grant MTM2010-17559

REFERENCES

Bräysy, O. and Gendreau, M. (2005). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118.

Calvete, H., Galé, C., and Oliveros, M. (2011). Bilevel model for production-distribution planning solved by using ant colony optimization. *Computers and Operations Research*, 38(1):320–327.

Cordeau, J., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119.

Crevier, B., Cordeau, J., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176:756–773.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambrigde, MA.

Dorigo, M. and Stützle, T. (2010). Ant colony optimization: Overview and recent advances. In Gendreau, M. and Potvin, J., editors, *Handbook of Metaheuristics*, pages 227–263. Springer, 2 edition.

Giosa, I., Tansini, I., and Viera, I. (2002). New assignment algorithms for the multi-depot vehicle routing problem. *Journal of the Operational Research Society*, 53(9):997–984.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

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

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435.

Renaud, J., Laporte, G., and Boctor, F. (1996). A tabu-search heuristic for the multi-depot vehicle routing problem. *Computers and Operations Research*, 23(3):229–235.

Tansini, L. and Viera, O. (2006). New measures of proximity for the assignment algorithms in the mdvrptw. *Journal of the Operational Research Society*, 57:241–249.

Yu, B., Yang, Z., and Xie, J. (2011). A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society*, 62(1):183–188.