**An ant colony system approach for variants of the traveling salesman problem with time windows**

Daniela Favaretto*

Elena Moretti†

Paola Pellegrini‡

*Università Ca' Foscari di Venezia*

*Dipartimento di Matematica Applicata*

*Dorsoduro 3825/E, I-30123 Venezia*

*Italy*

**Abstract**

The Traveling Salesman Problem with Time Windows has important applications in routing and scheduling and has been extensively studied in literature. In the paper, a mathematical formulation of the *temporal*–Traveling Salesman Problem with Time Windows is presented and a meta-heuristic based on Ant Colony System is proposed and implemented. Computational experience on a benchmark problem is reported and a case study is analyzed, where interesting results are obtained.

***Keywords :*** *Ant Colony System, temporal–TSPTW.*

## 1. Introduction

The Traveling Salesman Problem with Time Windows (*TSPTW*) consists in finding the minimum length circuit to be travelled by a vehicle, which must visit a set of nodes exactly once. The service at each node must begin within a specific time window. If the vehicle arrives too early, it has to wait until the window opens, while if it arrives too late the service is not possible [4].

* *E-mail*: favaret@unive.it

† *E-mail*: emoretti@unive.it

‡ *E-mail*: paolap@pellegrini.it

***Journal of Information & Optimization Sciences***

Vol. 27 (2006), No. 1, pp. 35–54

 0252-2667/06 $2.00 + 0.25

The objective of the problem is usually to minimize the total length of the tour. In this paper, the classical formulation of *TSPTW* is considered [4] and a variant of the problem is proposed, called *temporal-TSPTW*. The difference between the two problems consists in the objective functions. In the *temporal-TSPTW*, the total weighted time, traveling and waiting time, has to be minimized.

The *TSPTW* has been solved in literature both by exact algorithms [7], [8], [12] and heuristics [11], [13].

In this paper, a meta-heuristics based on Ant Colony System is proposed and implemented for both the Traveling Salesman Problem with Time Windows (*SACS*) and the *temporal*-Traveling Salesman Problem with Time Windows (*TACS*). Computational experience is presented solving instances of a benchmark proposed in literature [12], [14]. The instances of the benchmark have been tested for both *SACS* and *TACS* as will be discussed in the following. Moreover, a case study is also analyzed.

In section 2 the Traveling Salesman Problem with Time Windows is formulated also for the *temporal* variant. Section 3 deals with the approach of Ant Colony System while section 4 gives the description of the meta-heuristic proposed. Computational experience is presented in section 5 and in section 6. Finally, conclusions and hints for future research are reported in section 7.

## 2. The problem

Consider a logistic distribution problem having the following features.

- A unique vehicle has to serve a set of customers exactly once during a period $[0, T]$.
- Each customer may require to be served within a specific time window during the period $[0, T]$.
- A minimum cost path must be determined.

The problem can be formulated as a classical Traveling Salesman Problem with Time Windows (*TSPTW*).

Consider a set of customers $N = \{1, 2, \ldots, n\}$ to be visited and the additional single depot. Next, duplicate the depot into an origin depot, $o$, and a destination depot, $d$ [4].

Let $G = (V, E)$ be a graph where $V$ is the set of nodes and $E$ is the set of not oriented edges. More precisely, $V = N \cup \{o, d\}$ and $E = (\{o\} \times N) \cup (\{d\} \times N) \cup (N \times N)$, where $\{o\} \times N$ is the set of edges connecting the customers and the starting depot, $(\{d\} \times N)$ is the set of edges connecting the customers and the ending depot and $N \times N$ is the set of edges connecting the customers. The starting and the ending depots are represented with different nodes even if they are in the same location.

To each customer $i \in N$, a strong time window $[e_i, l_i] \subseteq [0, T]$ and a weight $s_i$, representing the service time to customer $i$, are associated.

To each edge $(i, j) \in E$, two weights $c_{i,j}$ and $t_{i,j}$, representing respectively the cost of travel from node $i$ to node $j$ and the time required to travel from node $i$ to node $j$, are associated.

Let $T_i$ be a variable indicating the instant of time in which the service at the customer $i \in N$ begins, and $x_{i,j}$ be a binary variable having the following meaning:

$$x_{i,j} = \begin{cases} 1, & \text{if the vehicle visits the customer} \\ & j \text{ immediately after the customer } i, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall\, i, j \in N\,.$$

The model can be formulated as follows

$$\min \sum_{i \in N \cup \{o\}} \sum_{j \in N \cup \{d\} \setminus \{i\}} c_{i,j} \cdot x_{i,j} \tag{1}$$

subject to

$$\sum_{j \in N} x_{i,j} = 1, \qquad \forall\ i \in N\,, \tag{2}$$

$$\sum_{i \in N} x_{i,j} = 1, \qquad \forall\ j \in N\,, \tag{3}$$

$$\sum_{j \in N} x_{o,j} = 1\,, \tag{4}$$

$$\sum_{i \in N} x_{i,d} = 1\,, \tag{5}$$

$$(T_j - T_i - t_{i,j} - s_i) \cdot x_{i,j} \geq 0\,, \quad \forall\ i \in N \cup \{o\},\ j \in \cup\{d\}\,, \tag{6}$$

$$e_i \leq T_i \leq l_i\,, \qquad \forall\, i \in N \cup \{o\} \cup \{d\}\,, \tag{7}$$

$$x_{i,j} \in \{0,1\}\,, \qquad \forall\ i \in N \cup \{o\},\ j \in N \cup \{d\} \tag{8}$$

$$x_{i,o} = x_{d,i} = 0\,, \quad \forall\ i \in N \cup \{o\},\ j \in N \cup \{d\}\,. \tag{9}$$

Constraint (2) requires the vehicle to leave each node exactly once and constraint (3) requires the vehicle to arrive at each node exactly once. Constraint (4) requires the vehicle to leave the depot only once and constraint (5) requires the vehicle to arrive at the depot only once. Constraints (4), (5), (6) and (9) eliminate subtours [4, p. 55], stating that it is impossible to come back to the same node. Constraint (7) ensures the schedule feasibility with respect to time windows. Binary conditions on the variables are expressed by the last constraints (8) and (9).

The objective (1) minimizes the routing cost.

The constraint (6) can be linearized as suggested in [4, p. 56].

A variant of the problem (1)-(9) is to consider a different objective function taking into account the costs deriving from the waiting times which can occur when the vehicle arrives at the customer before the opening of the time window.

Let $W_i$, $W_i \geq 0$, $i \in N$, and $C_i$, $i \in N$, be the waiting time and a weight associated to the node $i$, respectively.

The new objective function can be written as

$$\min \sum_{i \in N \cup \{o\}} \sum_{j \in N \cup \{o\} \setminus \{i\}} t_{i,j} \cdot x_{i,j} + \sum_{i \in N} C_i W_i\,. \tag{10}$$

Moreover, the constraint (6) becomes

$$(T_j - T_i - t_{i,j} - W_j - s_i) \cdot x_{i,j} = 0, \quad \forall\ i \in N \cup \{o\},\ j \in N \cup \{d\} \tag{11}$$

and the constraints

$$W_i \geq 0\,, \qquad \forall\ i \in N \tag{12}$$

have to be added.

Let's call this new problem *temporal-TSPTW*. It is easy to see that it becomes the *TSPTW*, if $C_i = 0$, $\forall\ i \in N$ and $t_{i,j} = c_{i,j}$, $\forall\ i, j \in E$ [4, p. 55], while it becomes the variant formulated by Desrsosiers et al. in [4, pp. 56-57], if $C_i = 1$, $\forall\ i \in N$. The parameter $C_i$ may express the importance of the customer $i$, $i \in N$. This parameter allows to weight in a different way the waiting time at the different customers. In fact if a customer $i$ is very important the corresponding value of $C_i$ is lower compared with the values of the parameters associated to the other customers, since it is worth while to wait to serve him as soon as possible.

## 3. Ant Colony System

Ant Colony System (ACS) studies a set of artificial ants cooperating to the solution of an optimization problem obtained exchanging information via pheromone deposited on graph edges [6].

Real ants are able to find the shortest path from a food source to their nest exploiting pheromone trail deposited by previous ants, developing what can be interpreted as collective learning. The trail of pheromone accumulates faster on the shortest path, so that eventually all ants will be going through it.

The first problem to which this method has been applied is the *TSP* [5], [6], [15] and recently an algorithm has been proposed also for the *VRP* [1], [2] and *VRPTW* [10]. This method has been applied to solve the *TSPTW* only under stochastic assumption [9].

The aim of this paper is to propose an algorithm to solve both *TSPTW* and *temporal-TSPTW* formulated in the previous section using the Ant Colony System.

According to the literature, two weights are associated to each edge: a desirability measure $\tau(i, j)$, called pheromone, which is modified at each run by artificial ants, and a heuristic measure $\tau(i, j)$, the meaning of which is related to the kind of the analyzed problem.

At the beginning, $f$ ants are located in the depot. Each ant generates a complete tour by choosing the nodes according to a probabilistic transition rule. Ants prefer to move to nodes connected by shortest edges with high amount of pheromone. When the ants move the level of pheromone, on the edges used, is modified (local updating rule). Once all the ants have completed their tours, a global pheromone updating rule is applied. Then the whole process, is repeated.

*ACS* ends when one of the following conditions becomes true: a fixed number of solutions has been generated, a fixed computational time has elapsed or a fixed number of iterations with no improvement of the objective function has been performed.

The state transition rule shows how ant $k$ in node $i$ chooses node $j$

to move to. Let $q_0$ be a fixed parameter $(0 \leq q_0 \leq 1)$ and $q$ be a random number uniformly distributed in $[0,1]$.

If $q \leq q_0$ the ant $k$ in node $i$ chooses node $j$ such that

$$j = \arg \max_{u \in J_k(i)} \{[\tau(i,u)] \cdot [\eta(i,u)]^{\beta}\}, \tag{13}$$

where

$\beta$ is a parameter which determines the relative importance of pheromone versus distance;

$\tau(i,u)$ is the level of pheromone associated to the edge $(i,u)$;

$\eta(i,u)$ is a function of the distance between $i$ and $u$;

$J_k(i)$ is the set of nodes which can be visited by the ant $k$ leaving from $i$.

If $q > q_0$, the choice of the ant $k$ in node $i$ is random. The ant $k$ in node $i$ chooses to move in node $j$ using the following probabilistic rule:

$$p_k(i,j) = \begin{cases} \dfrac{[\tau(i,j)] \cdot [\eta(,j)]^{\beta}}{\sum\limits_{u \in J_k(i)} [\tau(i,u)] \cdot [\eta(i,u)]^{\beta}}, & \text{if } j \in J_k(i), \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

It means that, if $q$ is greater than qo, the nodes have different probabilities to be chosen which are proportional to their desirability.

If $q$ is greater than $q_0$, this process is called *exploration*, otherwise it is called *exploitation*.

When $f$ hamiltonian circuits are determined, the pheromone level is modified on all edges by a global modification. In this context, only the ant that has found the shortest route deposits pheromone on the edges it went through, so that the choices of the following agents will be positively affected by those of the ant which has obtained the best solution.

## 4. The description of the meta-heuristic

The algorithms proposed to solve the problems formulated in section 2 are similar to the one proposed for *TSP* in [6] and differ from one another only for the evaluation of the objective function. They will be called respectively *SACS* (*Spatial Ant Colony System*) and *TACS* (*Temporal Ant Colony System*).

In the following, the algorithm *TACS* is described.

The first step of the procedure finds a feasible solution by an algorithm based on nearest neighbor [14]. This algorithm has been chosen because of its low complexity, even if it does not guarantee all nodes to be inserted in the tour: it may happen, owing to particular time windows, that some nodes are not included. Since the aim of this step is just providing a rough estimation of the cost of the tour, needed to fix the initial level of pheromone on edges, this problem has been faced by computing the average length of an edge used in the partial solution and adding to the total cost the length of as many edges as are the nodes not yet visited.

Let $\psi_{gb}$ be the current best solution (globally best).

The second step requires that a colony of ants is activated to find the shortest route.

The guideline of how a colony of ants works can be summarized in the following way: analyzing each node with respect to the constraints imposed by the model, each ant builds a list of feasible movements and chooses the one indicated by the probabilistic rule described in section 3. As explained before considering the first step, it is not guaranteed that all the nodes can be easily inserted in the tour; exploiting collective learning to reach this result, following Gambardella et al. [10], the best tour is, firstly, the one that reaches the highest number of nodes. Global pheromone update is made using the best solution found in this way.

In Figures 1 and 2 the algorithm is depicted.

### 4.1 *Set of feasible customers and time windows constraint*

Let $N_i$ be the set of feasible customers if the last node visited is $i$. $N_i$ will be the set of all not visited customers $j$ such that $T_i + t_{i,j} + s_i \leq l_j$. In this case, $j$ is reachable before its time window closes.

### 4.2 *Evaluation of $\eta$*

The value of $\eta(i, j)$, $i, j \in N \cup \{o, d\}$, represents not only the inverse of the time needed to go from node $i$ to node $j$, but also the urgency of serving customer $j$, given by the time interval between the present moment and the one in which his time windows closes; moreover, the number of times in which the node $j$ has not been inserted in the previous routes is taken into account.

**Figure 1**
**The algorithm**

*/* Main Procedure */*

1. */* Initialization */*

    */* $\psi_{gb}$ is the best current solution,*

    *$|N|$ is the number of nodes of the graph,*

    *$L_{NN}$ is the total travel time of the tour obtained by nearest neighbour,*

    *$L_{\psi_{gb}}$ is the total travel time of the best tour found */*

Initialize variables

Initialize $\psi_{gb}$

Start nearest neighbor

$\tau(i,j) = \dfrac{1}{|N| \cdot L_{NN}}, \forall\ (i,j) \in E$

2. */* Main Loop */*

Repeat

    For each ant $k$

        call TourBuildingProcedure

    */* $\psi_k$, $L_k$ are current solution and current travel time */*

    If #visited_customers($\psi_k$) $>$ #visited_customers($\psi_{gb}$)

        or (#visited_customers($\psi_k$)

            $=$ #visited_customers($\psi_{gb}$) and $L_k < L_{\psi_{gb}}$) Then

        $\psi_{gb} := \psi_k$

    End if

    End for each

    If #visited_customers($\psi_{gb}$) $= 0$ Then

        End

    Else

        */* Global pheromone updating */*

        If $(i,j)$ is an edge in the current best solution Then

            $\tau(i,j) := (1-\alpha) \cdot \tau(i,j) + \alpha(L_{\psi_{gb}})^{-1}$

        Else

            $\tau(i,j) := (1-\alpha) \cdot \tau(i,j)$

        End if

    End if

Until a stop criterion is met

**Figure 2**
**The tour building procedure**

```
/* Tour BuildingProcedure for the ant k */

    Initialize ψ_k
    Repeat
        1. /* Definition of compatible customers */
            For each not visited node j
                If T_i + t_{i,j} + s_i ≤ l_j Then
                    j is compatible
                    η(i, j) = 1 / max{1, δ_j(l_j − now) − IN_j^3}
                End if
            End for each
        If there are compatible customers Then
        2. /* Choice of the node to visit */
            Draw q
            If q ≤ q_0 Then
                exploitation
            Else
                exploration
            End if
            Insert j in ψ_k
        3. /* Local pheromone updating */
            τ(i, j) := (1 − ρ) · τ(i, j) + ρΔτ(i, j)
        End if
    Until no compatible customers are found
    For each node j ∉ ψ_k
        IN_j = IN_j + 1
    End for each
```

More precisely,

$$\eta(i, j) = \frac{1}{\max\{1, \delta_j(l_j - \text{now}) - IN_j^3\}} \tag{15}$$

where

now is the current time,

$\delta_j = \max\{\text{now} + t_{ij}, e_j\}$-now is the width of the time interval elapsing before the beginning of the service to customer $j$,

$IN_j$ is the number of times node $j$ has not been inserted in a tour [10],

$\gamma$ is a scale factor introduced to have homogeneous quantities.

The formula (15) for $\eta(i, j)$ is the same as the one proposed by Gambardella et al. in [10], apart from the raising to the third power of $IN_j$, which is introduced to privilege the choice of customers difficult to visit, and the presence of the coefficient $\gamma$. In the following $\gamma$ is considered equal to 1.

4.3 *Pheromone updating rules*

Pheromone updating rules are the following:

- *local updating rule*

$$\tau(i, j) := (1 - \rho) \cdot \tau(i, j) + \rho\Delta\tau(i, j),$$

with

$\rho$ parameter such that $0 < \rho < 1$,

$\Delta\tau(i, j)$ fixed equal to the initial level of pheromone [6];

- *global updating rule*

$$\tau(i, j) := (1 - \alpha) \cdot \tau(i, j) + \alpha\Delta\tau(i, j)\,,$$

with $\Delta\tau(i, j) = \begin{cases} (L_{\psi_{gb}})^{-1}, & \text{if } (i, j) \in \psi_{gb} \\ 0\,, & \text{otherwise,} \end{cases}$

$\alpha$ pheromone decay parameter, $0 < \alpha < 1$;

$L_{\psi_{gb}}$ length of the globally best tour $\psi_{gb}$.

The local updating rule does not modify the pheromone trail until a global update has been performed. Moreover, this modification allows the ants to choose different paths in the following iterations.

## 5. Computational results

The algorithm just described was coded in Visual Basic and run on a Athlon XP 1600, 1.39 GHz.

Preliminary tests have been performed for different values of $\alpha, \beta, \rho$ and $q_0$. The values tested were $\alpha \in \{0.1, 0.5, 0.7, 0.9\}$, $\beta \in \{0.5, 1, 5, 10, 20\}$, $\rho \in \{0.3, 0.5, 0.7, 0.8, 0.9\}$ and $q_0 \in \{0.8, 0.85, 0.9, 0.95\}$ and all the combinations were checked. The analysis indicates that satisfactory results can be obtained by setting:

$\alpha = 0,1$,

$\beta = 5$,

$\rho = 0,9$,

$q_0 = 0,9$,

$f = 30$,

maximum number of iterations $= 10000$, maximum

computational time allowed $= 60$ sec,

maximum number of iterations without improvement $= 55$.

Following Gendreau et al. [12], test problems were taken from the literature, namely single vehicle decomposition of Solomon's *VRPTW* instances in [14]. The travel costs and travel times between cities $i$ and $j$ correspond to the Euclidean distance separating them, using suitable coefficients to have homogeneous quantity. Results obtained both for *TSPTW* and *temporal-TSPTW*, setting $C_i = 1$, $\forall\ i \in N$, are compared with the ones reported by Gendreau et al. [12], who use an exact algorithm to solve *TSPTW*. As far as we know, in literature it is not possible to find results related to a benchmark for *temporal-TSPTW*. This is the reason why in Table 2 some results are presented which evaluate the objective function of *temporal-TSPTW* corresponding to the solution obtained by Gendreau et al. [12] for *TSPTW*. These results have been chosen as benchmark also because the two objective functions, (1) and (10), differ only from a term taking into account the total waiting time, while the other term coincides since $t_{i,j} = c_{i,j}$ is assumed.

In the following Table 1 the values of an optimal solution obtained by Gendreau et al. [12] and the values of the best solution obtained by *SACS* are reported.

In the last column the relative error of *SACS* is reported. As it can be seen, in more than one half of the cases *SACS* algorithm has found the optimal solution.

**Table 1**
**Results of *TSPTW***

| Instance | Number of nodes | Optimal solution value reported by Gendreau et al. | Solution value of *SACS* | $R_{SACS}$ |
|---|---|---|---|---|
| RC201-1 | 26 | 378.62 | 378.62 | 0 |
| RC201-2 | 29 | 374.70 | 374.70 | 0 |
| RC201-4 | 20 | 232.54 | 232.54 | 0 |
| RC202-1 | 26 | 246.22 | 256.59 | 0,0421 |
| RC202-2 | 23 | 206.53 | 214.13 | 0,0368 |
| RC202-3 | 28 | 341.77 | 341. 77 | 0 |
| RC202-4 | 26 | 367.85 | 410.56 | 0,1161 |
| RC205-1 | 27 | 251.65 | 313.54; 254.62 | 0,2459; 0,0118 |
| RC205-2 | 23 | 271.22 | 343.29 | 0,2657 |
| RC205-3 | 29 | 436.64 | 436.64 | 0 |
| RC205-4 | 25 | 361.24 | 361.24 | 0 |

In the following Table 2 comparisons between results of *temporal-TSPTW* are reported.

As it can be seen, in all cases except one, RC205-1, the solutions found by *TACS* algorithm are better than those deduced from Gendreau et al. results. For what concerns instance RC205-1, two solution values are reported, the second better than the first one. Such result was obtained postponing the beginning of the time window of the following five nodes: 98, 47, 14, 12, 5. The initial time $e_i$ of such nodes was set equal to 187, the initial instant of service of the first customer, node 69, in the solution presented by Gendreau et al. [12]. It may be that the first solution found was a local minimum; in fact also increasing the number of the ants of the colony it was not possible to jump out of it

The average relative improvement is 2,01%, considering the best value for instance RC205-1. In particular the solutions proposed reduce the average total waiting time of about 57% and increase the average total travel time of about 46%.

**Table 2**
**Results of *temporal-TSPTW***

| Instance | Solution value deduced from Gendreau et al.'s results | Solution value of *TACS* |
|---|---|---|
| RC201-1 | 821.43 | 821.02 |
| RC201-2 | 866.62 | 850.48 |
| RC201-4 | 715.95 | 699.61 |
| RC202-1 | 861.32 | 850.48 |
| RC202-2 | 726.39 | 702.28 |
| RC202-3 | 854.12 | 853.71 |
| RC202-4 | 775.04 | 771.48 |
| RC205-1 | 683.73 | 834.62; 656.84 |
| RC205-2 | 914.12 | 899.24 |
| RC205-3 | 944.66 | 908.79 |
| RC205-4 | 704.41 | 684.21 |

The instance RC201-3 has not been considered because the solution presented by Gendreau et al. is not feasible: the time window of customer 62 is violated. In fact, the service at customer 62 starts at the instant 294, while his time window closes at the instant 287.

The instances of problem RC207 presented by Gendreau et al. [12] have not been considered because the time window of customer 6 (instance RC207-2) is violated: the service starts at the instant 799,49 while his time window closes at the instant 646. The violation is very consistent, moreover also the run of the other instances RC207 have shown that sometimes it is not possible to find a feasible solution. The most probable motivation is that the clusters reported in Gendreau et al. [12] are not correct.

The average CPU time is about one second for every instance.

Another interesting remark is that in all the runs the stop criterion met is the number of iterations without improvements which have the following meaning: either the solution found is optimal, or it is not possible to jump out of a local minimum.

In Appendix 2 the solution of the instances of the benchmark are reported.

## 6. The case study

As a case study, a firm is considered which distributes door-to-door a wide number of food products in almost all regions of Italy.

The head office assigns each customer to one of the branch of the firm and decides how to split each of these areas in subareas that can be served by one agent. Each agent in this way deals always with the same customers, reaching higher levels of efficiency.

The selling strategy consists in visiting the customers and proposing them the goods in that moment available to the agent, without the need of placing any order. Agents receive a commission proportional to their volume of business.

The aim of the application of the algorithm to this case is to find an efficient way for visiting a predetermined set of customers respecting their time windows in the shortest time.

### 6.1 *The instance*

The instance solved considers 64 customers, the data of which are reported in Table 4 (see Appendix 1), which shows the cartesian coordinates referred to a suitable metric system and the time window of each customer. The node-depot is the one called $0$. The matrix of Euclidean distances is weighted by coefficient 2 to have a better approximation of the real distances, taking into account the fact that the roads are not straight lines. In any case the value 2 for the correction coefficient seems to be very conservative.

Each agent chooses by himself the starting time of his own tour, so the node-depot is considered available from 6:30 till 22:30; for all the customers that did not request a time window, the firm indicates the interval between 8:30 and 20:30.

Each vehicle has a speed of 20 Km/h, and each service requires 5 minutes.

The solution obtained is reported and compared with the result reached by the firm in Table 3.

It can be seen that the firm proposes a tour that requires 10 hours and 37 minutes, and cannot visit three customers owing to the late arrival time. Moreover, 8 time windows are violated, which means that the time constraints of about 30% of visited customers are not satisfied.

On the other hand, *TACS* algorithm gives a solution that respects all the time windows and includes all the customers, requiring a travel time of 10 hours and 23 minutes. Taking into account that the distances and travel time are overestimated, the relative improvement of the solution proposed with respect to the one chosen by the firm, can be considered highly satisfactory.

Computational time requested to reach the final result is 5 seconds.

## 7. Conclusions

It has to be pointed out that in all the considered instances the number of ants used is 30, which is high with respect to the number of nodes. In fact, the larger is the number of ants, the greater is the effect of the local updating rule of the pheromone and, then, the greater is the number of explored solutions. In this way, it may be that it is easier to jump out of a local minimum, even if of course this is not assured as it can be shown in the instance RC205-1. On the other hand, the fact that it is possible to fall in a local minimum is a typical problem of many meta-heuristics.

Since also the number of ants is one of the chosen parameters, it would be very interesting to perform a sensitivity analysis as in [3] to establish which are the values of parameters to fix. As an example, it would be very interesting to understand how the stop criteria should be chosen to guarantee a satisfactory solution.

Another very important decision is related to the construction of the data of the instance, that especially in real cases may be not so straightforward. In fact, it may be difficult to define the spatial distance between two nodes using real roads and in the same way to define the temporal distance between two nodes taking into account for example traffic and so on.

As it can be seen, the computational time is always very low; even if the size of the instances considered is generally small, nevertheless the computational performance can be considered very satisfactory.

**Table 3**
**Results reached by the firm and by the algorithm**

| Firm’s result | | | | *TACS*’s result | | | |
|---|---|---|---|---|---|---|---|
| Node | Arrival | Node | Arrival | Node | Arrival | Node | Arrival |
| 0 | 10.04 | 33 | 15.33 | 0 | 9.23 | 37 | 14.45 |
| 1 | 10.08 | 34 | 15.39 | 6 | 9.30 | 38 | 15.00 |
| 2 | 10.23 | 35 | 15.56 | 13 | 9.35 | 39 | 15.06 |
| 3 | 10.32 | 36 | 16.02 | 11 | 9.41 | 40 | 15.12 |
| 4 | 10.41 | 37 | 16.08 | 62 | 9.48 | 61 | 15.18 |
| 5 | 10.48 | 38 | 16.23 | 17 | 9.55 | 4 | 15.23 |
| 6 | 11.55 | 39 | 16.29 | 10 | 10.00 | 60 | 15.29 |
| 7 | 11.03 | 40 | 16.37 | 12 | 10.05 | 5 | 15.34 |
| 8 | 11.12 | 41 | 16.48 | 16 | 10.12 | 19 | 15.40 |
| 9 | 11.19 | 42 | 17.04 | 8 | 10.30 | 2 | 15.45 |
| 10 | 11.25 | 43 | 17.19 | 9 | 10.36 | 3 | 15.52 |
| 11 | 11.30 | 44 | 17.25 | 15 | 10.42 | 54 | 15.58 |
| 12 | 11.36 | 45 | 17.56 | 1 | 10.54 | 7 | 16.05 |
| 13 | 11.42 | 46 | 18.02 | 18 | 11.30 | 27 | 16.11 |
| 14 | 11.49 | 47 | 18.08 | 21 | 12.00 | 42 | 16.17 |
| 15 | 11.55 | 48 | 18.13 | 20 | 12.05 | 57 | 16.25 |
| 16 | 12.01 | 49 | 18.28 | 28 | 12.30 | 52 | 16.32 |
| 17 | 12.10 | 50 | 18.33 | 22 | 12.36 | 41 | 16.38 |
| 18 | 12.16 | 51 | 18.39 | 24 | 13.00 | 36 | 16.45 |
| 19 | 13.00 | 52 | 18.50 | 26 | 13.06 | 50 | 16.51 |
| 20 | 13.06 | 53 | 19.08 | 25 | 13.12 | 49 | 16.56 |
| 21 | 13.12 | 54 | 19.35 | 14 | 13.17 | 35 | 17.01 |
| 22 | 13.20 | 55 | 19.44 | 58 | 13.23 | 30 | 17.09 |
| 23 | 13.43 | 56 | 19.50 | 55 | 13.28 | 29 | 17.14 |
| 24 | 14.00 | 57 | 19.55 | 46 | 13.42 | 63 | 17.22 |
| 25 | 14.08 | 58 | 20.01 | 47 | 13.47 | 44 | 17.31 |
| 26 | 14.14 | 59 | 20.07 | 23 | 13.52 | 43 | 17.37 |
| 27 | 14.27 | 60 | 20.15 | 48 | 13.57 | 56 | 18.00 |
| 28 | 14.38 | 61 | 20.21 | 45 | 14.03 | 53 | 18.13 |
| 29 | 14.53 | 62 | not visited | 34 | 14.09 | 33 | 18.20 |
| 30 | 14.59 | 63 | not visited | 51 | 14.19 | 59 | 18.34 |
| 31 | 15.10 | 64 | not visited | 31 | 14.30 | 64 | 19.00 |
| 32 | 15.17 | | | 32 | 14.36 | | |

## 8. Appendix 1

**Table 4**
**Relevant data about the customers**

| Node | $x$ coord. | $y$ coord. | $e_j$ | $l_j$ | Node | $x$ coord. | $y$ coord. | $e_j$ | $l_j$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 3550 | 150 | 8:30 | 20:30 | 33 | 410 | 4925 | 6:30 | 22:30 |
| 1 | 2950 | 360 | 8:30 | 13:30 | 34 | 575 | 4990 | 8:30 | 20:30 |
| 2 | 2720 | 2090 | 8:30 | 20:30 | 35 | 1520 | 3220 | 8:30 | 20:30 |
| 3 | 2515 | 1525 | 8:30 | 20:30 | 36 | 1655 | 3300 | 8:30 | 20:30 |
| 4 | 2390 | 2240 | 8:30 | 20:30 | 37 | 1615 | 3370 | 8:30 | 20:30 |
| 5 | 2500 | 1970 | 8:30 | 20:30 | 38 | 2765 | 2160 | 15:00 | 16:00 |
| 6 | 2290 | 2285 | 9:30 | 10:30 | 39 | 2525 | 2065 | 15:00 | 16:00 |
| 7 | 2220 | 1820 | 8:30 | 20:30 | 40 | 2650 | 2500 | 15:00 | 16:00 |
| 8 | 2120 | 2535 | 10:30 | 11:30 | 41 | 2270 | 3425 | 8:30 | 20:30 |
| 9 | 2160 | 2350 | 10:30 | 11:30 | 42 | 1700 | 1630 | 8:30 | 20:30 |
| 10 | 2065 | 2210 | 8:30 | 13:30 | 43 | 960 | 155 | 15:30 | 20:30 |
| 11 | 2040 | 2270 | 8:30 | 13:30 | 44 | 915 | 230 | 8:30 | 20:30 |
| 12 | 2015 | 2120 | 8:30 | 13:30 | 45 | 730 | 4675 | 8:30 | 20:30 |
| 13 | 2125 | 2290 | 8:30 | 13:30 | 46 | 905 | 4720 | 8:30 | 20:30 |
| 14 | 2210 | 2600 | 8:30 | 20:30 | 47 | 925 | 4690 | 8:30 | 20:30 |
| 15 | 2330 | 2695 | 11:00 | 15:00 | 48 | 830 | 4710 | 8:30 | 20:30 |
| 16 | 2480 | 2700 | 8:30 | 13:30 | 49 | 1560 | 3320 | 8:30 | 20:30 |
| 17 | 2110 | 2200 | 8:30 | 13:30 | 50 | 1625 | 3240 | 8:30 | 20:30 |
| 18 | 2190 | 2410 | 11:30 | 12:30 | 51 | 1570 | 3400 | 8:30 | 20:30 |
| 19 | 2650 | 2035 | 13:30 | 20:30 | 52 | 2370 | 3020 | 8:30 | 20:30 |
| 20 | 2650 | 1890 | 12:00 | 16:00 | 53 | 810 | 4650 | 17:00 | 20:30 |
| 21 | 2700 | 1770 | 12:00 | 13:00 | 54 | 2860 | 1740 | 8:30 | 20:30 |
| 22 | 2740 | 2260 | 8:30 | 20:30 | 55 | 2200 | 2225 | 8:30 | 20:30 |
| 23 | 865 | 4660 | 8:30 | 20:30 | 56 | 2220 | 2310 | 18:00 | 19:00 |
| 24 | 1920 | 3010 | 13:00 | 14:00 | 57 | 2245 | 2240 | 8:30 | 20:30 |
| 25 | 2360 | 2730 | 8:30 | 20:30 | 58 | 2185 | 2270 | 8:30 | 20:30 |
| 26 | 2250 | 2860 | 8:30 | 20:30 | 59 | 2070 | 2480 | 18:30 | 19:30 |
| 27 | 2095 | 1480 | 8:30 | 20:30 | 60 | 2395 | 2125 | 8:30 | 20:30 |
| 28 | 2560 | 2340 | 12:30 | 13:30 | 61 | 2360 | 2290 | 8:30 | 20:30 |
| 29 | 820 | 2420 | 8:30 | 20:30 | 62 | 2710 | 2125 | 8:30 | 20:30 |
| 30 | 920 | 2420 | 8:30 | 20:30 | 63 | 465 | 1765 | 8:30 | 20:30 |
| 31 | 90 | 2950 | 14:30 | 15:30 | 64 | 170 | 2730 | 19:00 | 20:00 |
| 32 | 425 | 3085 | 14:30 | 15:30 | | | | | |

Coordinates are expressed in meters with respect to a fictitious origin

## 9. Appendix 2

In the following the solution of the instances of the benchmark are reported

*SACS solution for RC201*
**RC201-1:** 0 5 45 2 98 69 82
12 11 15 16 75 87 86 57 99 53 9 10
97 74 13 17 60 100 70
**RC201-2:** 0 65 14 47 59 52
83 64 19 23 21 18 76 85 84 51 49 22
20 66 56 96 54 37 43 35 93 91 80
**RC201-4:** 0 72 36 39 42 44
61 88 73 78 79 7 6 8 46 3 4 1 55 68

*SACS solution for RC202*
**RC202-1:** 0 91 92 95 63 85 33
28 26 27 29 31 30 62 67 71 72 41 40
43 35 54 93 94 96 80
**RC202-2:** 0 65 82 14 12 47 15
16 11 88 98 53 73 78 79 7 8 6 46 4 2 55 68
**RC202-3:** 0 45 5 3 1 42 39 36
37 38 44 61 81 90 99 57 86 87 9 10 97
59 74 13 17 60 100 70
**RC202-4:** 0 69 64 19 23 48 18
76 51 84 49 22 20 66 56 50 34 32 89
24 25 77 75 58 52 83

*SACS solution for RC205*
**RC205-1:** 0 98 47 14 69 11 15
16 12 88 78 73 79 7 6 8 46 5 3 1 4 43
37 35 93 96 80
**RC205-2:** 0 65 83 52 59 75 64
23 18 21 19 99 57 86 87 9 10 97 13 17
60 100 70
**RC205-3:** 0 92 95 33 28 27 29
31 30 63 76 85 67 84 22 49 51 50 34
32 26 89 20 24 74 77 58 25 48
**RC205-4:** 0 2 45 42 39 36 72
71 62 94 61 44 40 38 41 81 90 53 82
66 56 91 54 68 55

*TACS solution for RC201*
**RC201-1:** 0 5 45 2 98 69 82
12 11 15 16 75 87 86 57 99 53 9 10
97 74 13 17 60 70 100
**RC201-2:** 0 65 14 59 83 47
52 64 23 21 19 76 51 85 18 22 49 84
20 56 66 96 54 37 43 35 91 93 80
**RC201-4:** 0 72 42 36 39 44
61 88 73 78 79 7 6 8 46 3 55 4 1 68

*TACS solution for RC202*
RC202-1: 0 92 63 33 62 28 27
29 30 71 67 40 41 72 96 94 91 95 85 43
54 35 31 26 93 80
**RC202-2:** 0 65 14 47 12 15 11
16 88 73 78 53 79 8 7 6 46 2 98 82 55
4 68
**RC202-3:** 0 42 36 39 45 5 3 1
44 61 38 37 81 90 99 86 87 9 57 10 97
59 74 13 60 17 70 100
**RC202-4:** 0 69 64 23 19 76 75
18 51 84 22 49 20 66 56 32 34 50 89 24
48 25 77 58 52 83

*TACS solution for RC205*
**RC205-1:** 0 98 47 14 12 69 15
11 16 88 73 78 79 8 6 5 7 46 3 1 43 37
96 4 35 93 80
**RC205-2:** 0 65 83 52 75 59 64
23 19 21 18 99 86 87 9 57 10 97 60 13
17 70 100
**RC205-3:** 0 92 95 33 28 31 27
29 63 85 30 67 76 51 22 84 49 48 20 50
34 32 26 89 24 74 77 25 58
**RC205-4:** 0 45 2 42 39 36 72 71
62 94 61 44 40 38 41 81 90 53 82 91 54
56 66 55 68

**References**

[1] B. Bullnheimer, H. F. Hartl and C. Strauss, An improved ant system for the vehicle routing problem, *Annals of Operations Research*, Vol. 89 (1999), pp. 319–328.

[2] B. Bullnheimer, R. F. Hartl and C. Strauss, Applying the ant system to the vehicle routing problem, in *Meta-Heuristics: Advances and Trends in Local Search for Optimization*, S. Voss, S. Martello, I. H. Osman and C. Roucairol (eds.), Kluwer Academic Publishers, Boston, 1999.

[3] A. Colorni, M. Dorigo and V. Maniezzo, An investigation of some properties of an "ant algorithm", in *Proceedings of the Parallel Problem Solving form Nature Conference* (PPSN 92), Elsevier Publishing, Bruxelles, 1992.

[4] J. Desrosiers, Y. Dumas, M. M. Solomon and F. Soumis, Time constraint routing and scheduling, in *Network Routing*, M. O. Ball, T. L. Magnanti, C. L. Monma and G. L. Nemhauser (eds.), Elsevier, Amsterdam, 1995, pp. 35-139.

[5] M. Dorigo and L. M. Gambardella, Ant colonies for the traveling salesman problem, *BioSystems*, Vol. 43 (1997), pp. 73–81.

[6] M. Dorigo and L. M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transaction on Evolutionary Computation*, Vol. 1 (1) (1997), pp. 53–66.

[7] Y. Dumas, J. Desrosiers, E. Gelinas and M. Solomon, An optimal algorithm for the traveling salesman problem with time window, *Operations Research*, Vol. 43 (2) (1995), pp. 367–371.

[8] F. Focacci, A. Lodi and M. Milano, Embedding relaxations in global constraints for solving *TSP* and *TSPTW*, *Annals of Mathematics and Artificial Intelligence*, Vol. 34 (2002), pp. 291–311.

[9] W. J. Gutjahr, S-ACO: an ant based approach to combinatorial under uncertainty, *Technical Reports*, Department of Statistics and Decision Support Systems, University of Vienna, 2004.

[10] L. M. Gambardella, E. Taillard and G. Agazzi, MACS-VRPTW: a multiple ant colony system for vehicle routing problem with time windows, in *New Idea in Optimization*, D. Corne, M. Dorigo and F. Glover, McGraw-Hill, 1999, pp. 63–76.

[11] M. Gendreau, A. Hertz, G. Laporte and M. Stan, A generalized insertion heuristic for the traveling salesman problem with time windows, *Operations Research*, Vol. 43 (3) (1998), pp. 330–335.

[12] M. Gendreau, G. Pesant, J.-Y. Potvin and J.-M. Rousseau, An exact constraint logic programming algorithm for the traveling salesman problem with time windows, *Transportation Science*, Vol. 32 (1) (1998), pp. 12–29.

[13] M. W. P. Savelsberg, Local search in routing problems with time windows, *Annals of Operations Research*, Vol. 4 (1985), pp. 285–305.

[14] M. M. Solomon, Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints, *Operation Research*, Vol. 35 (1987).

[15] T. Stützle and M. Dorigo, ACO algorithms for the traveling salesman problem, in *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming and Industrial Applications*, K. Miettien, M. Mkel, P. Neittaanmki and J. Periaux (eds.), John Wiley & Sons, 1999.

*Received January, 2005*