SENSITIVITY ANALYSIS OF VEHICLE ROUTING SOLUTIONS TO UNCERTAINTY IN TRAVEL TIMES

Gerrit K. Janssens, An Caris and Katrien Ramaekers

Operations Management and Logistics
Hasselt University – Campus Diepenbeek
Agoralaan – Building D
B-3590 Diepenbeek, Belgium
e-mail: {gerrit.janssens,an.caris,katrien.ramaekers}@uhasselt.be

Abstract

Various algorithms and heuristics exist for solving the vehicle routing problem. When the problem is enriched with time windows (either from the depot or imposed by the customers) the methods of finding optimal solutions become harder and most analysts turn to heuristics. The methods mostly assume deterministic travel times between customers, which might be an unrealistic assumption. Once a heuristics solution for the vehicle routing problem is found, the solution might be evaluated against various scenarios including uncertainty in travel times. The paper first models a single route as a project network and links the notions of slack and criticality with notions in the vehicle routing solution. Further evaluation of the solution is investigated if the uncertainty in travel time can be expressed as an interval between finite lower and upper bounds.

Keywords

Vehicle routing, Time windows, Sensitivity analysis

1. Introduction

It is very well known that profitability is low in freight transport compared to other industries. Profitable companies offer value-added logistics to attain a reasonable level of profitability. Most customers require deliveries within pre-specified time windows, in which loading and unloading operations have to take place. Due to Just-in-Time and Zero Inventory policies those windows might be extremely strict. In other cases, time windows are induced due to working hours or other reasons of convenience put forward by the customer. In such cases, the time windows might be not so hard and even some flexibility in time window may be negotiated against price setting. Some customers are characterised by delivery time flexibility. They specify their time windows in terms of hours, or even of days. A carrier might use this type of flexibility order to design cost efficient routes. The discipline of Operations Research offers solutions to this problem, which is known as a special case of the standard Vehicle Routing Problem (VRP), called the Vehicle Routing Problem with Time Windows (VRPTW).

The *standard Vehicle Routing Problem* (VRP) is concerned with finding a set of routes for a fleet of m vehicles, which have to service a number of customers n. Each of the n customers has a non-negative demand q_i $(i=1..n)$. The demand is served by a homogeneous set of vehicles all having capacity Q. In general, the objective is to minimize the cost of serving all customers. This cost is measured as total travel time or total travel distance. Vehicle routing problems arise in many real-life applications such as school bus routing, postal deliveries and food distribution. Transportation costs represent a non-negligible fraction of the purchase price of many products and services. Consequently, an efficient distribution of goods and services is of paramount importance.

This paper considers the *Vehicle Routing Problem with Time Windows* (VRPTW). Time constraints may be related to service requests. A time window then represents the time interval in which the service at a customer must take place. It has a lower bound and an upper bound. Savelsbergh and Sol (1995) mention the existence of implicit time windows. These time windows refer to the desired delivery time of customers. Customer inconvenience can be controlled by taking the implicit time windows into account. Taillard et al. (1997) further distinguish between soft and hard time windows. When the time window is soft, the vehicle can arrive before the lower bound or after the upper bound. If the vehicle arrives early, it has to wait to start its service. If a vehicle is late, a penalty for tardiness is incurred. When the time window is hard, late services are not allowed. In our discussion we assume the existence of hard time windows. Time constraints may also be related to vehicles. In reality vehicles are not available all the time. A vehicle departs from and arrives at a single depot. In this case a time window refers to the time interval in which the vehicle is available. We refer to both types of time constraints respectively as customer windows and depot windows. In the VRPTW, the total cost does not only include the total travel time, but also the waiting time incurred when a vehicle arrives early at a customer, and the service time at the customer's site (loading or unloading).

The VRPTW has been subject of intensive research. As solving the NP-hard VRPTW to optimality remains hard, research has focussed on heuristic approaches. Bräysy et al. (2004) divide these heuristics in three categories. First, construction heuristics build routes sequentially or in parallel until the vehicle's capacity is reached, without violating time window constraints. A second class consists of improvement heuristics. These heuristics try to improve the incumbent solution. Construction and improvement heuristics are discussed in detail in Bräysy and Gendreau (2005). Finally metaheuristics are used to guide construction and improvement heuristics to escape local optima. More details can be found in a recent survey on metaheuristics by Bräysy and Gendreau (2005).

In this paper a sensitivity analysis of a solution obtained by one of the existing heuristics is proposed. The solution has been obtained using deterministic, known travel times. However travel times may be uncertain due to traffic jams, weather conditions or unexpected situations. By allowing some variability in travel times in a route, we investigate the effect on total route time and feasibility of the service on that route. The main idea behind this approach is to offer a help in designing robust routes instead of optimal routes in terms of a specific objective, like total travel time or distance. Travel times are given interval numbers, i.e. a finite interval containing possible durations of the travel time between two customers. The proposed sensitivity analysis can be used as a management tool to support negotiations with customers concerning purchase conditions.

This paper is organised as follows. In the next section it is shown how a route, the customer time windows and the depot window can be represented as a project network. Also an interpretation is given of the critical activities and of slack. In section 3, an illustration of the procedure is given. In section 4, dealing with uncertainty in travel times is discussed. Finally, some conclusions are formulated and directions for further research are indicated.

2. A network representation of a vehicle routing solution

In this section we show that the solution of a VRPTW may be represented as a set of project networks. Each project network represents a single route. First, a short introduction to project networks is given. Then, the modelling of routes as project networks is described and an example is given.

A *project network* is a directed, connected, acyclic graph $G(A,V)$. In this graph V is a set of nodes and $A \subset V x V$ is a set of arcs. Two alternative representations of project networks are available: activity-on-arc project networks and activity-on-node project networks. In an activity-on-arc project network, each activity is represented by an arc. A node represents an event. An activity-on-node project network represents each activity by a node. The notion of event does not exist here. In this paper, we use the activity-on-arc project network representation. Two nodes are specified in the graph *G:* a start node and a finish node.

A *path* in *G* is defined as a path from start node to finish node. Let us denote by $P(n)$ the set of all paths in *G* from start node to finish node. A deterministic duration time t_{ij} is associated with each activity $(i,j) \in A$. The length of a path is the sum of the (estimated) durations of the activities on the path. The minimum time for completion of the project is equal to the length of the longest path through the project network. A path $p \in P(n)$ is critical if and only if it is the longest path in the graph *G*.

A *route* is defined as a sequence of customers to be visited by a single vehicle. Each route may be modelled as a project network. The activities of a vehicle include travel and service. A route is a sequence of (travel, service)-pairs of activities. Waiting times are introduced due to the synchronisation with the customers' time windows. The opening and closing times can be expressed as pairs of virtual activities in the network. A special path exists from the start node to the finish node expressing the time window of the depot, defining the earliest leaving time and latest arrival time of a vehicle. In any feasible solution this path should be the critical path (or one of the critical paths) in the network. In case the time window of the depot is not the critical path or one of the critical paths, it means that the total time of the route exceeds the time window of the depot, which does not lead to a feasible solution.

Let the set of nodes $V = \{v_1, v_2, \ldots, v_{2k}\}$ represent events, which start an activity of either of two types: a travel activity or a service activity. Denote the set of events related to travel activities as V^t and those related to service activities as V^s, where $V = V^t \cup V^s$ and $V^t \cap V^s = \phi$. Further $|V^t| = |V^s| = k$. Two special nodes are defined in V: v_1 is a source node denoting start of the travel activity from the depot, and v_{2k} is a sink node denoting start of the service activity at the depot(which in fact means the end of the travel activity towards the depot). The set of arcs A consists of several subsets related to various concepts of the route and its customer. A partition $A = (A^a, ,A^d,A^{ce},A^{cl})$ is defined where:

A^a : the set of arcs related to travel activities and to service activities, with
$A^a \subset (V^t X\ V^s) \cup V^s X\ V^t$

A^d: a singleton representing the opening time of the depot (depot time window),

A^{ce}: the set of arcs related to the earliest service times of the customers (lower bound of the customer time window),

A^{cl}: the set of arcs related to the latest service times of the customers (upper bound of the customer time window).

The procedure to construct the network consists of the following steps:

1. Construct the backbone of the network (A^a)
2. Add the depot time window to the network (A^d)
3. Add the earliest customer arrival times to the network (A^{ce})
4. Add the latest customer arrival times to the network (A^{cl}).

The *backbone* of the network is made up of arcs belonging to A^a. The backbone consists of alternating travel and service activities. The weights of the arcs correspond to resp. travel times or service times.

The *depot time window* is represented by a single arc $A^d = (v_1, v_{2k})$. The weight of the arc is equal to the opening time of the depot.

The *earliest service times* are represented by a set of arcs A^{ce}. The start of a service at a customer's site, earlier than the lower bound of its time window, is prohibited (in a feasible network) by adding an arc from the depot node v_1 to the node starting a service activity at a customer's site. This means that $A^{ce} \subset (v_1\ X\ V^s)$. The weight of the arcs corresponds to the lower bound of the customer window, under the assumption that the opening time of the depot is put equal to zero.

The *latest service times* are represented by a set of arcs A^{cl}. The start of a service at a customer's site, later than the upper bound of its time window, is prohibited (in a feasible network) by adding an arc from the node starting a service activity at a customer's site to the depot node v_{2k}. This means that $A^{cl} \subset (\ V^s X\ v_{2k})$. The weight of the arcs corresponds to the difference between the closing time of the depot and the upper bound of the customer window, under the assumption that the opening time of the depot is put equal to zero.

3. Illustration of the procedure for building a VRP-network

In Figure 1 a small network is described with one depot and three customers. A route is shown in which the vehicle moves from the depot to customer K1 (10 time units travel time), to customer K2 (12 time units travel time), to customer 3 (8 time units travel time) and back to the depot (26 time units travel time). The service time at all customers is equal to 5 time units. The customer and the depot have time windows as shown in Table 1.

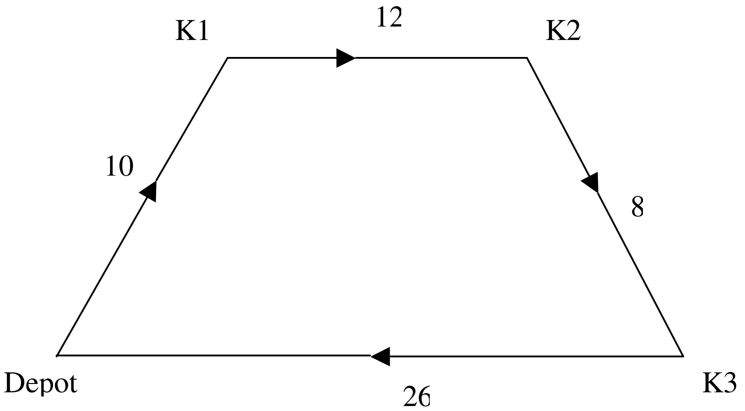

Figure 1: A single route visiting three customers

Location	*ETK*	*LTK*
K1	0	50
K2	10	50
K3	20	100
Depot	0	100

Table 1: Time windows

The route in Figure 1 can be represented as a sequence of activities. This is shown in the lower part of Figure 2. In the upper part of the same figure, the depot window is added to the network. In Figures 3 and 4 the customer windows are introduced: the earliest arrival times (denoted as ETK) and the latest arrival times (denoted as LTK) as given in Table 1. The latest arrival time for the customer K1 is equal to 50, which means the vehicle has to arrive before the time unit 50. The arcs, added in Figure 4, indicate the time left between the latest arrival of the vehicle and the upper bound of the depot window. For the customer K1 this means the arc has weight 100-50 = 50 time units.

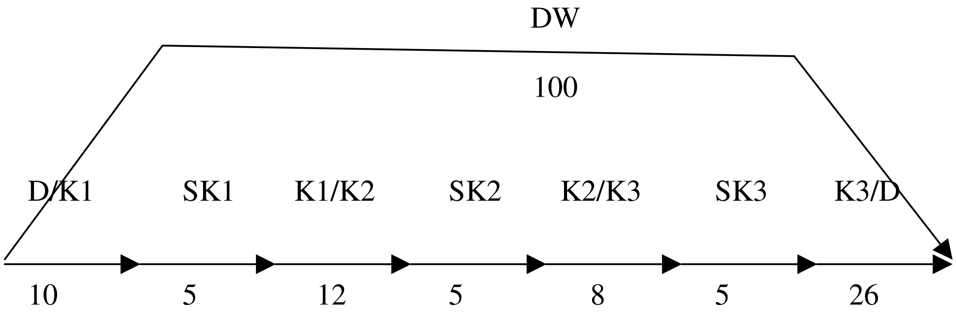

Figure 2: Travel and service activities and the Depot window introduction

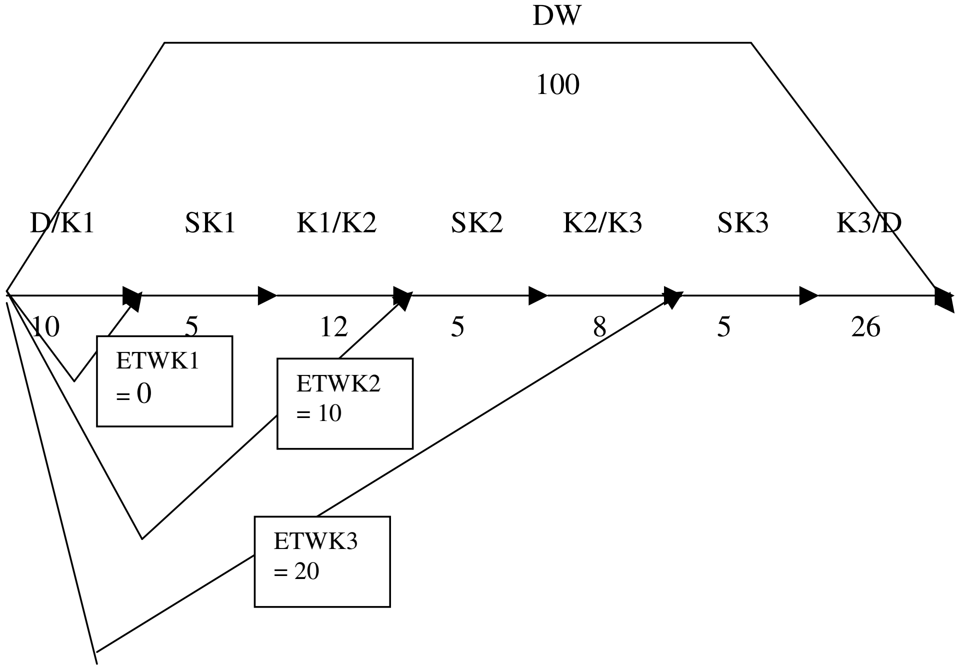

Figure 3: Customer window introduction: the earliest arrival time

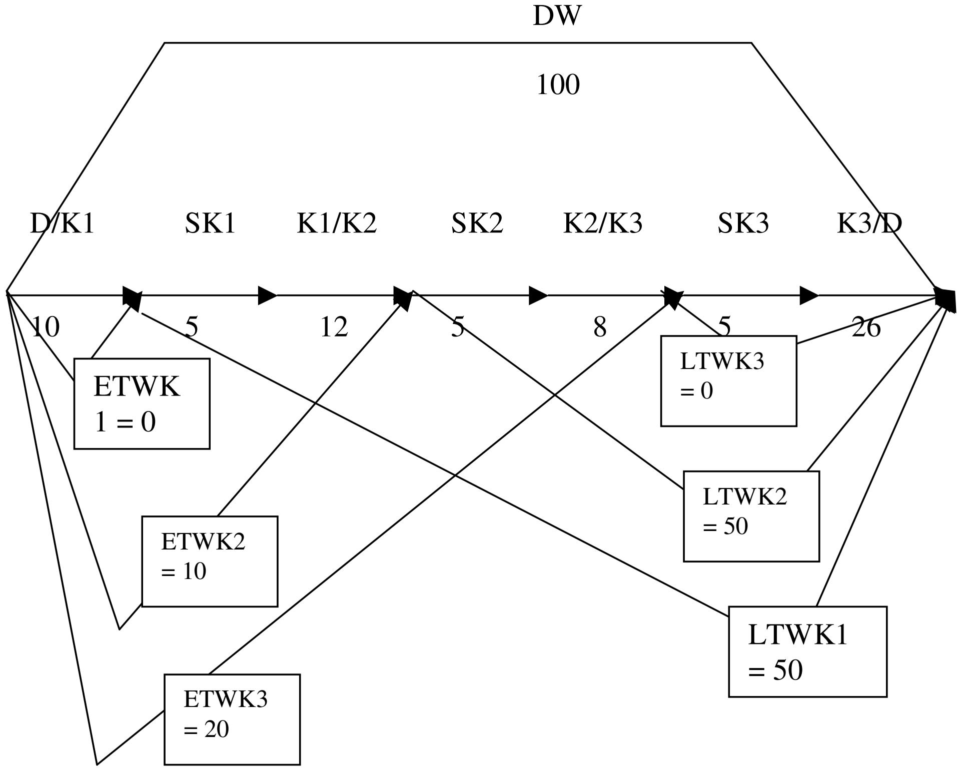

Figure 4: Customer window introduction: the latest arrival time

The first three columns in Table 2 show the information from Figure 5 in a form as required by a project planning algorithm. The last three columns show part of the output produced by such an algorithm. Table 2 shows that activity DW, indicating the depot window, forms the critical path and is the only critical path. This means that this solution is feasible in terms of the depot window.

Activity	Immediate Predecessors	Expected Time	Earliest Start	Latest Start	Slack
DW	-	100	0	0	0
D/K1	-	10	0	23	23
ETWK1	-	0	0	33	33
ETWK2	-	10	0	40	40
ETWK3	-	20	0	49	40
SK1	D/K1,EWTK1	5	10	33	23
LTWK1	D/K1,EWTK1	50	10	50	40
K1/K2	SK1	12	15	38	23
SK2	ETWK2,K1/K2	5	27	56	29
LTWK2	ETWK2,K1/K2	50	27	50	23
K2/K3	SK2	8	32	61	29
SK3	K2/K3,EWTK3	5	40	69	29
LTWK3	K2/K3,EWTK3	0	40	100	60
K3/D	SK3	26	45	100	29

Table 2: Route information ready to use in project network analysis

The project network offers a framework building scenarios for sensitivity analysis, for example by introducing other values for travel times to investigate their effect on feasibility. As an example, assume that the travel time from the depot to customer K1 takes 33 instead of 10 time units. A second critical path is found. The activities D/K1, SK1, K1/K2 and LTWK2 become the activities on the second critical path. The solution is still feasible, but several activities loose slack. When, for instance, the travel time, due to some uncertainty, from the depot to customer K1 takes more than 33 time units, the vehicle arrives late at customer K2. There is no more slack to counter uncertainty in the travel times from the depot to customer K1 and from customer K1 to customer K2. The latest arrival time at customer K2 has become critical.

4. Dealing with uncertainty in travel times

When representing the route as a project network, the essential assumption is that the duration times (either travel times or service times) are deterministic and known. In practice, this assumption is many times not fulfilled and the analyst should look for a representation with nondeterministic duration times represented with random variables or fuzzy numbers.

The uncertainty causes a problem when it comes to the notion of criticality. The PERT method finds critical activities using mean activity duration times. This method is a very simplified version of the reality and it is open to a lot of criticism. The evaluation of the probability with which a path is critical becomes very complicated from a computational point of view, even if all probability distributions of the duration times are known. The simplest way of representing uncertainty with respect to a duration time is by means of an interval.

Let the route again be represented as a graph $G(A,V)$. A travel or service activity is denoted by (i,j) where $(i,j) \in A$. But, in this case, duration times are given by means of interval numbers. The interval $I_{ij} = [a_{ij}, b_{ij}]$ contains possible duration times of (i,j) associated with a travel or a service activity.

Let us use the notion of interval-criticality, as introduced by Chanas and Zielinski (2002). A path p is interval-critical if there exists a set of times t_{ij} with $t_{ij} \in [a_{ij}, b_{ij}]$, $(i,j) \in A$, such that p is critical after replacing the interval times I_{ij} with the exact values t_{ij} by means of:

$$t_{ij} = \begin{cases} b_{ij} & \text{if} \quad (i,j) \in p \\ a_{ij} & \text{if} \quad (i,j) \notin p \end{cases}$$

In a route, the depot window should be a critical path. It can be checked easily, using the above method, whether this path is interval-critical. But, for practical purposes, this analysis does not solve the problem. The backbone may be interval-critical, but other paths, including time windows, may also be interval-critical. That would mean that the route cannot be executed in a feasible way. A guarantee for feasible execution can be realised only when the depot window is interval-critical, but no other path is interval-critical. The analysis would require to enumerate all paths in the route, which is a task of high computational complexity.

From a human-computer interaction point of view, the method however can be of some help. When a human analyst suspects that the interaction of two uncertainties might cause problems of feasibility, only a limited number of paths need to be generated and tested. If the analyst would like to have an answer including all types of uncertainties, the method is not feasible.

5. Conclusion and future directions

It has been shown that a single route of a solution to a Vehicle Routing Problem with Time Windows can be represented as a project network. In such a way the feasibility of the route can be tested when other values of travel times are realised. The notion of criticality is the key to this test. Sensitivity towards travel times can be performed for only one customer trip at a time. When upper and lower bounds on the travel times as an interval are known, a method has been described to find out whether a path is interval-critical, but the decision whether a route can be executed in a feasible way is a task of high computational complexity.

An algorithm needs to be developed in which an interval is given for each travel time and the feasibility of the route can be checked. The next paragraphs formulate some hints how a solution to this problem may be found.

The dynamic behaviour of the route and the assessment of its feasibility in the case of uncertain travel times might be studied using the Petri net formalism. In terms of structure, it has been shown that a project network might be represented as a marked graph, which is a special case of a Petri net. Ordinary Petri nets do not include any notion of time and are aimed to model only the logical behaviour of systems by describing the causal relations existing between events. A timing specification is required if we want to consider performance, scheduling or real-time control. Also in our case, in the presence of time windows and travel times, the Petri net model has to be extended with time aspects.

Two basic models for handling time have been defined: time Petri nets and timed Petri nets. A time interpretation can be associated with either places or transitions. The Time Petri net is more general (Merlin and Farber, 1976). In this case, Time Petri Nets are the most suitable class of Petri nets as they allow transitions to fire only within time intervals. A time interval includes an earliest and latest firing time, once the transition is enabled.

In Petri nets without timing information, transitions, which are enabled, can fire any moment but will not necessarily do. It is not specified when, if ever, transitions will fire, and why. The occurrence of transitions is not planned. The causal structure of transitions - pre and post conditions - is described, but the control over the occurrence of transitions is not specified by the design. The Time Petri Net case is slightly different. Each transition has a static timing interval (α_i, β_i) and a transition t_i may not fire for a period of time of at least α_i after it has been enabled, and it has to fire if it has been enabled for a period of time of β_i.

Continuity, a very specific nature of time implies that an exhaustive enumeration of possible states (with time-information) is not possible, as it will almost always be an infinite enumeration. Instead of handling individual states, states can be grouped in state classes. Formally, the *i*-th state class is defined by its marking M_i and its firing domain D_i , where D_i contains all enabled transitions $t_{i,j}$ with their relative firing interval ($\alpha_j < t_{i,j} < \beta_j$). Classes are pairs (M,D) in which M is a marking and D is a firing domain. Firing rules for states classes and ways to compare classes for equality have been defined (Berthomieu and Menasche, 1983). Using these state classes, a finite representation of an infinite number of reachable states can be generated, which will be mentioned further by the class reachability graph, similar to the reachability graph of states in (non-time) Petri nets.

Timing aspects in Petri nets have been applied to various real-life situations with success, mostly in performance analysis. Examples are abundant in the area of manufacturing systems, either with deterministic times or with stochastic times. In time-related risk assessment, Time Petri nets have been applied in Leveson and Stolzy [13] and in Deceuninck and Janssens [14].

The Petri net has as many starting places as there are customers (or depot) with time window constraints plus one additional starting place for representing the sequence of travel and service activities. In the partial schedule the transitions represent events with the meanings: distance travelled, service started, and service ended. The places between the transitions can have maximally one token. If the places contain a token, they have the meanings: transporting, waiting, and servicing.

In the paths, which model the time window progress, two transitions appear representing the beginning and starting events of the window. Between both transitions, a token in the place shows that the window is open. A second place behind the second transition indicates that the window is closed. This indicates the realisation of an infeasible schedule. Synchronisation between the partial schedule and the time window path appears as follows. The place with meaning 'waiting' and the place with meaning 'window open' appear as input places to the transition with meaning 'service started'. If both places contain a token, the schedule is still feasible and the dynamic process can continue. The final place in the partial schedule represents the vehicle's return at the depot.

By executing the Petri net, two types of information are obtained. If the final place in the partial schedule is not reached, then the partial schedule belongs to a non-feasible route. If the final place is reached, then the execution shows all time intervals in which a service may be started at a customer location. Both cases have their practical application.

It seems that this modelling approach might bring an answer to the feasibility question of a route in terms of uncertainty of travel times. The combination of representing the route as a project network with the inclusion of the uncertainty into the dynamical behaviour of the route by means of Time Petri nets will allow us to build an algorithm to solve the problem. The algorithm might be based on further ideas developed by Berthomieu and Diaz (1991). Also the complexity of the algorithm needs some further study.

References

Berthomieu, B. and Menasche, M. (1983) An enumerative approach for analyzing Time Petri Nets. In: R.E.A. Mason (ed.), *Information Processing 83*, North Holland, 41-46.

Berthomieu, B. and Diaz, M. (1991) Modeling and verification of time dependent systems using Time Petri Nets. *IEEE Transactions on Software Engineering* **17** (3) 259-273.

Bräysy, O. and Gendreau, M. (2005) Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science* **39** (1) 104-118.

Bräysy, O. and Gendreau, M. (2005) Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science* **39** (1) 119-139.

Bräysy, O., Hasle, G. and Dullaert, W. (2004) A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research* **159** 586-605.

Chanas, S. and Zielinski, P. (2002) The computational complexity of the criticality problems in a network with interval activity times. *European Journal of Operational Research* **136** 541-550.

Deceuninck, W. and Janssens, G.K. (1994) Risk analysis using Time Petri Nets. *Investigacao Operacional* **14** (1) 91-102.

Leveson, N.G. and Stolzy, J.L. (1987) Safety Analysis Using Petri Nets. *IEEE Transactions On Software Engineering* **SE-13** (3) 386-397.

Merlin, P.M. and Farber, D.J. (1976) Recoverability of communication protocols - Implications of a theoretical study, *IEEE Transactions on Communication*s **COM-24** 1036-1043.

Savelsbergh, M.W.P. and Sol, M. (1995) The general pickup and delivery problem. *Transportation Science* **29** (1) 17-29.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J. (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* **31** (2) 170–186.