Solution methods for a Decision Support System for vehicle routing with fluctuations in the travel time

Ana Osvald\(^a\), Raffaele Pesenti\(^b\), Walter Ukovich\(^a\)

\(^a\) DEEI and Excellence Center for Research in TeleGeomatics and Spatial Information
University of Trieste
Via A. Valerio, 10
I-34127 Trieste
Italy
Phone: (+39) 040 558 7848 Fax: (+39) 040 558 3460 E-mail: osvald@units.it, ukovich@units.it

\(^b\) DINFO – University of Palermo
Viale delle Scienze
I-90128 Palermo
Italy
Phone: (+39) 091 6598043 Fax: (+39) 091 6598043 E-mail: pesenti@unipa.it

Abstract – In this work we present a method to be used in decision support system (DSS), which solve the Vehicle Routing Problem with Time Windows. In order to find a good estimation of customers’ delivery time, we consider the fluctuations of the travel time. Solving the routing problems with time constraints without taking into account the fluctuations of the travel time during the day, can lead to serious errors in estimating customers’ delivery time with an additional cost for transportation companies. Initially we present a method to estimate the fluctuations of the travel time and than to solve the Vehicle Routing Problem with Time Windows and with Time Dependent travel time. In the estimation of the travel times, we do not use the Euclidean, but actual distance. The proposed method gives a solution in a short time and so it is suitable to be used also in web based DSS for logistic purposes.

I. INTRODUCTION AND PROBLEM STATEMENT

Many transportation companies, which have to deliver goods on time to their customers, use dedicated DSSs to schedule the fleet of vehicles and decide the best routes.

The DSSs solve Vehicle Routing Problem with Time Windows (VRPTW) \([1]\) i.e. have to find a route for each vehicle of the fleet in order to minimize the total distribution costs and satisfy each customer’s demand. Each vehicle must leave from and return to the central depot within a given time limit and serve the customers within the specific time window. Each customer must be visited exactly once by a vehicle, and the total customer demand for each route must not exceed the capacity of the vehicle. Some of the DSSs monitor also the location of the vehicles and perform a remote security control function.

Travel time cannot be considered fixed, e.g. the high amount of traffic leads to fluctuations in the travel time. Solving the problem without taking into account these fluctuations during the day can lead to serious errors in estimating customers’ delivery time. The use of the erroneous delivery time may reduce the satisfaction of the customers, and sometimes represents an additional cost for many transportation companies. Then in solving the problem, it is essential to consider the fluctuations of the travel time in order to find an acceptable route with a good estimation of customers’ delivery time.

The aim of this project is to develop a method to estimate the fluctuations of the travel time and to find a solution for the VRPTW with a good estimation of the customers’ delivery times.

II. REPRESENTATIONS OF THE TRAVEL TIME FLUCTUATIONS

The literature reports several representations of the travel time fluctuations. To each representation of the travel time fluctuation, a different representation of VRPTW corresponds. Given a network \(N = (V, A)\), a link travel time \(d_{ij}\) is associated with each arc \((i, j)\). The three most significant representations of the link travel times are as follows:

- Time-dependent link travel time; the travel time \(d_{ij}(t)\) from vertex \(i\) to vertex \(j\) is function of the departing time \(t\) from vertex \(i\). Such representation usually arises in models that try to capture time-of-day congestion effects.

- Stochastic times; travel times \(d_{ij}\) are random variables \([5]\). It usually arises in models, which are trying to capture some events, e.g., traffic light switching.

- Stochastic and time-dependent times \([8]\), \([5]\); the link travel times are random variables and with probability distributions functions of the time of the day. Consequently, the travel time on these types of links can be modeled as a continuous time stochastic process.

The most accurate representation of the fluctuations of the link travel time is the last one, but it is also the hardest to formulate and solve. Usually there is not enough deal with information (e.g. speed or traffic flow during the day) for each arc of the network.

An alternative method is to acquire the data from different fleets of vehicles and store the link travel times of each arc. Unfortunately, this method is applicable only for a restrict area and only few fleets are able to acquire and store this kind of data.
For a VRP including the fluctuations of the travel time, two representations are reported in literature: VRP with stochastic travel times reported in [6] and [4] and Vehicle Routing Problem with Time Dependent travel times (VRPTWTD). The VRPTWTD is an extension of the VRP, where the travel time between two locations depends on both the distance and on the time of the day.

Malandraki and Dial [13] introduced first in 1992 the time dependent Traveling Salesman Problem. Park et al. [10] presented the Time-dependent Vehicle Scheduling Problem (TDVSP) in which the travel speed between two locations depends on the passing areas and the time of the day. They proposed a model for estimating the time varying travel speed. Ichoua et al. [7] proposed a time-dependent model for a VRPTW, based on time-dependent travel speeds, which satisfies the FIFO assumption. They extended the tabu search heuristic developed by Taillard et al. [11] to solve the problem and performed some experiments to evaluate the model in static and dynamic environments.

Most of the presented work in the literature use the Euclidean distance to estimate the shortest time dependent travel time.

III. SOLUTION APPROACH

We split the solution of the problem into two parts. Initially we represent the road network and estimate the link travel times, and then we solve the VRPTW. To simplify the representation of the travel time (we have not a lot of information to deal with), we decided to represent the travel time fluctuations with time-dependent link travel time. We use the time-dependent link travel time to calculate the time-dependent Origin Destination (OD) matrix between each pair of customers, and, finally we solve the Vehicle Routing Problem with Time Windows and with time-dependent Travel Time.

The outline of algorithm is schematized as follows:

Estimate the time-dependent link travel time
Estimate the time-dependent minimum travel time OD matrix
Solve VRPTW with constant travel time approximating the real travel time with the average value of the time-dependent OD matrix
Correct the tours
Local optimization

A. Estimation of the time-dependent link travel time

To estimate the time-dependent link travel time, we describe the time dependencies of the travel times as follows: the arcs of the street network are classified according to their legal speed limit in 24 classes. Then, we divide a day into four time periods and to each period we assign a coefficient of variation of the speed. Four different periods are used, since during the day there are usually two periods of rush hour. In Fig. 1 the four time periods are shown (2nd and 4th periods represent the rush hours).

During each period, we consider constant the average speed (i.e. so speed is a step function of time). To assign the class of appertenance to the arc of the network, we consider the characteristics of the road (based on the division of the street in classes and their speed limits [13]) and the density of the population in the neighborhood of each street.

We estimate the shortest path between two nodes, given a start time, using an algorithm based on the FSM algorithm presented in [2]. This method guarantees the FIFO (first-in-first-out) propriety which is important in practice. The non-passing or FIFO propriety implies that two “identical” vehicles traveling on the same link arrive at the end of the link in the same order as they start, even if some congestion occurs during their route i.e. no vehicle cannot overtake others in the normal trip conditions.

Fig. 1. Division of the day in four time periods

B. Algorithm for the Vehicle Routing Problem with Time Windows and Time-Dependent travel time

Once evaluated the time-dependency of the network, we estimate the time-dependent travel time between each pair of customers and so we produce an OD matrix. To keep the time-dependency of the travel time, we calculate eight different minimum travel time OD matrices for eight different start times. For each time period defined on the road network (as in Fig. 1) we define two time periods (one at the beginning and one in the middle of each of four time periods).

Then the algorithm solves the classical VRPTW with soft time windows approximating the real travel time with the average value of the OD matrices. In the soft time windows [11] solutions that do not match the time windows constraints are allowed with an additive penalty.

In this step we minimize the following objective function:

\[ \text{o.f.} = \min \left\{ k_1 \sum_{k=1}^{K} \sum_{(i,j) \in A} \text{AvgTime}_{ij} X_{ij}^k + k_2 \sum_{k=1}^{K} \sum_{(i,j) \in A} \text{AvgDist}_{ij} X_{ij}^k + k_3 \sum_{i=1}^{N} \text{Penalty}_i \right\} \]

where:
- \( \text{AvgTime}_{ij} \) represents the average travel time between customers \( i \) and \( j \) calculated on eight OD matrices.
- \( \text{AvgDist}_{ij} \) represents the average distance between customers \( i \) and \( j \) calculated on eight OD matrices; also the distance may change in different periods
- \( \text{StDev}_{ij} \) represents the standard deviation of the travel time between customers \( i \) and \( j \).
- \( \text{Penalty}_i \); proportional to the violation of the time window of customer \( i \). In particular we define two coefficients: Advanced Cost Factor and Delayed Cost Factor and calculate the penalty as indicated in (2).

We use soft time windows since we think it is not reasonable to consider unfeasible a tour in which some time windows are violated for some minutes only. Above all, when traveling time cannot be known in advance, we make
many approximations in the estimation of the time-dependent travel time.

\[
\begin{align*}
\text{penalty} &= (a_i - S^i_k) \cdot \text{AdCostFactor}, \quad \text{for } a_i \leq S^i_k \\
\text{penalty} &= (S^i_k - b_i) \cdot \text{DeCostFactor}, \quad \text{for } S^i_k \geq b_i \\
\text{penalty} &= 0, \quad \text{for } a_i \leq S^i_k \leq b_i
\end{align*}
\]

Where

- \( S^i_k \): arrival time of vehicle \( k \) to customer \( i \)
- \( \text{AdCostFactor} \): Advanced Cost Factor
- \( \text{DeCostFactor} \): Delayed Cost Factor

To define an initial tour, we use a constructive heuristic based on the idea of Bräysy [9]. We construct different initial solution for different initial customers and for different insertion costs of the customers in the tour. Among the generated tours, we select the best one. The algorithm is schematized as follows:

```
Repeat for different Start Customers
  While (not Insert all Customers)
    Select Start Customer
    While ((Vehicle not full) & (exist possible insertions))
      select customer
      select position
      insert customer in tour
    Local optimization
  Select the best tour
```

Given an initial solution, the algorithm performs a corrective heuristic in order to take into account the actual time dependence of the travel time (the time indicated in the eight OD matrices). To this goal, it recalculates the customers’ delivery time for different start times and, when necessary, it relocates some customers in order to keep the tour feasible. If necessary, some customers are eliminated from the tour and inserted in another one.

```
Repeat for different start times and for all tours
  recalculate “time” and “cost” of all tour
Select the best tour
For all infeasible tour
  While tour not feasible
    Remove the customer with max penalty
Insert the removed customers
```

In the end, the algorithm performs a local search, which includes customer relocation, customer exchange, in tour 2-k opt and post insertion in order to improve the solution. In this step the travel time based on the eight OD matrixes is used.

### III. SOME PRELIMINARY RESULTS

In this section, some performance figures and examples of the algorithm are presented. The algorithm has been integrated in a “prototypal” DSS.

We divide the test of the algorithm in three parts. Initially, some examples of the time-dependent shortest path are presented and some computational results are given.

Then, the performances of the algorithm that solves the VRPTWTD are shown initially on the Solomon’s problem and finally on problems defined on real road network with randomly generated customers.

#### A. Some examples of the time-dependent shortest path

The algorithm for the time-dependent shortest path problem was tested initially on the road network of Italy and than on road network of Friuli Venezia Giulia. An Italian region networks were extracted from vector map. The network of Italy has 65,634 geo-referenced nodes; the network of Friuli Venezia Giulia has instead 54,677 geo-referenced nodes.

![Path with minimum travel time with and without fluctuations in the travel time](image)

When the coefficient of the variation on some streets during the rush hours is high (travel time during the rush hours is more than twice of the normal travel time), the time-dependent shortest path (minimizing the travel time) for a fixed start time may be very different from the time independent shortest path.

Figure 2 shows the shortest time independent path in superior part and the shortest time-dependent path in inferior part. Table II presents the travel times distances for both paths.

We tested the shortest time dependent path on a Pentium IV, 1.6GHz, 256MB PC. The algorithm takes less than a second to calculate the single pair and the single source (all shortest paths between one vertex in the graph and all other vertex in graph) problem on the Friuli Venezia Giulia network, and about 3 seconds on the Italy network. To calculate the shortest path between 100 nodes, it takes in average 10 seconds on the Friuli Venezia Giulia network.
TABLE I
EXAMPLE OF SHORTEST TIME DEPENDENT AND TIME INDEPENDENT PATH

<table>
<thead>
<tr>
<th></th>
<th>Travel time</th>
<th>Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time independent short path (SPP)</td>
<td>43 min 51 s</td>
<td>30.931</td>
</tr>
<tr>
<td>SPP calculated on time dependent network *</td>
<td>57 min 15 s</td>
<td>30.931</td>
</tr>
<tr>
<td>Time dependent shortest path</td>
<td>52 min 34 s</td>
<td>34.567</td>
</tr>
</tbody>
</table>

B. Test on the Solomon’s problems

To validate the static and time-dependent performance of the part of algorithm that solves the Vehicle Routing Problem with Time Windows and Time-Dependent travel times, we tested the algorithm on Solomon's 100-customer Euclidean problems [12]. In these problems, customer locations are generated within a $[0; 100]^2$ square. The problems are partitioned in six different sets, namely C1, C2, R1, R2, RC1 and RC2. The customers are uniformly distributed in the problems of type R, clustered in groups in the problems of type C and mixed in the problems of type RC. In the problems of type 1, only a few customers can be serviced on each route due to a narrow time window at the depot, as opposed to problems of type 2 where each route may have many customers.

TABLE II
TEST ON THE SOLOMON’S PROBLEM FOR VRPTWTD

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>C1</th>
<th>C2</th>
<th>RC1</th>
<th>RC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>1209,89</td>
<td>951,91</td>
<td>828,38</td>
<td>589,86</td>
<td>1384,16</td>
<td>1119,36</td>
</tr>
<tr>
<td>NV</td>
<td>11,92</td>
<td>2,73</td>
<td>10</td>
<td>3</td>
<td>11,5</td>
<td>3,25</td>
</tr>
<tr>
<td>Distance</td>
<td>1194,92</td>
<td>1030,36</td>
<td>1057,33</td>
<td>8505</td>
<td>1351,38</td>
<td>959,25</td>
</tr>
<tr>
<td>NV</td>
<td>10,92</td>
<td>2,82</td>
<td>10</td>
<td>3</td>
<td>11,88</td>
<td>3,88</td>
</tr>
</tbody>
</table>

TABLE III
COEFFICIENT OF VARIATION OF SOLOMON’S MATRIX

<table>
<thead>
<tr>
<th></th>
<th>1st period</th>
<th>2nd period</th>
<th>3rd period</th>
<th>4th period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st period</td>
<td>0,7</td>
<td>1,3</td>
<td>0,8</td>
<td>1,2</td>
</tr>
</tbody>
</table>

B. Algorithm for the Vehicle Routing Problem with Time Windows and Time-Dependent travel time

To validate the static and time-dependent performance of all parts of the algorithm, we tested the algorithm on the FVG road network. To this goal different sets of 30, 50 and 100 customers and a depot was randomly generated on the nodes of the interested area.

TABLE IV
COMPUTATIONAL RESULTS

<table>
<thead>
<tr>
<th></th>
<th>Estimation of the time fluctuations</th>
<th>VRPTW TD</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 customers</td>
<td>15</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>50 customers</td>
<td>20</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>100 customers</td>
<td>45</td>
<td>12</td>
<td>57</td>
</tr>
</tbody>
</table>

Table IV shows the computational results of the algorithm using a Pentium IV, 1.6GHz, 256MB PC.

Table V represents some examples when the problem was solved like a time independent problem and a time-dependent problem. For each problem, the average distance and average time are shown. For the time-dependent problem, also the penalty before and after the correction

TABLE V
NOT TIME DEPENDENT CASE

<table>
<thead>
<tr>
<th></th>
<th>Results obtained with VRPTW with constant travel time</th>
<th>VRPTW with time dependent travel time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>Distance</td>
<td>Time</td>
</tr>
<tr>
<td>235 km</td>
<td>337 km</td>
<td>859 km</td>
</tr>
<tr>
<td>489 min</td>
<td>457 min</td>
<td>2826 min</td>
</tr>
<tr>
<td>Penalty</td>
<td>Penalty</td>
<td>Penalty</td>
</tr>
<tr>
<td>3870 min</td>
<td>901 min</td>
<td>20429 min</td>
</tr>
</tbody>
</table>

The results of the test are indicated in Table II. The Advanced Cost Factor was fixed to 200 and Delayed Cost Factor was fixed to 300.

To test the time-dependent performance of the algorithm, we introduce the time-dependency in the Solomon’s matrixes as follows:

We introduce four different time intervals in order to represent the fluctuations of the traffic time.

*For the path obtained in the time independent case the “real” time for a given start time on the time dependent network was calculated.
calculated as in (2) with \textit{Advanced Cost Factor} = 1 and \textit{Delayed Cost Factor} = 1 is given.

IV. FUTURE DEVELOPMENT

Not all transportation companies have their own DSS. Especially the small transportation companies, which have to serve between 30 and 50 customers daily, usually use a web based DSS which integrates the same function like the classical dedicated DSS. The web based DSS acquires in real-time the information (position of the vehicle, vehicle speed, fuel level etc.) from different fleets of vehicles and for each of them performs a security remote control and calculates the best route. Collecting and analyzing all information allow obtaining additional information about the travel speed on each arc and of some unpredictable events (like traffic congestion, accidents etc.). So using historical data for these DSSs, it is possible to provide a more detailed representation of the fluctuations of the travel times.

In the same way, it is possible to monitor some important unpredictable events in real-time and to make a rerouting of the interested routes, in order to minimize the total delayed time, and suggest the vehicles to change the route on time.

V. CONCLUSIONS

In this paper we solve the VRPTW with the fluctuations of the travel time. We present a method to estimate the time-dependent road network and to solve the vehicle routing problem with time windows and time-dependent travel time.

The proposed method gives good initial solutions that may be used in dedicated DSSs for logistic purposes.

VI. REFERENCES