A two ant colony approaches for the multi-depot capacitated arc routing problem

Ali Kansou[1], **Adnan Yassine**[1,2]

[1]Laboratoire de Mathématiques Appliquées du Havre (LMAH), Université du Havre, France (ali.kansou@yahoo.fr)
[2]Institut Supérieure d'Etudes Logistiques (ISEL), Université du Havre, France (adnan.yassine@univ-lehavre.fr)

ABSTRACT

This paper deals with the Capacitated Arc Routing Problem with Multiple Depots (MD-CARP), a variant of the classical CARP problem. The MD-CARP is NP-hard, to resolve him efficiently, two ant colony approaches (ACs) are developed. The first AC1, is a hybrid approach consisting of two stages: optimization of the tasks order for the insertion by the ant colony optimization and the insertion of tasks by an heuristic saving method. A generalization for the splitting method of ulusoy is incorporated with the ant colony optimization in the second approach, called AC2. Computational experiments carried out on instances taken from the literature prove the performance of the proposed methods.
Keywords: Arc Routing Problem, Multiple Depots, Ant Colony Optimization.

1. INTRODUCTION

The purpose of this paper is to present two ant colony approaches for an important extention of the classical capacitated arc routing problem (CARP): the Multi-Depot Capacitated Arc Routing Problem (MD-CARP) on a mixed graph. The basic MD-CARP problem is defined on an undirected graph $G = (V_n, V_d, E)$, where $V_n = \{v_1, ..., v_n\}$ is a no-depot vertex set, $V_d = \{v_{1+n}, ..., v_{n_d+n}\}$ is the set of depots, $E = \{(v_i, v_j) : v_i, v_j \in V_n, i < j\}$ is an edge set and n_d is the number of depots in G. Each edge $e = (v_i, v_j) \in E$ has a demand $q_e = q_{ij} \geq 0$ and a no-negative traversal cost $c_e = c_{ij}$. Let $R = \{e \in E : \ q_e > 0\}$; $|R| = m_r$, a subset of required edges which must be served. At each depot $v_{n+d} \in V_d$ are based a fleet of m_d identical vehicles of capacity Q. The MD-CARP problem consists of designing a set of vehicle routes on G such that: (1) each vehicle route starts and ends at the same depot, (2) each edge is visited once and exactly once by one vehicle, (3) the total demand of the route does not exceed the capacity Q and (4) the total cost of routing is minimized. In this paper, we focus on the MD-CARP problem with the following three points: (1) G is a mixed graph, then we have two sets of links. A set of arcs A which have a required and no required arcs and the set of edges E; (2) two different costs by required link (a link is an arc or edge and a required link is called task): a cost to visit a task and another cost for serving it; (3) windy edge: an edge have a different cost by each direction.

The MD-CARP arises naturally, e.g, in garbage collection, in street sweeping, in emergency etc. It is NP-hard problem since it reduces to the CARP, whenever $V_d = \{v_1\}$. Originally, the CARP was proposed by Golden and Wong [12] (1981) and we refer to the survey of Assad and Golden (1995) for more details about the CARP problem. Among the used metaheuristics, we cite the simulated annealing (Li, 1992 and Eglese, 1994), the taboo search (Belenguer, 2000; Hertz, 2000 and Brandâo [6], 2008), the scatter search of Greistorfer [13], the memetic algorithms by Prins and Lacomme [16]. Belengeur et al. [3] presente a lower and upper bounds, and a new lower bounds was proposed by Wϕhlk [25] on the mixed CARP. Recently, the algorithms based on ant colony are applied effectively on the CARP by Lacomme [17], and on the mixed CARP by Bautista [2]. There exist heuristics for the CARP with Intermediate Facilities CARPIF, we refer to the works of Ghiani et al. [10], Polacek et al. [21] based on the variable neighborhood search algorithm VNS and to the Ghiani et al. [11] for the arc routing problem with intermediate facilities under capacity and lenth restrictions CLARPIF. In 2006, Bouhafs et al. [5] proposed a combination of simulated annealing (SA) and ant colony system (ACS) for the capacitated location-routing problem CLRP, where the (SA) searching the good facility configuration and the (ACS) constructs a good routing that corresponds to this configuration. The Ant Colony Otimization (ACO) is used for the Bin Packing problem by Yalaoui et al. [26] and by Levine and Ducatelle [19], so that the order to put objects into bins is optimized by the ACO and implementing an insertion method to insert objects into bins. The good results obtained on the problem of Bin Packing motivate us to apply the same principle to MD-CARP problem. On the other hand, the approach by ants colonies was never applied for the resolution of MD-CARP problem.

In the context of nodes routing, the multiple-depot vehicle routing problem MD-VRP is very studied in the literature. Laporte et al. [18] studied a family of multi-depot asymmetrical problems. A tabu search heuristic is developed for the MD-VRP by Renaud et al. [22] and Cordeau et al. [8] which is probably the best know algorithm. A hybrid genetic algorithm for the MD-VRP is also developed by Ho et al. [15]. Amberg et al. [1] studied the undirected MD-CARP with heterogeneous vehicle fleet. Sanne Wϕhlk [24] in his thesis considered the undirected MD-CARP where the vehicles have various fixed costs that arise when the vehicles are used. She considered the MD-CARP from a theoretical point of view and she gave a mathematical model with two lower bounds for the routing cost and for the fixed cost.

Our aim in this work is to develop two simples approaches based on the ant colony optimization and capable of solving the MD-CARP. The problem formulation is presented in Section 2. The two approches AC1 and AC2 are explained respectively in the two sections 3 and 4. Computational results are presented in Section 5. At the end, a general conclusion of our results will be presented in the last section.

2. Mathematical model of MD-CARP

In this section, we give a mathematical model for the mixed MD-CARP. Our model formulation uses two binary variables: $x_{ijk} = 1$ if the node j is visited after i by the vehicle k (or route k) and equal 0 otherwise; $l_{ijk} = 1$ if the vehicle k serves the link (i,j) from i to j and equal 0 otherwise. Suppose that w_{ij} is the cost to serve a task $(i,j) \in A_r \cup E_r^1 \cup E_r^2$, where E_r^1 and E_r^2 are respectively the set of all the first and the second directions of required edges ($E_r = E_r^1 \cup E_r^2 \subseteq E^1 \cup E^2 = E$). We take $E_r \cup A_r$ the set of all required links, called tasks and we denote by K the set of all vehicles. L is the new set of links which contains all the arcs in G and L_r is the new set of tasks (required arcs to be served). Then we will take $L = E^1 \cup E^2 \cup A$ and $L_r = E_r^1 \cup E_r^2 \cup A_r$. Mathematical model:

$$\text{Min} \quad \sum_{k\in K}\sum_{(i,j)\in L_r} w_{ij} l_{ijk} + \sum_{k\in K}\sum_{(i,j)\in L} c_{ij}(x_{ijk} - l_{ijk}) \quad (1)$$

$$\text{s.t.} \quad \sum_{k\in K}(l_{ijk} + l_{jik}) = 1 \quad \forall (i,j) \in E_r \quad (2)$$

$$\sum_{k\in K} l_{ijk} = 1 \qquad \forall (i,j) \in A_r \quad (3)$$

$$\sum_{(i,j)\in L} x_{ijk} = \sum_{(j,i)\in L} x_{jik} \quad \forall k \in K, \forall i \in V_n \cup V_d \quad (4)$$

$$\sum_{w\in V_d}\sum_{j\in V_n} x_{wjk} \leq 1 \quad \forall k \in K \quad (5)$$

$$\sum_{(i,j)\in L_r} q_{ij} l_{ijk} \leq Q \quad \forall k \in K \quad (6)$$

$$x_{ijk} \geq l_{ijk} \forall k \in K, \qquad \forall (i,j) \in L_r \quad (7)$$

$$\sum_{(i,j)\in L(S)} x_{ijk} \geq l_{rsk} \quad \forall k \in K, \forall (r,s) \in L_r(S), \forall S \subseteq V_n; |S| \geq 2 \quad (8)$$

$$x_{ijk}, l_{ijk} \in \{0,1\} \quad \forall k \in K, \forall w \in V_d, \forall i,j \in V_n \cup V_d \quad (9)$$

The objective function (1) minimizes the sum of the servising cost and visiting cost without servising. Constraints (2) assure that each link edge is assigned to a single route and by unique direction. Constraints (3) require that each link arc must be assigned to a single route. (4) are the flow conservation constraints. Constraints (5) ensures that each vehicle can be used at most once from a single depot. Limits on vehicle capacity are imposed by through constraints (6). Constraints (7) ensures that each required link can not be served unless it is visited. Constraints (8) are the standard sub-tours elimination constraints, where $L_r(S) = \{(i,j) \in L_r;\ i \in S \text{ and } j \notin S\}$.

For that follows, we transform G into a graph $\Gamma = (V, W, B)$, where V is a new set of nodes. Each node p of V represents a link arc of G ($\overline{p} = 0$ is the inverse of p in this case) or it is a one direction of a link edge (the another direction for the same task is noted by $\overline{p} \in V$ which is $\neq 0$ in this case). Each task p in V has a routing cost c_p, a serving cost w_p and a demand q_p. For each node depot in V_d, we create a fictitious arc $d \in W$ ($c_d = w_d = q_d = 0$) and then we have the new set of depot $W = \{1 + 2 * m_r, ..., n_d + 2 * m_r\}$. B is a set of the news fictitious arcs that are evaluated by the costs of shortest paths calculated by Dijkstra algorithm and linking new nodes of $V \cup W$. Then for each link $(p,q) \in B$ we associate a cost $D(p,q)$.

3. An Hybrid Ant Colony System AC1 for the MD-CARP

In this approach, the MD-CARP is divided into two phases: choose tasks order phase and insertion tasks phase. We propose an hybrid approach based on the ant colony and an insertion method to solve the MD-CARP problem. The Ant Colony Optimization (ACO) was first proposed by Colorni and Dorigo ([7], 1992) and applied to the Traveling Salesman Problem (TSP). Its principle is based on a population of artificiel ants who is trying to build and improve solutions for the Combinatorial Optimization problem. Real ants communicate between them by leaving a chemical substance, called the pheromone, on the paths which they traverse. Over a path used by ants, more than there are deposed pheromones more this path becomes attractive to the other ants. Then, the quantity of the pheromone deposited on paths depends on both, the length of the paths as well as the quality of the food source found. The ant colony system is used to find the order to insert the tasks in a current solution, and the insertion heuristic is used to insert each task selected by the (ACO) in this current solution. The two phases will be tackled repeatedly until the algorithm termination condition is done.

3.1. Initialization

We represent a current solution S of MD-CARP by $S = (S_1, ..., S_{n_d})$, where S_d is a sub-solution of S which represent a route associated to the depot d, for all d from 1 to n_d. Then, S_d contain the node depot d at the beginning and the ending, and between these, the sequence tasks already inserted into the solution. I_{pd} is the insertion cost of p in the route d and i is a counter which indicates the position of a given task $S_{d,i}$ in a sub-solution S_d. The best insertion cost of p in a route d is:

$$I_{pd} = Min_{i\in\{1,...,|S_d|-1\}}[D(S_{d,i-1}, p) + D(p, S_{d,i}) + w(p) - D(S_{d,i-1}, S_{d,i})] \quad (1)$$

Initially, a solution S^0 is calculated by the best insertion method: we construct n_d sets NO^d, where each task is in NO^d if it is closer to d among all depots. For a fixed depot d^*, repeatedly, we choose a task p^* from NO^{d^*} to insert it in $S^0_{d^*}$. For that, we scan each task in NO^{d^*} and we find the best position i^* and direction p^* of p that gives $I_{p^*d^*} = Min\{I_{pd^*}, I_{\overline{p}d^*}\}$ if p is a edge task. Then we insert p^* in the route $S^0_{d^*}$ and we remove p and $\overline{p}$ from NO^{d^*}. We repeat this for each $d \in W$ until NO^d will be empty. $C(S^0)$ is the cost of this initial solution, it is calculated by applying the *split* procedure of Lacomme et al. [16] in each sub-solution S_d. Then, the current best solution S^{best} for AC1 algorithm is S^0 and the according best cost is $C^{best} = C(S^0)$.

We denote the pheromone quantity at iteration t on an arc (p,q) by $\tau_{pq}(t)$ and the visibility associated with this arc is η_{pq} which is the inverse of the travel cost between the two tasks p and q ($\eta_{pq} = \frac{1}{D(p,q)}$ $\forall (p,q) \in V \times V$). Early, at the iteration 0 of AC1 algorithm, we have $\tau_{pq}(0) = 1/C(S^0)$ for all $(p,q) \in V \times V$. We proposed a colony of n_f ants where each ant f represent the order of tasks, called $sequence$, which will be included in the associated solution.

3.2. Choose tasks order phase

In each iteration of AC1 method, we will obtain the $sequence$ of tasks to insert it in a solution S. For this, we use the ant colony system algorithm. Initially, each ant f is positioned on a task randomly and it construct $sequencef$ by successively choosing a task to insert, continuing until each task has been in the $sequencef$. So at each iteration t, each ant f chooses task q after p from a set NO^f which contains tasks of V that are not yet in the $sequencef$, using the pseudo-random rule:

$$q = \begin{cases} argmax_{r \in NO^f} \left[\tau_{pr}(t)\right]^{\alpha} \left[\eta_{pr}\right]^{\beta} & \text{if } pr \leq pr_0 \\ T & \text{otherwise} \end{cases} \tag{2}$$

With α and β are respectively the coefficient of pheromones and the coefficient of visibility, $0 < pr$, $pr_0 < 1$ and T is a task calculated by the standard probability rule:

$$P^f_{(p,T)}(t) = \begin{cases} \frac{[\tau_{pT}(t)]^{\alpha} [\eta_{pT}]^{\beta}}{\sum_{r \in NO^f} [\tau_{pr}(t)]^{\alpha} [\eta_{pr}]^{\beta}} & \text{if } T \in NO^f \\ 0 & \text{otherwise} \end{cases}$$

After the construction of each $sequencef$, the second phase is called. After that, there is the updating of pheromones using the following formula:

$$\tau_{pq}(t+1) \leftarrow \rho \tau_{pq}(t) + \sum_{f=1}^{f_e} \Delta\tau^f_{pq}(t) \tag{3}$$

with $\Delta\tau^f_{pq}(t) = \frac{1}{C^f(t)}$ if (p,q) is in $sequencef$ and equal to zero otherwise. The last formula (3) includes a term for evaporation (where ρ is the rate of evaporation, $0 < \rho < 1$) and to strengthen. f_e and $C^f(t)$ are respectively the number of elitist ants and the cost of the solution that will be obtained using the $sequencef$ of ant f at iteration t.

3.3. Insertion phase (Routing phase)

At a given iteration t of AC1, once the first phase will be finished and we have the $sequencef$ for each ant f, the second phase starts. We insert each task p by order of $sequencef$ in the associated current solution $S^f(t)$, more precisely in the sub-solution $S^f_{d^*}(t)$ where $d^* = argmin_{d \in W} \{I_{pd}\}$ (see equation 1, section 3.1). Each solution calculated by an ant undergoes a process improvement (local search). We consider the following changes in a trip or two trips: the 2-optimale procedure, move a task befor/after another, move two consecutive tasks, change a task by its inverse if possible and remove a task from a sub-solution and put it in another. Local search runs until changes are improving the solution. Then, we obtain n_f solutions and the cost of each solution S^f is calculated by applying the $split$ procedure of Lacomme et al. [16] on each sub-solution of S^f. Finaly, there is the pheromone update by (3).

4. An Ant colony method AC2 for the MD-CARP

Here, we develop another version of the ant colony optimization for solving the MD-CARP. Firstly, we generalize a route-first-cluster-second approach (Ulusoy, [23] of the single depot CARP problem) for the MD-CARP in mixed graphs and we will adapted the efficient algorithm of Lacomme et al. [16] for the single depot to the case of multi-depot.

4.1. Splitting giant solution

Here, the objective is to transforming a RPP (Rural Postman Problem) solution to a MD-CARP solution. Let S a RPP solution (no depot in S) where $t_s = (v_{i_s}, v_{j_s})$ the s^{th}-servised task of S (s from 1 to $|R|$). We construct an auxiliary directed graph $H = (N, F)$ as follows: N is a new set of nodes n_{yz} in H where each vertex n_{yz} corresponds to each sequence $t_y, ..., t_z$ of serviced tasks in S so that $\sum_{s=y}^{z} q_{i_s j_s} \leq Q$. N contains also a original node o_1 and a final node o_2. F is a new set of arcs between the nodes of N such that: F contains an arc between o_1 and every node n_{1z} evaluated by the cost $min_{d \in W} \{c_{di_1} + c_{j_z d}\} + \sum_{s=1}^{z-1} c_{i_s j_{s+1}}$, an arc between every $n_{i_y,|R|}$ and o_2 evaluated by the cost $min_{d \in W} \{c_{di_y} + c_{j_{|R|} d}\} + \sum_{s=y}^{|R|-1} c_{i_s j_{s+1}}$ and finally, an arc of cost 0 from every n_{xy} to every $n_{y+1,z}$. Implicitly, we can provid a feasible solution of MD-CARP, which respect the capacity of vehicle, by finding a shortest path $P_{o_1 o_2}$ between o_1 and o_2. In Algorithm1, we calculate least-cost of feasible solution MD-CARP, which is $C(P_{o_1 o_2}) + \sum_{s=1}^{|R|} w_{i_s}$, by generalize the algorithm of Lacomme et al. [16].

Algorithm1: $mdsplit$ algorithm of the MD-CARP

$P[0] = 0,\ Z[0] = 0,\ Z[r] = \infty\ \forall r = 1, ..., m_r;$
$\forall r = 1, ..., m_r$ do
  $s = r,\ load = 0;$
  while $s \leq m_r$ and $load \leq Q$ do
    $load = load + q_{t_s}$
    if $r = s$ then
      $cost = D(A_{t_r t_r}, t_r) + w_{t_r} + D(t_r, A_{t_r t_r})$
    else
      $cost = cost - D(A_{t_r t_{s-1}}, t_r) - D(t_{s-1}, A_{t_r t_{s-1}})$
      $+ D(A_{t_r t_s}, t_r) + D(t_s, A_{t_r t_s}) + D(t_{s-1}, t_s) + w_{t_s}$
    if $load \leq Q$ then
      if $(Z[r-1] + cost < Z[s])$ then
        $Z[s] = Z[r-1] + cost;$
        $P[s] = r - 1;$
      $s = s + 1;$
$Z[m_r]$ is final cost of the MD-CARP solution S.

4.2. The efficient algorithm

In the multi-depot split version, we will calculate a matrix A of size $(2m_r + 1, 2m_r + 1)$, where m_r is the number of tasks that must be served and each A_{pq} contains the value of the depot such that: $A_{pq} = argmin_{d \in W} \{D(d,p) + D(q,d)\}$, with p and q are two tasks in V. We can calculate $C(P_{o_1 o_2})$ in $O(|R|^2)$ without generate explicitly H. For each task t_r in a given giant solution S, we take $Z[r]$ the cost of the optimal path from o_1 to t_r and $P[r]$ its predecessor task. Each trip $t_r ... t_s = n_{rs}$

corresponds to an arc $(r-1, s)$ in H. If there is an amelioration, we update the label of s instead of storing the arc. For a fixed r, we examine each possible trip n_{rs} $(s \geq r)$ and we calculate its $cost$ and $load$ until $load$ exceeds Q or t_s reached the end of S.

4.3. A general algorithm AC2 for the MD-CARP

At each iteration of AC2, each ant designs the giant solution for all vehicles at the same time. Each ant f starts its solution from a different task t_f and the remainder of tasks is chosen according to transition rule (2) where NO^f is $R - \{t_f, \overline{t_f}\}$ at the begin. After that, each giant solution calculated is splitted into a faiseable solution to the MD-CARP by the Algorithm1 and all costs are calculated after the process improvement. Only f_e elitist ants that obtained the best solutions are take into account for the updating by (3). AC2 procedure stops when there is not improvement on the solution after several iterations or when max_{iter} number of iterations is reached.

5. Computational results

The two methods were coded in C language and tested on two sets of benchmark instances adapted for the MD-CARP problem. The basic CARP instances [4] contains only edges and the cost of passage is equal to the cost of service for each task. This lack of mixed instances MD-CARP problem pushes us to adapt the CARP instances to the MD-CARP problem by adding deposits in the next section.

table1: results on md-Golden instances

I	$InitH$	AC1	SEC_1	AC2	SEC_2
g_1	351	300	< 1	300	< 1
g_2	392	331	< 1	**329**	1.2
g_3	354	**267**	< 1	273	< 1
g_4	352	**266**	< 1	269	< 1
g_5	460	369	< 1	**364**	1.3
g_6	373	285	< 1	385	< 1
g_7	391	325	< 1	325	< 1
g_8	392	**351**	1.5	359	1.8
g_9	360	316	1.7	**314**	2
g_{10}	319	275	< 1	275	< 1
g_{11}	482	407	< 1	407	1.7
g_{12}	555	**450**	< 1	454	< 1
g_{13}	582	540	< 1	540	1
g_{14}	113	98	< 1	98	< 1
g_{15}	68	58	< 1	**56**	< 1
g_{16}	137	127	< 1	127	< 1
g_{17}	95	91	< 1	91	1
g_{18}	172	160	< 1	160	1.5
g_{19}	69	55	< 1	55	< 1
g_{20}	131	**122**	< 1	123	< 1
g_{21}	172	158	< 1	158	< 1
g_{22}	208	202	1.3	202	1
g_{23}	253	**235**	1.6	236	1.5

5.1. MD-CARP instances

The first set will be called md-Golden set. It contains the 23 CARP instances of DeArmon (the size of theses instances tends from 11 to 55 required edges). We add two depots, at the first node 1 and at the final node V for each md-Golden instance. The second set adapted to the MD-CARP problem is called md-Benavent set. It contains the 28 CARP instances of Benavent (the size of theses instances tends from 34 to 92 required edges). Theses instances can be downloaded from the Internet [4]. We add three depots, at the begin node 1 and at the two vertex $\lfloor |V|/2 \rfloor$ and $2\lfloor |V|/2 \rfloor$ for each md-Benavent instance.

table2: results on md-Benavent instances

I	$InitH$	AC1	SEC_1	AC2	SEC_2
b_{1a}	449	415	1.3	415	1.2
b_{2a}	471	368	0.8	368	1.1
b_{3a}	204	167	1	167	1.5
b_{4a}	843	**710**	3.1	748	3.2
b_{5a}	801	**702**	3	737	2.5
b_{6a}	632	**530**	3	544	2.4
b_{7a}	754	**634**	3.5	666	3.1
b_{8a}	719	**634**	3.3	663	2.9
b_{9a}	887	**772**	7.5	824	6.3
b_{10a}	918	**821**	8.7	883	7.4
b_{1b}	449	**415**	1.2	421	1.5
b_{2b}	471	**368**	1	380	1.7
b_{3b}	204	**167**	1	169	2
b_{4b}	843	**717**	3.5	764	3.2
b_{5b}	801	**697**	3.2	749	3
b_{6b}	632	**530**	2.2	547	2.7
b_{7b}	754	**634**	3.7	670	3.6
b_{8b}	719	**642**	3.1	671	3.3
b_{9b}	887	**773**	7	859	6.2
b_{10b}	918	**826**	7.6	896	6.5
b_{1c}	449	**415**	1.7	435	1.5
b_{2c}	505	**410**	1	422	1.2
b_{3c}	204	**167**	1.4	174	1.5
b_{4c}	843	**710**	3	770	3.3
b_{5c}	801	**716**	3	756	2.9
b_{6c}	684	**595**	1.8	612	2.5
b_{7c}	778	**640**	2.9	666	3.5
b_{8c}	739	**695**	2.5	722	3.2
b_{9c}	887	**776**	6	825	6.6
b_{10c}	918	**836**	7.3	903	7.2
b_{4d}	857	**754**	3.1	759	3.4
b_{5d}	841	**749**	2.4	795	3.3
b_{9d}	911	**796**	6	846	6.7
b_{10d}	964	**887**	6.5	931	7.2

The notations used in the two tables of results are: I, which means an instance; g_i is the i^{th} md-Golden instance for $i = 1, ..., 23$; b_{ik} is the (i^{th}, k) md-Benavent instance for $k = a, b, c$ or d and $i = 1, ..., 10$; $InitH$ means the algorithm used in the initialization method for the two methods AC1 and AC2 (see section 3.1); SEC_1 and SEC_2 are respectively the computational time in seconds of the two methods AC1 and AC2.

The parameters of the AC1 for the MD-CARP problem are: On the md-Golden instances $max_{iter} = 150$, $n_f = 10$, $f_e = 5$, $\alpha = \beta = 1$, $\rho = 0.9$ and $pr = 0.5$. On the md-Benavent instances $max_{iter} = 170$, $n_f = 15$, $f_e = 7$, $\alpha = \beta = 1$, $\rho = 0.9$ and $pr = 0.4$.

The parameters of the AC2 for the MD-CARP problem are:

On the two sets of instances $max_{iter} = 200$, $n_f = 15$, $f_e = 10$, $\alpha = \beta = 1$, $\rho = 0.8$ and $pr = 0.5$.

5.2. Results on the md-Golden and md-Benavent instances

Both algorithms are initialized by the same heuristic method $InitH$ (cf. the procedure in the section 3.1). AC1 continues by using an heuristic of insertion (cf. section 3.3) and AC2 continues by using the splitting ant colony method (cf. section 4). We notice a strong improvement of the optimal solution compared to the initial solution given by a very known heuristic method in the literature ([22], [8], [15], [9]). The value of the best solution of the methods (ACs) is indicated in bold characters in the two above tables.
On the Golden instances (table 1), both methods AC1 and AC2 are almost equivalents. More precisely, AC1 gets 6 best solutions on the isntances g_3, g_4, g_8, g_{12}, g_{20} and g_{23} compared with those obtained by AC2, against by AC2 we obtained 4 best solutions on the instances g_2, g_5, g_9 and g_{15}, and on the remaining 13 instances both methods come to get the same results and at the same time of calculation which is very small (≤ 1.6 second).
On the other hand, on the Benavent instances (table 2), AC1 always marks its superiority concerning the quality of the obtained solution. Then, the $mdsplit$ procedure on the MD-CARP problem is very efficient on the instances of small sizes (md-Golden instances) and it becomes less efficient when the size of the instance increases (md-Benavent instances).

Then several factors indicate the efficiency and the robustness of our approaches: 1) our methods provide a significant improvement on solutions obtained by an insertion heuristic, effective and widely used in literature ([22], [8], [15], [9]); 2) we applied both methods on the CARP instances where $n_d = 1$, i.e. we have a single depot in each instance, we find the same results obtained by other methods developed in the literature of CARP problem ([14], [20], [16], [25], [6]).

6. Conclusion

In this work, we treated a variant of the CARP problem in arc routing, the Multi-Depot CARP. Our approach was motivated by two reasons: 1) it is the first time when we use this approach for resolve the MD-CARP problem, 2) its good obtained results by applying it to the Bin Packing problem. To resolve effectively this problem, two new methods based on the ant colony optimization AC1 and AC2 were developed. The numerical simulations show the efficiency of these two methods and the superiority of the algorithm AC1.

At present we implement an exact method to resolve the MD-CARP problem. This work is going to allow us to estimate the deviation of our solutions to the optimal soltuions.

REFERENCES

[1] Amberg A., Domscheke W., Voβ S., "Multiple center capacitated arc routing problem: A tabu search algorithm using capacitated trees", 124, 360-376 (2000).

[2] Bautista J., Fernandez E., Pereira J., "Solving an urban waste collection problem using ants heuristics", Computers and Operations Research, 35(9), 3020-3033 (2008).

[3] Belenguer J.M., Benavent E., Lacomme P., Prins C., "Lower and upper bounds for the mixed capacitated arc routing problem", Computers and Operations Research, 33(12), 3363-3383 (2006).

[4] Belenguer J.M., "Directory of CARP instances", http://www.uv.es/ belengue/carp.html, (1997).

[5] Bouhafs L., Hajjam A., Koukam A., "A combination of simulated annealing and ant colony for the capacitated location-routing problem", KES (1), 409-416 (2006).

[6] Brandô J. and Eglese R., "A Deterministic Tabu Search Algorithm for the Capacitated Arc Routing Problem", Computers and Operations Research, 35(4), 1112-1126 (2008).

[7] Colorni A., Dorigo M., Maniezzo V., "Distributed optimization by ant colonies", Proceedings of the first European, Conference on artificial life. Cambridge, USA: The MIT Press, 134-142 (1992).

[8] Cordeau J.F., Gendreau M., Laporte G., "A tabu search heuristic for the periodic and Multi-depot vehicle routing problems", Networks, 30, 105-119 (1997).

[9] Crevier B., Cordeau J.F., Laporte G., "The multi-depot vehicle routing problem with inter-depot routes", 176, 756-773 (2005).

[10] Ghiani G., Guerriero F., Laporte G., Musmanno R., "The capacitated arc routing problem with intermediate facilities", Networks, 37(3), 134-143 (2001).

[11] Ghiani G., Guerriero F., Laporte G., Musmanno R., "Tabu search heuristics for the arc routing problem with intermediate facilities under capacity and lenth restrictions", Journal of Mathematical Modelling and Algorithms, 3, 209-223 (2004).

[12] Golden B.L. and Wong R.T., "Capacitated arc routing problems", Networks 11, 305-315 (1981).

[13] Greistorfer P., "A tabu scatter search metaheuristic for the arc routing problem", Computers and Industrial Engineering, 44(2), 249-266 (2003).

[14] Hertz A., Laporte G., Mittaz M., "A Tabu search heuristic for the capacitated arc routing problem", Operations Research, 48, 129-135 (2000).

[15] Ho W., George T.S. Ho, Ji P., Henry C.W. Lau, "A hybrid genetic algorithm for the multi-depot vehicle routing problem", Engineering Applications of Artificial Intelligence, 21(4), 548-557 (2008).

[16] Lacomme P., Prins C., Ramdane-Chérif W., "Competitive Memetic Algorithms for Arc Routing Problems", Annals of Operations Research, 131, 159-185 (2004).

[17] Lacomme P., Prins C. and Tanguy A., "First Competitive Ant Colony Scheme for the CARP", Research Report LIMOS/RR-04-21, (2004).

[18] Laporte G., Nobert Y., Taillefer S., "Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems", Transportation Science, 22, 161-172 (1988).

[19] Levine J. and Ducatelle F., "Ant Colony optimization and local search for bin packing and cutting stock problems", Journal of Operational Research Society, 55, 705-716 (2004).

[20] Mazzeo S., Loiseau I., "An Ant Colony Algorithm for the Capacitated Vehicle Routing Problem", Electronic Notes in Discrete Mathematics, 18, 181-186 (2004).

[21] Polacek M., Doerner K.F., Hartl R.F., Maniezzo V., "A variable neignborhood search for the capacitated arc routing problem with intermediate facilities", (2007).

[22] Renaud J., Laporte G., Boctor F.F., "A tabu search heuristic for the multi-depot vehicle routing problem", Computers and Operations Research, 23(3), 229-235 (1995).

[23] Ulusoy G., "The Fleet Size and Mix Problem for Capacitated Arc Routing", European Journal of Operational Research, 22, 329-337 (1985).

[24] Wϕhlk S., "Contributions to Arc Routing", PhD thesis, University of Southern Denmark, 2005.

[25] Wϕhlk S., "New lower bound for the Capacitated Arc Routing Problem", Computers and Operations Research, 33(12), 3458-3472 (2006).

[26] Yalaoui A., Chu C., "Optimisation par Colonies de fourmis hybride : Decoupe à deux dimentions 2BP/O/G", MOSIM$^{\prime}$06-Rabat-Maroc, (2006).