Finding a minimum cost path between a pair of nodes in a time-varying road network with a congestion charge

Liang Wen[a,*], Bülent Çatay[b], Richard Eglese[a]

[a] Lancaster University Management School, UK

[b] Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

Abstract

The minimum cost path problem in a time-varying road network is a complicated problem. The paper proposes two heuristic methods to solve the minimum cost path problem between a pair of nodes with a time-varying road network and a congestion charge. The heuristic methods are compared with an alternative exact method using real traffic information. Also, the heuristic methods are tested in a benchmark dataset and a London road network dataset. The heuristic methods can achieve good solutions in a reasonable running time.

Keywords: Heuristic; Minimum cost path; Time-varying; Congestion charge

1. Introduction

There has been much research to provide solutions for solving vehicle routing and scheduling problems. However, most of the research published is based on models where the time between nodes on a road network is considered as fixed. In practice, this is not the case and the speed taken for any journey may vary significantly by the time of the day, the day of the week and the season of the year in which the journey takes place. For example, the traffic conditions at 1am are often different from those at 8am which is in the rush hour for commuters and as a result a journey starting at

*Corresponding author:
E-mail address: l.wen@lancaster.ac.uk (Liang Wen)
Telephone: ++44 1524 592758
Fax: ++44 1524 592060

8am may take a much longer time than the same one starting at 1am. The results from fixed speed models may produce schedules which lead to more vehicles spending time and fuels in congested traffic, which gives rise to further congestion and associated environmental costs. The fixed speed models may even lead to infeasible solutions for the practical problem. There are also economic and social costs due to missing delivery time windows and overtime costs when routes take longer time than planned.

In this paper, we propose two heuristic methods to determine the minimum cost path between a pair of nodes on a time-varying road network. Our cost structure includes three parts to the cost of the journey. One is the fuel cost which is influenced by the speed; the second is the driver cost which is related to the travelling time. The last is the congestion charge, when applicable. A congestion charge scheme (CCS) is a scheme of surcharging users of a transport network in periods of peak demand to reduce road congestion and decrease travelling times within the congestion charge zone. CCS helps to reduce pollution factors within the zone. CCS may be applied in a certain area during a certain time of the day. In general, some tolls may be collected on certain roads at certain times or at rates that change with time.

The implementation of CCS is an important factor when designing vehicle routing and scheduling systems. It can greatly affect the minimum cost paths on a real transportation network in a time-varying setting. This paper is motivated by the need for determining the cost minimizing paths on real size networks fast and with little computational effort as the existing algorithms are inefficient due to their CPU memory and/or computational time requirements. We test the performance of the proposed heuristic methods against an exact method to validate their applicability using real traffic information. The rest of the paper is organized as follows: the next section provides a literature review of previous work that utilizes time-varying (time-dependent) travel times. Section 3 describes the optimization problem of finding the minimum cost paths in a time-varying road network and presents the two heuristic methods. Section 4 investigates the performances of the heuristics on a benchmark dataset and compares them to the exact method devised in Chabini (1998). The following section presents a case study where we discuss the computational results obtained through the proposed heuristic methods on a real-life London data set. The last section presents conclusions and directions for further research.

2. Literature related to Time-Dependent Travel Time Models for Vehicle Routing and Scheduling

Ichoua et al. (2003) give a brief literature review of the time-dependent vehicle routing problem (TDVRP). They point out that the TDVRP models proposed by Hill and Benton (1992) do not satisfy the "first-in-first-out" (FIFO) property as they represent the travel time as a step function of time. Ichoua et al. (2003) introduce a time-dependent model for the vehicle routing problem with time windows based on time-dependent travel speeds which satisfies the FIFO property. They implement a parallel tabu search approach and test its performance both in dynamic and static environments. The scheduling horizon is divided into three time intervals by taking into account the rush hours and three types of road are considered. The results show that the time-varying model provides significant improvements compared to the model with fixed travel times. Ichoua et al. (2003) also develop a dynamic vehicle routing model to adjust the vehicle routes that react to continuously changing traffic conditions in real time.

Eglese et al. (2006) show how the use of time-varying data can affect results for a hypothetical distribution operation and develop a model to use the historical data to construct a Road Timetable that shows the shortest time between nodes when the journeys start at different times. The shortest times and routes may vary as the speed of travel on individual roads may differ significantly by the hour of the day, by the day of the week and by the season of the year. The paper describes a case study using real speed data on a road network in the north of England.

Eglese and Black (2012) demonstrate the importance of speed with reference to vehicle routing. A route generated for optimizing distance may emit more CO_2 or other polluting gases due to slower speeds than a longer alternative route. So, reducing the travelling distance does not always reduce the CO_2 emissions.

Bektas and Laporte (2011) concern vehicle routing problems (VRPs) with different objective functions, but does not consider time-dependent travel times. They compare four different models with different objectives including distance minimizing, energy minimizing, weight load minimizing and cost minimizing objectives. They provide some numerical analyses on some small instances and conclude that minimizing the energy consumption is not equivalent to minimizing the cost. As the labour cost constitutes a major proportion of the total cost, the cost minimizing model focuses on the labour cost in order to reduce the total costs. Advances in engine technology lower the amount and cost of emissions hence lowering the overall total cost. Minimizing the cumulative weight load only does not necessarily imply energy minimization, particularly when time window restrictions are applied.

Cooke and Halsey (1966) present a dynamic programming algorithm for solving the all-to-one (from all nodes to one destination for any possible departure time) fastest path problem with time-dependent travel times over the discrete time horizon (0, T]. The algorithm is based on Bellman's optimality conditions for the shortest path problem (SPP) with time-dependent travel times. Based on the formulation proposed by Cooke and Halsey, Ziliaskopoulos and Mahmassani (1993) develop a label-correcting algorithm to solve the time-dependent SPP. Labels are stored in a vector, one for each time interval, and are maintained for every node and updated in a label-correcting fashion, i.e. the labels are upper bounds to the optimum path label until the algorithm terminates. The main characteristic of the algorithm is to scan all labels of a node for all possible departure times. A scan-eligible list is created to maintain all the nodes with the potential to improve at least one label of any node in the network. Note that Bellman's optimality principle is not satisfied on time-varying transportation networks when the objective is to determine the minimum cost path. In other words, real transportation networks do not satisfy the Cost Consistency property when waiting at the nodes is not allowed, i.e. leaving a node earlier does not necessarily cost less than leaving it later, in particular if a CCS is applied.

The stochastic dynamic network extension of the problem has been addressed by Hall (1986), Cheung (1998), Fu and Rilett (1998), Miller-Hooks and Mahmassani (1998, 2000, 2003), Pretolani (2000), and Huang and Gao (2012). These studies deal with finding the path with the shortest expected travel time on a dynamic network where the arc travel times are time-dependent random variables and the probability distributions vary with time. Multi-objective approaches have also been proposed within the context of hazardous materials transportation for finding the non-dominated paths by considering uncertain attributes such as travel time, population exposure, accident probability (e.g. Chang et al., 2005). Since this topic is beyond the scope of the paper, we refer the interested reader to Erkut et al. (2007) for a comprehensive review.

Although the literature on finding the shortest or fastest path is vast, there are few articles that attempt to determine the minimum cost path on a time-varying network environment. Pallottino and Scutellà (1998) present an algorithmic paradigm, namely Chrono-SPT, for the dynamic shortest path problems using discrete models, i.e., they assume that the time varies in a discrete set. They analyze different implementation schemes by performing chronological type visits only on the non-redundant portion of an acyclic space-time network. Based on the reverse implementation of the Chrono-SPT and time-dependent SPP algorithm of Ziliaskopoulos and Mahmassani (1993) Miller-Hooks and Yang (2005) present reoptimization techniques to determine the updated fastest paths from all origins to a single destination when future travel times on the time-varying networks change. Their experimental results show that these techniques may provide substantial savings in the computational effort over determining the paths starting from scratch.

Chabini (1998) proposes an algorithm called the Decreasing Order of Time (DOT) algorithm to solve all-to-one fastest path problem and minimum cost problem in a time-varying road network by applying a backward labeling algorithm visiting the entire space-time network also using time discretization. The DOT algorithm has an exact computing complexity of ($SSP + nM + mM$) where SSP is static shortest path, n is the number of nodes, m is the number of arcs and M is the number of time intervals. It compares the performance of DOT with three dynamic adaptations of label correcting algorithms using three types of data structures for node candidates list: the Deque data structure of Pape (1974) as described in Ziliaskopoulos and Mahmassani (1993), the 2-queue data structure of Gallo and Pallotino (1988) and the 3-queue data structure in Chabini (1998). The DOT algorithm has a better performance than the other three algorithms. However, both Chrono-SPT and DOT algorithms fail on real-size networks. We shall compare the performances of DOT and our heuristic methods using a real road network in a later section of the paper.

3. Finding the Minimum Cost Path between Two Nodes

3.1 Preliminaries

We assume the driver will drive as fast as possible, subject to the speed of the traffic and any given maximum speed. So, the speed of the vehicle is always equal to the link speed in the dataset and is not a decision variable in the model.

The FIFO property means that if a vehicle leaves node i to go along arc (i,j) starting at time t, the time to arrive at node j for any other vehicle leaving node i and travelling along (i,j) after time t is later than the first vehicle. Provided that the FIFO property holds, Dijkstra's algorithm is able to find the optimal path between locations when the objective is to minimize the time. However, when the objective is minimum cost rather than shortest time where the cost changes with time, Dijkstra's algorithm cannot guarantee finding the optimal path.

In the following example we illustrate how Dijkstra's algorithm fails to find the optimal solution in a time-varying road network. The objective is to find the least cost path from node *A* to node *E*. Each arc is labeled with (*c*,*t*) where *c* is the cost of traversing the arc and *t* is the travel time in minutes. Each node is given a label which keeps the cost of a path from the source node to that node and the corresponding time is recorded. Figure 1 shows that Dijkstra's algorithm gives the optimal path as *A*→ *B*→*C*→*D*→*E* and its cost is 6. The labels of the nodes that are made permanent are shown in bold and the dashed arcs represent the optimal path.

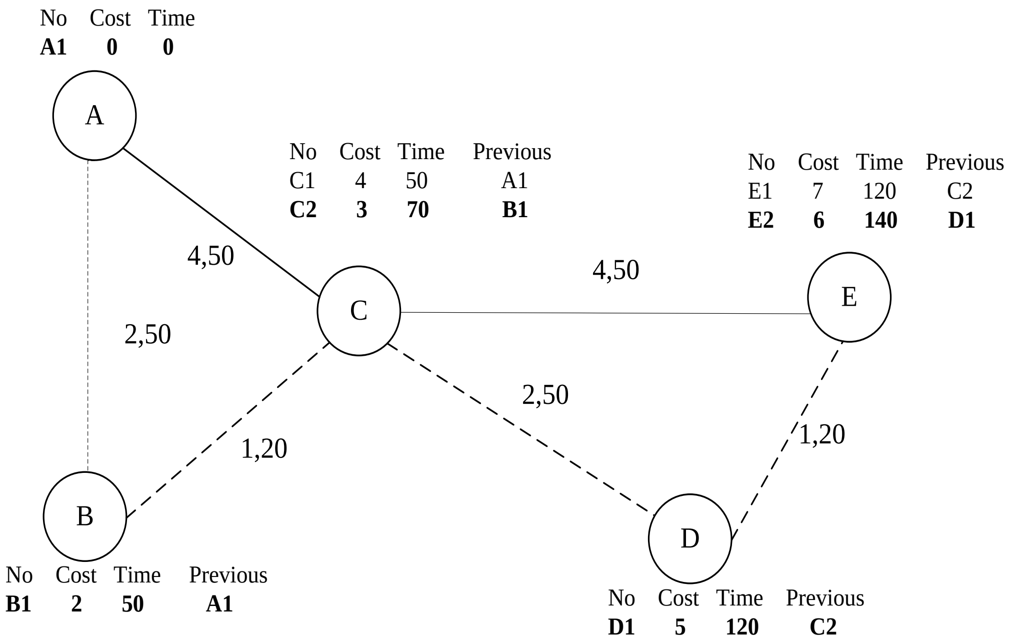

Figure 1 Example using Dijkstra's algorithm

Now suppose that if travel along the arc *CE* starts at or before time 60, then the cost is only 1 unit and the travel time is 30. The standard Dijkstra's algorithm finds the same path as in the previous example; however, the optimal path should be *A*→*C*→ *E* with a cost of 5. So, Dijkstra's algorithm fails to provide the minimum cost path in this case.

3.2 Model

3.2.1 Assumptions

To facilitate the understanding of the model we list all of the underlying assumptions as follows, although some of them may have been mentioned in previous sections:

- Time is divided into discrete time bins and the speed on an arc is assumed constant during each time bin.
- Waiting at any node is prohibited.
- A route is allowed to cycle back through a previously visited node. When the time varying road toll is applied, the driver may cycle in order to avoid the road toll.
- The driver will drive as fast as possible, subject to the maximum speed on an arc which is determined by the traffic conditions and is estimated from traffic data for the same time period in the past. Thus, the speed is not a decision variable in the model.
- A congestion charge may be applied in a certain area during a certain time period.

3.2.2 Definitions

Let $G=(N,A,D,C,V,K)$ be a directed network where $N=\{1,..,n\}$ is the set of nodes, $A=\{(i,j)\in N\text{x}N\}$ is the set of arcs, $D=\{d_{ij} \mid (i,j)\in A\}$ is the set of arc lengths, $C=\{c_{ij}(t) \mid (i,j)\in A\}$ is the set of time-dependent arc travel costs, $V=\{v_{ij}(t) \mid (i,j)\in A\}$ is the set of time-dependent speeds which are calculated in km/h and $K=\{k_{ij}(t) \mid (i,j)\in A\}$ is the set of time-dependent arc travel times. Speed is constant on an arc during each time bin as in the model used by Sung et al. (2000) and Ichoua et al. (2003). If a vehicle crosses one or more time bin boundaries before reaching the end of an arc, the speed is changed at each boundary and the fuel cost for traversing the arc is obtained by summing the fuel used for each section travelled at a different speed.

The travel cost includes three parts: the fuel cost, driver cost and congestion charge. The fuel cost $F_{ij}(t)$ denotes the fuel cost on arc (*i*,*j*) departing at time *t*. $L_{ij}(t)$ is the cost of the driver on arc (*i*,*j*). *C* is the congestion charge and λ is a binary scalar. As the congestion charge is paid once per day, $\lambda = 1$ if it is the first time for vehicle to enter the congestion charge zone when the congestion charge is applied, otherwise $\lambda = 0$.Then the travel cost of a vehicle on an $arc(i, j)$ is calculated as follows:

$$c_{ij}(t) = F_{ij}(t) + L_{ij}(t) + \lambda * C \tag{1}$$

The detailed calculations of the travel cost are described in the Appendix.

3.2.3 Formulation

Let $C_i(t)$ denote the minimum total cost from node *i* to destination node *q* departing at time *t*. The minimum total travel costs are then defined by the following relationship:

$$C_i(t) = \begin{cases} \min_{j \in B(i)} \ c_{ij}(t) + C_j[t + k_{ij}(t)], & \text{if } i \neq q \\ 0\ , & \text{if } i = q \end{cases} \tag{2}$$

$B(i)$ denotes the set of nodes $\{j\}$ where the arc $(i,j) \in A$.

Two heuristic methods are introduced to solve the problem in the following sections.

3.3 Heuristic 1

The method is first described for a case where there is no congestion charge but the cost and time for traversing an arc depends on the starting time for traversing the arc.

In the standard Dijkstra's algorithm, nodes in the network are assigned labels. Each label represents the minimum cost to travel from the origin node to the corresponding node when costs are fixed. In the standard Dijkstra's algorithm, when a label is calculated for a node after it is reached from a new path, only the minimum cost label is retained. But in this heuristic method, all labels are kept at the intermediate nodes after using Dijkstra's algorithm to get the initial solution and path. The method also records the corresponding time that the node is reached for each label.

The method then examines the nodes on the initial path to determine whether using one of the labels saved at the node might lead to a lower cost solution in this problem where costs depend on the times when the arcs are traversed. Such a label is referred to as a potential label. The key to this heuristic method is how to identify potential labels.

For each node on the initial path, the method examines the labels one by one. Let $n_i^{'}$ be the label at node *i* with the least cost. At node *i*, $p(n_i^k, t_i^k)$ is the total remaining cost starting from label *k* at node *i* to the destination node with corresponding departure time t_i^k along the initial path. $p(n_i^{'}, t_i^{'})$ is the total remaining cost starting from the best label at node *i* to the destination node with corresponding departure time $t_i^{'}$ along the initial path.

The differences between these two costs are compared with the differences in the costs recorded in the labels that represent the costs of the path from the origin to node *i*. At each node we calculate the difference between the cost of the best label and the cost of other labels, ΔC^k. $\Delta C^k = C^k - C^{'}$, where C^k represents the cost of label *k* and $C^{'}$ is the cost of the best label at each node. At each node, we compare ΔC^k with the difference in cost for the remainder of the journey from node *i* to the destination using the same departure time and the same route.

$$\Delta C^k \leq p(n_i^{'}, t^{'}) - p(n_i^k, t_i) \quad (3)$$

If the inequality (3) is satisfied, we consider label *k* at node *i* to be a potential label. Then the standard Dijkstra's algorithm is run starting from each potential label to determine whether a better final solution can be obtained. After this has been done for all potential labels, the least cost solution is retained.

The model of Ziliaskopoulos and Mahmassani (1993) is based on the Bellman's principle of optimality, which does not hold in finding the min cost path on real time-dependent transportation networks. It will scan all labels of a node for all possible departure times. The algorithm proposed is to correct the labels on the path found by Dijkstra's algorithm only. It will not scan all possible departure times.

Consider the 5-node network shown in Figure 2 and the data given in Table 1 in a similar setting as in the example illustrated in Section 3.1

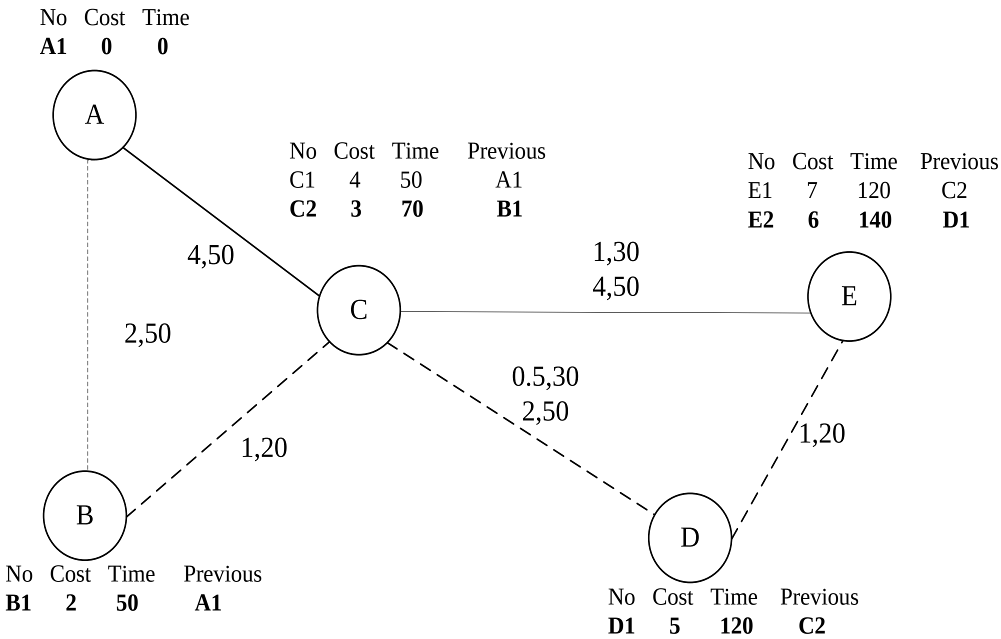

Figure 2 Example for Heuristic 1

Table 1 Data for Heuristic 1 Example

	Time Bin 1		Time Bin 2	
Arc	**Cost**	**Travel time**	**Cost**	**Travel time**
AC	4	50	4	50

AB	2	50	2	50
BC	1	20	1	20
CE	1	30	4	50
CD	0.5	30	2	50
DE	1	20	1	20

Let *A* be the source node and *E* be the destination. The information relating to the labels kept for *A*, *B*, *C*, *D* and *E* are shown in Figure 2 next to the corresponding node. The time units are minutes. Each time bin represents a time span of 60 minutes, i.e. time bins 1 and 2 correspond to time slots of (0,60] and (60,120], respectively. The costs for relevant departure time ranges for each arc are given in Table 1. In Figure 2, the previous node corresponding to each label is shown by a letter representing the node name and a number corresponding to the label; e.g. *C*1 refers to the first label for node *C*. For example, if the vehicle leaves node *C* at time 50 which is in time bin 1, a cost of 1 is applied when traversing arc *CE*. The standard Dijkstra's algorithm only keeps the best labels in the intermediate nodes which are highlighted in boldface in Figure 2. The initial path is *A*→*B*→*C*→*D*→*E*. The cost of the initial solution is 6 and the travel time is 140.

The method examines the labels along the initial path *A*→*B*→*C*→*D*→*E* to check if there are any better solutions. For example, we examine node *C* and label *C*1. The best label from Dijkstra's algorithm is *C*2 (C_{C2}=3). If starting from label *C*1 (C_{C1}=4), $\Delta C^{C1} = C_{C1} - C^{'}$ = 4 - 3 = 1. Following the initial path determined by the standard Dijkstra's algorithm and starting at the times indicated by the labels, it can be quickly calculated that $p(n_C^{'}, t^{'})$ is equal to 3, while $p(n_C^1, t_1)$ is equal to 1.5, so $p(n_C^{'}, t^{'}) - p(n_C^1, t_1) = 1.5$. Since $\Delta C^{C1} < p(n_C^{'}, t^{'}) - p(n_C^1, t_1)$ we consider *C*1 as a potential label that may be part of a better solution. The standard Dijkstra's algorithm is applied starting from *C*1 to check whether any better solution can be found. In this case, a better solution with a cost of 5 is found and the corresponding path is *A*→*C*→*E*.

When a congestion charge is imposed for traversing arcs within a congestion charge zone, two initial paths are generated using the standard Dijkstra's algorithm. The first is one where the path may enter the congestion charge zone and the second is where arcs within the congestion charge zone are avoided. Comparing the total costs of these two paths, if the least cost path is the one with the congestion charge only one least cost path is recorded.

3.4 Heuristic 2

The standard Dijkstra's algorithm selects the node *i* with the minimum temporary label, makes it permanent, and reaches out from that node, that is it scans arcs adjacent to node *i*. Heuristic 2 is a Dijkstra's type heuristic method which uses the time-space expanded network. The time horizon is divided into several time intervals and only the minimum cost labels associated with each time interval are kept at intermediate nodes. While Dijkstra's algorithm uses the minimum label only, Heuristic 2 considers all of the labels at a node to reach out to label adjacent nodes. As the number of intermediate nodes on the network increases the number of labels will increase dramatically which may require significant computational memory and time. The number of labels to be considered at each node may be limited to circumvent this problem. The example in Figure 3 illustrates the working mechanism of Heuristic 2.

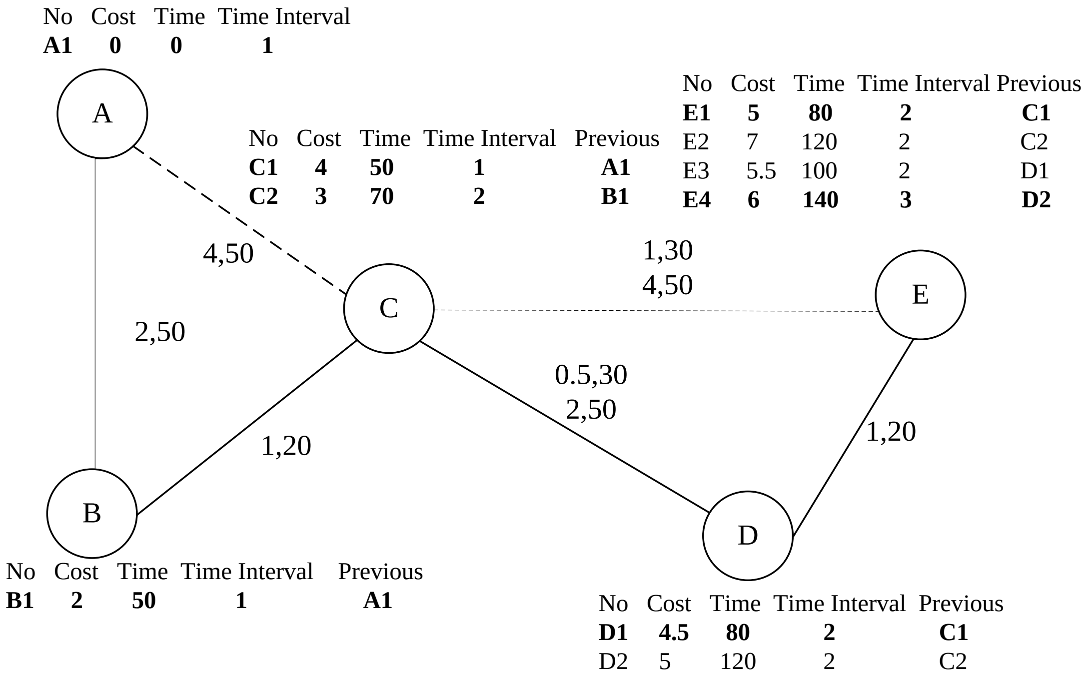

Figure 3 Example for Heuristic 2

For simplicity, a time interval is set equal to the length of 1 time bin. Different costs and travel times may apply in different time intervals. For example, for node *C*, labels *C*1 and *C*2 correspond to time intervals 1 and 2, respectively. *C*1 is the minimum cost label for time interval 1 while label *C*2 is the minimum cost label for time interval 2. As the minimum cost labels associated with different time intervals are kept at any node, Heuristic 2 will consider both labels *C*1 and *C*2 in node *C* for determining labels at the subsequent nodes. All labels that are going to be kept in nodes are given in bold in Figure 3. Heuristic 2 provides the same solution as Heuristic 1.

4. Numerical Investigation

In order to investigate the performances of the heuristics we first generate four benchmark instances of different sizes and with known optimal solution. Next, we use two real-world datasets, namely Bristol and London data, to compare the computational effort required by the heuristics against the DOT algorithm of Chabini (1998). Then, we compare the quality of the solutions obtained by the heuristics and that of DOT algorithm using Bristol data only since the latter algorithm cannot be implemented within the available memory to solve the London data.

4.1 Computational Analysis using Benchmark Dataset

Each instance in the benchmark dataset consists of $N*N$ nodes, where N is varied to generate networks with different sizes. The total number of arcs in the network is $(N\text{-}1)*N*2$. Each arc is 1 unit in length. The direction of travel for arc (i,j) is from i to j where $i < j$. In the first N-1 units of time, the speeds on all arcs are 1 per unit time and all costs are 1. From time period N and later, the speed changes to 1/2 per unit time and costs are 2 for each arc except the arcs in the bottom row. From time period N, the speeds for these arcs are 1 per unit time and the costs for these arcs are 1.5 each. Node 1 is the starting point and node N^2 is the destination.

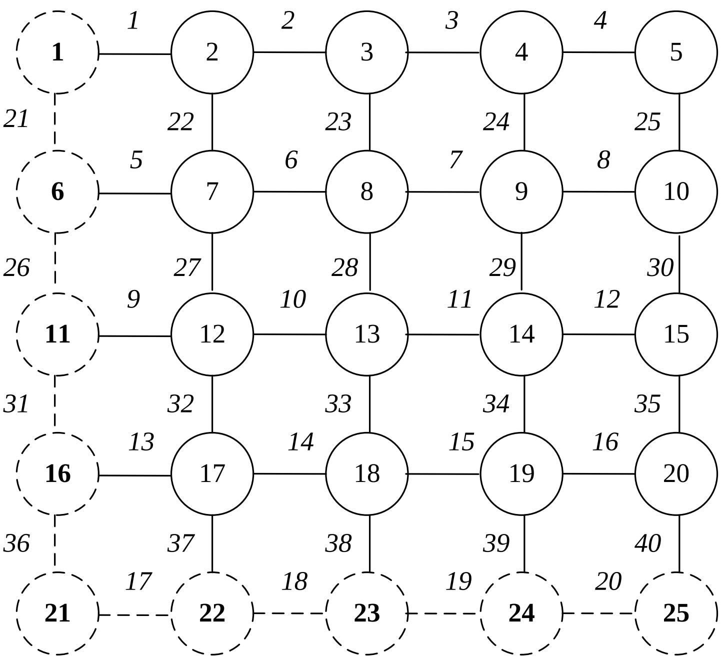

Figure 4 A 5*5 benchmark dataset

An example instance for N=5 is shown in Figure 4. We use the benchmark dataset to test the performance of heuristic methods against the optimal solutions. Note that the cost of the optimal path is always equal to $(N\text{-}1)*1+ (N\text{-}1)*1.5 = (N\text{-}1)*2.5$. So, in the case of *N*=5, the optimal solution is equal to 10. The minimum cost path is 1→6→11→16→21→22→23→24→25 and is shown in dashed links and bold node reference numbers. Time intervals equal to 1, 2 and 3 were used for Heuristic 2. Both Heuristic 1 and Heuristic 2 are able to find the optimum solutions for N=25, 50, 75 and 100.

Table 2 Computing times in seconds for different instances

Method	**Time Interval Length**	**N**			
		25	**50**	**75**	**100**
Heuristic 1	N/A	0.13	0.64	2.08	3.92
Heuristic 2	1	0.13	1.45	7.03	23.03
	2	0.09	1.69	2.75	8.36
	3	0.07	0.39	1.11	2.95

In Table 2 we compare the computational efforts required by Heuristic 1 and Heuristic 2 using instances with different sizes. Since the performance of Heuristic 2 depends on the length of the time interval, we analyze the cases where the time interval duration is 1, 2 and 3 time units (specified in parentheses in the table) and the maximum number of labels considered at any node is 100. All of the computations were carried out on a Dual Core processor PC of 3.00GHZ with 3.25GB of RAM and computing times are expressed in seconds. From Table 2, we observe that the algorithms require reasonable running times. The length of the time interval has a significant effect on the computation time of Heuristic 2; the longer the time interval is, the faster it finds the solution.

4.2 Comparing the heuristics with Chabini's method

The DOT method of Chabini (1998) is described using discrete intervals of time. When the problem can be defined using discrete time intervals then Chabini's method gives an optimal solution. However, if the network includes road segments that are short, then the time intervals used need to be very small in order to model the times required to travel along the short road segments at different speeds. Using very small time intervals may lead to long computation times and significant memory requirements. Table 3 shows the running time and the memory used for Chabini's method, Heuristic 1 and Heuristic 2. DOT was used to calculate the minimum cost paths from all nodes to one destination node. Heuristic 1 and Heuristic 2 calculate the minimum cost paths from one node to all other nodes, stopping when all customers have been reached. From Table 3, we can see that the heuristic methods are faster and require less memory.

Table 3 Comparing the running time and memory used for DOT, Heuristic 1 and Heuristic 2 on the London and Bristol networks

Data	**Method**	**Time Interval Length (seconds)**	**Running Time (seconds)**	**Memory Used (MB)**
London	DOT	120	7175	12518
- 5 customer locations		60	8031	36579
- 208448 nodes		30	NA	Out of memory
- 257531 arcs	Heuristic 1	-	128.85	893
	Heuristic 2	1	39.27	671
Bristol	DOT	60	15.68	25
- 15 customer locations		30	30.17	2219
- 4208 nodes		6	162.33	9102
- 4628 arcs		1	906.30	63175
	Heuristic 1	-	1.88	26
	Heuristic 2	1	0.63	27

In a real network, the travelling time between any two nodes cannot be expressed in the positive integer range unless the time intervals are extremely small. As a result, Chabini's method cannot guarantee to generate the optimal solution for a real network. The actual costs of the paths provided by Chabini's method when recalculated using real numbers for times and costs may be different to those calculated within Chabini's method, where the times and costs are based on every arc being traversed in a discrete number of time intervals. The smaller the time interval length is, the higher quality solution is obtained. If the time interval length is set to be a very small number, then the computing times increase as shown in Table 3. For example, if a street segment is only 20m in length (which may happen for some linking segments) and the time interval used is one second, then this implies a minimum speed of 72 km/h for the street segment to be passed over in one second.

An experiment is carried out based on the Bristol dataset by setting the time interval length for Chabini's DOT algorithm to be 1 second. Notice that this algorithm cannot be used in London data because of memory problems using realistic time intervals. The Bristol dataset is relatively small containing 4208 nodes and 4628 arcs. The experiment is based on a pair of customer nodes for a single journey. 96 instances were tested using different start times over 24 hours at 15 minute intervals. Figure 5 shows the shows the results for 21 of these instances over a period of time where differences were most notable. The chart illustrates the cost generated by Heuristic method 1 or 2, the cost generated by Chabini's method and the actual cost using real numbers for the times and costs generated by the paths provided by Chabini's method for different starting times.

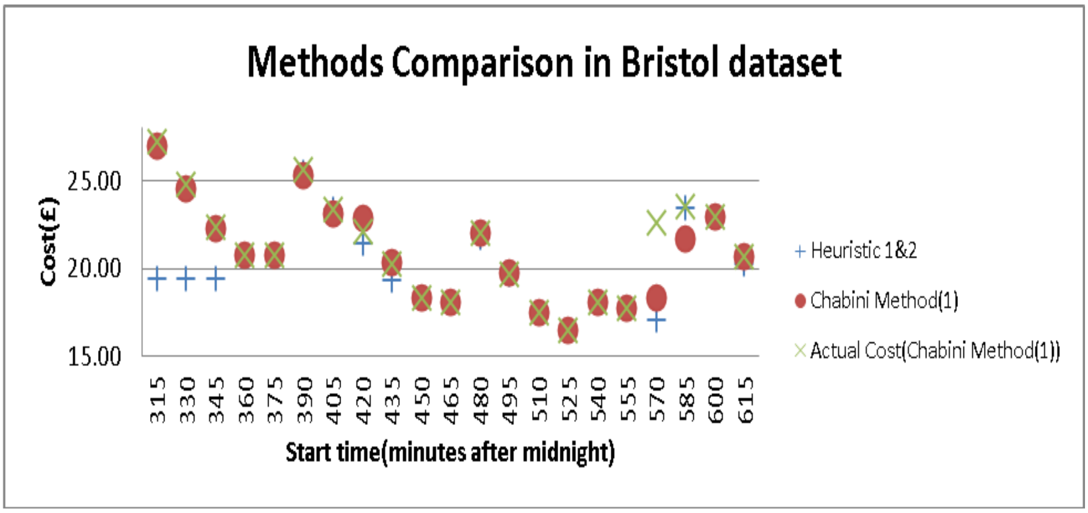

Figure 5 Comparison of costs for Heuristics 1 and 2 with those obtained using DOT

Figure 5 shows that the actual costs for Chabini's method may sometimes be more or less than the costs used internally in Chabini's method. The figure also shows that Heuristic 1 or 2 always gives an equal or lower cost than the actual cost provided by Chabini's route evaluated with the same cost function. This pattern was also observed for other start times over 24 hours.

These comparisons suggest that the heuristic methods can be more accurate and effective than Chabini's method for datasets containing a large number of short street segments.

5. Case Study

London is one of the places where a congestion charge is applied, so a case study is conducted using a London dataset. In practice, CCS has come into operation in parts of Central London on 17 February 2003 and it was extended into parts of West London on 19 February 2007. The western extension was officially removed from the charging zone beginning 4 January 2011. The congestion charge area data was accessed in 2010 when the congestion charge was equal to £8 and the western extension zone was still part of the congestion charge zone.

5.1 Data Structure

A scenario is modeled in which it is assumed to have 8 customers located in London as illustrated in the map in Figure 6. The red area shows the congestion charge zone. A congestion charge of £8 per vehicle per day is applied to all roads in this area from 7:00am to 6:00pm. The customers are located outside the congestion charge area; however, the journey between a pair may take place on the roads in the congestion charge area. In this case study, we consider 4 such pairs: customer 4 to customer 1, customer 6 to customer 2, customer 0 to customer 5 and customer 3 to customer 7. The experiment works on these 4 pairs of customers. The objective is to minimize the total cost.

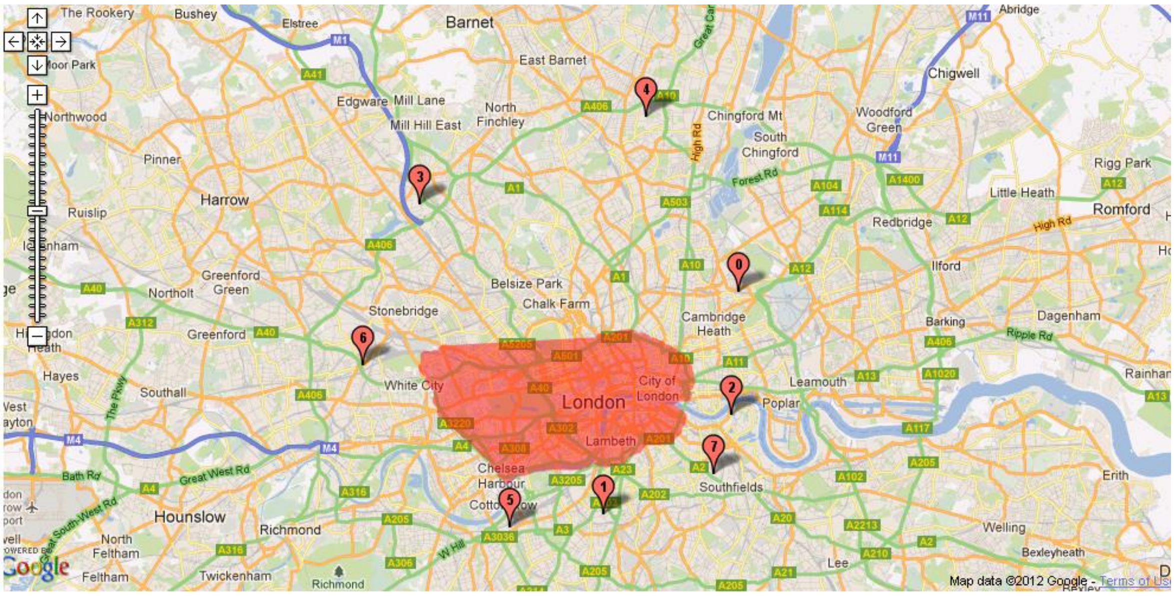

Figure 6 Customer locations

The representation of the road network is similar to that in the Road Timetable proposed by Eglese et al. (2006). Bidirectional roads are represented by two arcs of single direction. The time-varying speed limit data is also the same as that used by the Road Timetable. There are 15 time bins to cover a 24 hour period.

There are 208488 nodes and 257531 arcs in the road network considered. 18737 arcs are located within the congestion charge area. The arc lengths vary between 1m and 2848 m, and the mean arc length is 91.8m. About 71% of the arcs are shorter than

100m. As it is a time varying network, the speeds relating to each link are different in different time slots. The fuel consumption (g/km) is calculated by considering a Diesel LGV Euro II type vehicle as follows (National Atmopheric Emissions Inventory (2003)):

$$EF(v) = 77.43 + 0.009v - 0.015v^2 + 0.00015v^3 + 519v^{-1} - 70v^{-2} \quad (4)$$

where v is the speed of the vehicle in km/h. 1 litre diesel is approximately equal to 840g. In this case study, the diesel price is set at £1.2 per litre. In order to obtain the fuel cost, $EF(v)$ is transformed from g/km into litres and then multiplied by the unit price of the diesel and travelling distance. The driver cost is £8 pounds per hour.

The problems are solved for different start times at 15 minute intervals. 96 different starting times cover a day (24 hours) and the total cost associated with each one may be different.

5.2 Results

The heuristic methods are capable of obtaining a low cost route between two customers with any starting time. Figure 7 shows the total cost generated by Heuristic 1 and Heuristic 2 from customer 6 to customer 2.

Heuristics 1 and 2 generally gave the same results for this data set. However Heuristic 1 occasionally produced routes that involved cycling in order to avoid entering the congestion charge zone while the charge still applied when this would produce a lower total cost.

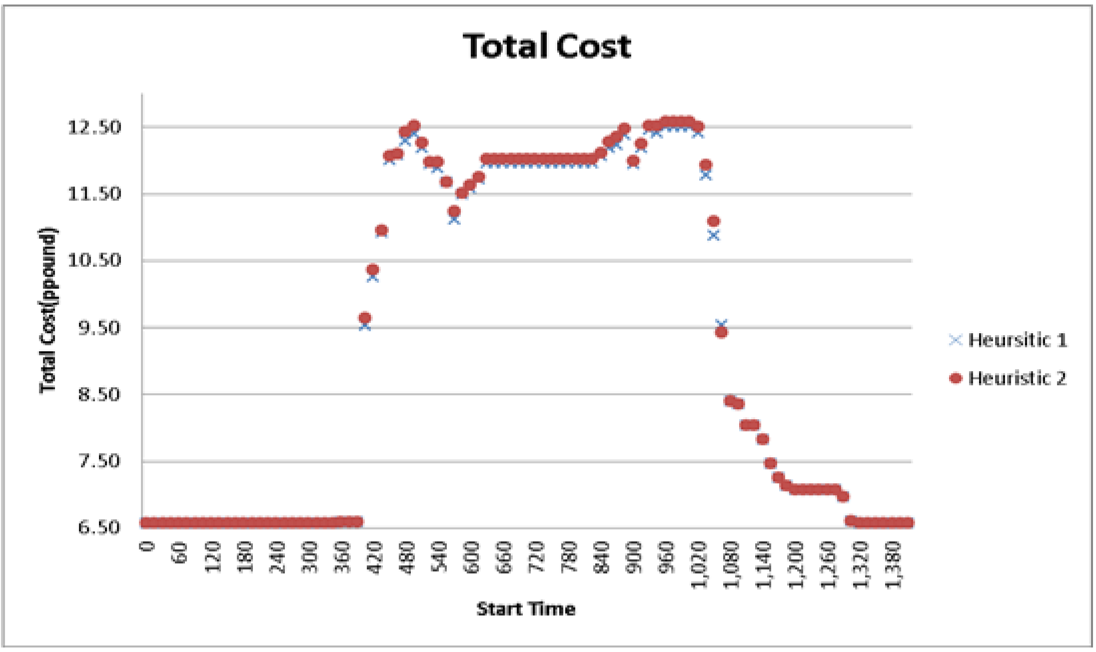

Figure 7 Total cost for route 6 to 2

Table 4 compares the costs obtained from shortest-time routes using the LANTIME algorithm described in Eglese et al. (2006) and the minimum cost routes for starting times of 1am and 8am from customer 6 to customer 2. It includes the time, fuel cost, congestion charge, distance and total cost. The driver cost is always higher than the fuel cost. If the driver has a fixed monthly wage, then the least cost routes may change.

Table 4 Results for route 6 to 2 using Heuristic 1

	1am	**1am (shortest time)**	**8am**	**8am (shortest time)**
Time (min)	33.64	33.64	67.55	44.24
Fuel Cost (£)	2.10	2.10	3.29	2.29
Driver Cost (£)	4.49	4.49	9.01	5.90
Congestion Charge (£)	0.00	0.00	0.00	8.00
Distance (m)	18672	18672	24947.9	18542.2
Total Cost (£)	6.59	6.59	12.30	16.19

In Table 5, the times, fuel costs, distances and total costs are compared for the routes obtained by the new heuristic methods with the shortest-time routes. The ratio for each attribute is calculated using the following formula:

$$ratio = \frac{a_1}{a_2} \qquad (5)$$

where a_1 is an attribute of the minimum cost solution, a_2 is the same attribute of the shortest time path solution. For example, if the route obtained from the shortest time scheme has time = 10 min, cost = £10, distance = 10 km, while the minimum cost solution is time = 13 min, cost = £9, distance = 8 km, then the ratios for time, cost and distance are 1.3, 0.9 and 0.8, respectively.

Table 5 Comparing ratio between 1am and 8am for all routes using Heuristic 1

	From 6 to 2		**From 1 to 4**		**From 7 to 3**		**From 5 to 0**	
	1am	**8am**	**1am**	**8am**	**1am**	**8am**	**1am**	**8am**
Time	1.00	1.53	1.00	1.21	1.00	1.44	1.00	1.13
Fuel Cost	1.00	1.44	0.99	1.27	0.98	1.32	0.99	1.13
Distance	1.00	1.35	0.99	1.30	0.97	1.27	0.99	1.14
Total Cost	1.00	0.76	1.00	0.68	1.00	0.74	1.00	0.60

Starting at 1am, the optimum solutions for the shortest time scheme and minimum cost scheme yield a similar cost performance.

Starting at 8am, in order to avoid the congestion charge, the minimum cost scheme runs longer distances and spends more time on the road than the shortest time scheme. This solution yields a higher fuel cost compared to the shortest-time one, but despite this fact, the total cost is significantly lower than the shortest time scheme.

6. Conclusions and Future Research

Finding minimum cost routes is a complicated optimization problem when the costs change with time. The results depend on the speeds, paths taken and the starting time of the journey. Getting the minimum cost is useful for vehicle operators to improve their operation performance so that they can cut environmental and economic costs. The paper demonstrates two heuristic methods to generate the low cost routes between nodes. A benchmark dataset has been designed to verify the performance of the heuristic methods.

The heuristics have also been implemented for a network of roads in London. From the case study, it is demonstrated that shortest paths and minimum cost paths may be significantly different in the rush hour, although they are quite similar in the off-peak time. In order to avoid paying the congestion charge, a least cost path leads to longer distances and more time on the road. In the London dataset, both heuristic methods are able to find low cost routes between nodes and help the driver to find a route to avoid going through the congestion charge area. The routes produced can be substantially cheaper than those produced on the basis of minimizing time such as described in Eglese et al. (2006).

The new algorithms show how costs for freight distribution may be significantly influenced by traffic conditions and the presence of a congestion charging scheme. Furthermore, the new algorithms could generate sets of road timetable and stored the data for solving the full VRP problem. Further research will consider how to embed the results from the new algorithms into a method to solve the full vehicle routing problem in the presence of a congestion charge.

Acknowledgments

The first author is supported by a grant under the LANCS Initiative, from the EPSRC. (Grant number: EP/F033613/1).

Appendix

The appendix describes the calculation of the travel cost. Total travel cost consists of three components: fuel cost, driver cost and congestion charge.

Fuel Cost

Formula (2) shows the calculation of the fuel cost:

$$F_{ij}(t) = d_{ij} * EF(v_{ij}(t)) * p^{f} \quad \text{(A.1)}$$

where p^{f} is the fuel cost in £/g and $EF(v_{ij}(t))$ is the fuel consumption function in g/km.

At different speeds, vehicles consume fuel at different rates. So the fuel used per kilometer is not constant and varies by the speed of the vehicle. According to the vehicle emission factor database of the National Atmospheric Emissions Inventory (2003) the fuel consumption function for Light Goods Vehicles (LGVs) is formulated as follows:

$$EF(v) = a + bv + cv^{2} + gv^{3} + hv^{-1} + iv^{-2} + jv^{-3} \quad \text{(A.2)}$$

where v is the speed in km/h and a, b, c, g, h, i and j are constant parameters that depend on the type of vehicle. Figure A.1 illustrates the fuel consumption for varying speeds. We can observe that the fuel consumption is a non-linear, convex function of speed.

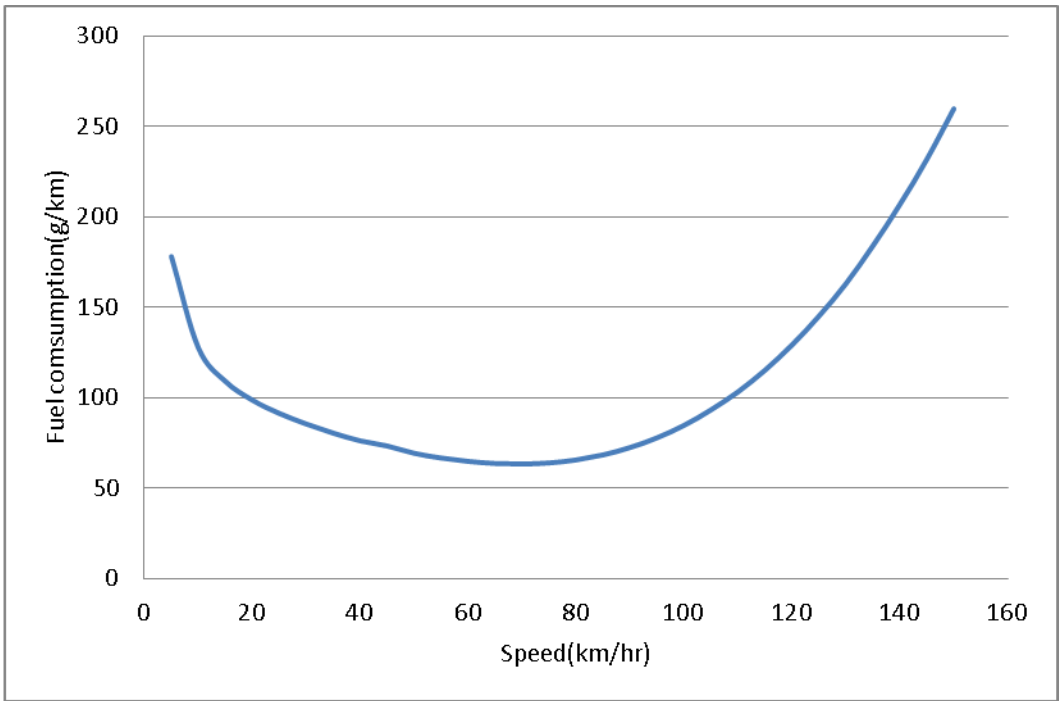

Figure A.1 Fuel consumption function for Euro II LGV

Driver Cost

The driver cost is calculated at a fixed rate *p* per unit time (usually per hour). The total amount paid to the driver is shown as follows:

$$L_{ij}(t) = p^{l} * k_{ij}(t) \qquad \text{(A.3)}$$

where p^{l} is the unit labor cost in £.

Congestion Charge

A congestion charge is imposed in a certain time in a certain area. In other words, it will be equal to zero or a certain constant value in the model. It depends on when the vehicle enters the congestion charge zone. The congestion charge is paid once per day if the vehicle travels into the congestion charge zone.

References

BEKTAS, T. & LAPORTE, G. 2011. The pollution-routing problem. *Transportation Research Part B: Methodological,* 45**,** 1232-1250.

CHANG, T. S., NOZICK, L. H. & TURNQUIST, M. A. 2005. Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments. *Transportation Science,* 39, 383–399.

CHEUNG, R. K. 1998. Iterative methods for dynamic stochastic shortest path problems. *Naval Research Logistics,* 45, 769–789.

CHABINI, I. 1998. Discrete dynamic shortest path problems in transportation applications : Complexity and algorithms with optimal run time. *Transportation Research Record,* 1645**,** 170-175.

COOKE, K. L. & HALSEY, E. 1966. The shortest route through a network with time-dependent intermodal transit times. *Journal of Mathematical Analysis and Applications,* 14**,** 493-498.

EGLESE, R. & BLACK, D. 2012. Optimizing the routeing of vehicles. *In:* MCKINNON, A., WHITEING, A. & BROWNE, M. (eds). *Green logistics: Improving the environmental sustainability of logistics.* Kogan Page, London, UK, 223-235.

EGLESE, R., MADEN, W. & SLATER, A. 2006. Road Timetable (TM) to aid vehicle routing and scheduling. *Computers and Operations Research,* 33**,** 3508-3519.

ERKUT, E., TJANDRA S.A. & VERTER, V. 2007. Hazardous materials transportation. In: Barnhart, C. & Laporte, G. (eds). *Handbooks in Operations Research and Management Science, 14,* Elsevier, 539-621.

FU, L. & RILETT L. R.. 1998. Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B-Methodological*, 32, 499–516.

GALLO, G. & PALLOTINO, S. 1988. Shortest paths algorithms. *Annals of Operational Research,* 13**,** 3-79.

HALL, R. 1986. The fastest path through a network with random time-dependent travel times. *Transportation Science,* 20, 182-186.

HILL, A. V. & BENTON, W. C. 1992. Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society,* 43**,** 343-351.

HUANG, H. & GAO, S. 2012. Optimal paths in dynamic networks with dependent random link travel times. *Transportation Research Part B-Methodological*, 46, 579-598.

ICHOUA, S., GENDREAU, M. & POTVIN, J. Y. 2003. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research,* 144**,** 379-396.

MILLER-HOOKS, E. D. & MAHMASSANI, H. S. 1998. Least possible time paths in stochastic, time-varying networks. *Computers and Operations Research*, 25, 1107–1125.

MILLER-HOOKS, E. D. & MAHMASSANI, H. S. 2000. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science,* 34, 198–215.

MILLER-HOOKS, E. & MAHMASSANI, H. 2003. Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks. *European Journal of Operational Research,* 146**,** 67-82.

NATIONAL ATMOPHERIC EMISSIONS INVENTORY. 2003. *Emission factors database* [Online]. Available: http://naei.defra.gov.uk/ [Accessed 10 April 2013].

PALLOTTINO, S. & SCUTELLA, M. G. 1998. Shortest path algorithms in transportation models: classical and innovative aspects. *In:* MARCOTTE, P. & NGUYEN, S., (eds). *Equilibrium and Advanced Transportation Modelling.* Kluwer Academic Publishers, Boston, MA, 245–281.

PAPE, U. 1974. Implementation and efficiency of moore-algorithms for the shortest route problem. *Mathematical Programming,* 7**,** 212-222.

PRETOLANI, D. 2000. A directed hypergraph model for random time dependent shortest paths. *European Journal of Operational Research,* 123**,** 315-324.

SUNG, K., BELL, M. G. H., SEONG, M. & PARK, S. 2000. Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research,* 121**,** 32-39.

ZILIASKOPOULOS, A. K. & MAHMASSANI, H. S. 1993. Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record*, 1408, 94-100.