Time Dependent Vehicle Routing Problem with an
Ant Colony System

Alberto V. Donati, Luca Maria Gambardella,
Andrea E. Rizzoli, Norman Casagrande,
Roberto Montemanni

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, 6928 Manno, Switzerland
alberto@idsia.ch

November 19th, 2002

Abstract
The Time Dependent Vehicle Routing Problem, TDVRP, consists in optimally routing a fleet of vehicles of fixed capacity when the traveling times between nodes are depending on the time of the day that the trip on that arc was initiated. The delivery to a customer must satisfy the customer delivery time window. The optimization consists in finding the solution that minimizes the number of tours (the number of vehicles used) and the total traveling time. The traveling time is calculated by knowing the departing time and the speeds distribution on the arc, which is supposed to be known at the beginning of the optimization.

This version of the VRP is motivated by the fact that in so many circumstances, traffic conditions can not be ignored in order to carry out a realistic optimization.

In this paper we present a new approach to this problem based on a Ant Colony System using a time dependent pheromones space, and introduce a novel methodology to improve local exploration of feasible moves in the search space.

We finally show some interesting aspects of this model, such as how it is favored traveling on longer legs when traveling faster, how different factors can affect the solutions obtained, and the importance of taking into account time dependent traveling times, not just because speeds profiles can affect the objective(s) of the optimization, but also because best solutions known for a non time dependent problem, in a time dependent context, are in general unfeasible.

1. Introduction
Vehicle routing problem has been largely studied because of the interest in its application in logistic and supply chains management. Many different versions of this problem have been formulated to take into account the many possible aspects. One of the most intriguing ones is to take into account the added complexity deriving from traffic conditions in the road network, not just in the context of an optimization problem, but also when the assumption of constant traveling times can not be considered reliable, like
when timely deliveries are to be done and costs of operation are on the edge of a feasible and effective solution to be found. Not taking into account these factors might greatly affect the layout of planned operations. Besides, optimization objectives might not be any longer satisfied.

We present in this paper an approach based on Ant Colony System. This approach has also been successfully used for the non-time dependent version of the vehicle routing problem with hard time windows, in [1], where ants of two different colonies are collaboratively searching in the solutions space, so that one colony aims to minimize the number of vehicles used, while the other the total distance traveled of the solution. In this case, the first objective is to minimize the number of tours first, and then the total distance traveled. The results presented in [1] show the effectiveness of this method.

The presence of diversified conditions of viability on the roads depending on the time of the day, is taken into account in [2], where one the formulation of the time dependent VRP is done (as well as for the TSP) by considering on each arc a step-function like distribution of the traveling times depending on the time when the travel is initiated. A mixed integer programming approach and a nearest neighbor heuristic are used for the optimization.

In [3] a time dependent vehicle routing problem with soft time windows (except the one of the depot) is presented. A very interesting property is introduced in the model, the First Leaves, First Arrives (FLFA, called in [3] First In First Out) that is, if two vehicles leave from the same node to the same destination, the one that leaves first will always arrive first, no matter the change in the traffic situation on that arc (change in speeds). This property is assured by considering instead of a step function of traveling times, a step function of the speeds, and by calculating the traveling time by integration once the departing time is known.

We have found that the FLFA property is quite important as we will see later, not only because guarantees some inconsistencies to be avoided, such as in the traveling time model (like [2]), it would be better to wait at some location because of a imminent lower traveling time coming, and arrive first at the next location! It is based on the solely fact that on the same arc, even if speeds change, the change will reflect uniformly on the arc, and it does not depend on the position on the arc, so it will be the same for both the vehicles. So all the vehicles will have the same speed on the same arc at the same time. In [3] a time dependent model is analyzed, and approached thru a taboo search heuristic similar those presented in [6] and [7], and based on the use of approximation function to evaluate in linear time the goodness of local search moves. In [3] the model is also formulated in a dynamic environment, where not all the service requests are known before the start of the optimization.

In this model we adopt the FLFA propriety, where the speeds distribution on arcs of the network are known a priori, before starting the optimization. The speeds distribution induces a partition of the time scale in so called time slices, or periods of time over which
the speed can be considered/is constant. Then number of slices is then used to create a
corresponding number of search subspaces, were the colony of ants search optimal
solutions, passing from a subspace to the next during the tour construction (see Figure 1). The pheromones are deposited on these subspaces, and then are also time dependent, because they are relative to the time subspace where the ant was moving at a given
moment. So in this case, the pheromone temporal distribution on the arc represents the
goodness of that move when initiating the trip at that time.

The FLFA propriety, besides giving consistency to the model, also has as a consequence,
the very desirable property of enabling of back propagating slack times, as we will show
in the §4. This fact can be used to keep O(1) the search of feasible moves of local search
procedures. Procedures such as tour reduction and post insertion, customers' relocation,
and exchange are also presented, to perform post-optimization and local search.
The objective of the optimization, like in [1] is to minimize the number of tours first, and
then the total traveling time.

![Figure 1: process of two routes construction thru time subspaces.](image)

### 2. Problem Description

In the Time Dependent Vehicle Routing Problem, TDVRP, like in the case of the classic
Vehicle Routing Problem with time windows, VRP-TW, a fleet of \( M \) initial vehicles of
fixed capacity \( C \), has to be scheduled to serve \( N \) customers and hit them no later than the
ending time of the respective time window. In this case each customer has one single
time window, while each vehicle can be used only once in the time horizon considered. If
the arrival time at the customer is earlier than the beginning of the time window, the vehicle will incur in a waiting time at the customer site. Each customer, beside the quantity requested and the delivery time window, is also characterized by a service time. The quantity requested by a customer must be delivered in full at one time. Tours must originate and end at the depot within the depot opening time and a closing time, so that no tours can originate or end outside this timeframe. This also determines the time scale for the model.

The problem is to find those feasible solutions, that is, a set of routes that satisfy all above constraints, not exceeding \( M \), so to minimize the number of tours first and then the objective function, defined here as the total traveling time.

The problem can be formalized as follow: given an incomplete directed graph \( G(V,E) \), customers' locations are represented by nodes \( V \), while roads with the set of oriented arcs \( E \). To maintain scalability on large instances, for each node we introduce a set of neighbors, that is a set of \( n \) closest connected nodes, in the sense that exists an arc going from the node to its neighbor. The numbers of neighbors here is fixed and is usually set to 30-35, while every customer has a connecting arc to the depot, and the depot is also connected to all the customers.

On each oriented arc is specified a distance and a speed distribution. A speed distribution is a step like function, like the one shown in Figure 2, so it is defined by the intervals on which it is constant and its values. In this model it is in principle possible to define a speed distributions different in intervals and values for each arc, but for simplicity we classify each arc with a given type (like a road type) and use a unique speed distribution per each type.

![Figure 2: an example of speed distribution](image)
The traveling time deriving from this type of distribution is continuous distribution, and since the distance between two nodes is fixed, it only depends on the time of the day the travel originates. If the distance would be equal 2 the traveling time would be a distribution like the one shown in Figure 3.

![Travel time distribution](image)

**Figure 3:** travel times for an arc of length 2, induced by the speed distribution of Figure 2

### 3. The algorithm Ant-TDVRP

Recently a variety of new heuristics have been developed to solve NP-complete optimization problems. Ant based algorithms use a colony of artificial ants, or cooperative agents, to explore and find optimal solutions, and in recent years, it has been applied successfully to a variety of problems in operational research, and in a variety of variants.

The idea of Ant Colony System is inspired by observations of social insects [5], in particular ants. In real world ants are able, among other things, to find and establish trails to minimize the time to travel between their nest and a food source found nearby. The formation of a trail is based on two processes: the deposition and the evaporation of the pheromones. As computational device is based on this following things: 1. shorter paths have a shorter travel times, so higher chances of being traveled, or traveled more frequently, 2. ants go back to their nest. This last is as important as the first, because if no ant would came back to the nest, all paths would be traveled with exactly the same frequency (even if the ant on the shorter path would arrive first), so no reinforcement would happen.

An Ant-based optimization is a parallel, distributed optimization process. Similarly to real ants, the basic mechanism is to encode the information not in a population of individuals, but in a modification of the environment, an information that is locally accessible to all individuals, a shared space of pheromones. Once the formulation of the
problem is given, and a proper representation is found, artificial ants move over the space (in this case the space of solutions) step by step. Each step represent a completion of a task, and is decided by the ant by examining the pheromones trails and weighting them against the immediate convenience of doing a that choice (e.g. for the TSP, this immediate convenience would be the inverse of the distance to the city). The choice of which next step to do is done using both these information. To create diversification in the search, once the probability distribution is given, the choice to take among all the possible, can be taken in different ways: exploiting, exploring, randomly exploring.

Once the solution is finished, that is all the customers are serviced, or there are no more trucks available, a post optimization procedure is applied (as described in the next paragraph), and if the solution is feasible, the fitness $f$ of the solution is calculated (in this case it is the inverse of the total travel time), and the pheromones are updated in the following way: a uniform evaporation, which uniformly reduces them, and the deposition, proportional to $f$, on the arcs belonging to the path the ant took.

Because the deposition is proportional to the fitness, pheromones encode locally global information. Since pheromones also evaporate over time, only paths that keep to be visited will be emerging among the all possible, and the ants will tend to search around their neighborhood, making this search mechanism very effective.

In this problem, the fitness defined as the inverse of the total travel time, that is:

\[
\text{Equation 1} \quad f = \frac{1}{\sum_i T_i}
\]

where $T_i$ is the travel time of the tour $i=0, \ldots, m$ where $m$ is the number of tours.

In other optimization contexts the fitness can be defined in different ways to reflect the optimization objective(s), like it could be a combination of distance travel (if there are costs per mile traveled), the travel time, and the waiting times (if there are non fixed-cost associated with drivers working times), and so on.

\textbf{Probability distribution}

The probability distribution that an ant evaluate to go to a node $j$ from node $i$ at a time $t$, with index $k$ (the time slice index corresponding to $t$) is given by:

\[
\text{Equation 2} \quad p(j) = H(i,j, t)^\alpha \cdot \varphi(i, j, k)^\beta
\]

where $\varphi$ are the pheromones, $\alpha$ and $\beta$ are power weighting constants, and $H$ is often referred as the heuristic function, that expresses namely the level of immediate convenience of doing a certain choice.
In this model, it is calculated as follow. For all the undone customers, $H$ is set to zero if going to $j$ violates any of the constraints, that is:

1. the quantity requested by $j$ exceeds the quantity left on the truck,
2. the arrival time at $j$, leaving at $i$, is greater than the end of the time window of $j$,
3. the returning time from $j$ to the depot, is after the depot closing time.

If there is no violation of constraints, then $H$ has the following form:

\[
H(i, j, t) = \frac{1}{\text{travelTime}(i, j, t)} \cdot \frac{1}{\text{waitingTime} + 1} \cdot A
\]

that expresses the tendency to go to a customer $j$ for which the travel and the waiting times are smaller, and the attractiveness $A$, (that is the inverse of difference from the time the time window ends and the arrival time at $j$) is greater, that means, customer for which the end of the time window is getting closer are more important.

We notice that the second and third factor can have some disturbance on the optimization and sometimes need to be discounted by some factor.

**Types of steps**

1. *exploiting*: pick the $j$ that maximizes $p(j)$
2. *probabilistically exploring*: pick the $j$ randomly distributed as $p(j)$,
3. *randomly exploring*: pick the $j$ completely randomly (in this case is not necessary to calculate and $H$, but only verify the constraints)

Which one is used is determined by the colony “profile”: 2 numbers $q_o$ and $r_o$ (with $0 \leq q_o \leq r_o \leq 1$) that determines the percent of each steps. At each step, an ant would generate a random number $r$ in [0,1) and do 1, 2, or 3 respectively whether $r < q_o$, or $q_o \leq r < r_o$, or, $r \geq r_o$. Typical values used are $q_o = 0.9$ and $r_o = 0.02$.

**SCHEME of the algorithm**

The scheme of the algorithm is then the following, and it has been developed in an object oriented architecture in C++, in a fashion that is abstracted from the particular type of the problem (TDVRP). In this case, all the ants know is that there is an instance of some problem to be solved, and that there are a number of task to be completed.

**Initialization**

From the main class, the problem type is specified, in this case the TDVRP, and a storage class, keeping all the system objects instantiated and initialized, by reading the data and initializes the proper variables/objects, and set the pheromones to a uniform initial value.

**Ant loop**
The ant starts to loop, by calling abstract methods of the abstract problem, that is (omitting methods’ arguments here):

1. `prepareToStart()`: does a sort of an ant initialization of all the variables that were changed by the previous ant and need to be reset, and it also tells to the ant to start a new tour.
2. `preliminaryStep()`: at each iteration an abstract method is called, that does some eventual preliminary operations before doing a step (e.g. in this case, we check if any tour need to be initiated),
3. `greedyStep()`, `probStep()`, or `randomStep()`: depending on the value of \( r \) then one of these types of steps is done. Note that
4. 2 to 3 are repeated till every task is complete, or there are no more available resources to accomplish them.
5. `finalize()`: a finalizing method of the problem is called (e.g. for the TSP this could be to close the loop).

Note that a tour is closed (and added to the solution) in 2. when: a. there is not a sufficient quantity in the truck for any of the remaining neighbors customers, b. the tour would end after the depot closing time, c. there are no more customers in the neighbors that can be done without violating the time windows constraints (or no more customers at all), or the ant has chosen in \( p(j) \) to go back to the depot.

After 5. the fitness \( f \) is calculated.

**Post optimization**
If the solution is not feasible, a post insertion is attempted. If it results feasible, post optimization procedures (see for details next paragraph) are applied. One can chose the way to run them, being quite and heavy computational effort. E.g. one can choose to run them only on good solution (a good solution is that is within a small percent in fitness from the best, or with a number of tours less that the best).

**Pheromones update**
- **evaporation**: uniform everywhere with a constant \( \rho \):
  \[
  \phi(i, j, k) \leftarrow \rho \cdot \phi(i, j, k) \quad \forall i, \forall j, \forall k
  \]
- **deposition**: on the arcs that belong to the path of the solution so far obtained, proportionally to the fitness, as defined in Equation 1.
  \[
  \phi(i, j, k)^+ = \epsilon \cdot f \quad i, j, k \in S
  \]
  where \( S \) is the solution found.
Note that if the solution it is a good solution, the pheromones are boosted, that the value of \( \epsilon \) is of a factor 10 to 100 greater.
If the solution is the new best solution, it is stored. This procedure is repeated till a fixed temporal limit is reached.

4. **Local Search and Other Considerations**

Local search procedures are very useful search procedures to improve the quality of the solution found, by using, usually, some simple criteria. A local search procedure attempts to modify a solution thru a specified move, and evaluate 1. if the move is feasible, and 2. how good it is.

In order to be fast and effective thou, given the number quite large of possible combinations, these procedures should at least perform 1. in constant time.

The two basic operations for a local search procedure are: a. insertion of a new delivery in a tour, b. removal of a delivery from a tour.

In the case of the TDVRP, doing a. or b. will generate a time shift of all the deliveries following the one inserted, such that, also travel times might change as a result of the shift of all the departing times, that is the time of the day when a trip was initiated. In particular with different speeds distributions on arcs, also doing b. could create a delay and then an infeasible solution.

Since a delay added will shift all the deliveries, also all the following travel times will change, so it would seem that to test the feasibility of an insertion, one should recalculate all the times relative to the deliveries following.

We present here a method which is an extension of a method used for non time dependent case, that is based on the use of a delay variable, called the slack time, calculated (and kept updated after each operation a. and b.) per each customer, that tells how much a delivery can be delayed without causing any time window of the following customer to be missed.

The slack times are calculated backward, starting with the ending depot, once a tour is complete, in the following way:

\[
s_i = \min(s_{i+1}, TWend_i - t_{di})
\]

where \(i\) is the customer’s index in the tour, \(TWend_i\) is the time when the delivery time windows ends for \(i\), \(t_{di}\) is the time the vehicle arrives at \(i\). The slack time for the depot is the depot closing time minus the time the tour ends.

In the case of time dependent model, since the travel times change, we have to see if it is possible “back propagate” the slack time of the next customer \(s_{i+1}\), and calculate, given the speed distribution, the corresponding delay on customer \(i\), so that not to violate \(s_{i+1}\).
If \( g \) is the function that gives the arrival time at the next location, we would have to find the \( \Delta \), the maximum delay on arrival time at \( i \), such that:

\[
\text{Equation 5} \quad g(t_{Ai} + \Delta) - g(t_{Ai}) \leq s_{i+1}
\]

from which we would have:

\[
\text{Equation 6} \quad \Delta = g^{-1}(s_{i+1} + g(t_A)) - t_{Ai}
\]

We have to prove that \( g \) is invertible. If the speed distribution \( v(t) \) is given, from the definition of speed \( v = ds/dt \), we have by integration between 2 points A and B:

\[
\text{Equation 7} \quad D(A, B) = \int_{t_A}^{t_B} v(t) \cdot dt \quad \Rightarrow \quad V(t_B) = D(A, B) + V(t_A)
\]

so if \( V \) is the primitive of \( v \), is namely the function \( g \) we are looking for, \( g \) is invertible if it is monotonic (or locally monotonic), that is if its derivative never changes sign, which it is always the case when a speed distribution \( v \) is always greater than zero. In case \( v \) does not admit a primitive, the monotonic behavior is guaranteed again by the fact that it will never change sign. Intuitively, this is a consequence of the FLFA principle, that keeps monotonically increasing the arrival times with the departing times.

This property, give us the possibility of back propagating the slack times, and then to test the feasibility of a move in constant time, by using the corresponding of the Equation 4 but with the back propagated slack time, that is:

\[
\text{Equation 8} \quad s_i = \min(\Delta, TW_{end} - t_{Ai})
\]

where \( \Delta \) is given by Equation 6.

A procedure to calculate the backward slack time is given in appendix B.

\textit{Post optimization procedures}

The post optimization procedures used in this analysis are the following:

1. \textit{Tour reduction and post insertion}: when the total remaining quantity on the vehicles exceed the fixed vehicle capacity, it can be possible to reduce the number of tours. The tour with the least deliveries is chosen, and its customers are tried to be inserted in other tours, by minimizing the difference between the previous arrival time at the next customer and the new one. If for all the customers to insert, an insertion is found, the new solution is accepted.
2. **Customer relocation**: for a subset of customers (usually 10% random sample), the procedure evaluates the convenience of doing the delivery to that customer in a different tour, or at a different moment in the same tour (in-tour relocation), by calculating the delay on the next customer on arrival time, like in the post insertion procedure.

3. **Customer exchange**: a subset of customers (10%) are chosen to try an exchange from a tour to another one with some other customer. The best position in the respective tours is calculated for both customer such to minimize the variation in total travel time of the newly obtained solution, if feasible.

5. **Experiments**

Some experiments have been conducted to show some interesting features of a time dependent model and its advantages. The experiments presented here aim to show the two following things:

1. the algorithm tends to favor routes for which longer legs are possibly traveled in periods of time with higher speeds, and shorter legs in periods of time with lower speeds.

2. if solutions known for non time dependent model are used in the context of the time dependent model, their feasibility and optimality might drastically change. They are compared to the solutions found for the time dependent model.

To enlighten 1. we have created some *ad hoc* experiments, using the following settings. There are 2 time intervals with the same speed distribution for all the arcs, a two value speed distribution. The customers locations are chosen so that there are subsets of customers that are fairly close one to another one, and other subsets that are in average as double distant. The working times and travel times (or speeds) are such that it is possible to cover entirely the subset if it would be possible to choose the right speed (in this case the low speed or the high speed).

It is crucial to eliminate in this context all those factors that could mask this adaptive behavior, such as the time windows (which constitute a hard constraints here), that have been removed in these type of experiments. Note that in this case the calculation of a starting time for the tours is not needed, Also working times play a role, since they can shift the tour completion up to the end, no matter how fast one can go, so they needed to be chosen in a way not to mask the time dependent features. The experiment consists in using 2 distinct speed distributions: one with a profile type low/high (meaning the first period with a low speed, the second with high speed, the double), the other one of the type high/low.

The following figures Figure 4 and Figure 5 show that three tours are formed but the orientation of the tours from Figure 4 to Figure 5 is inverted as a consequence of the different speed distribution, so it appears clear that in order to obtain an optimized solution, the tours are formed so to use the high speed period to complete the longer legs, and the low speed period to complete shorter legs.
Figure 4: LOW → HIGH speed distribution.

Figure 5: HIGH → LOW speed distribution.
It is worth to note that sometimes the best solutions obtained by “inverting” the speeds have not necessarily the same “look”. Indeed while in Figure 5 the best solution for problem is shown, for the first problem (Figure 4) that has a total travel time of 253.64 there is a better solution, shown in next Figure 6, that shows another important thing: since no waiting times are introduced by the optimization, to take full advantage of the high speed periods of time, it is better to make an initial delivery to a quite not on route customer.

In tests for 2., we can use any arbitrary speed distribution on each arc, since it is not only the objective value that is important, but also, and first of all, the feasibility of a solution. For various known problems we have used a 4-value speed distribution,

It is important to notice that a cost comparison between the 2 models not necessary will find the time dependent model more convenient over the non time dependent case, since it all depends on the assignment of speeds distribution and their average values in time. But in order to analyze the feasibility of a non-TD solution in a TD context, we have to use a speeds distribution that averages to 1 (the non TD case is a TD case with constant speed equal 1). We have showed that for the most constrained Solomon problems, the
degree of unfeasibility of the best non-TD solution known is proportional to the degree of variation of speeds.

6. Conclusions
In conclusion, we have presented a time dependent model for the vehicle routing problem based on the ACS with a time dependent component of the pheromones space. ACS algorithm is supported also by local search procedures, and it is showed how it is possible to maintain in constant time the search of feasible moves for these procedures. A set of vehicle routing problems is examined in the time dependent case, and it is shown that in most cases either the best solution known for non time dependent case, might be unfeasible in a time dependent context, or have traveling times that do not reflect the real situation. Computational results suggest that in those system where the constant speed approximation is no longer valid, it is crucially important, not only for the optimization, to explicitly consider those variability in the model, to make it more suitable for applications to some real world situations.

Acknowledgements
This work was co-funded by the European Commission IST project MOSCA: Decision Support System For Integrated Door-To-Door Delivery: Planning and Control in Logistic Chain", grant IST-2000-29557. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

References


