Journal of the Operational Research Society (2010) 61, 515–522 0160-5682/10

www.palgrave-journals.com/jors/

Vehicle routing and scheduling with time-varying data: A case study

W Maden[1], R Eglese[2]* and D Black[2]

[1]University of Huddersfield, Huddersfield, UK; and [2]Lancaster University Management School, Lancaster, UK

A heuristic algorithm is described for vehicle routing and scheduling problems to minimise the total travel time, where the time required for a vehicle to travel along any road in the network varies according to the time of travel. The variation is caused by congestion that is typically greatest during morning and evening rush hours. The algorithm is used to schedule a fleet of delivery vehicles operating in the South West of the United Kingdom for a sample of days. The results demonstrate how conventional methods that do not take time-varying speeds into account when planning, except for an overall contingency allowance, may still lead to some routes taking too long. The results are analysed to show that in the case study using the proposed approach can lead to savings in CO_2 emissions of about 7%.

Journal of the Operational Research Society (2010) **61,** 515–522. doi:10.1057/jors.2009.116
Published online 14 October 2009

Keywords: vehicle routing; distribution; heuristics; environment

Introduction

Vehicle Routing and Scheduling algorithms have traditionally been developed for road networks, where the average speed of the vehicles on each road link is estimated as a constant value. Sometimes different average speeds are used for different types of road or roads in particular areas, but often the average speed is not altered over the planning period. In practice, traffic flows may be subject to congestion that leads to lower average speeds at particular times of the day or night. The average speeds may also vary due to the day of the week, the time of year and other influences such as weather conditions.

There is now much more traffic information available that makes it possible to plan vehicle journeys taking account of congestion that is predictable from the traffic patterns of the past. This approach will not be able to take account of unexpected events that may cause congestion such as an accident, but regular congestion due to volume of traffic or long-term road works can be predicted from past data.

Such data can be used to create a Road Timetable that shows the shortest time between customers when the journey is started at different times. In some cases the shortest-time path may change at different times of day due to the pattern of congestion on the road network. Issues regarding the construction of such a Road Timetable are discussed in Eglese *et al* (2006).

This paper describes a vehicle routing and scheduling algorithm, called LANTIME, that is able to accept data from a Road Timetable and will construct a set of vehicle routes that aims to minimise the total time required to deliver goods from a depot to a set of customers subject to a set of constraints. The possible constraints include the capacities of the vehicles, the times available for each driver and vehicle and time-window constraints for the customer deliveries.

The main aim of this paper is to present a case study where the algorithm has been applied using real data for a vehicle fleet delivering electrical wholesale items in the South West of the UK. The purpose of this case study is to analyse the effects of using Road Timetable data compared with routing and scheduling where this information is not available. The results are compared in terms of the total distances travelled, the time required for each vehicle route and the CO_2 emissions.

The paper is organised as follows. The next section discusses the academic literature relevant to vehicle routing and scheduling using time-varying data. In the following section, the details of the LANTIME scheduler are described. The case study and results are described next followed by the conclusions and remarks on further research.

Literature related to time-varying vehicle routing models

Vehicle Routing Problems have been studied extensively in the Operational Research literature. A good overview of exact and heuristic methods, together with descriptions of some application areas is to be found in Toth and Vigo (2002). Most models treat the time between customers as constant values, but a relatively small number of articles have been published that consider these times as varying according to the time of travel. One example is the paper by Fleischmann

*Correspondence: Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX, UK.

et al (2004). Modern traffic information systems are described and a framework is presented for using time-dependent travel times within dynamic vehicle routing problems of different types. Computational tests based on the traffic in Berlin show that the use of constant average travel times can lead to significant underestimation of the total travel times. The paper also reports on how this can lead to missing delivery time-window constraints. Ichoua *et al* (2003) examine a model for vehicle routing and scheduling based on time-dependent travel times. These travel times are derived from speeds that vary according to different times of the day and it is shown that this approach has the advantage of maintaining the First In, First Out (FIFO) property. This means that a vehicle starting to travel down a single road will always reach the end of the road before another similar vehicle that starts its journey down the same road at a later time. The model is tested on generated data and illustrates the benefits from taking time-varying travel times into account. Eglese *et al* (2006) demonstrate how the use of time-varying data can affect results for a hypothetical distribution operation, using real speed data on a road network in the north of England.

Van Woensel *et al* (2008) describe an approach incorporating queueing models for vehicle routing where travel times may vary dynamically due to congestion. The models are built on the relationships between the speed, flow and density of the traffic on any road. The approach can be used where direct observations of speeds are not available, but data on flows can be obtained or easily estimated.

There have also been developments in dynamic vehicle routing for adjustments to vehicle routes reacting to changing traffic conditions in real time. Examples include Ichoua *et al* (2006) and Taniguchi and Shimamoto (2004). When the customer demands are for specific orders that need to be loaded at the depot, as occurs in our case study, then although these approaches may allow routes to be improved after a vehicle has left a depot, planning which customer is to be served on each vehicle route still needs to be determined when the vehicles are loaded based on expected travel times.

Formulation and algorithm design

The problem considered is a conventional single-depot Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW, where the speeds over each arc or edge of a network are not varying with time, can be formulated in a variety of ways including a multi-commodity network flow model, where the binary variables indicate whether an arc between two nodes is traversed by a certain vehicle, or a set-partitioning formulation, where the variables used in the master problem indicate possible paths. The formulation used in this paper is based on a different standard formulation that is capable of being expanded later to include more practical routing requirements as well as variable travel time information.

The set of customers is denoted by N and the set of vehicles by K. For each customer $i \in N$, the service time requirement is $s(i)$, the demand required is $w(i)$ and a time window $[e_i, l_i]$ is specified for beginning of service. The set of vehicles K represents a homogeneous fleet of vehicles. For each vehicle $k \in K$, its capacity is W, its starting time is τ and its maximum working time is D.

The travelling times between locations are all known and fixed and defined as $c(i, j)$ where $\{i, j\} \subset N \cup \{0\}$, and where by convention 0 represents the depot. The cost associated with traversing an arc is represented as the travelling time.

For each individual vehicle, k, $R_k = [v_0^k, \ldots, v_{m^k}^k]$ the path of locations required to be serviced by the vehicle k where $v_i^k \in N$ for $i = 1, \ldots, m^k - 1$, and, v_0^k and $v_{m^k}^k$ are identified with 0. That is, the route starts and ends at the depot. m^k represents the number of stops on the complete path, including depot stops, for vehicle k.

The starting time of the service for a customer $i \in N$ is denoted by $a(i)$ and the waiting time before a service is denoted as $b(i)$. The starting time is defined as the maximum of the leaving time from the previous stop plus the travelling time from the previous location and the start of the time window e_i. By convention $a(0) = \tau$ and $s(0) = 0$. The waiting time is dependent on both thestarting time and the time-window start time.

$$a(v_i^k) = \max \left\{ \begin{array}{c} e_{v_i^k} \\ a(v_{i-1}^k) + s(v_{i-1}^k) + c(v_{i-1}^k, v_i^k) \end{array} \right\} \quad (1)$$

$$b(v_i^k) = \max \left\{ \begin{array}{c} e_{v_i^k} - a(v_i^k) \\ 0 \end{array} \right\} \quad (2)$$

The only constraint applied directly to each individual customer is that the service is started within the given time window. This is a time window on beginning of service and not on completion of service.

$$e_i \leqslant a(i) \leqslant l_i, \qquad \forall i \in N \quad (3)$$

The working time required on each route must be under the total work time available, D. In the VRPTW the working time includes travelling time, service time and waiting time.

$$\sum_{i=0}^{m^k-1} c(v_i^k, v_{i+1}^k) + \sum_{i=1}^{m^k-1} s(v_i^k) + \sum_{i=1}^{m^k-1} b(v_i^k) \leqslant D \qquad \forall k \in K \quad (4)$$

Constraint (4) ensures that the vehicle returns to the depot on time. Equation (5) defines the variable $\tilde{R}_k$ as the path of locations required to be serviced by the vehicle k with the depot stops removed.

$$\tilde{R}_k = R_k \backslash \{v_0^k, v_{m^k}^k\} \qquad \forall k \in K \quad (5)$$

$$\bigcup_{k \in K} \tilde{R}_k = N \quad (6)$$

The equality Constraint (6) ensures that all customers N are dealt with by the individual customer sets $\tilde{R}$.

$$\sum_{i=1}^{m^k-1} w(v_i^k) \leqslant W \qquad \forall k \in K \quad (7)$$

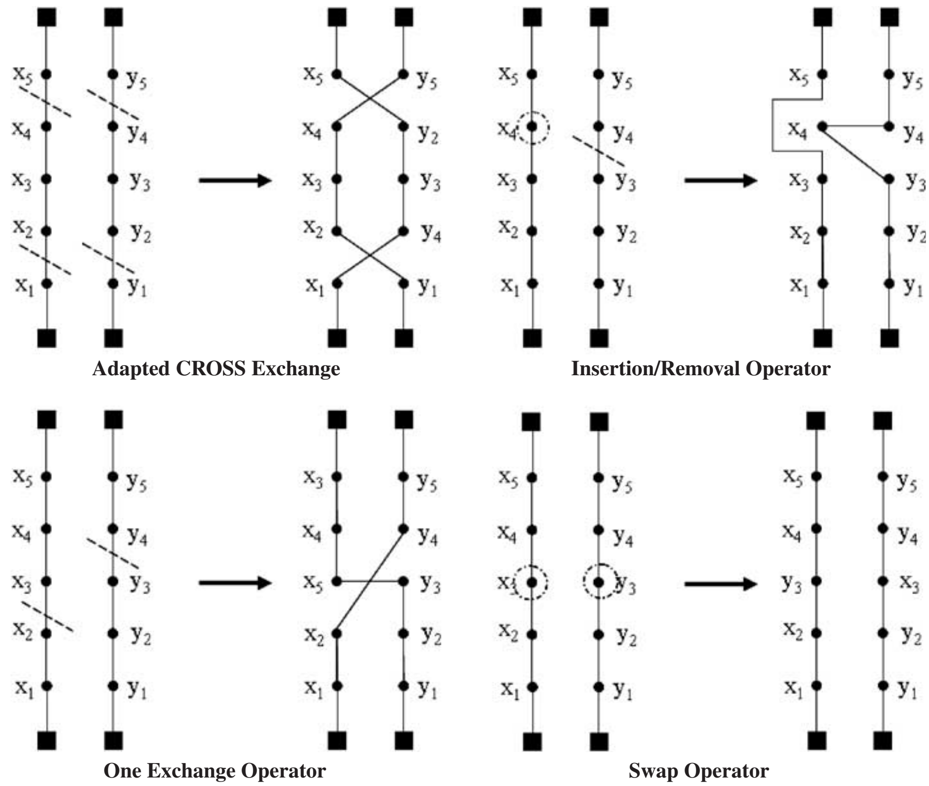

Figure 1 The four types of neighbourhood move.

Constraint (7) ensures the capacity of the vehicle is not exceeded by the demand requirements of the individual deliveries placed on route R_k.

The objective function (8) is for a fixed-fleet version of the problem where the objective is to minimise the travelling cost over all the routes.

$$\text{Min} \sum_{k \in K} \sum_{i=0}^{m^k-1} c(v_i^k, v_{i+1}^k) \tag{8}$$

Full details of the algorithm used can be found in Maden (2006). The main points are summarised below.

An initial solution is created for the VRPTW using a parallel insertion algorithm following Potvin and Rousseau (1993). This initial solution is a starting solution from which tabu search explores neighbouring solutions. The insertion algorithm must ensure that the solution is feasible with respect to all constraints.

The insertion algorithm forms a five-step process as outlined below.

Step 1: Seed the vehicles' routes with one delivery each (using a seeding criterion).

Step 2: For each remaining delivery, find the cost if a vehicle were to complete that delivery from the depot with no other customers on its route.

Step 3: Find best location on best vehicle's route for all remaining deliveries.

Step 4: Make the best insertion based on a savings criterion.

Step 5: If all routes are full or all jobs are scheduled, terminate; else go to step 3.

The initial vehicle route is started by seeding it with the delivery farthest from the depot, though other variations are possible.

The initial solution is then improved using a tabu search algorithm.

The algorithm described here uses four possible neighbourhood operations: CROSS Exchange, insertion/removal, one exchange and swap. The first of these is an adaptation of a neighbourhood move proposed by Taillard *et al* (1997). The four neighbourhoods are illustrated in Figure 1.

The insertion/removal, one exchange and swap operators can be considered as special cases of the adapted Cross-Exchange operator. They represent smaller neighbourhoods and so take less time to search completely than the adapted Cross Exchange. In all these moves, when a set of deliveries is moved together, a check is carried out to discover whether it is better to reverse the sequence of deliveries for those moved.

The algorithm randomly selects which neighbourhood to explore at each stage, according to probabilities assigned in advance.

The tabu list is not fixed but varies as proposed by Gendreau *et al* (1994); a move added to the tabu list at time t remains restricted until time $t + \theta$ where θ is randomly selected from an interval $[\underline{\theta}, \overline{\theta}]$. Gendreau *et al* (1994) suggests that this approach virtually eliminates the probability of cycling between solutions. A standard aspiration criterion overrides the restriction implied by the tabu list if the move leads to a new best solution.

A long-term memory structure is used to help diversify the search into new areas. The tabu search objective has an additional component to represent the long-term memory cost $M(x_{trial})$ of the proposed move that leads to solution x_{trial}.

$$cost_{trial} = M(x_{trial})f(x_{trial}) \tag{9}$$

The long-term memory factor reflects the number of times each delivery that is involved in the move, leading to the trial solution x_{trial}, has been involved in previous completed moves. The memory structure keeps a tally of the number of times a delivery has been involved in a completed move. For a move only involving swapping a single delivery, a value of 1 is added to that delivery tally; for a move involving more than one delivery, $1/(Num\_deliveries\_involved)$ is added to the tally for all deliveries involved in the completed move.

The component $M(x_{trial})$ for Equation (9) is calculated using a sum of the tally counts of deliveries that are involved within the move, divided by the number of deliveries involved in the move; then this value is expressed as a fraction of the total number of iterations, *total_num_iterations*. β is a constant used to control the effect that long-term memory has on the cost function.

$$M(x_{trial}) = \left(\frac{\sum tally\_count/num\_deliveries\_involved}{total\_num\_iterations} + 1\right)\beta \tag{10}$$

The memory component of the cost function is set to equal 1 when determining if the possible solution meets the aspiration criteria.

The tabu search objective tries to locate a solution x which leads to the minimum search cost. The search cost includes the original VRPTW objective $f(x)$, the memory cost $M(x)$ and the function $P(x)$ that is a measure of the infeasibility of solution x.

$$M(x)f(x) + \alpha P(x) \tag{11}$$

The parameter α is dynamically adjusted throughout the search. α is initially set at 1 and in the same way as Gendreau *et al* (1994), every ξ iterations, $\alpha = 2\alpha$ if all previous ξ solutions were infeasible, and $\alpha = \alpha/2$ if all previous ξ solutions were feasible.

In expression (11) the cost of the solution $f(x)$ is calculated using Equation (8) and the memory cost is calculated using Equation (10). The penalty cost is calculated by determining how much extra time, including the time windows for each delivery and the working time allowance for each vehicle, is needed to complete the solution. The total extra time is denoted as $P(x)$.

The algorithm can be summarised as follows:

Step 1: *Initialisation*
Generate initial solution, x_{now},
Copy initial solution to best overall feasible solution, $x^* = x_{now}$,
Ensure tabu list is empty, and long-term memory values are defaulted to zero

Step 2: *Neighbourhood selection & termination*
If termination criterion is met, **End** Search and **Return** x^*,
Else Randomly select the neighbourhood N to be used this iteration

Step 3: *Choice*
Randomly generate x_{trial} from $N(x_{now})$
If first trial value in neighbourhood, $x_{best} = x_{trial}$
If $P(x_{trial}) = 0$ *and* $f(x_{trial}) < f(x^*)$, (Aspiration criterion)
    Copy $f(x_{best}) = f(x_{trial})$, $P(x_{best}) = P(x_{trial})$, $x_{best} = x_{trial}$ *Go to* Step 4,
Else if move from x_{now} to x_{trial} is not set tabu,
    if $M(x_{trial})f(x_{trial}) + \alpha P(x_{trial}) < M(x_{now})f(x_{now}) + \alpha P(x_{now})$,
        Copy $f(x_{best}) = f(x_{trial})$, $P(x_{best}) = P(x_{trial})$, $x_{best} = x_{trial}$ *Go to* Step 4,
    Else if $M(x_{trial})f(x_{trial}) + \alpha P(x_{trial}) < M(x_{best})f(x_{best}) + \alpha P(x_{best})$,
        Copy $M(x_{best}) = M(x_{trial})$, $f(x_{best}) = f(x_{trial})$, $P(x_{best}) = P(x_{trial})$, $x_{best} = x_{trial}$
Continue Step 3 until search of neighbourhood $N(x_{now})$ is exhausted

Step 4: *Update*
$x_{now} = x_{best}$
Update Memory table and Tabu List, increment iteration count
If $P(x_{best}) = 0$ and $f(x_{now}) < f(x^*)$, $x^* = x_{now}$
Restart Step 2

Experiments were carried out on standard benchmark VRPTW test problems from Solomon (1987) to determine the best values of the parameters to be used with this algorithm. Full details are in Maden (2006).

Some further extensions were made to the algorithm so that it could be used in practice for case study problems.

The most important extension is to allow the time to travel between locations $c(i, j)$ to vary according to the time that the journey is started. The optimum times, distances and routes were previously calculated and stored as a Road Timetable, using the approach described in Eglese *et al* (2006).

In addition, the algorithm was modified to ensure that the routes obeyed current driving legislation by inserting breaks for a driver when required. The law lays down several conditions:

(i) there must be a driving break of 45 min every 4.5 h (270 min),
(ii) if the total working time is greater than 6 h (360 min) then a 30-min break must be taken,
(iii) if the total working time is greater than 9 h (540 min) then a 45-min break must be taken.

The two types of breaks can be taken simultaneously. For example, if a driver takes a 45-min driving break then he would have taken enough rest to work over 9 h. Conversely, if a driver works over 6 h and takes a 30-min break, then the next driving break need only be for 15 min.

Finally, a penalty function is included in the objective if any deliveries are not included in the routes for the fixed-vehicle fleet. However the final solutions produced need to service all deliveries. For the experiments in the case study, if any deliveries were not included then the vehicle fleet was increased until a feasible solution was found that included all deliveries. In other practical situations, some deliveries may have to be rescheduled for another time or passed on to another carrier.

The algorithm was also modified to allow a heterogeneous fleet, the capacity constraint was extended to cover two elements (eg weight and volume) and problems involving pickups and deliveries were included, but these extensions were not required for the case study described in this paper.

The implementation of the algorithm changes significantly with the inclusion of variable travel time information. As highlighted in papers notably Malandraki and Daskin (1992), Fleischmann *et al* (2004) and Ichoua *et al* (2003) there are other complications that need to be considered when extending an algorithm to make use of time-dependent travel times. This is because a local neighbourhood move involving deliveries near the start of a route could have a significant effect on the timings later on, and makes it more difficult to determine the effect of the neighbourhood move on the objective function efficiently.

Ichoua *et al* (2003) use an approach of approximating the effect on later jobs by using a similar approach to the one outlined when using static data, using the creation of an additional time window to give an approximation of the effect. The M best moves given by the approximation are then calculated exactly. The final selection is based on the exactly calculated values.

Malandraki and Daskin (1992) determined that using a typical insertion algorithm with variable travel time information was too computer intensive to be of practical use, but as computer speeds and power increase, this problem becomes less acute. In this algorithm, the effect of a neighbourhood move on the objective function is determined exactly.

Another issue arises from the approximation of a continuous change in the time to travel between any two nodes on the network using a set of discrete time bands. The issue is similar to that of ensuring that the FIFO property holds when constructing the Road Timetable as described in Eglese *et al* (2006). When travelling from A to B, the time of arrival at B could be earlier if the vehicle waits at A until the starting time falls in a time band where the speeds are faster. Although the construction of the Road Timetable ensures that this will not happen when the best route between A and B does not change, if the best route between A and B changes with the change to a new time band, then this phenomenon can occur. In practice, delaying the departure from A should never lead to an earlier arrival at B for any particular route. If the time bands used in constructing the Road Timetable are relatively narrow, then this should not be a significant problem in practice. The experiments were all conducted using 15-min time bands that are narrow enough so that the changes in speed on a road between neighbouring time bands are generally small and provide a good approximation to the continuous case. Using narrower time bands would provide an even better approximation but increase the computational burden.

Case study

The case study is based on the distribution system of an electrical goods wholesaler. For its operation in the South West of the UK, items need to be taken from its regional distribution centre in Avonmouth to a set of customers. The area covered includes Worcester, Swindon and Portsmouth to the east, the whole of south Wales and the south west of England to the tip of Cornwall. The operation is carried out on a daily basis Monday–Friday. The vehicles used are all 3.5 tonne GVW box vans, so there are no restrictions on the roads on which they may travel. As the items of electrical equipment are relatively small and light there are no effective constraints on the capacity of the vans. However each driver is available only for a maximum 10-h working day including the statutory breaks for driving time and working time. There are no time-window constraints for the deliveries, other than that they must all be delivered on a particular day.

Demand data were obtained for a sample of nine separate days. The number of customers served per day ranged between 40 and 64. The number of vans required is normally

Table 1 Results for run sets A and B

Date	*N*	*Run*	*Total dist. (km)*	*Total time (min)*	*Work Time per vehicle (min)*					
					1	2	3	4	5	6
9/6/2008	53	A	1990	2295	599	581	557	598		
		B		2619	**660**	**680**	**631**	**648**		
11/6/2008	51	A	2037	2377	180	485	599	573	541	
		B		2736	198	539	**729**	**640**	**630**	
12/6/2008	64	A	2466	2934	538	571	573	598	152	501
		B		3342	**605**	**628**	**637**	**716**	168	587
13/6/2008	49	A	1964	2337	595	596	569	577		
		B		2620	**668**	**674**	**628**	**649**		
16/6/2008	55	A	2458	2853	472	555	593	397	588	249
		B		3186	518	**622**	**648**	450	**666**	282
17/6/2008	57	A	1987	2454	195	514	575	589	582	
		B		2744	214	570	**637**	**651**	**671**	
18/6/2008	48	A	2076	2393	160	457	589	590	597	
		B		2672	179	505	**662**	**656**	**670**	
19/6/2008	40	A	1523	1782	438	574	195	575		
		B		2018	481	**654**	218	**666**		
20/6/2008	43	A	1668	2045	593	511	390	551		
		B		2381	**754**	567	445	**615**		

up to seven, though additional vans and drivers are available if required.

In order to construct the corresponding Road Timetables, data were supplied from ITIS Holdings whose Floating Vehicle Database contains speeds of vehicles on roads that have been captured through tracking devices on the vehicles. Road Timetables were constructed for each day's set of customers based on the speeds observed in 96, 15-min time bins averaged over a 3 month period in 2007. For comparison purposes, Road Timetables were also constructed using the speeds found at times of the day when the traffic was free flowing or uncongested.

In cases where the location of a customer was off the main road network covered by the ITIS data (typically on an estate or very minor road) then the time for a vehicle to transfer between the location and the main road network was estimated based on the straight line distance to a node on the network in the way described in Eglese *et al* (2006). This time was generally a very small proportion of the total journey time.

For each day's data, initially two runs were made using the LANTIME algorithm. The first set of runs (A) used the uncongested speeds that did not vary by time of day. The results from this correspond to what would be expected from a conventional vehicle routing and scheduling system where the speeds on each road are constant. The second set of runs (B) give the results of using the routes planned in (A), but with the varying speeds taking account of the effects of congestion at different times of day.

The results are shown in Table 1.

For each of the 9 days sampled, when the routes that were constructed using constant uncongested speeds from A were used and tested using the actual time-varying speeds in B, at least one of the routes constructed became infeasible, because the total time required exceeded the 10 h allowed, sometimes by a considerable margin. These instances are indicated by bold type in the table. Over all the runs, the percentage of routes that went over time was 65% and the total extra time required to finish those routes was an average of 57 mins. In practice this may require the payment of overtime payments and could also lead to delivery problems if some deliveries are delayed beyond the normal time when customers can accept deliveries.

To overcome this problem, one strategy used by planners is to use constant speeds, but slower than the uncongested speeds to make an allowance for congestion. With a constant speed model, this will not reflect the actual variations in speed at different times of the day, but the approach might be expected to make sufficient allowance so that the actual route lengths do not exceed the 10 h allowed. Using slower speeds may lead to plans requiring more vehicle routes and drivers than strictly necessary.

In order to analyse this strategy, the algorithm was run again using constant speeds, where the original uncongested speeds were reduced by 10%. The resulting plans were then tested using the actual time-varying speeds in the same way as the previous set of runs in B. The results from these runs are shown in Table 2 and are labelled 'P-10%'.

The results from these runs show that even with this allowance, many of the routes planned still exceed the 600-mins time limit. The percentage of routes that went over time is 44% and the total extra time required to finish these routes is an average of 20 mins. In this case, the allowance has not been enough to provide a set of routes that are likely to be satisfactory.

Another set of runs was then carried out, again using constant speeds, but this time where the speeds were reduced

Table 2 Results for run set P-10%

Date	*Total dist. (km)*	*Total time (min)*	*Work Time per vehicle (min)*					
			1	2	3	4	5	6
09/06/2008	1979	2659	236	582	**620**	**640**	581	
11/06/2008	2328	2966	**624**	500	**606**	**602**	**634**	
12/06/2008	2875	3665	**631**	**600**	653	572	**600**	**608**
13/06/2008	1884	2633	450	**603**	**636**	435	508	
16/06/2008	2513	3213	547	595	240	**608**	**643**	579
17/06/2008	1995	2757	**604**	597	593	**634**	328	
18/06/2008	2034	2670	**609**	324	**627**	506	**604**	
19/06/2008	1485	2003	**607**	286	502	**607**		
20/06/2008	1679	2328	598	578	583	570		

Table 3 Results for run sets P-20% and C

Date	*Run*	*Total dist. (km)*	*Total time (min)*	*Total CO_2 (kg)*	*Work Time per vehicle (min)*							
					1	2	3	4	5	6	7	8
09/06/2008	P-20%	2309	3005	497	565	**613**	463	541	545	279		
	C	2155	2862	463	586	438	247	600	397	594		
11/06/2008	P-20%	2558	3237	548	484	516	**607**	519	552	560		
	C	2458	3076	528	580	536	505	332	528	595		
12/06/2008	P-20%	3546	4373	763	542	577	548	551	539	551	533	532
	C	2802	3632	604	596	198	597	566	501	595	578	
13/06/2008	P-20%	2027	2801	438	557	475	505	593	557	113		
	C	1955	2692	424	588	499	520	505	580			
16/06/2008	P-20%	3171	3928	680	541	554	582	557	516	589	589	
	C	3153	3864	676	513	600	482	589	544	596	540	
17/06/2008	P-20%	2145	2945	465	299	448	563	560	516	559		
	C	2226	3064	483	452	460	546	583	512	512		
18/06/2008	P-20%	2581	3257	557	536	593	564	538	484	543		
	C	2210	2896	477	458	600	173	592	514	559		
19/06/2008	P-20%	1749	2300	376	493	**604**	362	296	545			
	C	1504	2002	324	596	318	582	506				
20/06/2008	P-20%	1710	2386	370	206	541	576	467	596			
	C	1777	2361	383	593	299	600	532	337			

by 20%. The resulting plans were then tested using the actual time-varying speeds as before. The results from these runs are shown in Table 3 and are labelled 'P-20%'.

A final set of runs (C) show the results from planning the routes using the LANTIME algorithm with the time-varying speed data and these are also given in Table 3.

When the change in speeds is further reduced by 20% for planning, then in all instances apart from three, the routes are within the 600 mins maximum and the extra time required is only an average of 8 mins. However on each sample day, this requires the use of an additional van route compared to run set P-10%.

In contrast with the previous results, using the LANTIME algorithm with the time-varying speeds (set C) produced results where all routes were completed within the 10 h limit. As for run sets P-10% and P-20%, in many cases an additional van route was needed compared to the original plans in run sets A and B. The results from set C demonstrate that using LANTIME provides a more reliable basis for planning routes in terms of the time needed to complete each route.

Table 3 also presents an estimate of the CO_2 emissions for each of the run sets. These have been calculated using the speed along each road in the route using the emissions function provided in the National Atmospheric Emissions Inventory. This can be accessed online (at www.naei.org.uk). For this case study, the figures used are for Euro II Diesel LGVs. The tables provided allow an estimate to be made of various emission factors in terms of emissions per kilometre for different average speeds. They may not fully reflect actual emissions that may be affected by the amount of irregularity in speed, weight of load, road inclines and other factors. For this case, as the customer orders are relatively light compared to the weight of the van, no attempt has been made to modify the function for the weight of goods carried at each stage of the route. A good discussion of the issues involved in the estimation of CO_2 emissions from road freight transport can

Table 4 Summary statistics

Run	*Total dist. (km)*	*Total time (min)*	*Total CO_2 (kg)*
P-20%	21796	28232	4694
C	20236	26431	4363

be found in McKinnon and Piecyk (2009). The evaluations could have been made for other harmful pollutants, but only CO_2 emissions have been evaluated as a major contributor to the greenhouse effect. All the calculations have been made in grams and then rounded in the results to the nearest kilogram.

When the total emissions per day are compared for run sets P-20% and C, the total emissions for run set C were usually lower than those for run set P-20%, though not in every case.

Table 4 summarises the total distance, total time required and the total CO_2 emissions for run sets P-20% and C. It shows that the total distance travelled and the total time required for run set C were less than those for P-20%. The reduction is about 7% when compared with the P-20% run set. This is because the LANTIME algorithm using the time-varying speed data tends to avoid routes where congestion is high, speeds are low andCO_2 emissions are relatively high. By searching for the fastest routes, it tends to avoid congestion and only uses longer routes when the vehicles can travel faster at a speed closer to the optimum for emissions per kilometre.

Conclusions and further research

The case study results demonstrate the effect that consideration of time-varying speeds can have on a real distribution operation. Ignoring this issue can lead to routes that suffer delays producing duties for drivers that are unacceptably long. In this case, there were no time-window constraints on service deliveries, but where these are important then ignoring time-varying speeds can lead to missing delivery time windows as well.

The analysis has shown that being able to plan routes using traffic information that provides time-varying speeds for the roads in the network can also lead to some reduction in the levels of CO_2 emissions produced compared with plans based on constant speeds and a general contingency allowance for congestion. The reduction observed for the case study was about 7% when compared with a planning method using constant speeds and a common contingency allowance for all roads.

Although this study suggests that finding the set of routes that minimise the total time in a network with time-dependent travel times may lead to reductions in emissions compared to conventional approaches, the LANTIME algorithm still does not plan routes that directly minimise pollution. The next step in this research is to modify the algorithm to find the set of routes that will produce the set of routes that directly minimise the pollution, still taking account of the time-varying speeds due to patterns of congestion.

Acknowledgements—This research was supported by the Engineering and Physical Sciences Research Council Green Logistics project: Grant No. EP/D043328/1.

References

Eglese RW, Maden W and Slater A (2006). A Road Timetable™ to aid vehicle routing and scheduling. *Comput Opns Res* **33**: 3508–3519.

Fleischmann B, Gietz M and Gnutzmann S (2004). Time-varying travel times in vehicle routing. *Trans Sci* **38**: 160–173.

Gendreau M, Hertz A and Laporte G (1994). A tabu search heuristic for the vehicle routing problem. *Mngt Sci* **40**: 1276–1290.

Ichoua S, Gendreau M and Potvin J-Y (2003). Vehicle dispatching with time-dependent travel times. *Eur J Opl Res* **144**: 379–396.

Ichoua S, Gendreau M and Potvin J-Y (2006). Exploiting knowledge about future demands for real-time vehicle dispatching. *Trans Sci* **40**: 211–225.

Maden W (2006). *Models and heuristic algorithms for complex routing and scheduling problems*. PhD Thesis, Lancaster University Management School.

Malandraki C and Daskin MS (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Trans Sci* **26**: 185–200.

McKinnon AC and Piecyk MI (2009). *Measurement of* CO_2 *emissions from road freight transport: A review of UK experience*. Working Paper, Logistics Research Centre, School of Management and Languages, Heriot-Watt University, Edinburgh EH14 4AS. Also available to download from http://www.greenlogistics.org/ (accessed 3 July 2009).

Potvin J-Y and Rousseau J-M (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur J Opl Res* **66**: 331–340.

Solomon M (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Opns Res* **35**: 245–265.

Taillard ED, Badeau P, Gendreau M, Guertin F and Potvin J-Y (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Trans Sci* **31**: 170–186.

Taniguchi E and Shimamoto H (2004). Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Trans Res Part C* **12**: 235–250.

Toth P and Vigo D (eds) (2002). *The Vehicle Routing Problem*. SIAM: Philadelphia.

Van Woensel T, Kerbache L, Peremans H and Vandaele N (2008). Vehicle routing with dynamic travel times: a queueing approach. *Eur J Opl Res* **186**: 990–1007.

Received October 2008;
accepted August 2009 after one revision