

UDK: 656.07

A Metaheuristic ACO to Solve the Multi-Depot Vehicle Routing Problem with Backhauls

Jhon Jairo Santa Chávez

Universidad Libre Seccional Pereira, Pereira (Colombia), jj santa@unilibrepereira.edu.co, Universidad Tecnológica de Pereira, Pereira (Colombia), jj santa@utp.edu.co

Mauricio Granada Echeverri

Universidad Tecnológica de Pereira, Pereira (Colombia), magra@utp.edu.co

John Willmer Escobar*

Pontificia Universidad Javeriana Cali, Cali (Colombia), jwescobar@javerianacali.edu.co

César Augusto Peñuela Meneses

Universidad Libre Seccional Pereira, Pereira (Colombia), capenuela@unilibrepereira.edu.co

Received (11.05.2015.); Revised (17.06.2015.); Accepted (03.07.2015.)

Abstract

In this paper, a metaheuristic algorithm based on the ant colony optimization is presented to solve the multi-depot vehicle routing problem with delivery and collection of package. Each performed route consists of one sub-route in which only the delivery task is done, in addition to one sub-route in which only the collection process is performed. The proposed algorithm tries to find the best order to visit the customers at each performed route. In addition, the proposed approach determines the best connection between the sub-routes of delivery and collection, in order to obtain a global solution with the minimum travelling cost. The efficiency of the proposed algorithm has been evaluated by considering a set of instances adapted from the literature. The computational results have been compared with a greedy heuristic algorithm based on the nearest neighborhood approach. Finally, conclusions and suggestions for future works are presented.

Key words: Ant Colony, Backhauls, Combinatorial Optimization, Computational Simulation, Multi-Depot Vehicle Routing Problem.

1. INTRODUCTION

The transportation and logistics companies are facing demanding situations with fewer resources. Currently, an important aspect is to optimize the use of existing resources in order to perform specific tasks. Many of these tasks in real-world problems can be solved by a network representation containing nodes, connections, variables, parameters, and their interactions. In recent decades, there have been various proposals for solving vehicle routing problems with backhauls by considering only capacity constraints. Toth and Vigo [1] propose the first approach to the formulation of a vehicle routing problem based on a fuel distribution problem. In Clarke and Wright [2], the first effective heuristic algorithm for solving the Capacity Vehicle Routing Problem (CVRP) is proposed.

This method starts with a set of vehicles located at the depot. Some of these vehicles are selected to deliver the cargo to the customers through predefined routes. Then, two routes are selected iteratively in order to determine if a lower distance can be generated as a

consequence of merging them. In this case, the routes are merged to create a route that

will be attended by only one vehicle. In all possible scenarios, the capacity of the vehicle must be ensured.

The Multiple-Depot Vehicle Routing Problem with Backhauls (MDVRPB) could be defined as follows:

Given two sets of predefined customers, where L is the set of *Linehaul* customers (customers to be delivered products), and B the set of *Backhaul* customers (customers to be collected products). In addition, given a set of geographically dispersed depots; the objective is to consider the routes to be performed from the selected depots to the customers by a fleet of homogeneous vehicles in order to satisfy the demand of the customers (products to be collected or products to be delivered). In such case, the vehicles must attend first the customers with delivery requirements before the customers with collection requirements.

The features of the customers, depots and vehicles, as well as the different operating constraints on the performed routes, leads to different variants of the problem: The Vehicle routing Problem with Backhaul (VRPB) in which only one depot is considered, the

Mixed Vehicle routing Problem with Backhaul (MVRPB) in which a fleet of homogeneous capacity vehicles is concerned, and the VRPB with time windows (VRPBTW) in which the expected time to serve the clients is constrained. The MDVRPB is considered as NP-hard, and has not been deeply explored in the literature. The variants of the MDVRPB can be classified into three categories: (i) simultaneous collecting and dispatching of products; (ii) collecting first, following the delivery of products; and (iii) collecting after of the delivering process.

Recently, several algorithms have been proposed to solve the VRPB. In [1], an exact model for the formulation of the VRPB in a fuel distribution scenario is proposed. The first heuristic algorithm for VRPB has been proposed by Deif and Bodin [3]. This algorithm is an extension of the known method of saving method proposed by Clarke and Wright [2]. The algorithm starts with a feasible solution, where each customer is visited by a different route.

Then, the routes are combined iteratively by considering the possible reduction in the objective function, i.e. the traveling cost of the route that can be achieved by serving different type of customers with the same vehicle. The results show that the evolution of the algorithm is affected by the constraint of precedence of customers with delivery requirements. This fact substantially reduces the number of feasible links between different types of customers. Deif and Bodin [3] experimentally show that the best results are obtained when mixed routes are initially constructed. Therefore, [3] propose a modification of definition of savings in order to penalize the edges connecting customers of different types.

Thus, avoiding the union of backhauling customers at the end of the algorithm. The savings generates with the use of the edge S_{ij}^* is defined in equation (1).

$$S_{ij}^* = \begin{cases} S_{ij} - pS_i & \text{if } i \in L, j \in B \text{ otherwise} \\ S_{ij} & \text{otherwise} \end{cases} \quad (1)$$

Where S is an estimate of the maximum savings between a *Linehaul* customer ($i \in L$) and a *Backhaul* customer ($j \in B$), and p is a penalty factor set to [0,1]. Deif and Bodin in [3] have tested the proposed algorithm on cases randomly generated with instances between 100 and 300 customers, and a percentage of backhaul customers among 10% to 50% of the total of customers. The Multi Depot Vehicle Routing Problem with Backhauls (MDVRPB) is an extension of the well-known VRPB by considering several depots, which are not necessarily used. Salhi and Nagy [4] propose a heuristic approach to solve the MDVRPB. The algorithm is based on the idea of "Border Customers", defined as the customers geographically located in a middle point between two depots. Min [5] introduced the MDVRPB by considering collection process after delivery of products.

A unified heuristic for different vehicle routing problems with backhaul is presented in [6]. In this work, the version of mixed pickup and delivery with multiple

depots is considered. Two algorithms based on ant colony for the multi depot vehicle routing problem with multiple depots and mixed pickup and delivery are presented in [7]-[8]. Finally, genetic algorithms for solving the MDVRPB have been proposed by Chunyu & Xiaobo [9] and Chunyu [10].

This paper proposes an algorithm based on an Ant Colony System to solve the MDVRPB with collection of products exclusively after delivery of products. The efficiency of the proposed algorithm has been compared in 33 instances taken from the literature, proposed by Salhi and Nagy [4].

These instances have been used to test the two first categories of the MDVRPB (simultaneous collecting and dispatching of products and collecting first, following the delivery of products). However, there is no relevant literature on the subject of the variant solution of collecting products after of the delivering process. Therefore, for comparison purposes, we have implemented a greedy algorithm based on the nearest neighbor idea. The paper is organized as follows: Section 2 provides a brief description of the MDVRPB. Section 3 shows described the metaheuristic Ant Colony System. In Section 4, the proposed algorithm is introduced. Finally, in Sections 5 and 6, we present the analysis of computational results and the conclusions for future research, respectively.

2. PROBLEM STATEMENT

The vehicle routing problem with Backhaul, is an extension of the well-known VRP involving demand nodes and collection of products [1]. The VRPB can be defined as the problem of determining a set of routes to visit every customer concerning the following constrains: each vehicle develops a single route; each route starts and ends at the depot; the routes must firstly deliver products to customers defined as Linehaul, and finally pick up products from customers defined as Backhaul. The MDVRPB is an extension of VRPB by considering m potential depots from which we can generate different routes. In the MDVRPB all deposits are not necessarily used. The MDVRPB can be formulated as a Graph Theory Problem. Let $G = (V, A)$ be a complete undirected graph, where $V = \{1, \dots, n\}$ is the set of vertices, and A is the set of arcs. The set V is partitioned into two subsets: the set of customers $N = \{1, \dots, n\}$, and the set of potential depots $M = \{1, \dots, m\}$. Additionally, the set N is divided into a subset of Linehaul nodes (Linehaul customers - L), and the Backhaul nodes (Backhaul customers - B). Therefore, $N = L \cup B$.

Each customer has a nonnegative amount D_i ($i \in N$) of product to be delivered ($i \in L$) or to be picked up ($i \in B$). Each depot has a fictitious demand, i.e. $D_i = 0$, with $i \in M$. A set of K identical vehicles with a given capacity Q_k is initially placed at each depot. It must be clarified that all vehicles are not necessarily used. For each edge $(i, j) \in A$ merging the vertices i with the vertices j , a nonnegative cost c_{ij} is associated,

where $c_{ij} = +\infty$ for each arc $(i, j) \in A$. As the symmetrical condition is assumed, for each arc $(i, j) \in A$ we assume that $c_{ij} = c_{ji}$, where $i \neq j$. The main goal of the MDVRPB is to find the minimum cost (traveled distance) of r routes, where the following conditions must be satisfied:

- a) Each vehicle must start from a depot m , and return to the same depot m ;
- b) Each customer must be visited exactly once.
- c) The sum of the demands of customers belonging to a determined route must not exceed the vehicle capacity Q ;
- d) For each performed route, the Linehaul customers must precede the Backhaul customers in the sequence;
- e) The flow between depots is not allowed.
- f) The objective is to minimize the total traveled distance, defined as the sum of the costs of the arcs belonging to the performed routes. [1]

A mathematical model for the MDVRPB has been introduced in [11]. The objective function is calculated as the minimization of the total traveled distance by the performed routes (1). So, the binary variable x_{ij}^k is set to 1 when the vehicle k passes by the edge $(i, j) \in A$. Otherwise x_{ij}^k is set to 0

$$\text{Minimize Total Cost} = \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k \in K} x_{ij}^k \tag{1}$$

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1 \quad \forall i \in N \tag{2}$$

$$\sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k \quad \forall j \in V, k \in K \tag{3}$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij}^k = 1 \quad \forall k \in K \tag{4}$$

$$\sum_{i \in S} d_i \sum_{j \in V} x_{ij}^k \leq Q_k \quad \forall k \in K \tag{5}$$

$$\sum_{i \in B} d_i \sum_{j \in V} x_{ij}^k \leq Q_k \quad \forall k \in K \tag{6}$$

$$\sum_{i \in S} d_i \sum_{j \in B} x_{ij}^k \leq 1 \quad \forall k \in K \tag{7}$$

$$x_{ij}^k \in \{0, 1\}$$

Equations (2) constrain that a single vehicle visits each customer once. Additionally, the set of equations (3) impose the flow constraints for each customer. Each performed route must be started and finished at the same depot (4). The capacity of the vehicles is

constrained by equations (5) and (6) for linehaul and backhaul customers, respectively. Finally, linehaul customers must be addressed before the backhaul customers (7).

Solving the mathematical model (1) – (7) by exact methods, the computational time grows exponentially as the number of vertices V is increased. For this reason, metaheuristic algorithms based on Ant Colony System appear as an alternative for the solution of MDVRPB.

3. METAHEURÍSTIC ALGORITHM BASED ON ANT COLONY SYSTEM

The Ant Colony metaheuristic is based on the natural behavior of ants searching food. The logical tendency of each ant is to reduce the effort and time required to gather food. This goal is achieved by reducing the distance between two specific points for collecting food. The ants are individuals with relatively simple features. However, they perform highly complex work in a simple way. The success lies in the interaction of many individuals with the environment and by indirect communication between them through chemicals substances called pheromones [12].

This behavior is emulated in artificial intelligence to find good quality solutions to optimization problems characterized by a wide space of solutions. The concept makes the equivalence between the natural ant colony and the artificial system moving within a computational environment [12].

3.1 Selecting arcs

The proposed algorithm interprets an ant as the emulation of a vehicle in the process of performing its route. An artificial ant begins its travel from a depot selected randomly to a first customer. The decision of the next client to be visited is based on heuristic preferences biased by the distance among nodes and the emulation of the natural pheromone.

A probabilistic transition rule defines the likelihood that the ant k (one ant for each available vehicle), placed at node i , decides to move to the node j (selecting the arc $(i, j) \in A$). The rule is defined according to the equation (8).

$$P_k(i, j) = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{s \in S} [\tau_{is}]^\alpha \cdot [\eta_{is}]^\beta} \tag{8}$$

Where τ_{ij} is the amount of pheromone on the edge $(i, j) \in A$, and η_{ij} is the heuristic information of (i, j) , i.e. the distance to travel from the node i to the node j . The level of importance on the rule decision from both the heuristic information and the amount of pheromone is given by parameters α and β , respectively. Generally, these parameters must have optimized at an off-line process. Finally, S is the set of neighborhood

nodes of node i , which have not visited yet by the ant k .

3.2 Setting of parameters α and β

In particular, if $\alpha = 0$ the nearest nodes have more probability to be selected (well-known algorithm of the gradient with multiple starting points). If $\beta = 0$ only the level of pheromone is considered, generally obtaining solutions with low quality, especially if $\alpha > 1$. Indeed, this case causes a stagnation of an algorithm, because all the ants follow the same way generating sub optimal solutions. A proper update pheromone matrix directly impacts the diversity in decision-making process by an ant for the following arcs. We have considered that when an ant travels by an arc a certain level of pheromone is deposited, which is evaporated with the time.

In particular, an additional positive feedback of the amount of pheromone is used to reinforce future components of good solutions. For a better solution, more pheromone is provided at its arcs.

The equation (9) determines the rule for increasing the quantity of pheromone, where $\Delta\tau_{ij}^k$ is the amount of pheromone deposited at the arcs visited by the ant k , where $C(\tau_k)$ is the total cost of the solution generated by the ant k , i.e., the total distance of the route developed by the vehicle k . The arcs visited by the entire ants in the current solution will receive an extra contribution of pheromone.

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{C(\tau_k)} & \text{if the edge } (i,j) \text{ is used by the ant } k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

In addition, pheromone evaporation is used to prevent an unlimited increasing of pheromone trails and to forget low quality solutions. The evaporation level is the same for all pheromone trails, eliminating a predefined percentage of the current value $\tau_{ij}(t)$ for each arc (i,j) , through a predefined rate ρ , with $0 \leq \rho \leq 1$.

Thus, the evaporation mechanism could exhibit a behavior more active than the presented in natural environments. The equation (10) describes the general rule for updating the pheromone matrix at the edge $(i,j) \in A$, at the iteration t .

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_k \Delta\tau_{ij}^k \quad (10)$$

3.3 Ant System (AS)

In the ant system algorithm, an artificial colony of ants cooperate to find good solutions for discrete, static, and dynamic combinatorial optimization problems.

Three different variants for the AS have been proposed: Ant-Density, Ant-Quantity and Ant-Cycle [13]. In the first two variants, the pheromone updating process is performed after each selection arc process, while in the third variant (Ant-Cycle), the amount of pheromone is only updated once the ants have completed the entire

routes. The first two versions generally obtain worse results than the third variant [13].

Therefore, we consider the process for updating the pheromone by assigning a fixed amount of pheromone at each iteration. The equation (11) is generally used to determine this value, where K is the number of ants, i.e. the number of alternative of solutions, and C_{min} is the tour length obtained at the construct solution procedure.

$$\tau_{ij} = \frac{K}{C_{min}} \quad (11)$$

4. PROPOSED ALGORITHM FOR THE MDVRPB

4.1 Encoding Solution

The problem of MDVRPB is solved by an ant colony algorithm [13]. Then, the implemented encoding is presented. The distance matrix is constructed from the coordinates of the vertices, i.e. the matrix contains the Euclidean distances among depots and customers. In order to improve the efficiency in the search process, the information is arranged in a redundant way. So, the Figure 1 shows the distance matrix is a three-dimensional array with dimensions $(n+1) \times (n+1) \times (n)$. It is shown that this matrix can be understood as a set of submatrix each one dedicated to a particular depot. At any submatrix, the element (i,j) represents the cost for travel through the edge merging the customer i with the customer j . In the plane corresponding to the distances between customers, for each deposit, the diagonal is greater as possible, and the no diagonal values are symmetrical. The Figure 1 sketches the structure of the distance matrix.

The Visibility Matrix is obtained from each element of the matrix elements of distances. It contains the inverse of the distance. This matrix is not modified during the execution of the algorithm and it is symmetrical. Again, each plane corresponds to a particular depot, and its diagonal is high enough. The Linehaul and Backhaul customers use the same representation because the algorithm identifies the delivery or pick up of products (Figure 2).

Finally, we proceed to generate the pheromone matrix with all positions initialized with the same amount of pheromone. As shown in the Figure 3, all the elements at the diagonal are zero, and it is initially a symmetric matrix. Their values are generated with the equation (11), and the updating process is restricted to thus elements in the route traveled by each ant.

Note that the proposed algorithm not considers the distance among depots, since these distances should not be considered in the distance of the performed routes. Indeed, the algorithm uses a new vehicle to continue visiting other customers. Therefore, the visibility matrix or net matrix is replaced by a squared array of random values that provides diversity in the searching of the new depot to be aggregated to the solution. The total number of depots gives the dimension of the visibility matrix, and its values are

initialized using the equation (11). The nodes are updated only if they belonging to the route traveled by a vehicle.

In addition, a vector of not visited customers is required. It contains the feasible customers to be selected. Initially, it contains Linehaul customers, and each selected customer is removed from the vector. The first element of the vector is 1, and the final element is set to the total number Linehaul customers (NL). Similarly, it is needed a vector for Backhaul not visited customers. It starts with the total Backhaul customers (NB). When a feasible customer is selected, it is removed from the vector. The first element of vector for Backhaul not visited customers is the total number of Linehaul customers + 1, and the final element is set to total number of customers ($NL + NB$).

The solution vector is also required. Its first element is set to the number of a depot, which is randomly selected. It is followed by a sequence of customer until a new depot has been selected again. This shape is continued until the entire customers have been selected. However, at each sequence of customers, a Linehaul customer must not appear between two Backhaul customers, and vice versa. Additionally, the Backhaul customers are visited before the Linehaul customers. The Figure 4 shows an example of a route of the proposed solution. The process to construct an alternative of solution is repeated until the population of the algorithm is complete. The number of the ants to create is a parameter to be established by an off-line process.

Finally, an array stores the solution vectors found by different ants. This array represents the population matrix, for which every row is able to store only one alternative of solution found by an ant. The number of rows is the number of iterations by the number of depots and the number of columns is at least the number of customers plus the number of depots to satisfy the collections and delivery of products. In the Figure 5, a set of alternative of solutions is shown. At this example, there are three (3) depots, four (4) Linehaul customers, and three (3) Backhaul customers.

4.2 Types of constructed routes

Let us define the total number of depots as m : i.e. M_1, M_2, \dots, M_m , the total number of linehaul customers as nl : L_1, L_2, \dots, L_{nl} and the number of backhaul customers as nb : B_1, B_2, \dots, B_{nb} . As a example, the proposed algorithm performs the following routes for a system with three (3) depots: M_1, M_2 and M_3 ; with seven (7) linehaul customers: $L_1, L_2, L_3, L_4, L_5, L_6$ and L_7 ; and seven (7) backhaul customers: $B_1, B_2, B_3, B_4, B_5, B_6$ and B_7 . We can consider the following types of routes:

- **A linehaul – backhaul route.** The Figure 6 describes this type of route. Its main feature is that both the linehaul sequence and the backhaul sequence are reaching the capacity of the vehicle. The performed route is $M_2-L_1-L_6-L_7-B_5-B_1-B_6-M_2$.

- **A single linehaul route.** This route contains only linehaul customers. This route is shown in the Figure 7. The performed route is $M_1-L_4-L_2-L_3-L_5-M_1$ (Figure 7).
- **A single backhaul route.** This type of route is depicted in the Figure 8. It contains only backhaul customers. The performed route is $M_3-B_4-B_3-B_2-B_7-M_3$.

The alternative of solution is conformed as a combination of the possible types of routes. In the proposed algorithm, the construction of a multi-depot solution requires that the ant performs as many subroutes as needed to visit all the customers, and each sub-route comprises a tour considering linehaul and backhaul customers. A possible alternative of solution is shown in the Figure 9. The dashed line represents the entire tour performed by the ant, and it is equivalent to an alternative of solution.

4.3 Pseudocode for the proposed algorithm

Procedure MDVRPB (depots, customers, vehicleNumber, vehicleCapacity)

- Calculate distance matrix
- Initialize the Pheromone matrixes
- Initialize two Net matrixes
- Initialize Pheromone contribution array
- Initialize Net contribution array
- It = 0
 - While it < 100 do
 - ants = 0
 - While ants < size(depots) do
 - // Starting MDVRPB procedure
 - initialDepot = choose(depots)
 - If anyLinehaul(customers) then
 - Process Linehaul customers
 - If anyBackhaul(customers) then
 - Process Backhaul customers
 - Evaporate(pheromones)
 - update(pheromones)
 - ants++
 - End While
 - End While

5. COMPUTATIONAL RESULTS

5.1 Instances

The proposed algorithm has been evaluated in 33 instances MDVRPB described by Salhi and Nagy [5]. The respective set has a diverse number of Linehaul and Backhaul customers (from 50 to 250), a number of depots is between 2-5, and a homogeneous set of vehicles is considered. Euclidean distance for each arc is performed. The set of benchmark instances has been used to solve the problem of allowing MDVRPB percentages in the delivery and pickup of 10%, 25% and 50% (Salhi and Nagy [5] and Ropke and Pisinger [6]). However in the reviewed literature, there is not found papers that solve the problem MDVRPB with Backhauls at the end of the routes. Therefore, we have implemented a simple greedy heuristic based on the

nearest neighborhood in order to make a comparison with the proposed approach.

5.2 Description of the Greedy algorithm

The greedy algorithm is a constructive approach based on the nearest neighbor, i.e. the lower distance to the current location. Hence, starting from a particular depot, the route in construction receives the customer with the lower distance to the depot among the customers not assigned yet.

This rule is sequentially repeated, but the current location regards a last customer added to the solution. So, the feasible customer is evaluated through the distance to the last customer in the route in consideration. When a backhaul customer is added to the solution, the next location cannot be a linehaul customer, and only backhaul customers are considered, as well as coming back to the initial depot. A random

decision determines whether the next route will be started from the current depot, or at other depot.

At each case, the procedure described before is repeated. The overall process is executed until all customers had been attended.

5.3 Obtained Results

The proposed algorithm has been implemented in MatLab, and computational experiments have been run on a PC with Core i5 1.4 GHz processor and 8GB of RAM. The Table 1 shows the obtained results in comparison with the heuristic nearest neighborhood. We report the computing time and the values reached by the metaheuristic objective proposed. The algorithm has been executed for 10 runs over 100 iterations, reporting the average results and the best results of all runs.

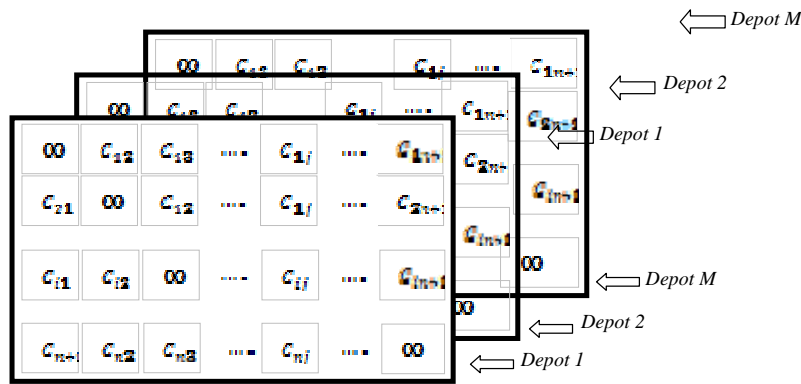


Figure 1. Encoding of the distance matrix.

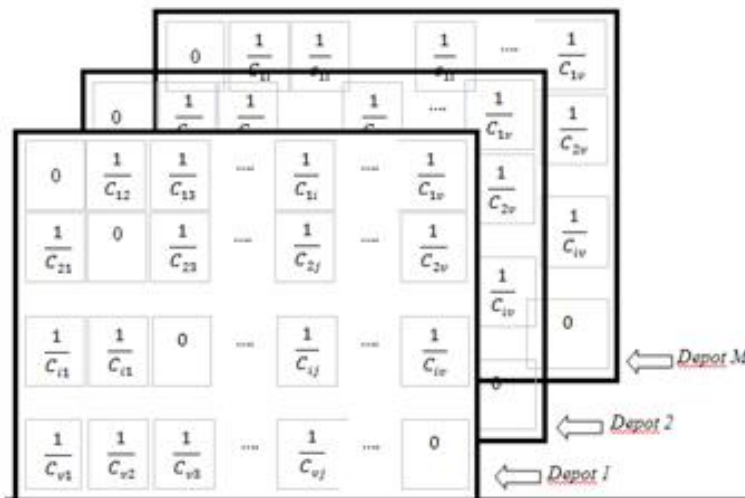


Figure 2. Encoding of the Visibility matrix

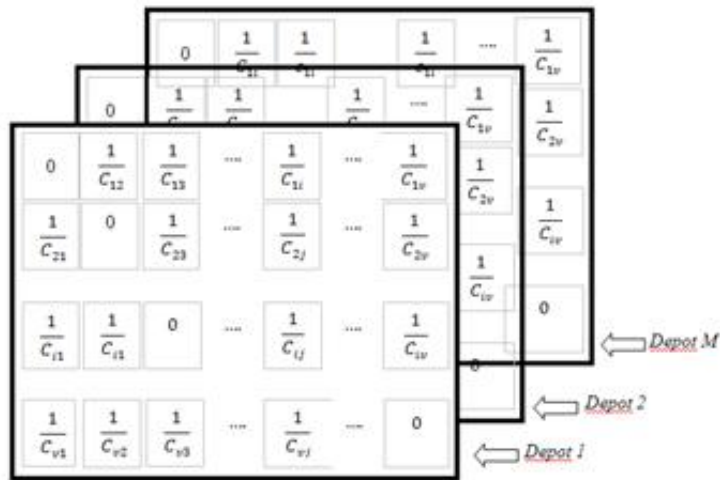


Figure 3. Encoding the pheromone matrix

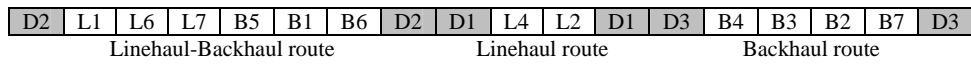


Figure 4. Encoding the solution Vector

Solution 1	D2	L4	L3	B1	B2	D2	D3	L1	L2	D3	D1	B3	D1
Solution 2	D1	L1	L2	D1	D2	L3	B2	D2	D3	L4	B1	B3	D3
Solution i	D3	L1	L2	L3	B1	D3	D1	L4	B2	B3	D1		
Solution k	D2	L2	L1	D2	D1	L3	B2	D1	D3	L4	B1	B3	D3

Figure 5. Encoding of the Matrix of the solution

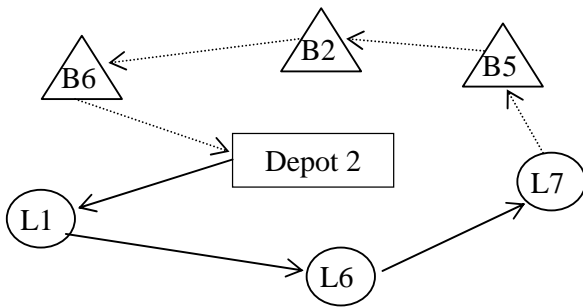


Figure 6. linehaul - backhaul route

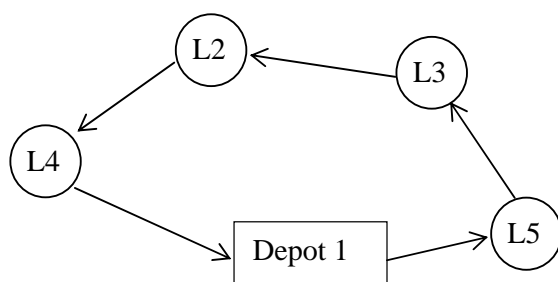


Figure 7. Single linehaul route.

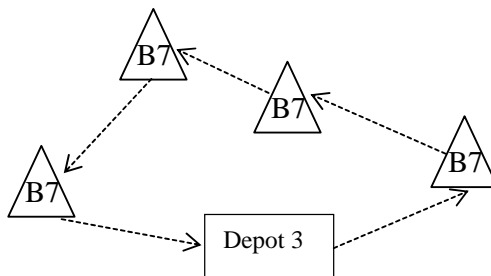


Figure 8. Single backhaul route.

The parameters used to execute instances are the following: Number of Iterations: 100, Distance parameter, $\alpha = 1$, Pheromone parameter, $\beta = 3$, Evaporation effect parameter, $\rho = 0.01$, contribution factor of the visited arc = Number of Customers / Best Solution Found.

6. CONCLUDING REMARKS

In this paper, an effective algorithm for the Multi-Depot Vehicle Routing Problem with Backhauls (MDVRPB) is proposed. The development of the proposed algorithm has been evaluated by considering benchmarking instances proposed in the literature. However comparison of the results obtained have been performed with a simple heuristic based on the nearest neighborhood due to that there is not algorithms in the literature that have considered the same variant of the problem. Computational results show the proposed algorithm is capable of producing, within short computational times, several solutions of high quality. In fact, the former algorithm exceeds the results obtained by the greedy heuristic based on the nearest neighborhood in an 19.01%. The results suggest that the proposed algorithm could be applied to other routing problems such as the Multi Depot Vehicle Routing Problem (MDVRP), the Periodic Location Routing Problem (PLRP), the Muti-Depot Vehicle Routing Problem with Heterogeneous Fleet (HMDVRP), among others.

7. REFERENCES

- [1] Toth P., Vigo D. (2002), The Vehicle Routing problem. Università degli Studi di Bologna, Bologna, Italy.
- [2] Clarke G., Wright J. (1964), "Scheduling of Vehicles from a Central Depot to a Number of Delivery Point", Operations Research, Vol. 12, pp. 568-581.

- [3] Deif I., Bodin L. (1984). "Extension Of The Clarke And Wright Algorithm For Solving The Vehicle Routing Problem With Backhauling". in A. E. Kidder, Hrsg., Proceedings of Babson Conference on Software Uses in Transportation and Logistics Management, pages 75-96, University of Maryland, College Park, MD 20742, USA, College of Business and Management Babson Park.
- [4] Salhi S., Nagy G. (1999), "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling", Journal of the Operational Research Society, Vol. 50, pp. 1034-1042.
- [5] Min, H., Current, J., Schilling, D., & Current, J. (1992). "The multiple depot vehicle routing problem with backhauling". Journal of Business Logistics, Vol. 13, pp. 259-288.
- [6] Ropke S., Pisinger D. (2006). "A unified heuristic for a large class of vehicle routing problems with backhauls". European Journal of Operational Research, Vol. 171, pp. 750-775.
- [7] Wade, A., Salhi S., (2003). "An ant system algorithm for the mixed vehicle routing problem with backhauls", in M.G.C. Resende and J.P. de Sousa (eds.): Metaheuristics: Computer Decision-Making, Chapter 33, pp. 699-719, Kluwer.
- [8] Wade A., Salhi S. (2001). "An ant system algorithm for the vehicle routing problem with backhauls", in MIC'2001 - 4th Metaheuristic International Conference.
- [9] Chunyu, R., Wang X. (2009), "Study on hybrid genetic algorithm for multi-type vehicles and multi-depot vehicle routing problem with backhauls". in Second International Conference on Intelligent Computation Technology and Automation, 2009. Vol. 1. IEEE.
- [10] Chunyu, R., Song Z., Wang X. (2009), "Study on Single and Mixed Fleet Strategy for Multi-depot Vehicle Routing Problem with Backhauls". In International Conference on Computational Intelligence and Natural Computing, CINC'09. Vol. 1. IEEE.
- [11] Rizka R. (2013). "Multiple Depot Vehicle Routing Problem with Backhauls Menggunakan Algoritma Clark and Wright dengan 2-Opt dan Penerapannya". SKRIPSI Jurusan Matematika-Fakultas MIPA UM. In <http://karya-ilmiah.um.ac.id/index.php/matematika/article/view/24501>
- [12] Dorigo M. (1992), "Optimization, Learning and Natural Algorithms", PhD Thesis, Politecnico di Milano, Italia.
- [13] Walter J. Gutjahr. ACO Algorithms with Guaranteed Convergence to the Optimal Solution. Information Processing Letters, 2002, 82: 145-153.

Metaheuristički ACO za Rešavanje Problema Trase Vozila za Više Skladišta sa Povratnim Utovarom

Jhon Jairo Santa Chávez, Mauricio Granada Echeverri, John Willmer Escobar,
César Augusto Peñuela Meneses

Primljen (11.05.2015.); Recenziran (17.06.2015.); Prihvaćen (03.07.2015.)

Apstrakt

U ovom radu, metaheuristički algoritam zasnovan na optimizaciji kolonije mrava predstavljen je radi rešavanja problema trase vozila za više skladišta sa dostavom i prikupljanjem pošiljki. Svaka izvršena trasa sastoji se od jedne pod-trase u kojoj je vožena jedino dostava, pored jedne pod-trase u kojoj je izvršen jedino proces prikupljanja. Predloženi algoritam pokušava da pronađe najbolji redosled posete klijentima prilikom svake izvedene trase. Pored toga, predloženi pristup određuje najbolju vezu između pod-trasa dostave i prikupljanja kako bi se dobilo globalno rešenje sa minimumom putnih troškova. Efikasnost predloženog algoritma ocenjena je posmatranjem niza primera adaptiranih iz literature. Izračunati rezultati upoređeni su sa pohlepnim heurističkim algoritmom zasnovanim na principu najbližeg okruženja. Na kraju, predstavljeni su zaključci i predlozi za budući rad.

Ključne reči: kolonija mrava, povratna pošiljka, kombinatorna optimizacija, izračunata simulacija, problem trase vozila za više skladišta

Table 1. Obtained results over 10 executions with 100 iterations of the proposed algorithm

Instance Number	Instance	Greedy Neighborhood Heuristic				Proposed Methodology with ACO				Percentage Greedy - ACO
		Number of Used Vehicles	Average Distance	Computing Time (sec)	Best Solution	Number of Used Vehicles	Average Distance	Computing Time (sec)	Best Solution	
1	GJ01H	6	749,8	0,18	719,5	6	693,6	6,92	654,03	9,09%
2	GJ01Q	8	846,8	0,26	808,5	8	777,2	3,55	744,61	7,91%
3	GJ01T	9	826,6	0,27	795,2	9	782,0	7,68	760,45	4,37%
4	GJ02H	3	836,1	0,18	831,9	3	645,4	0,92	598,82	28,02%
5	GJ02Q	4	820,2	0,19	816,3	5	700,2	1,63	685,86	15,98%
6	GJ02T	5	787,5	0,20	770,3	5	665,9	2,45	662,80	13,96%
7	GJ03H	6	1081,1	0,24	1062,0	6	848,6	8,13	827,87	22,04%
8	GJ03Q	8	1066,3	0,26	1012,4	8	923,1	9,96	902,97	10,81%
9	GJ03T	9	996,0	0,26	953,7	9	896,0	6,66	874,41	8,31%
10	GJ04H	8	1422,9	0,41	1416,4	8	1245,7	1,84	1219,61	13,90%
11	GJ04Q	12	1426,0	0,41	1409,9	12	1303,6	4,69	1274,70	9,59%
12	GJ04T	14	1500,5	0,23	1489,7	14	1332,4	6,63	1288,87	13,48%
13	GJ05H	4	1519,2	0,09	1517,1	4	1028,4	0,25	1028,40	32,21%
14	GJ05Q	6	1467,0	0,20	1464,3	6	1136,5	2,34	1069,80	26,94%
15	GJ05T	7	1166,0	0,36	1155,7	7	1049,5	3,16	1027,80	11,07%
16	GJ06H	8	1440,3	0,30	1407,5	8	1155,1	9,05	1112,83	20,93%
17	GJ06Q	12	1445,0	0,44	1408,7	12	1243,0	8,10	1197,10	15,02%
18	GJ06T	14	1406,6	0,71	1373,7	14	1200,1	16,68	1165,34	15,17%
19	GJ07H	8	1342,2	0,29	1306,7	8	1143,9	11,75	1129,42	13,56%
20	GJ07Q	12	1359,4	0,60	1340,6	12	1195,3	18,97	1170,17	12,72%
21	GJ07T	14	1286,1	0,88	1262,4	14	1164,9	11,65	1140,64	9,64%
22	GJ08H	13	7078,1	0,58	6943,6	13	5081,3	24,66	5000,06	27,99%
23	GJ08Q	19	7168,1	0,69	7073,6	19	5951,4	25,29	5808,30	17,89%
24	GJ08T	22	6791,0	0,80	6756,6	22	5874,3	50,19	5779,57	14,46%
25	GJ09H	13	7341,7	0,48	7111,0	13	5208,5	16,92	5164,06	27,38%
26	GJ09Q	19	7257,6	0,67	7036,6	19	6079,8	15,56	5857,96	16,75%
27	GJ09T	22	7229,4	0,54	7117,6	22	6079,0	38,41	5893,50	17,20%
28	GJ10H	13	6830,0	0,57	6404,2	13	4911,6	38,30	4765,45	25,59%
29	GJ10Q	19	7017,2	1,91	6908,1	19	5555,0	128,75	5438,90	21,27%
30	GJ10T	22	6049,2	3,66	5964,2	22	5443,2	112,43	5384,66	9,72%
31	GJ11H	13	6950,8	1,29	6790,2	13	4930,4	84,36	4768,98	29,77%
32	GJ11Q	19	6583,1	3,12	6446,3	19	5299,4	134,72	5193,38	19,44%
33	GJ11T	22	5877,4	7,07	5616,1	22	5123,5	250,39	5039,75	10,26%
			Average		3166,4				2564,6	19,01%