4

A Hybrid Ant Colony System Approach for the Capacitated Vehicle Routing Problem and the Capacitated Vehicle Routing Problem with Time Windows

Amir Hajjam El Hassani, Lyamine Bouhafs and Abder Koukam
Systems and Transportation Laboratory
University of Technology of Belfort-Montbliard,
France

1. Introduction

The Vehicle Routing Problem (VRP) is a class of well-known NP-hard combinatorial optimisation problems. The VRP is concerned with the design of the optimal routes, used by a fleet of identical vehicles stationed at a central depot to serve a set of customers with known demands. In the basic version of the problem, known as a Capacitated VRP (CVRP), only capacity restrictions for vehicles are considered and the objective is to minimize the total cost (or length) of routes.

The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW), which is a generalization of the CVRP, is one of the most studied variants of the VRP. In the CVRPTW, the vehicles must comply with constraint of time windows associated with each customer in addition to capacity constraints.

The study of the VRP is very important. The VRP contributes directly to a real opportunity to reduce costs in the important area of logistics. Logistics can be roughly described as the delivery of goods from one place (supplier) to others (consumers). Transportation management, and more specifically vehicle routing, has a considerable economical impact on all logistic systems.

Due to the nature of the problem, it is not viable to use exact methods for large instances of the VRP. Therefore, most approaches rely on heuristics that provide approximate solutions. Some specific methods have been developed to this problem. Another option is to apply standard optimization techniques, such as tabu search, simulated annealing, constraint programming, genetic algorithms and ant systems.Our main interest is about the metaheuristics used to solve the VRP and more particularly about the ant colony system.

The first algorithm based on the ant colony system, applied to the CVRP, was proposed by [Bullnheimer & al. 1999] known as « Ant System » (AS), applied first for the TSP in [Dorigo & al. 1996]. The pheromone and the nearest neighbor heuristic are used to build the routes of the vehicles. To improve the routes, a heuristic of 2-OPT is combined with the AS. This algorithm was tested on 40 benchmark but the performances are inferior to those relative to the tabu search algorithm. Yet, in terms of time accomplishment, the AS is a serious rival for the existing metaheuristics.

Source: Vehicle Routing Problem, Book edited by: Tonci Caric and Hrvoje Gold, ISBN 978-953-7619-09-1, pp. 142, September 2008, I-Tech, Vienna, Austria

The same authors [Bullnheimer & al. 1999] proposed an improved AS which consists of replacing the nearest neighbor heuristic, in the transition rule of the basic algorithm (AS), by the Savings heuristics of [Paessens 1988]. The same process used in their first algorithm is applied for the pheromone updating rule. At the end of each iteration, a local search heuristic (2-OPT) is applied to improve the paths/routes. A second local search heuristic is used in the choice of clients. For each client, the distances are sorted out in an increasing manner ; the list of clients not yet visited. The tests confirm an improvement in the results in comparison to the first version of the algorithm of [Bullnheimer & al. 1999]. However, the algorithm is still limited ; the best solutions published are rarely achieved.
The algorithm proposed by [Doerner & al. 2002] is almost identical to the algorithm proposed by [Bullnheimer & al. 1999]. The only différence is that the algorithm of [Doerner & al. 2002] combines the AS with the Savings heuristic of [Clarke & Wright 1964]. The results obtained from this algorithm do not improve in a significant manner those of the previous approaches.
Contrasting to all the methods based upon "Ant System", we have proposed an approach, [Bouhafs & al. 2004], based on the new version of the « Ant Colony System» [Dorigo & Gambardella 1997] proposed to improve the «Ant System» and namely for the problems with large number of instances.
In this chapter, we propose an improvement of our approach [Bouhafs & al. 2004] by integrating an algorithm of Simulated annealing to the pheromone updating rule of the Ant Colony System.
In Section 2 we present the formulation of the CVRP. Section 3 deals with the description of the algorithm proposed to solve the CVRP. Section 4 describes the experimental results for the CVRP. Section 5 presents the formulation and description of the algorithm proposed to solve the CVRPTW. Section 6 describes the experimental results for the CVRPTW and Section 7 concludes the works.

2. CVPR formulation

The CVRP can be represented as a weighted directed graph $G = (V, A)$ where $V = \{v_0, v_1, v_2, ..., v_n\}$ represents the set of the vertices and $A = \{(v_i, v_j) : i \neq j\}$ represents the set of arcs.
The vertex v_0 represents the depot and the others represent the clients. To each arc (v_i, v_j) a non negative value d_{ij} is associated. This value corresponds to the distance between the vertex v_i and the vertex v_j in terms of cost or time between the two vertices. A demand q_i and time service δ_i $(q_0 = 0, \delta_0 = 0)$ are associated with each client (vertex) v_i.
In this case, the objective is to minimize the total cost of routing and at the same time respect the following constraints: (1) Every client is visited exactly once by exactly one vehicle , (2) all the vehicles paths/routes start and end at the depot, (3) the total demand of clients of each path/route should not exceed the capacity of each vehicle.
The number of vehicles is supposed to be illimited ; it is calculated during the construction of the routes of vehicles.
The figure 2.1 shows the graphic representation of a CVRP example.

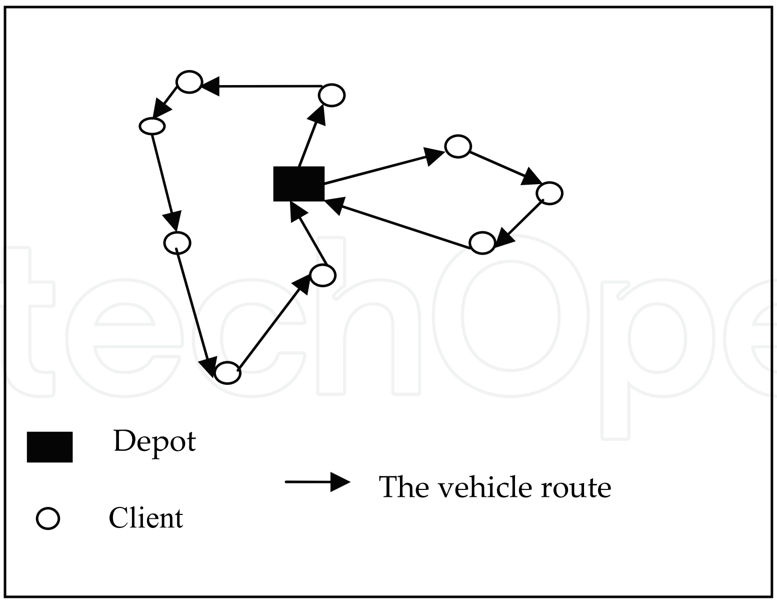


Figure 2.1 Example of CVRP solution

Here, we give the CVRP formulation found in literature. But, before that, we describe the list of variables that will help us in the formulation :

- $G = (V, A)$
- $V = \{v_0, v_1, v_2, ..., v_n\}$ where v_0 represents the depot and $v_1, ..., v_n$ all the clients
- q_i the demand of the client i, $i \in V$
- d_{ij} the distance (cost) between the vertices(clients) v_i and v_j
- $K = \{k_1, k_2, ..., k_m\}$ represents the vehicles fleet
- Q the capacity of each vehicle $k_i \in K$ (the fleet is homogeneous)

In order to find the clients visit order, we define the decision variables as follows :

$$x_{i,j}^k = \begin{cases} 1 & \text{if the vehicle k visits the client j directly after the client } i \\ 0 & \text{otherwise} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{if the client } i \text{ is served by the vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

The objective function is

$$Min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{i,j}^k \tag{2.1}$$

with the constraints

$$\sum_{k \in V} \sum_{j \in N} x_{i,j}^k = 1, \quad \forall i \in V \tag{2.2}$$

$$\sum_{j \in V} x_{i,j}^k - \sum_{j \in V} x_{j,i}^k = 0, \ \forall i \in V, k \in K \tag{2.3}$$

$$\sum_{j\in V} x_{0,j}^{k} = 1\,,\ \forall k \in K \tag{2.4}$$

$$\sum_{j\in V} x_{j,n+1}^{k} = 1\,,\ \forall k \in K \tag{2.5}$$

$$x_{i,j}^{k} = 1 \Rightarrow y_i - q_j = y_j\,,\ \forall i, j \in V, k \in K \tag{2.6}$$

$$y_0 = Q,\ \ 0 \le y_i\,,\ \forall i \in V \tag{2.7}$$

$$x_{i,j}^{k} \in \{0,1\}\,,\ \forall i, j \in V, k \in K \tag{2.8}$$

The function of the euclidean cost solution $X = \left(x^{k}{}_{ij}\right)\ \ \forall i, j \in V, k \in K$ is defined by :

$$Cost(X) = \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} d_{ij} x_{i,j}^{k} \tag{2.9}$$

The number of vehicles used by the solution X is defined by

$$\text{NB vehicles } (X) = \sum_{k\in K}\sum_{i\in V} x_{0,j}^{k} \tag{2.10}$$

3. A hybrid algorithm based on ant colony system and annealing for the capacited vehicle routing problem (CVRP)

Our algorithm is based on a hybrid ACS with an algorithm of Savings and a local search (2-opt). In the transition rule, we introduce the Savings heuristic that allows an enrichment of the search and to calculate the utility to combine 2 clients in the same route or to put them in two different paths/routes. The 2-opt is applied at each definition of the routes by the ants, juste before the global updating rule of pheromones. The simulated annealing intervenes in the phase relative to the updating of the pheromones where two rules are applied as in « the Ant Colony System » (ACO) introduced in [Dorigo & Gambardella 1997]. Before developing the different steps of the algorithm, we explain first the 2 heuristics 2-opt and Savings used in our approach.

3.1 The 2-OPT heuristic

The principle of the 2-OPT is to delete 2 arcs of the same route and to replace them by 2 other arcs in order to improve the cost of this route and to delete the road junctions.
In the figure 3.1, the two scattered arcs are deleted in the left graph. They are replaced by the two arcs labeled (A and B) in the right graph in order to find a new route. The direction of some arcs in directed graphs, that are not concerned by a 2-OPT operation, can be modified during the construction of the new route.

3.2 The savings heuristic

This heuristic was proposed by [Clarke & Wright 1964] and improved by [Paessens 1988]. It is the basis of most of the commercial software used to solve the vehicle routing problems in the industrial applications. The objective of this heuristic is to determine whether it is better to combine the clients v_i and v_j in the same route (when the value of γ_{ij} is big) or to put them in two different routes. The Savings value of the clients v_i and v_j is calculated as follows :

$$\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.\left|d_{i0} - d_{0j}\right|$$

where

- d_{ij} represents the distance between the vertice i and the summit j ,
- The index 0 corresponds to the depot
- g and f represent 2 parameters of the heuristic.

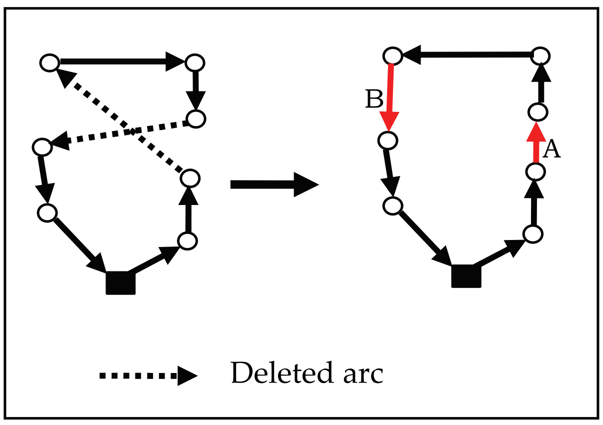


Figure 3.1 The heuristic principle of local search 2-OPT

3.3 Description of the algorithm

3.3.1 Construction of the routes/paths

The algorithm ACS of [Dorigo & Gambardella 1997], presented for the first time to solve the TSP, takes into consideration the pheromone and the heuristic of the nearest neighbor for the construction of the routes. Our approach ACS hybrid, presented here, takes into consideration the Savings heuristic as well for such construction.

Initially in the ACS hybrid, m ants are positioned on all the graph vertices and a quantity of initial pheromone is applied on the arcs. Each ant takes its departure from the depot to visit the clients. Each client is visited once and only once by an ant, however, the depot can be visited several times. If the load stored by the ant exceeds the vehicle constraint capacity, the ant must return to the depot. We then get a complete route for a vehicle. When an ant goes back to the depot, it starts from scratch again. It initializes another route to visit other new clients. This operation is repeated over and over again until all clients are visited. This means that a solution to the Capacitated Vehicle Routing Problem (CVRP) has been found.

During the process of building a route, the ant modifies the quantity of pheromone on the chosen arc by applying a local updating rule. Once all the ants are done with the building of their routing, the quantity of pheromone on the arcs belonging to the best routing found is updated according to the global updating rule.

The rule used for the construction of the routes is described hereafter. An ant k, positionned on a vertice i, chooses the next vertice j to visit by applying the probabilistic rule $p_k(i, j)$ given in the equations (3.1) and (3.2).

$$j = \begin{cases} \arg\max\limits_{u \in F_k(i)} \left\{ \left(\tau_{iu}\right)^{\alpha} . \left(\eta_{iu}\right)^{\beta} . (\gamma_{iu})^{\lambda} \right\} & \text{if } q \leq q_0 \\ J & \text{if } q \succ q_0 \end{cases} \tag{3.1}$$

J is a random variable generated according to the function of distribution given by :

$$p_{ij} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha} . (\eta_{ij})^{\beta} . \ (\gamma_{ij})^{\lambda}}{\sum\limits_{u \in F_k(i)} \left(\tau_{iu}\right)^{\alpha} . \left(\eta_{iu}\right)^{\beta} . \ (\gamma_{iu})^{\lambda}} & \text{if } u \in F_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where

- $F_k(i)$ is the list of the vert not yet visited by the ant k positioned at the vertex i,
- q is a random variable that follows a uniform distribution on $[0,\ 1]$,
- q_0 is a parameter $(0 \leq q_0 \leq 1)$ that determines the relative importance of the exploitation versus the exploration. Before an ant visits the next vertex, q is generated randomly. If $q \leq q_0$, the exploitation is then encouraged, otherwise the process of exploration is encouraged,
- τ_{ij} is the quantity of pheromone associated to the arc (i, j)
- η_{ij} is the heuristic of visibility that is the inverse of the distance between the vertices i and j,
- $\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.\left|d_{i0} - d_{0j}\right|$ is the heuristic of Savings where f and g are two parameters.

α, β and λ are three parameters that determine the relative importance of the pheromone, the distance and Savings respectively.

3.3.2 Pheromones updating

The originality in our algorithm is to introduce the simulated annealing in the phase of the pheromones updating. While an ant is building its solution, the level of pheromone on each arc (i, j) visited is modified according to the following updating local rule :

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\Delta\tau_{ij} \tag{3.3}$$

where

ρ is the parameter of evaporation $(0 < \rho < 1)$

$\Delta\tau_{ij} = \tau_0$ is the quantity of initial pheromone.

Once each ant has defined its tours, a 2-opt local search is applied in order to improve the vehicles routes. This heuristic is applied at the end of each iteration and just before the global updating rule of the pheromones.

In the algorithm ACS, the ant allowed to lay down a quantity of pheromones to guide the search is the one that has found the best solution.

In our algorithm ACS, each ant that finds a solution $S^{'}$ where

$\Delta S(= Cost(S^{'}) - Cost(S)) \prec 0$ or $\exp(-\Delta S / T) \succ \mu$ is given permission to lay down a quantity of pheromone.

where

- S is the best current solution
- μ is a random variable that follows a uniform distribution on $[0,1]$
- $Cost(S)$ is the cost of the solution S,
- $\exp(-\Delta S / T)$ is the criteria of Metropolis used for the simulated annealing
- T parameter that represents the temperature in the simulated annealing

The integration of this choice criterion in the global updating rule permits to accept some solutions which are not inevitably better than the current solution, and hence to diversify the search.

The overall updating rule is given in the equation (3.4)

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\Delta\tau_{ij} \quad (3.4)$$

Where ρ is the parameter of evaporation of the pheromone $(0 < \rho < 1)$

If the arc (i, j) is used by an ant whose solution is accepted (meets the criteria defined above), the quantity of pheromone is then increased on this arc by $\Delta\tau_{ij}$ that is equal to $1/L^{*}$ with L^{*} is the length of tours found by this ant.

4. Experimentations and results

After the description of the ACS hybrid algorithm, we evaluate its performance by considering a set of instances relative to the CVRP problem. The algorithm is coded in java language and executed on a laptop PC, equipped with Windows XP system, a P4 processor at a 2.4 GHz speed, a RAM of 512 Mo. Our experimental study was about a set of instances Of different sizes. These instances are detailed in the Table 4.1. They can be grouped in 3 categories according to their originators. instances (A, B et P) : [Augerat & al. 1995], instances E : [Christofides & Eilon 1969], instances C : [Christofides & al. 1979].

For each instance, we give the number n of clients and the capacity Q of a vehicle.

4.1 The algorithm parameters adjustment

We initialize the algorithm parameters by the following values :

- The number of ants $m = 10$ initially placed at random on all the summits,
- $\alpha = 1$, $\beta = 2$ with α , β and λ are three parameters that determine respectively the relative importance of the pheromone, of the distance and of Savings,
- $\rho = 0.1$ is the parameter of evaporation of the pheromone used in updating rules of the pheromones,
- $f = g = 2$ with f and g the two parameters of the heuristic of Savings,

- $q_0 = 0.75$ is the parameter that determines the relative importance of the exploitation versus the exploration. Before that an ant visits the next summit, q is generated randomly. If $q \le q_0$ the exploitation is then encouraed, otherwise it is the exploration process that is encouraged,
- $\tau_0 = 10^{-6}$ represents the initial quantity of the pheromone put down on all the arcs.
- $T = 100$ represents the initial value of the temperature,
- Initially the parameter of the temperature T is T_0 $(T \leftarrow T_0)$,
- $\theta = 0.9$, at each iteration, the temperature is modified by the formula $(T \leftarrow \theta * T)$.

Instance	n	Q	Instance	n	Q
B-n68-k9	68	100	C1	50	160
B-n78-k10	78	100	C2	75	140
B-n66-k9	66	100	C3	100	200
B-n50-k8	50	100	C4	150	200
B-n57-k9	57	100	C5	199	200
B-n63-k10	63	100	C11	120	200
B-n67-k10	67	100	C12	100	200
P-n51-k10	51	80	A-n63-k9	63	100
P-n76-k5	76	280	A-n63-k10	63	100
E-n76-k10	76	140	A-n69-k9	69	100
E-n76-k15	76	100	A-n80-k10	80	100
A-n60-k9	60	100			

Table 4.1 details relative to the intances of tests (n : number of clients, Q : capacity of a vehicle)

4.2 Tests results

In this paragraph, we present a study of the tests performed of the ACS hybrid algorithm we have proposed. The tests results for the instances detailed in the table 4.1 are given in the table 4.2. The first column of the table represents the name of the instances dealt with. The second column specifies the best solution known found by the heuristics (Dist : distances & Nbv : Number of vehicles). The column with the header H_ACS gives the solution found by our approach. Finally, the last column gives the deviation of our solution in comparison with the best solution published found by other heuristics. It is calculated by the following formula :

$$deviation = \frac{Solution(H\_ACS) - Solution(best\ known)}{Solution(best\ known)} * 100$$

From table 4.2, we note that the results found by our approach have a deviation average of (0.022%) in comparison with the best solutions known regarding all the tests. What explains this point is that the solutions found by our ACS hybrid are very close to the best solutions published so far. Moreover, our method allows even an improvement of certain solutions relative to some instances (B-n68-k9, B-n78-k10, B-n66-k9, A-n60-k9, P-n51-k10).

We also draw a comparison between our approach and that of [Bullnheimer & al. 1999] to evaluate the quality of our solution and the CPU time. The tests of comparison are done on the benchmarks of [Christofides & al. 1979], and presented in the table 3.1 (C1, C2, C3, C4, C5, C11, C12). At the initialization of this approach, N ants are placed on the graph summits, N represents thus the number of clients. For each instance of the problem, the algorithm is executed $2*N$ iterations.
The results of comparison are given in table 4.3. The first column of the table represents the name of the instances dealt with. The column "best publ." represents the best solution known in literature. The third column specifies the best solution found by AS of [Bullnheimer & al. 1999] (COST: cost of the solution, CPU: computing time in seconds and Dev : deviation in comparison with the best solution known). The column with the header H_ACS gives the solution found by our approach after 500 iterations for each instance.

Instance	Best known		H_ACS		deviation
	dist.	nb v.	dist.	nb v.	
B-n68-k9	1304	9	**1300.91**	9	-0,23%
B-n78-k10	1266	10	**1238.28**	10	-2,19%
B-n66-k9	1374	9	**1371.67**	9	-0,17%
B-n50-k8	1313	8	1317.34	8	0,30%
B-n57-k9	1598	9	**1598.0**	9	0%
B-n63-k10	1537	10	1540.71	10	0,19%
B-n67-k10	1033	10	1035.46	10	0,19%
A-n60-k9	1408	9	**1357.70**	9	-3,57%
A-n63-k9	1634	9	1636.94	9	0,18%
A-n63-k10	1315	10	1322.93	10	0,60%
A-n69-k9	1168	9	1177.36	9	0,80%
A-n80-k10	1764	10	1787.05	10	1,30%
P-n51-k10	745	10	**743.26**	10	-0,23%
P-n76-k5	631	5	631.0	5	0%
E-n76-k10	832	10	836.37	10	0,52%
E-n76-k15	1032	15	**1030**	15	-0,19 %

Table 4.2 CVRP experimentations results

Instance	**best publ.**	AS **Bullnheimer et al. 1999]**			**H_ACS**		
		cost	cpu(s)	dev.	cost	cpu(s)	dev.
C1	524.61	524.61	6	0%	524.61	2	0%
C2	835.26	844.31	78	1.08%	838.86	20	0.43%
C3	826.14	832.32	228	0.75%	834.77	35	1.04%
C4	1028.42	1061.55	4048	3.22%	1035.23	80	0.66%
C5	1291.45	1343.46	5256	4.03%	1304.25	200	0.99%
C11	1042.11	1065.21	552	2.22%	1046.81	32	0.45%
C12	819.56	819.56	300	0%	819.56	30	0%

Table 4.3 Test of comparison between the ACS hybrid algorithm and the AS algorithm of [Bullnheimer & al. 1999]

Through Table 4.3, we deduct that the average deviation of our approach (0,51%), in comparison with the best solutions known, is smaller than the approach of [Bullnheimer & al. 1999] (1.61%). This proves that our approach gives better performanes. Moreover, the solutions are found in far less computing time than [Bullnheimer & al. 1999]. However, we are unable to make a direct comparison in terms of computing time because the approaches are tested on different machines.
To sum up, our approach finds solutions of good qualities within reasonable computing time.

5. The CVRPTw (The Capacitated Vehicle Routing Problem With Time Windows)

The CVRPTV is one of the most studied variants of the vehicle routing problems. We can say that the CVRPTW is a generalization of the CVRP. In fact, in addition to the capacity constraints introduced in the CVRP, the vehicles must respect the constraints of time windows associated with each client.

5.1 The CVRPTV formulation

As the CVRP, the CVRPTW can be represented by a weighted directed graph $G = (V, A)$, where

- $V = \{v_0, v_1, v_2, ..., v_n\}$ represents the whole set of vertices,
- $A = \{(v_i, v_j) : i \neq j\}$ represents all the arcs between the vertices.

The vertex v_0 represents the depot and the others represent the clients. For each arc (v_i, v_j) is associated a non negative value d_{ij} that corresponds to the distance between the vertex v_i and the vertex v_j that is the cost between the two vertices.
Each client (vertex) v_i has

- A demand q_i $(q_0 = 0)$
- A time window, which means that each client *i* has a predefined time with an earliest arrival time e_i (lower bound) and a latest arrival time l_i (upper bound) for the visit of vehicles.

The vehicles must arrive before the upper bound of the window l_i and if they arrive before the lower bound e_i they must wait awhile w_i (time wait). Each client imposes a service time δ_i that corresponds to the goods loading/unloading time. The time window of the depot means that each vehicle that leaves the depot at a time e_0 go back to the depot before the time l_0. We suppose that the vehicles fleet is homogenous and that the capacity of each vehicle is Q.
The objective function of the CVRPTW is to minimize the total cost of routing (the distance or the total time traveled) while respecting the constraints concerning the CVRP as well as those of time windows. The graphical representation is the same as the CVRP.

5.2 ACS improved for the CVRPTW

Our algorithm for the CVRPTW is an algorithm of hybrid optimization. It is based on ACS and local searches. In addition of the heuristic of Savings introduced for the CVRP, we use a new local search in order to take into account informations specific to the CVRPTW. We name this heuristic a wait heuristic that favors the clients choice whose wait time is smaller. It is calculated by the following formula :

$$\omega_{ij} = 1/w_j \tag{5.1}$$

and

$$w_j = \begin{cases} e_j - t_j & if \quad (e_j - t_j) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{5.2}$$

where

- e_j is the time of arrival as earliest to the client j (Time Windows lower bound of client j)
- t_j is the actual arrival time to client j

Once the ants find routes, we improve them by a local search heuristic $2-opt^*$ whose objective, contrasting to the 2-opt heuristic that tries to improve one route, is to improve 2 routes in the same routing. A $2-opt^*$ operation consists of deleting 2 arcs from 2 different routes (one bone from each route/path) to divide each route in two parts and exchange the second parts of the 2 routes/paths while respecting the constraints of both the time windows and the capacities (Figure 5.1).

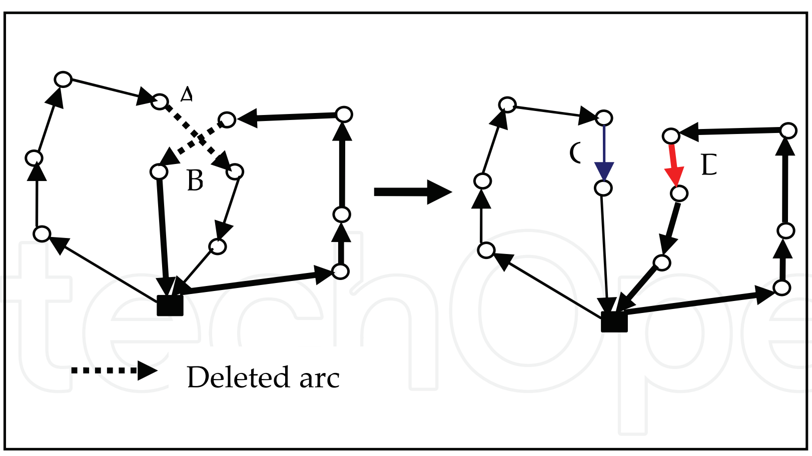


Figure 5.The local search heuristic 2-OPT *

Each ant's point of departure is the depot to visit the clients. Each client is visited once and only once by an ant, however, the depot can be visited several times. If the load stored by the ant exceeds the vehicle constraint capacity or violates the constraint of time windows of the depot, the ant must return to the depot. We then get a complete route for a vehicle. When an ant goes back to the depot, it starts from scratch again in terms of its accumulated load and time. It initializes another route to visit other new clients. This operation is repeated over and over again until all clients are visited. This means that a thorough

solution to the Capacitated Vehicle Routing Problem Time Windows (CVRPTW) has been found.
When an ant k is localized to client i, the transition rule for the client j is given as follows.

$$j = \begin{cases} \arg\max_{u \in F_k(i)} \left\{ (\tau_{iu})^{\alpha} . (\eta_{iu})^{\beta} . (\gamma_{iu})^{\lambda} . (\omega_{iu})^{\theta} \right\} & if\ q \le q_0 \\ J & if\ q \succ q_0 \end{cases} \tag{5.3}$$

J is a random variable generated according to the following function of distribution

$$p_{ij} = \begin{cases} \dfrac{(\tau_{iu})^{\alpha} (\eta_{ij})^{\beta} (\gamma_{ij})^{\lambda} (\omega_{ij})^{\theta}}{\sum_{u \in F_k(i)} (\tau_{iu})^{\alpha} (\eta_{iu})^{\beta} (\gamma_{iu})^{\lambda} (\omega_{iu})^{\theta}} & \text{if } u \in F_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

where

- P_{ij} is the probability to select the client j,
- $F_k(i)$ is all the clients not yet visited by the ant k positioned to client i,
- q is a random variable that follows a uniform law on $[0,1]$ & q_0 is a parameter $(0 \le q_0 \le 1)$,
- $(\alpha, \beta, \lambda, \theta)$ represent the parameters that have an impact respectively on the relative importance of the pheromone level, of the distance, of Savings and the wait heuristic

When an ant k goes to client j, we use the following formula to update locally the track of pheromone on the arc (i, j):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + (\rho)\tau_0 \tag{5.5}$$

As soon as an ant generates all the routes/paths (CVRPTW solution), the local search $2-opt^*$ is applied to improve the routes when possible.
Finally, an overall updating of the pheromone tracks – that consists in reinforcing the arcs related to the best solution found by one of the ants – is carried out by using the following formula:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \tag{5.6}$$

where

- ρ $(0 < \rho < 1)$ est le paramètre d'évaporation de la phéromone.
- ρ $(0 < \rho < 1)$ is the parameter of evaporation of the pheromone.

If the arc (i, j) is used by the ant that has found the best solution, the quantity of pheromone is increased on this arc by $\Delta\tau_{ij}$ that is equal to $1/L^*$ with L^* is the length of the routes found by the best ant.

6. Experimentation and results

Our algorithm has been tested on a classical set benchmark Solomon problems. Solomon has generated sets of VRPTW benchmark composed of six different problem types (C1, C2, R1, R2, RC1, and RC2) available from:
http://w.cba.neu.edu/~msolomon/home.htm, each data set contains between eight to twelve 100-node problems. The names of the six problem types have the following meaning. Sets C have clustered customers whose time windows were generated based on a known solution. Problem sets R have customers location generated uniformly randomly over a square. Sets RC have a combination of randomly placed and clustered customers. Sets of type 1 have narrow time windows and small vehicle capacity. Sets of type 2 have large time windows and large vehicle capacity. Therefore, the solutions of type 2 problems have very few routes and significantly more customers per route.
For the setting parameters we use 20 artificial ants, $\alpha = 1$, $\beta = 2$, $\lambda = 1$, $\theta = 0.75$, $\rho = 0.1$, $q_0 = 0.75$, and $f = g = 2$ (parameters of the Savings function). Table 6.1 gives the computational results for the test problems obtained by our approach.
The average deviation of our approach is 0,30% in comparison with the best solutions known. This deviation is very feeble because our algorithm finds in many cases the best value published. In addition, the solutions are found in a reasonable computing time. We believe that the modified ACS algorithm needs further refinements. In particular, the suitable values of algorithm parameters.

7. Conclusion

In this chapter, we have presented two new algorithms for the CVRP and the CVRPTW. The two algorithms are based on the hybrid ant colony with local searches. The experimental results we have displayed show the quality of solutions achieved by our approaches. The objective of this study is to show the efficiency of the ant colony algorithms for the vehicle routing problems.
This study has shown that the combining of the ant colonies with specific informations to the problem under study enables the finding of competitive results. The experimental results we have displayed show the quality of solutions achieved by our approach. The objective of this study is to show the efficiency of the hybrid ant colony algorithms for the vehicle routing problems. This study has shown us that by combining the ant colonies with the simulated annealing, competitive results are found.

8.References

Augerat P, Belenguer J M, Benavent E, Corberan A, Naddef D, Rinaldi G. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report RR 949-M, University Joseph Fourier, Grenoble (1995).

Bouhafs, L., Hajjam, A., Koukam, A.: "A Hybrid Ant Colony System Approach for the Capacitated Vehicle Routing Problem", 4th International Workshop, ANTS 2004, volume 3172 de LNCS, pages 414–415. Springer, September (2004).

Bullnheimer, B., Hartl, R. F. and Strauss, Ch.: Applying the ant system to the vehicle routing problem. In: Voss, S., Martello, S., Osman, I. H. and Roucairol, C. (Eds.): Meta-

Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston (1999).

Bullnheimer, B., Hartl, R. F. and Strauss, Ch.: An improved ant system algorithmfor the vehicle routing problem. Annals of Operations Search 89 319–328, (1999).

Christofides, N., Eilon, S.: An algorithm for the vehicle dispatching problem. Operational Search Quarterly 20 (3), 309–318, (1969).

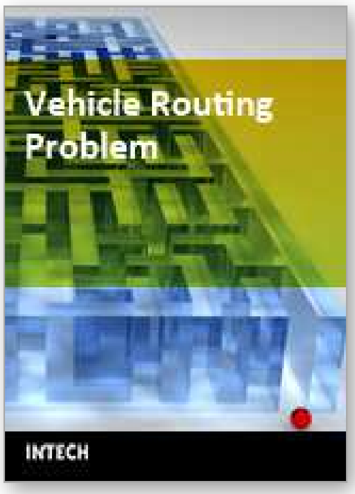
Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), Combinatorial Optimization, Wiley, Chichester, pp. 315-338, (1979).

Clark, G., and Wright, J. W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Search 14, 568-581, (1964).

Dorigo, M., Maniezzo, V., and Colorni, A.: "Ant system: Optimization by a Colony of Cooperating Agents". IEEE Trans. Sys., Man, Cybernetics 26 1, pages 29-41, (1996).

Doerner, K., Gronalt, M., Hartl, R.F., Reimann, M., Strauss, C., and Stummer, M.: "SavingsAnts for the Vehicle Routing Problem". In LNCS 2279, pp. 11—20, Springer, April (2002).

Paessens, H.: The savings algorithm for the vehicle routing problem, Eur. J. Oper. Res. 34, 336-344, (1988).



Vehicle Routing Problem

Edited by Tonci Caric and Hrvoje Gold

ISBN 978-953-7619-09-1
Hard cover, 142 pages
Publisher InTech
Published online 01, September, 2008
Published in print edition September, 2008

The Vehicle Routing Problem (VRP) dates back to the end of the fifties of the last century when Dantzig and Ramser set the mathematical programming formulation and algorithmic approach to solve the problem of delivering gasoline to service stations. Since then the interest in VRP evolved from a small group of mathematicians to a broad range of researchers and practitioners from different disciplines who are involved in this field today. Nine chapters of this book present recent improvements, innovative ideas and concepts regarding the vehicle routing problem. It will be of interest to students, researchers and practitioners with knowledge of the main methods for the solution of the combinatorial optimization problems.