CIMSA 2003 - International Symposium on
Computational Intelligence for Measurement Systems and Applications
Lugano, Switzerland, 29-31 July 2003

Integration of a Robust Shortest Path Algorithm with a Time Dependent Vehicle Routing Model and Applications

Alberto V. Donati, Roberto Montemanni,
Luca M. Gambardella, Andrea E. Rizzoli
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, 6928 Manno, Switzerland
Phone: +41-(0)91-610-8568, Fax: +41-(0)91-610-8661, Email: alberto@idsia.ch

Abstract ***– This paper describes a way of combining two techniques, in order to define a framework that can deal with the following problem: find an optimized set of routes when the customers set is a proper subset of an entire network, and variable traffic conditions have to be taken into account.***

This is accomplished on one hand by extending the vehicle routing problem (VRP) to a time dependent case (TDVRP), on the other hand by using an appropriate algorithm, the robust shortest path (RPS) that can provide itineraries when moving to a location to another, and guarantee good performance under any possible road network situation.

Once a proper description of the TDVRP model is given, we discuss the optimization technique, based on the ant colony system (ACS), and the robust shortest path (RPS) algorithm.

Different ways of integrating these techniques are discussed. The one presented here consists in using the RPS algorithm for the calculation of the paths among each pair of customers, so that this information can to be used by the TDVRP optimization in a very efficient way.

In the case of a real road network, some tests have been made, that show that the optimal solutions obtained for the classic VRP case are sub optimal when considered in a time dependent context, revealing that the approximation of constant speeds is sometimes inadequate.

I. INTRODUCTION AND PROBLEM DEFINITION

The vehicle routing problem (VRP) has been largely studied because of the interest in its applications in logistic and supply-chains management. Different versions of this problem have been formulated to take into account different aspects/issues.

In the Time Dependent Vehicle Routing Problem, TDVRP, as in the case of the classic Vehicle Routing Problem with time windows, VRP/TW, a fleet of vehicles of fixed capacity has to be scheduled to visit a given set of customers, each by a time no later than the ending time of the customer's time window. If the arrival time at the customer is earlier than the time window, the vehicle will incur in a wait at the customer's location. Each customer is also characterized by a service time, the time to complete the unload/work. Other assumptions of the problem are that 1. the quantity requested by a customer must be delivered in full, and at one time; 2. all tours must originate and end at the depot, within the depot opening time; 3. the total quantity delivered in each tour can not exceed Q. The novelty of the model is that instead of having fixed travel times, like in the VRP case, the travel times are dependent on time.

The problem can be represented with an incomplete directed graph G(V, A), where A is a set of oriented arcs connecting pairs of nodes, and V is the set of nodes of which one represent a depot, and the rest the customers. Each node is characterized by a location, and each customer c_i, i=1,..,N has a requested quantity q_i, a the delivery time window $tw_i = [b_i, e_i]$, and a service time wt_i. For the depot the opening and closing times $[T_o, T_c]$ and the fixed truck capacity Q, are given.

For each oriented arc, a speed distribution $v(t)$ is given, defined on $[T_o, T_c]$, where t is the time when the travel begins from the starting node.

A feasible solution is a set of routes not violating any of the described constraints, and visiting all the customers once. The optimization consists in finding the solution that minimizes the number of tours and (then) the total traveling time.

The TDVRP optimization is a combinatorial NP-hard optimization problem, so an exact approach, if exists, is often inconvenient if relative short times are available for computations. Heuristic algorithms can find solutions of very high quality, close to an exact algorithm, but in a considerably shorter time.

II. ANT COLONY OPTIMIZATION

Ant Colony Optimization (Dorigo, Di Caro, and Gambardella in [1]) is based on the idea that a large number of simple artificial agents are able to build solutions via low-level based communication, inspired by the collaborative behavior of a colony of ants. A variety of ACO algorithms has been proposed to solve NP optimization problems, and have been successfully applied to the traveling salesman problem (TSP), the quadratic assignment problem, graph-coloring problem, job-shop, flow-shop, sequential ordering, and vehicle routing problems. As showed by Dorigo, Maniezzo, Colorini in [2], on the TSP problem, the prototype of the NP-hard problems, an ant system (AS) is able to obtain results comparable to the best heuristics, and can be very effective even for large instances, when combined with specialized local search procedures.

Real ants cooperate in their search for food by depositing chemical traces, the pheromones, to establish trails to minimize the time to exploit a food source. Ants tend to follow in their walk directions where the pheromones are higher. The formation of a trail is due to the fact that each ant during its walk deposits a small fixed amount of pheromone per unit length, which is also subject to constant evaporation. Shorter paths have shorter travel times, so intrinsically higher chances of being traveled more frequently. This process brings to a rapid consolidation the most efficient path.
The basic idea of an ACO is to use a positive feedback mechanism (as discussed in [1]) to reinforce those portions that contribute/belong to a good solution, while discarding those that would bring to poor solutions, with the possibility to temporally store this information so that it will be locally available/accessible to all the individuals. The way to accomplish this is through a modification of the environment, the artificial pheromones. Once the graph G(*V*, *A*) representing the problem is defined, the pheromones are associated to each arc. Each artificial ant will use this information weighted with some appropriate local heuristic function (e.g. for the TSP, the inverse of the distance) during the construction of a solution. The pheromones are updated proportionally to the goodness of the solutions found, so they will tend to encode global information about the goodness of the choices made in the past. Since artificial ants tend to move following the higher pheromones trail, the search for improved solutions will tend to concentrate in the neighborhoods of the good solutions found.

III. TDVRP

We briefly introduce here the TDVRP model and the relative optimization algorithm.

A. The model

A first coherent extension to a time dependent version of the VRP is presented in [4], where for each arc $a \in A$, a speed distribution *v(t)* is given, which from which the travel times are derived. In this model tough, the customers' time windows are soft constraints, except for the depot.
The speed distributions used for each arc, are step-like functions defined over specified time intervals T_k, like the one shown in Figure 1.

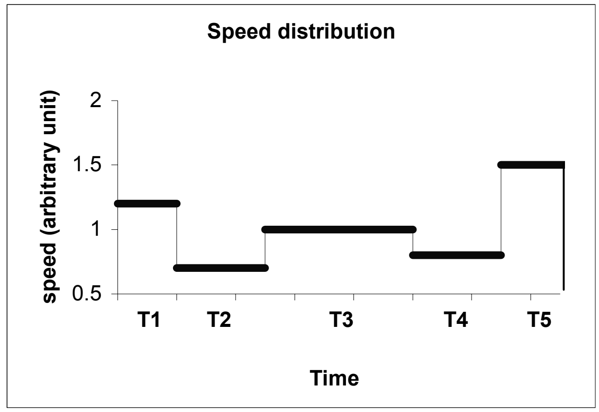

Figure 1: example of a step-like speed distribution.

The TDVRP model is also based on the idea of creating temporal partitions of the model time. The model time in this case, is the time within the model horizon, set by the depot time window. Once a speed distribution is given on an arc, it will induce a partition of the time in subspaces, namely the T_k. For each oriented arc $a \in A$, one of such distribution is given, so once the distance between the starting and ending nodes is known, it is possible to calculate the relative traveling time distribution, and so to know the travel time through that arc, once the departing time from the starting node is known. Note that if during the travel a speed change is incurred, the travel time will reflect the change. It derives that the travel time distribution is a continuous function of time.

The use of speed distributions, instead of travel time distributions, has many advantages, and thus has been adopted here. First, it guarantees some inconsistencies to be avoided, such as departing from the same location at a later time, in some schemes could end up in arriving first (see also [4] and [5]). Second, in this way it is guaranteed that the arrival time $AT(t)$ function is a monotonic increasing function of the departing time t. This property is fundamental, as discussed in [5], for some considerations on local search and post-optimization procedures, that is they can be run in constant time, like in the constant speeds (classic) model.

B. The algorithm

As presented by Gambardella et al. [3] in the MACS-VRP algorithm for the VRP with time windows, an ant-based algorithm is very effective to solve this problem. In the case of MACS-VRP two colonies of ants are used, for two optimization objectives of the problem: one deputed to tour minimization, the other to distance (or travel time, since here speeds are constant) minimization.

Inspired by this approach, we have used a travel time minimization colony adapted to time dependent case, while tour minimization is done implicitly.

The time partition induced by the speeds distribution, allows to formulate the optimization problem in time subspaces, as schematically represented in Figure 2 (where only 3 subspaces are shown).

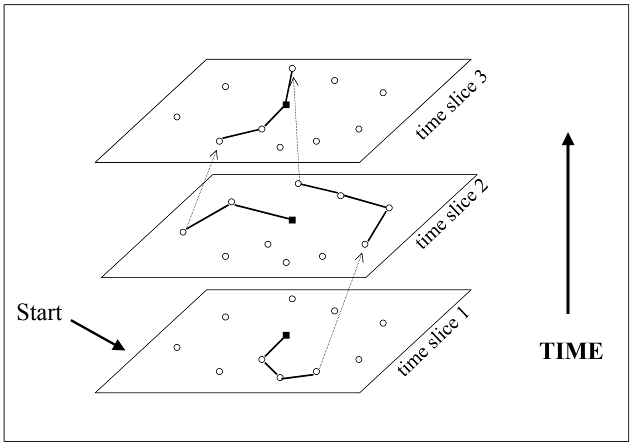

Figure 2: the partition of the model time in time subspaces, and the construction of two tours, though time. Transition though time is indicated with the arrow. The dots are the customers while the square is the depot.

The pheromones are represented in this way: to each existing oriented arc connecting the nodes (***i***, ***j***) is associated a temporal distribution $\tau_{ij}(k)$, where the index ***k*** refers to the index of the subspace T_k.

The basic ant loop is then the following.
Each ant constructs a solution by completing number of tours, until all the customers are serviced. A tour starts at the depot with a fully loaded truck. The ant move from a node ***i*** (a customer or a depot) to the next ***j*** (a customer or a depot) by choosing among all the possible ***j***, that have not visited yet (except depot) and which do not violate any of the constraints (see below), with the following probability distribution:

Equation 1 $p(j) = h(i,j,t)^{\alpha} \cdot \tau_{ij}(k)^{\beta}$

where t is the departing time from ***i*** (e.g. the time the work at ***i*** is completed) in T_k, and ***k*** is the corresponding time index, $h(i,j,t) = \frac{1}{f(i,j,t)} \cdot \frac{1}{wt(j)+1}$ is the local heuristic function, $f(i,j,t)$ is the travel time and waiting time *wt(j)* at ***j***, and α and β are power weighing constants.
The next location ***j*** is considered a valid possible one, if it satisfies all the following constraints: 1. arrival time at ***j*** $at(j) \le e_j$ customer's time window; 2. $q_j \le Q_{left}$ the quantity left on the truck; 3. returning time at the depot from ***j,*** once the work is complete at ***j***, cannot be greater than the depot closing time.
An ant uses the probability given by Equation 1 in different ways, determined by the cut-off parameters q_o and r_o ($0 \le q_o \le r_o \le 1$), and a random number $r \in [0,1)$:

1. exploiting: pick the ***j*** which maximizes *p(j),* if $r<q_o$
2. exploring: pick the ***j*** distributed as *p(j),* if $q_o \le r < r_o$
3. randomly exploring: pick the ***j*** randomly, if $r \ge r_o$

We used the following values for the parameters: α=0.5, β=2, q_o=0.95, r_o=0.01, and τ_o=0.001 (constant for the initialization of the pheromones).
Once the ant returns to the depot, a new tour is initiated if more customers need to be serviced.

The complete optimization algorithm is organized like in the following scheme, assuming the proper data is given.

```
begin:
  initialize  τij(k)   // for all the existing arcs
  init optimization objectives and best
    solution Sbest

loop:
while (time is not expired )
{
    ant loop      // find a solution S

    if(S is not feasible)
       try post-insertion

    if(S is not feasible) continue

    if ( S is good )
       run post optimization on S   // see below

    if ( S > Sbest)        // to be > means to be better
       Sbest=S             //  see objectives definition.

    update the pheromones

    check time
}

return Sbest
```

A solution is considered *good* if it has the same number of tours of S_{best} and it is within 5% from the total travel time T_{best}. The post optimization/local search (LS) procedures are applied only to the good solutions, and are: tour reduction and post insertion (if the number of tours can be reduced, otherwise this is turned off), customers' relocation, customers' exchange, and 2-k optimization.

A solution S is considered the new best if: 1. it has a smaller number of tours; 2 it has the same number of tours and a smaller total travel time T.

Note that tours minimization is accomplished only with the tour reduction procedure, or with best solution selection criteria.

The pheromones are updated in the following way: 1. a uniform evaporation, 2. increment on the arcs of the solution found, proportionally to $1/T$. In this case, also the information about the time index ***k***, that the arc was traveled, is necessary to increment the element $\tau_{ij}(k)$ of the time dependent pheromone distribution.

IV. ROBUST SHORTEST PATH, RSP

A road network is usually modeled in mathematical terms as a weighted digraph, where each arc is associated with a road and costs represent travel times.

Unfortunately, it is not always easy to estimate arc costs exactly, because they are dependent on many factors, which are difficult to predict, such as traffic conditions, accidents, traffic jams or weather conditions. For this reason, the fixed cost model previously introduced may be inadequate, as it may not be realistic enough.

We then considered a more complex model, when the robust deviation criterion (Kouvelis and Yu [7]) is applied to the shortest path problem defined on an interval data (Karasan et al. [6]) directed graph $G = (V, A)$, where N is a set of nodes and A is a set of arcs. A starting node $s \in V$ and a destination node $t \in V$ are given. An interval $[l_{ij}, u_{ij}]$, with $0 \leq l_{ij} \leq u_{ij}$, is associated with each arc $(i,j) \in A$. Each interval represents the range of possible travel times for the respective arc.

A robust shortest path guarantees good performance under any possible road network situation (according to interval data). A robust path is then more desirable than a classic shortest path, because it exploits the extra information guaranteed by interval data to give an output, which is more reliable for the user (who is going to use this calculation to reach t from s).

The algorithm developed (described in details in Montemanni and Gambardella [8]) is based on the conjecture that a robust shortest path is also one of the first paths in a shortest path ranking in a simple directed graph where the cost on each arc (i,j) is equal to u_{ij}. The algorithm works then in the following way: a procedure to rank paths in the simple directed graph defined above is run. For each path retrieved, the respective robustness cost is calculated. The algorithm stops when a lower value for the robustness costs of the paths not yet examined matches an upper bound for the same paths (both the lower and the upper bound are defined in [8]).

V. INTEGRATION OF THE TDVRP WITH THE RSP

In order to deal with a problem where the set of customers is a proper subset of the set of nodes, like in the real cases, paths need to be calculated for each customers' pairs.

Since the time dependent nature of this model, these paths are also time dependent.

There are two alternative ways to proceed: 1. calculate shortest paths on the fly, that is, at departure time, from the current location; 2. store one or a set of paths that can represent a suitable approximation of the problem, so that given a departing time, a pre-calculated path from the list will be used.

The first option would imply the added computational effort to calculate at each location the paths and travel times for going to all the next possible remaining locations. For this reason, we adopted the second method, pre-computing the robust shortest paths among all the customers' pairs before the start of the optimization. This calculation is done by knowing the interval graph (that is the *min* and *max* travel times on the arc).

A more complex computation can be done also by considering a time dependent interval graph, and by computing for each arc, a proper set of time intervals on which the path can be considered fixed.

In this first version of the model, we have assumed that the interval graph used by the RSP algorithm is not time dependent. The complete graph $G = (V, A)$ with all the nodes and arcs of the real road network, is considered for the computation, and for each customers' pair, the robust shortest path is calculated. The TDVRP is then initialized by considering among all nodes, only those relative to the depot and the customers, and by creating a set of oriented arcs A_C among each customers' pair. Each arc $a_C \in A_C$ is associated with a RSP path, and a time dependent travel time distribution is derived for each of such composite arcs, by considering the speed distribution on the arcs $a \in A$ belonging to the path. In this sense the time dependent features of the TDVRP are preserved, since traveling times are still time dependent.

Note that when considering the time dependent paths case, this scheme is still valid with the same set A_C, except that each arc will have a list of paths and an extra time partition, such that once the departing time is known, the proper path can be chosen. A proper adjusted travel time distribution for a_C will be derived to reflect the path followed.

VI. COMPUTATIONAL RESULTS

A number of tests have been conducted on a real context, using the data collected on the Padua road network by mean of the automated traffic control system “Cartesio”. The system records in real time the volumes of traffic and speeds of the vehicles, air and noise pollution data, and can provide traffic information, though different channels such as VMS (Variable Messaging System) panels, RDS (Radio Data System) messages, SMS messages. Logistic centers can access “Cartesio” on the Internet.

The data consists in a set of 1,522 geo-referenced locations and 2,579 arcs. Road types (sections) and relative traffic data are specified, in particular hourly travel time distribution.

A set of customers and a depot (in these tests, the Interporto Padova) are given, as long with their demand, opening times, and available trucks and their capacity (the model can deal with non-constant truck capacity, but no optimization is done in this sense at this point). No customers’ time windows were specified, while the service time was set to ½ hour for all. We used for the time subspaces 4 intervals of the same width, for all the arcs, partitioning the depot opening time, [8.00, 18.00].

We made two types of runs: the non-time dependent case using constant speeds (briefly CS) obtained by suitable average of the travel times and the time dependent case (briefly TD).

We have considered for each test a sub-sample of 30 customers extracted randomly from a given sample of 60. For each test, the optimization was run 5 times, first for the CS model, then for the TD model, evaluating the best solution found for the CS model. The average total travel time <*T*> and standard deviation with respect to the runs, respectively for the CS solutions, the CS solutions in a TD context, and the TD solutions, are shown in Table 1 for each test. Times are expressed in hours, and fractions of hours. The number of tours is always equal to 2 in all the cases, so it has been omitted from the table. The computation time of each run was 5 minutes on a Pentium IV, 1.5GHz machine.

The last column gives the difference between the <*T*> of best TD-solutions with the <*T*> of best CS-solutions, for the data considered. In other words, Δ evaluates the level of sub optimality of a CS-solution in a time dependent context. The average of Δ, over all the tests, is *758%*, but it is possible to notice that can be as much as 12%.

If customers’ time windows are present, CS solutions might also become unfeasible. The degree of sub optimality/unfeasibility is related to the deviation from the constant speed case.

In the following Figure 3, a TD solution is shown, with visualization software developed at IDSIA. Some information about the solution (white box) and the tours (gray boxes) is also displayed. Small/darker labels are the intermediate nodes’ and their IDs, larger/lighter labels are the customers and their IDs, with arrival times, while the circle represents the depot.

	Constant Speeds		**CS in TD**		**Time Dependent**		**Δ**
	<*T*>	σ	<*T*>	σ	<*T*>	σ	
test1	1.5186	*0.0008*	1.5255	*0.0187*	1.4396	*0.0178*	5.97%
test2	1.5558	*0.0002*	1.6641	*0.0630*	1.5042	*0.0426*	10.63%
test3	1.5974	*0.0023*	1.5784	*0.0445*	1.5000	*0.0070*	5.23%
test4	1.6270	*0.0005*	1.6768	*0.0131*	1.4973	*0.0063*	11.98%
test5	1.3472	*0.0032*	1.3750	*0.0046*	1.2892	*0.0075*	6.65%
test6	1.54005	*0.00319*	1.5360	*0.0442*	1.4652	*0.0284*	4.83%
test7	1.481206	*0.00584*	1.4061	*0.0503*	1.2728	*0.0991*	10.48%
test8	1.16595	*0.01211*	1.1846	*0.0311*	1.1185	*0.0106*	5.91%
test9	1.435494	*0.00888*	1.4305	*0.0580*	1.3427	*0.0093*	6.54%

Table 1: comparison of the best solutions obtained for the constant speeds model and the time dependent model.

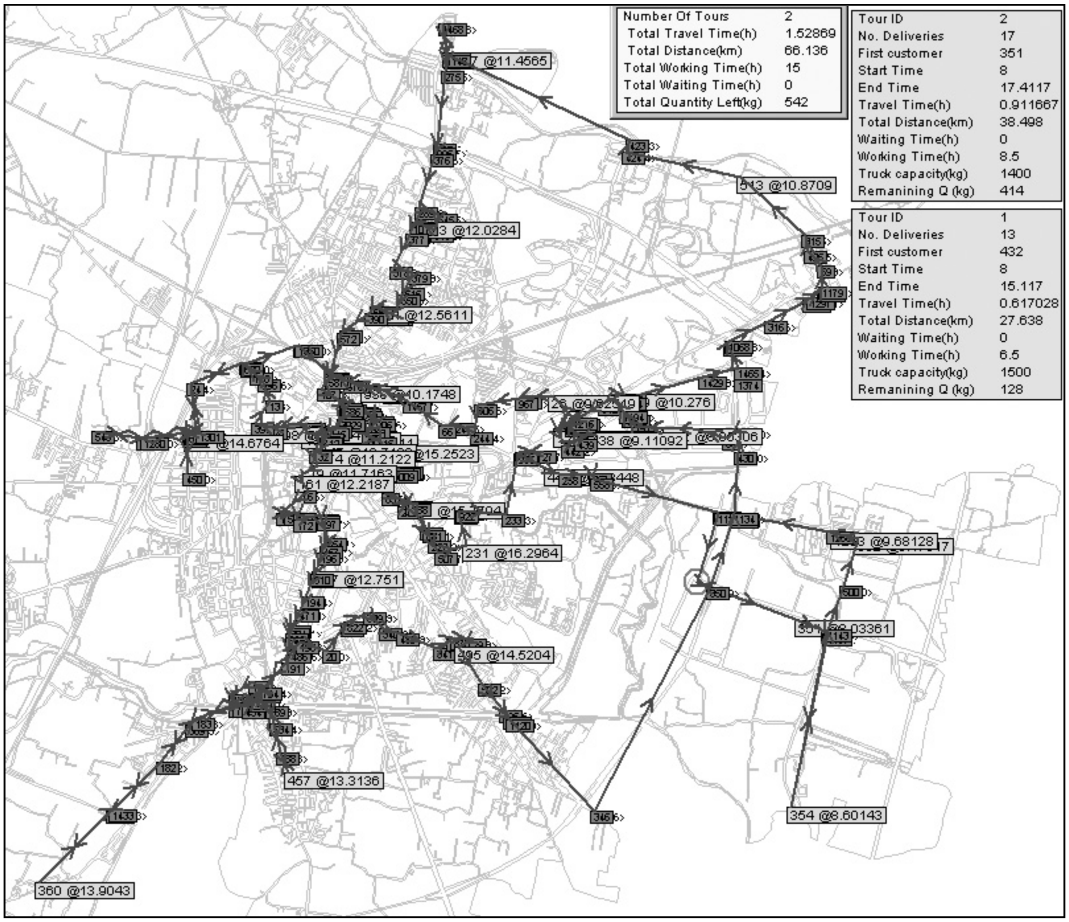

Figure 3: visualization of a TD solution. Small/darker labels are the intermediate nodes' IDs, larger/lighter are the customers' IDS with arrival times, while the circle represents the depot.

VII. CONCLUSIONS

We have presented a way to solve the VRP in presence of traffic conditions and taking into account a complete road network, through the integration of a time dependent VRP model (and algorithm) with a robust shortest path algorithm.

An application of the model to a real urban context has been done. From the results obtained, with the assumptions discussed and the data used, it is shown that the model is more accurate with respect to the optimization objective(s). We believe that the impact of time dependent components on VRP models is an important aspect to be taken into account, mostly when relating this model to urban contexts.

ACKNOWLEDGMENT

The work was co-funded by the European Commission IST project MOSCA, grant IST-2000-29557.

The authors would like to thank L. Lucietti (Fit Consulting), G. Valenti (ENEA), N. Garbin and P. Pandolfo (Interporto Padova) for the support in the data collection process.

REFERENCES

[1] M. Dorigo, G. Di Caro, L. M. Gambardella, *Ant Algorithms for Discrete Optimization*, Artificial Life, vol.5, n.2, 137-172 (1999)

[2] M. Dorigo, V. Maniezzo, A. Colorni, *The Ant System: Optimization by a Colony of Cooperating Agents.* IEEE Trans. Syst. Man Cybern., B 26(1) (1996), pp. 29-41.

[3] L.M. Gambardella, E. Taillard, G. Agazzi "*MACS VRPTW: Vehicle Routing Problem with Time Windows*." In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization, McGraw-Hill, London, UK, pp. 63-76, 1999.

[4] S. Ichoua, M. Gendreau and J-Y Potvin, *Vehicle Dispatching With Time-Dependent Travel Times*. To be publish in European Journal of Operational Research.

[5] A. V. Donati, L. M. Gambardella, N. Casagrande, A. E. Rizzoli, R. Montemanni *"Time Dependent Vehicle Routing Problem with an Ant Colony System"* internal IDSIA report IDSIA-02-03, 20-Jan-03.

[6] O.E. Karasan, M.C. Pinar and H. Yaman. The robust shortest path problem with interval data. *Computers and Operations Research*, to appear.

[7] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its applications*. Kluwer Academic Publishers, 1997.

[8] R. Montemanni and L.M. Gambardella. *An exact algorithm for the robust shortest path problem with interval data*. Computers and Operations Research, to appear.

[9] Comune di Padova, IL SISTEMA CARTESIO ANALISI DEI FLUSSI DI TRAFFICO, DATI 1999 – 2000, Padua Municipality.