Ant Colony System for the Periodic Capacitated Arc Routing Problem

Ali KANSOU* & Adnan YASSINE**

*Laboratoire de Mathématiques Appliquées du Havre (LMAH), Université du Havre, France
25 rue Philippe Lebon, B.P. 540, 76058 LE HAVRE cedex

*Institut Supérieur d'Etudes Logistiques (ISEL), Université du Havre, France
ali.kansou@yahoo.fr ; adnan.yassine@univ-lehavre.fr

Abstract

In this paper, we present a new Ant Colony System (ACS) approach for the Periodic Capacitated Arc Routing Problem (PCARP) on a mixed network. In the CARP problem the objective is to find routes starting from the depot such that each required arc/edge (called task) is serviced, capacity constraints are satisfied and total travel cost is minimized. The PCARP extends the classical CARP problem to a planning horizon of several days. Each task requires a certain number of visits within this time horizon. Hence, the objective of the problem is to choose the visit days for each task and to solve a CARP for each day in order to find efficient routes over the whole horizon. In our approach, we suggest a new technique for solving the PCARP problem. We have combined the ACS algorithm with a heuristic for insertion of tasks in a current solution. Hence, the ACS is used to optimize the order of tasks for insertion and the heuristic is used for insertion the tasks. The results of computational experiments carried on problem taken from literature indicate that the proposed approach outperforms existing algorithms in most cases. A comparison with the best memetic methods is presented and the results prove the robustness, the rapidity and the performance of our method.

Keywords: *Periodic Capacitated Arc Routing Problem, Ant Colony System, Insertion Heuristic.*

1 Introduction

The Periodic Capacitated Arc Routing Problem (PCARP) generalizes the Capacitated Arc Routing Problem (CARP) by extending the single work period (day) to a planning period H of several days of work. The well know NP-hard problem CARP is introduced by Golden et al. in [10]. In this paper, the CARP is defined on a mixed network M in which a fleet of identical vehicles, with limited capacity, is based at a depot node. Each link (arc or edge) in M has a non-negative traversal cost and a non-negative service cost which are different and can be traversed at any time. A subset of required links, called tasks (a task has a demand strictly positive), must be serviced by one single vehicle. Then, the Mixed CARP (MCARP) consists of determining a set of feasible vehicle trips that minimizes the total cost of traversed links. Each vehicle trip starts and ends at the depot and the total demand serviced by any vehicle must not exceed the vehicle capacity. In this paper, the treated problem is the Periodic Mixed Capacitated Arc Routing Problem (PMCARP). Each task t must be serviced f_t times over H in such a way that each time in a one single day and by one single vehicle in this day. The objective to PMCARP is to minimize the total distance travelled over all days of H. This problem is also NP-hard and arise in several applications like municipal waste collection, street sweeping, winter gritting, mail delivery, etc.

The first methodological paper devoted to a periodic arc routing problem was realized by Ghiani et al. in [9]. It presents a heuristic method for the periodic rural postman problem. Chu et al. proposed a linear programming model and constructive heuristics to solve the PCARP. The scatter search method is presented in [3]. Three heuristics methods are presented by Chu et al. in [4] to resolve this problem. The memetic algorithm is proposed by Lacomme et al. in [12] where the two decisions of planning (affectation of tasks to the days of H) and scheduling, are simultaneously treated. The principle of this memetic algorithm is based on a sophisticated crossover operator. The Periodic Vehicle Routing Problem PVRP is pioneered by Beltrami and Bodin in 1974 and several heuristics (Russell and Gribbin in [16], Chao et al. in [2]) and metaheuristics (see Cordeau et al. in [7], Vianna et al. in [17], and Hemmelmayr et al. in [11]) were used to solve this problem. All those methods were considered as multilevel combinatorial optimization problem, with the exception of Beltrami and Bodin procedures and the memetic algorithms [12]. In the first level an appropriate day is allocated to every task and the second level is dedicated to the resolution of the classic VRP for the subset of tasks by taking into account every day of H.

The Ant Colony Optimization (ACO) is a metaheuristic succesfully used for several combinatorial problems. Its principle is inspired from the behavior of ants who cooperate in search of food. The ACO method is applied to the PVRP [14] using two phases. The first one is the construction of a good infeasible solution to PVRP by ACO model, and the second phase is devoted to convert this solution into a feasible one. Another technique for using of ACO algorithm on the Bin packing problem with two dimentions (2BP/O/G) was presented by Yalaoui and Chu in [18] and by Levine and Ducatelle in [13]. The principle of this technique is the combination of ACO method and the SHF-FF heuristic (Ben Messaoud et al. 2003). The ACO algorithm is used to choose the object candidate to put it in the truck and the heuristic is used to place the object. We propose a similar hybrid approach which combines the ACO method to choose such a task with an insertion method, called Insertion In Best Combination $IIBC$. This insertion heuristic combines the idea for insertion tasks in the days service combination and the idea of choosing the task for insertion (Chu et al. in [4], Lacomme et al. in [12], ...).

The remainder of this paper is organised as follows: In Section 2, we describe the PMCARP problem and the notations used in this work. The Section 3 is devoted to the presentation of our insertion heuristic $IIBC$. In Section 4, we develop a new hybrid method based on the Ant Colony Optimization. A numerical analysis and experimental results are shown in Section 5. The general conclusions of our work will be presented in the Section 6.

2 Problem description

The academic CARP problem focuses on servicing edges of an undirected network graph. Our goal in this paper is to solve the periodic CARP problem based on more realistic network G, with the following three points: (1) G is a mixed graph, comprises two types of links, arcs and edges; (2) two distinct costs per link, i.e. a *deadheading* cost (whether the link is traversed without treatment) and a *treatment* cost; (3) windy edge: if at least one of these two costs depends on the direction. Consider a network $G = (N, E \cup A)$, where N is a vertex set including a depot (node 0), E is a set of edges (i, j) $(i, j \in N)$ including a subset E_r of required edges and A is a set of arcs (i, j) $(i, j \in N)$ including a subset A_r of required arcs. A vehicle fleet is given, all vehicles have identical capacity Q and are stationed at the depot. Each link $(i, j) \in E \cup A$ has a *deadheading* cost c_{ij} and a demand $d_{ij} \geq 0$. We call each link in $R = E_r \cup A_r = \{(i, j) \in E \cup A; d_{ij} > 0\}$ by task, where $|R| = m_r$ and each task (i, j) has a *treatment* cost w_{ij} ($w_{ij} \neq w_{ji}$ if $(i, j) \in V_e$).

We transforme G into a graph $\Gamma = (V, L)$, where each edge is remplaced by two opposite arcs. V is a novel set of nodes remplaced $E \cup A$ and we add a fictitious arc 0 around node 0 which represents the depot. V is composed of two other sets V_a and V_e, where V_a is a set of a_r required arcs identified by indexes from 1 to a_r and V_e is a set of $2 * e_r$ arcs (the two directions of each required edge) identified by indexes from $a_r + 1$ to $a_r + 2 * e_r$. For any task p in V ($|V| = r = a_r + 2 * e_r$), we denote by $inv(p) = \overline{p}$ its inverse such as $\overline{p} = 0$ if and only if $p \in V_a$, $\overline{p} \neq 0$ if and only if $p \in V_e$ (if $p \in V_e$ then $\overline{p} \in V_e$ and

$\overline{\overline{p}} = p$). L is a set of new fictitious arcs linking each two nodes in V. We create for this a square matrix D, $(r+1) \times (r+1)$, where $D(p,q)$ is the cost of the shortest feasible paths from p to q for each $(p,q) \in L$, which are calculated by Dijkstra's algorithm described in [8]. Then c_p, w_p and d_p are respectively a *deadheading* cost, a *treatment* cost and a demand for a task $p \in V$ ($c_0 = w_0 = d_0 = 0$).

Consider a discrete time horizon H of n_d days where each task $p \in V$ has a frequency f_p ($1 \leq f_p \leq n_d$) and must be treated f_p times, but at most once per day. The PCARP amounts to deciding on which days each task has to be serviced and to design a CARP trips for each day of H. The objective is to minimize the total distance travelled over H. Each task $p \in V$ has a set of allowed combinations C_p. A combination in C_p is a set of f_p possible service days.

3 The proposed heuristic $IIBC$: Insertion In Best Combination

This heuristic was used by Lacomme et al. [12], Chu et al. [4], etc. on the PCARP problem. They used a criteria $ChooseTask$ to choose each task to be included in a solution as a descending order of frequency tasks. In our heuristic $IIBC$, the $ChooseTask$ procedure used is very similar to that used by Paletta in [15] and by Bertazzi et al. in [1] on the PTSP problem (Periodic Traveling Salesman Problem). It makes a task p chosen for insertion, independently of the choice of combination by the following rule:

$$p = argmax_{p \in NO}\big(f_p.Min_{q \notin NO}(D(p,q) + w_p)\big) \tag{1}$$

where NO is the set of tasks in V and which are not yet included in a current solution S. We represent a current solution S of PMCARP problem by $S = (S_1, ..., S_{n_d})$, where S_d is a sub-solution of S that represents a route associated to day d. After choosing a task p by $ChooseTask$, the procedure $ChooseCombination(p)$ is capable to make a combination $k \in C_p$. This procedure was used in [12], [4], etc... on the indirected PCARP problem. Then, we insert p in current solution by $InsertionTask(p,k,S)$ procedure. In the last step, we apply the $Split()$ procedure made by Lacomme et al. in [12] on each sub-solution S_d to get the different trips of each day d in H and then a feasible solution S of PMCARP problem. The general algorithm used for this heuristic $IIBC$ is explained in **Algorithm 1.**

Each task existing in solution S is represented by $S_{d,i}$ where i is its position in the sub-solution S_d. Then, the best cost of inserting p in position i in day d of k is:

$$I_{pdk} = Min_{i \in \{1,...,|S_d|-1\}}\big(D(S_{d,i-1}, p) + D(p, S_{d,i}) + w_p - D(S_{d,i-1}, S_{d,i})\big) \tag{2}$$

We denote by I_{pk} the best insertion cost of p in k, such as $I_{pk} = \sum_{d \in k} Min\{I_{pdk},\ I_{\overline{p}dk}\}$ if $p \in V_e$ and $I_{pk} = \sum_{d \in k} I_{pdk}$ otherwise. The existing of w_p in (1) and (2) is justified to treat the case of windy edge. Depending on the situation of p we have two forms of procedure $ChooseCombination()$, the first is the case if $p \in V_e$ and the second case is done where $p \in V_a$. The following algorithm illustrates the case where $p \in V_e$ (in the other form, we remove the vector of directions VS and the cost $I_{\overline{p}dk}$):

Procedure $ChooseCombination(p)$**:**

1. Let VP and VS respectively a vector of positions and directions of p, of size f_p,
   Let k^* which will be contain the value of combination to make, $I = infini$,
2. For all $k \in C_p$ repeat:
   a) $I_{pk} = 0$, $counter = 0$
   b) for all $d \in k$ repeat:
      find the best direction and position of $\{p, \overline{p}\}$ in d,
      $I_{pk} = I_{pk} + Min\{I_{pdk}, I_{\overline{p}dk}\}$, $counter = counter + 1$
   c) if $I_{pk} < I$ then
      $I = I_{pk}$, $k^* = k$,
      update VP and VS by the values of the latest and best positions and directions of p.
3. End $ChooseCombination(p)$ and recovery k^*.

After that we apply $InsertionTask(p, k, S)$ procedure, we insert a task p chosen by $ChooseTask$ in the combination k^* of C_p, i.e. in the f_p days composing k. Then for each $d \in k^*$ we insert $VS[i]$ in $VP[i]$ of S_d and we increment i from 0 to $f_p - 1$ with d which follows an ascending order. We call a null solution, a solution that contains only the deposit in the begin and the end of each sub-solution.

Algorithm 1. Outline of the $IIBC$ heuristic.

1. $NO \leftarrow V$, $S = (S_1, ..., S_{n_d})$ a null solution, $counter = 0$
2. While $counter < m_r$ repeat:
   a) $p \longleftarrow ChooseTask$,
   b) $k^* \longleftarrow ChooseCombination(p)$,
   c) Apply $InsertionTask(p, k^*, S)$,
   d) $NO \leftarrow NO - \{p, \overline{p}\}$ if $p \in V_e$ and $NO \leftarrow NO - \{p\}$ if $p \in V_a$,
   e) $counter = counter + 1$,
3. Apply $Split(S)$ and recovery S the final feasible solution of PMCARP problem.

4 The proposed hybrid approach $ACOBC$ to $PMCARP$ problem

In this approach, the PMCARP is divided into two phases: choose tasks order phase and insertion tasks phase. We propose an hybrid approach based on the Ant Colony Optimization ACO and an insertion method to solve the PCARP problem. The ACO is a metaheuristic proposed by Colorni and Dorigo ([6], 1992) to solve the travelling salesman problem. It is inspired by the ability of ants to find the shortest path between their nests and food sources. Naturally, ants deposited chemical quantity, called pheromones, on the tracks they borrowed. Pheromones are visible by ants and therefore more a road is used by ants there are more deposited pheromones and then it becomes more attractive to other ants. The ACO is used to find the order to insert the tasks in a current solution, and the $IIBC$ heuristic is used to insert each task choosed by ACO in this current solution.

We proposed a colony of n_f ants where each ant f represents the order of tasks, called $sequencef$, which will be included in the associated solution S_f. We note by $\tau_{(i,j)}$ the amount of pheromone filed between two tasks i and j, $\eta_{(i,j)}$ is the visibility measur which is the inverse of the travel cost between i and j. At the iteration 0 of $ACOBC$ algorithm (see **Algorithm 2.**), we have $\tau_{(i,j)} = 1/C(S_0)$ for all $(i, j) \in V \times V$ where S_0 and $C(S_0)$ are respectively the initial solution calculated by $IIBC$ heuristic and the associated cost.

The first phase is called at each iteration, where each ant f is positioned on a task randomly and it constructs $sequencef$ by successively choosing a task to insert, continuing until each task has been in $sequencef$. f chooses the task j after i from a set NO^f which contains the tasks that are not yet in $sequencef$, using the following rule of probability:

$$j = \begin{cases} argmax_{r \in NO^f} \left[\tau_{(i,r)}(t)\right]^{\alpha} \left[\eta_{(i,r)}\right]^{\beta} & \text{if } q \leq q_0 \\ J & \text{otherwise} \end{cases} \tag{3}$$

With $0 < q,\ q_0 < 1$ and J is a task calculated by the following standard probability:

$$P^f_{(i,J)}(t) = \begin{cases} \frac{\left[\tau_{(i,j)}(t)\right]^{\alpha}\left[\eta_{(i,j)}\right]^{\beta}}{\sum_{r \in NO^f}\left[\tau_{(i,r)}(t)\right]^{\alpha}\left[\eta_{(i,r)}\right]^{\beta}} & \text{if } \ j \in NO^f \\ 0 & \text{otherwise} \end{cases}$$

After the construction of all $sequencef$ the second phase is called. For each task p in $sequencef$ (by order of $sequencef$) we will obtain a combination k using $ChooseCombination(p)$ procedure and we apply $InsererTask(p, k, S_f)$. After that, we will obtain n_f giants solution S_f (capacity constaints not satisfied on each day of H).

At each iteration, the $Improvement$ process (local search) is applied. With a probability p_r, we improve a solution S obtained by an ant using five procedures: 1) Change each task $p \in V_e$ by its inverse $\overline{p}$; 2) Change the current combination for each task p by another combination in C_p; 3) Exchange

combinations of two tasks p and q who have the same frequency and two different combinations used by them in S; 4) Exchange positions tasks p and q who are in the same S_d of S for each $d \in H$ and finaly 5) Remove a task p from its position and put it elsewhere in the same S_d of S and for every day $d \in H$.

Then, we apply the $Split()$ procedure of Lacomme et al. [12] to obtain n_f feasible solutions and the cost of each solution S_f is calculated by the same procedure. Finaly, there is the pheromone update by (4). Suppose that we have $u^f_{(i,j)}(t) = 1$ if (i,j) is in $sequencef$ at iteration t and equal to 0 otherwise, then the rule for global updating pheromones is given as follows:

$$\tau_{(i,j)}(t+1) \longleftarrow \rho\tau_{(i,j)}(t) + \sum_{f=1}^{n_e} \Delta\tau^f_{(i,j)}(t) \quad \text{with} \quad \Delta\tau^f_{(i,j)} = u^f_{(i,j)}(t)\frac{1}{C^f} \tag{4}$$

where ρ is the evaporation rate of pheromones ($0 < \rho < 1$), n_e represents the number of elitist ants and C^f is the cost of obtained solution S_f using $sequencef$ for insertion phase. The second term represents the amount of pheromones added to the next iteration between tasks for the n_e best ants.

Algorithm 2. Outline of the $ACOBC$ method.

1. Find $S_0 \leftarrow S_{best}$ an initial solution by $IIBC$ heuristic, $C^{best} \leftarrow C(S_0)$ its cost,
   Initialize $\tau_{(p,q)} = 1/C^{best}\ \forall\ (p,q) \in L$, n_f empty ants, $iter = 1$,
2. While $iter \leq iter_{max}$ repeat:
   a. for all $f = 1,\ ...,\ n_f$ do
      a.1 $NO^f \leftarrow V$, prepare a null solution S_f
      a.2 choose randomly p by f, $k \leftarrow ChooseCombination(p)$ and apply $InsererTask(p,k,S_f)$,
      a.3 $NO^f \leftarrow NO^f - \{p,\overline{p}\}$ if $p \in V_e$ and $NO^f \leftarrow NO^f - \{p\}$ if $p \in V_a$
      a.4 while NO^f is not empty
         a.4.1 choose a task q for f by (3)
         a.4.2 $k \leftarrow ChooseCombination(q)$ and apply $InsertionTask(q,k,S_f)$,
         a.4.3 $NO^f \leftarrow NO^f - \{q,\overline{q}\}$ if $q \in V_e$ and $NO^f \leftarrow NO^f - \{q\}$ if $q \in V_a$
      a.5 apply process $Improvement$
      a.6 apply $Split(S_f)$ and calculate $C(S_f)$
      a.7 if $C(S_f) < C^{best}$ then $S_{best} \leftarrow S_f$ and $C^{best} \leftarrow C(S_f)$
   b. do update pheromones by (4) after finding the n_e best ants
3. End While and recovery S_{best} the final solution achieved with the cost C^{best}.

So the ACO algorithm chooses each task to insert in a current solution and the $IIBC$ heuristic chooses the combination and the suitable positions and directions of the task by $ChooseCombination()$ and by $InsertionTask()$ we insert it in each day and position selected in the current solution. The optimization of the order of insertion of tasks by $IIBC$ is assured then by the Ant Colony.

5 Computational results

Our method has been implemented in C language and tested on a 1.4 GHz Pentium 4 with windows 98. Detailed results are given in **Table 1.** for the 23 periodic instances of Golden manufactured by Lacomme et al. (see [12], 2005) which contain basic graphs for the PCARP problem. To prove its effectiveness we compared his results with those of the better memetic methods, noted $BestMA$ and presented in [12]. The parameters of the $ACOBC$ for the problem are: $iter_{max} = 185$, $\alpha = \beta = 1$, $\rho = 0.9$, $n_f = 20$, $n_e = 10$, $q_0 = 0.5$, $p_r = 0.75$, $n_d = 5$ and $n_c = 24$ (n_c is the total number of combinations). In **Table 1.**, the columns Nom, n and n_s concern respectively a name of instance, a size of vertex set, and a sum of frequencies (number of task occurrences in a solution).

Three methods in [12] are applied on Golden instances of the PCARP problem which are called PMA, DMA and $IPMA$. The values in column $BestMA$ in **Table 1.** are the best values obtained by all three methods and after changing all the parameters on each forum of these 23 instances of Golden, for example by increasing the number of cross-over applied. Then in terms of the costs achieved by our

solutions with the $ACOBC$ method, we managed to beat the best method $BestMA$ in 13 instances (the costs are marked in bold in the **Table 1.**). Note also that our method providing the initial solution is more efficient than the BIH method (see [12]) in most cases.

Table 1. Results of our hybrid method on mixed periodic instances of Golden.

Nom	n	e_r	Q	n_s	BIH	$IIBC$	$BestMA$	$ACOBC$	$Time$
G_1	12	22	5	65	1109	1080	890	**886**	7
G_2	12	26	5	76	1339	1129	1030	**1007**	10
G_3	12	22	5	61	1147	926	794	**757**	7
G_4	11	19	5	52	1117	973	858	**806**	5
G_5	13	26	5	75	1454	1247	1097	1121	13
G_6	12	22	5	67	1282	1054	1001	**979**	7
G_7	12	22	5	65	1209	1162	905	909	7
G_8	27	46	27	143	1447	1370	1000	1206	43
G_9	27	51	27	155	1318	1194	934	1098	57
G_{10}	13	23	35	65	925	854	743	**720**	10
G_{11}	22	45	50	133	1501	1403	1172	1207	49
G_{12}	12	25	10	67	1669	1525	1218	1371	9
G_{13}	10	28	41	81	1813	1745	1597	1667	11
G_{14}	7	21	21	64	350	351	298	**292**	6
G_{15}	7	21	37	64	200	182	182	**176**	5
G_{16}	8	28	24	85	452	406	372	**370**	9
G_{17}	8	28	41	85	299	281	267	**263**	11
G_{18}	9	36	37	106	574	585	502	523	32
G_{19}	8	11	27	30	195	203	173	**173**	9
G_{20}	11	22	27	63	455	393	374	**363**	6
G_{21}	11	23	27	101	609	552	525	**519**	20
G_{22}	11	44	27	129	663	623	598	606	38
G_{23}	11	55	27	165	798	743	690	712	75

The running times require some remarks. The last column $Time$ in the above table presents the execution time needed to obtain solutions by our method $ACOBC$ (Time in second). The average of time on all instances required by $ACOBC$ is almost 0.34 minutes. But the average time by the three methods genetic algorithm [12] are respectively: $Time(PMA) = 2.98$ minutes, $Time(DMA) = 4.70$ minutes and $Time(IPMA) = 7.78$ minutes. The average for the method $BestMA$ is almost 748.7 minutes and this is justified by the very large number of cross-over applied and the change of lots of other parameters. So our method is much faster, because in our simple method, we apply a certain number of times $iter_{max}$ the two steps $ChooseCombination()$ and $InsertionTask()$ of the $IIBC$ heuristic and the duration of each time is a lot smaller than a second.

Hence, our approach is very simple but very effective and quick. The robustness of $ACOBC$ method is even assured, because unlike the $BestMA$ method, we used the same parameters on all instances to get our results.

6 Conclusion

We proposed a new hybrid approach $ACOBC$ based on the Ant Colony Optimization ACO combined with an insertion heuristic $IIBC$ for the Periodic Capacitated Arc Routing Problem on a mixed network. The problem is interesting and the applied methodology looks appropriate. This paper shows that the $ACOBC$ is able to provide high quality solutions to the PCARP problem with a very limited computing time and a very efficient manner. The presented results demonstrate the good performance, rapidity and robustness of the $ACOBC$ compared to memetic algorithms approach $BestMA$ of Lacomme et al. [12]. Our approach has found thirteen new best solutions compared to the solutions obtained in [12] on the 23 instances of Golden.

References

[1] L. Bertazzi, G. Paletta, M. Grazia Speranza, *An improved heuristic for the period traveling salesman problem*, Computers and Operations Research (2004) Vol. 31, No. 8, pp. 1215–1222.

[2] I. M. Chao, B.L. Golden, and E. A. Wasil, *An improved heuristic for the period vehicle routing problem*, Networks (1995) 26: pp. 25–44.

[3] F. Chu, N. Labadi, C. Prins, *A scatter search for the periodic capacitated arc routing problem*, European Journal of Operational Research (2006) Vol. 169, No. 2, pp. 586–605.

[4] F. Chu, N. Labadi and C. Prins, *Heuristics for the periodic capacitated arc routing problem*, Journal of Intelligent Manufacturing (2004) Vol. 16, No. 2, pp. 243–251.

[5] F. Chu, N. Labadi and C. Prins, *The Periodic Capacitated Arc Routing Problem, Linear Programming Model, Metaheuristic and Lower Bounds*, Journal of Systems Science and Systems Engineering (2004) Vol. 13, No. 4, pp. 423–435.

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

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

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, MA (1990).

[9] G. Ghiani, R. Musmanno, G. Paletta and C. Triki, *A heuristic for the periodic rural postman problem*, Computers and Operations Research (2005) Vol. 32, No. 2, pp. 219–228.

[10] B.L. Golden, R.T. Wong, *Capacitated arc routing problems*, Networks 11 (1981), pp. 305–315.

[11] V. C. Hemmelmayr, K. F. Doerner, R. F. Hartl, *A variable neighborhood search heuristic for periodic routing problems*, European Journal of Operational Research (2007).

[12] P. Lacomme, C. Prins and W. Ramdane-Chérif, *Evolutionary algorithms for periodic arc routing problems*, European Journal of Operational Research (2005).

[13] J. Levine and F. Ducatelle, *Ant Colony optimization and local search for bin packing and cutting stock problems*, Journal of Operational Research Society (2004) Vol. 55, pp. 705–716.

[14] A. C. Matos and R. C. Oliveira, *An Experimental Study of the Ant Colony System for the Periodic Vehicle Routing Problem*, Springer-Verlag Berlin Heidelberg (2004).

[15] G. Paletta, *The period traveling salesman problem: a new heuristic algorithm*, Computers and Operations Research (2000) Vol. 29, No. 10, pp. 1343–1352.

[16] R. A. Russell and D. Gribbin, *A multiphase appraoch to the period routing problem*, Networks (1991) 21: pp. 747–765.

[17] D. Vianna, L. Ochi, L. Durummond, *A parallel hybrid evolutionnary metaheuristic for the period vehicle routing problem*, Proc.IPDPS (1999).

[18] A. Yalaoui, C. Chu, *Optimisation par Colonies de fourmis hybride : Decoupe à deux dimentions 2BP/O/G*, MOSIM'06-Rabat-Maroc (2006).