

Vehicle Dispatching With Time-Dependent Travel Times

Soumia Ichoua^{1,2}, Michel Gendreau^{1,2}
and Jean-Yves Potvin^{1,2}

1 Centre de recherche sur les transports, Université de Montréal, C.P. 6128,
succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

2 Département d'informatique et de recherche opérationnelle,
Université de Montréal, C.P. 6128, succ. Centre-ville,
Montréal, Québec, Canada H3C 3J7.

Abstract. Most of the models for vehicle routing reported in the literature assume constant travel times. Clearly, ignoring the fact that the travel time between two locations does not depend only on the distance traveled, but on many other factors including the time of the day, impact the application of these models to real-world problems. In this paper, we present a model based on time-dependent travel speeds which satisfies the “first-in-first-out” property. An experimental evaluation of the proposed model is performed in a static and a dynamic setting, using a parallel tabu search heuristic. It is shown that the time-dependent model provides substantial improvements over a model based on fixed travel times.

Keywords. Transportation, vehicle routing, time-dependent, tabu search.

1 Introduction

The routing and scheduling of a fleet of vehicles to service customers plays an important role in the distribution chain. This research domain has thus been widely studied in the literature (see for example, the survey in Ball et al. (95)). However, there is still a lack of modeling approaches that more closely represent real-life conditions. One practical aspect that has seldom been addressed is the time dependency of travel times on the time of the day. Many available models assume that the travel times are constant throughout the day or exploit simple procedures to adjust them, like multiplier factors associated with different periods of the day. Unfortunately, these assumptions are weak approximations of the real-world conditions where travel times are subject to more subtle variations over time. These variations may result from predictable events (e.g., congestion during peak hours) or from unpredictable events like accidents, vehicle breakdowns, and others. Therefore, the optimal solution to a formulation of the problem that assumes constant travel times may be suboptimal or even infeasible for the time-dependent problem (e.g., if time windows are considered).

Time-dependent vehicle routing problems have seldom been addressed because they are harder to model and to solve. These problems can be stated as follows. Let assume that a fixed size fleet of m identical vehicles of fixed capacity is available to service customers (nodes) with fixed demand and that the time horizon is partitioned into p time intervals $T_1, T_2, \dots, T_p$. Given a network of n nodes, a $n \times n$ time-dependent matrix $C(T_k)_{k=1,2,\dots,p} = [c_{ij}(T_k)]$ contains the travel times between each pair of nodes (i, j) when the vehicle departs from node i within time interval T_k . The goal is to find a set of minimum cost vehicle routes that service every customer, such that :

- Each vehicle route originates from and terminates at a fixed depot;
- Each vehicle services one route and the service point of each request is visited once by exactly one vehicle;
- The capacity of each vehicle is not exceeded and the demand of each customer is satisfied;

Furthermore, each service point i (including the depot) may have its own time window $[e_i, l_i]$, where e_i is the earliest service time and l_i is the latest service time. When the service points have "soft" time windows, a vehicle can arrive before the lower bound or after the upper bound. If the vehicle is too early, it must wait to start its service; if the vehicle is too late, a penalty for lateness is incurred in the objective function.

The problem considered here is motivated from a courier service application found in the local operations of long-distance shipping services, where parcels and envelopes are collected at different customers' locations and brought back to a central office for further processing and shipping. In this case, the service points have "soft" time windows, but each route must start and end within the time window associated with the depot. Also, no capacity constraint is enforced since only small items are transported. The cost to be minimized is a weighted sum of the total travel time over all routes, plus the total lateness over all customers. A time-dependent model for predictable variations in travel times is proposed and analyzed in this context. It is implemented in a static environment where the customer requests are known in advance (i.e., before the routes are constructed), and in a dynamic environment where new customer requests are unveiled as the routes are executed.

The paper is organized as follows. Section 2 presents a brief literature review dedicated to time-dependent vehicle routing and two other problems closely related to it, the time-dependent shortest path and the path choice problems. Section 3 presents our time-dependent model and discusses several issues related to it. Section 4 briefly introduces a tabu search heuristic developed by Taillard et al. (97) for a version of the problem with fixed travel times and explains how it was modified to account for time-dependency. Section 5 reports experimental results obtained with the new algorithm in a static context. Section 6 addresses the dynamic version of the problem. Finally, section 7 concludes and proposes future avenues of research.

2 Literature review

The only papers that we are aware of in the literature of time-dependent vehicle routing are the ones of Malandraki (89), Malandraki and Daskin (92), Hill and Benton (92) and Malandraki and Dial (96). Malandraki and Daskin (89, 92) examine both the time-dependent vehicle routing problem (TDVRP), and the time-dependent traveling salesman problem (TDTSP) which is a special case of the TDVRP when the fleet size is equal to one. They provide mixed integer linear programming formulations which include time windows, capacities and allow for waiting at a customer location. The travel times are computed using step functions. Nearest-neighbor (greedy) heuristics for the TDTSP and the TDVRP without time windows are proposed, as well as a branch-and-cut algorithm for solving small problems with 10 to 25 nodes. In Malandraki and Dial (96), a dynamic programming algorithm is proposed to solve the TDTSP. Although it is argued that many different types of travel time functions can be handled by this algorithm, results are only reported for step functions like those found in Malandraki (89) and Malandraki and Daskin (92).

Hill and Benton (92) consider a time-dependent vehicle routing problem (without time windows) and propose a model based on time-dependent travel speeds. Computational results are reported on a small example with a single vehicle and five locations. The authors also mention the implementation of a simple greedy heuristic for the multi-vehicle traveling salesman problem with capacity constraints and no time windows for a city with 210 locations. A validation of the model within a commercial courier scheduling package is also mentioned, but no details are provided.

The major weakness of the above models is that they do not satisfy the "first in first out" (FIFO) property. The FIFO property guarantees that if a vehicle leaves a node i for a node j at a given time, any identical vehicle leaving node i for node j at a later time will arrive later at node j (which is common sense). This will be discussed in section 3.

Time-dependency has seldom been addressed in the literature on vehicle rout-

ing. However, it has been widely studied in three related problems: the time-dependent traveling salesman problem, the shortest path problem and the path choice problem. They are briefly described below.

Time-dependent shortest path problem

The earliest models which account for time-dependency were developed by the end of the fifties for solving shortest path problems (Ford and Fulkerson (58), Cooke and Halsey (66), Dreyfus (69)). Since then, these problems have been the most widely studied. The goal is to find minimum cost paths from origin nodes to destination nodes, through a network where travel times and costs are time-dependent.

Time-dependent path choice problem

The path choice problem is part of traffic equilibrium models. Here, many travelers "compete" on a transportation network to get to their destination. These travelers are typically distributed among several paths, besides the shortest ones, based on route choice models which simulate user behavior. The earliest work that we are aware about time-dependent path choice problems is the one by Marguier and Ceder (84) for common bus stops with overlapping routes.

Time-dependent traveling salesman problem (TDTSP)

The earliest papers in the literature related to time-dependent vehicle routing problems appeared in the sixties and were dedicated to the time-dependent traveling salesman problem (Miller et al. (60) and Hadley (64)). The TDTSP constructs a Hamiltonian tour of minimum travel cost over n cities, where the travel cost from city i to city j depends on the time of the day.

According to the properties of the travel time and cost functions, the work related to time-dependent problems can be classified in four main categories. They are briefly presented in the following subsections.

2.1 Models based on "simple" travel time and cost functions

Many researchers have used simple rules to integrate time-dependency components in their models. In their decision support system for dispatching and processing customer orders for gasoline, Brown et al. (87) first produce a solution where travel time fluctuations are ignored. Then, the loads for each truck are resequenced

”manually” to take into account various factors such as traffic congestion during rush hours, road and weather conditions, etc. Other researchers use multiplier factors to represent variations in travel times (Fisher et al. (82), Hill et al. (88), Rousseau and Roy (88), Shen and Potvin (95)). Clearly, this is just a rough approximation of actual conditions.

2.2 Models based on discrete travel time and cost functions

In this kind of formulation, the horizon of interest is ”discretized” into small time intervals. The travel time and cost functions for each link are assumed to be step functions of the starting time at the origin node. This scheme is widely used in many time-dependent transportation problems. However, the assumption that travel times vary in discrete jumps is just an approximation of real-world conditions since travel times change continuously over time.

Many researchers used this kind of model to solve time-dependent shortest path problems (Cooke and Halsey (66), Dreyfus (69), Ziliaskopoulos and Mahmassani (93), Ziliaskopoulos (94) and Chabini (96, 97)). This framework was also used to formulate time-dependent traveling salesman problems (Picard and Queyranne (78), Fox et al. (80), Malandraki (89) and Malandraki and Daskin (92)). Finally, Natchigall (93) used discrete transit functions to model a time-dependent path choice problem in a railway context.

The major drawback of using models that are based on discrete travel time and cost functions is that the FIFO property does not hold (see section 3).

2.3 Models based on continuous travel time and cost functions

In real-life, travel times vary continuously over time. The work reported in this section is thus aimed at modeling the reality more accurately. In their formulations of time-dependent shortest path problems, Halpen (77) and De Palma et al. (90) calculate the travel times using nonnegative and piecewise linear functions while Orda and Rom (90) assume that the travel times are arbitrary. In their models dedicated to time-dependent path choice problems, Hall (86), Hickman and Wilson (95), and Hickman and Bernstein (97) assume that the travel times are stochastic

and time-dependent. Finally, Marguier and Ceder (84) study the same problem for common bus stops with overlapping routes. Time-dependent distributions are used to represent passengers' waiting times.

Continuous travel time functions seem to be more appropriate to model real-world conditions. Unfortunately, models based on such functions are confronted with the following limitations: **(i)** simplifying assumptions are often made to obtain a more tractable model (e.g., differentiability, piecewise linearity, etc...); **(ii)** they are still an approximation of what is observed in the real world; **(iii)** in the case of the path choice problem, using continuous time-dependent distributions to represent travel time or waiting time functions results in complicated integrations that are difficult to solve analytically. Furthermore, this kind of formulation often suggests that passengers make their boarding decision using a fairly complicated logic (Hickman and Bernstein (97)).

2.4 Models based on Markovian formulations of travel time and cost functions

The only work that we are aware of in this category is the one of Psaraftis and Tsitsiklis (93). The authors investigate the shortest path problem in a stochastic and time-dependent setting. The cost of each arc (i, j) is a known function $f_{ij}(e^i)$ of the state e^i of some environment variable at node i at the time of departure from node i to node j . Environment variables are mutually independent and governed by a finite state Markov process where state transitions occur in discrete time. The goal is to seek a policy that minimizes the expected total cost on a path between two specific nodes. A dynamic programming algorithm is proposed to solve the problem. However, given that its complexity depends on the number of Markov states at each node, the state space grows quickly with problem size, thus preventing the model from being applied to realistic problem instances.

In section 3, we propose and analyze a model that focuses on travel speed variations from one time period to the next.

3 A time-dependent travel speed model

3.1 Motivation

As mentioned previously, the literature related to time-dependent vehicle routing problems is very scarce. In fact, the only papers that we are aware of in this category are those of Malandraki (89), Malandraki and Daskin (92) and Hill and Benton (92). They were briefly discussed in section 2. The major shortcoming of Malandraki's model, which represents the travel time as a step function of time, is that the FIFO assumption does not hold. To illustrate this, consider figure 1 which represents a travel time function on a link (i, j) of length 1.

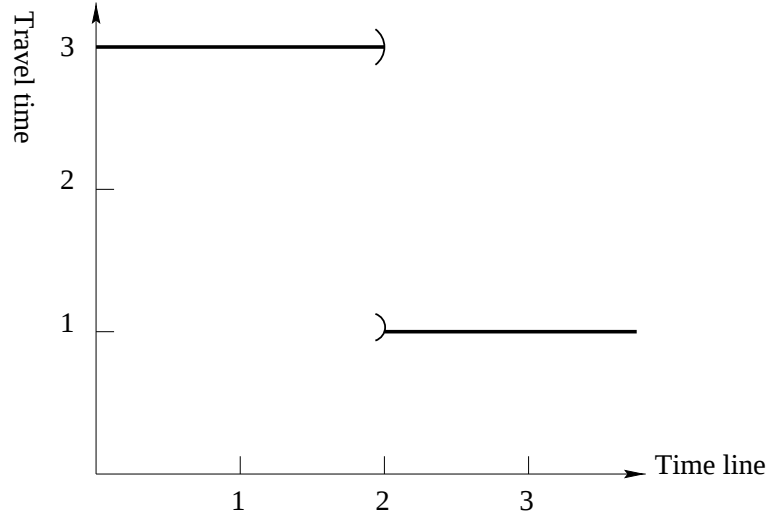


Figure 1: Travel time function on a link.

If the vehicle leaves node i at instant $t_1 = 1$, it will reach node j at instant $t'_1 = 4$; but it will arrive at instant $t'_2 = 3$ ($< t'_1$) if it leaves node i at instant $t_2 = 2$ ($> t_1$). To overcome this weakness, Malandraki and Daskin (89, 92) suggested to allow vehicles to wait at a node to smooth the travel time function. However, this suggestion concerns decreasing step functions only and induces useless waiting at nodes. In real-life applications (e.g., distribution of industrial products) it is hard to convince a dispatcher that it is advantageous to force a driver to wait at

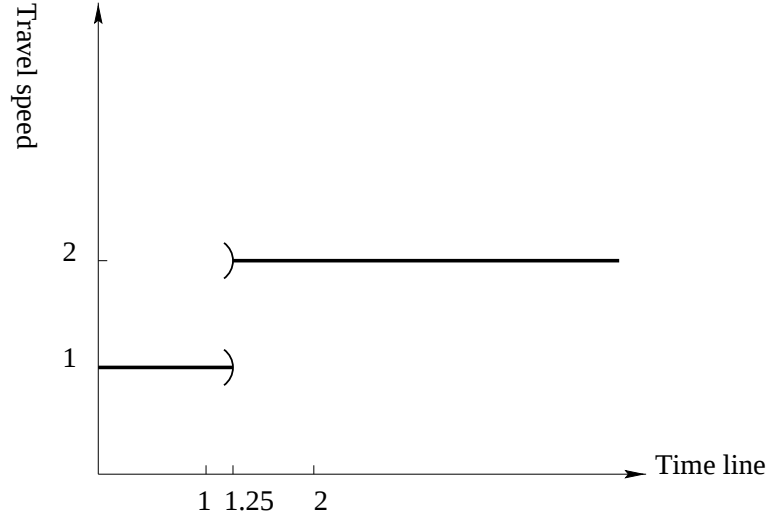


Figure 2: Travel speed function at a node.

a customer location even if he is ready to depart.

We recall that Hill and Benton (92) developed the only model that we are aware of based on time-dependent travel speeds. In this formulation, the travel time on a given link (i, j) , starting during time period T , is

$$d_{ij}/r_{ijT}, \text{ with } r_{ijT} = (r_{iT} + r_{jT})/2, \quad (1)$$

where d_{ij} is the distance between locations i and j , and r_{iT} is the average speed associated with the "area" around location i during time period T . Thus, r_{ijT} is an average travel speed for a trip from i to j starting during time period T . But, since the speed along a given link (i, j) is an average speed based on a single period, the FIFO assumption is not necessarily satisfied either. To illustrate this, consider the arrival time at node j from node i when all nodes have the same speed value r_{iT} for a given time period T . Figure 2 illustrates the travel speed function for a node.

Assuming a link (i, j) of length 1, we have the following. If the vehicle leaves node i at instant $t_1 = 1$, it will arrive at node j at instant $t'_1 = 1 + 1 = 2$. However, if it departs from node i at instant $t_2 = 1.25 (> t_1)$, it will reach node j at instant $t'_2 = 1.25 + 0.5 = 1.75 (< t'_1)$.

In fact, associating a time-dependent travel speed or a time-dependent travel time with a link is equivalent since it is always possible to deduce the travel speed from the travel time, and conversely, for a given travel distance. To better model time-dependency, one has to take into account the adjustment of the travel speed when the vehicle crosses the boundary between two consecutive time periods. In the following, we propose a model that addresses this issue.

3.2 The model

In vehicle routing problems and more generally in the transportation field, an important area that remains very challenging is the conception of efficient models to achieve a good trade-off between the implementation requirements and the ability to reflect the complexity of real-world conditions such as fluctuations in travel times. A "natural" and simple way to take time-dependency into account is to work with time-dependent travel speeds and to adjust the speed when the vehicle crosses a boundary between two time periods.

Let assume that the horizon is divided into p time periods $T_1, T_2, \dots, T_p$. Given a network of n nodes, a symmetric distance matrix $D = (d_{ij})$ and travel speed matrices $V_T = (v_{ijT})$, $T \in \{T_1, T_2, \dots, T_p\}$ are defined. In contrast with the formulation proposed by Hill and Benton (92) where travel speeds are indexed by time periods and nodes, here, the travel speeds are indexed by time periods and arcs. This reduces the computational effort at the cost of more storage (in Hill and Benton (92), we recall that the travel speed on a link (i, j) is calculated as $r_{ijT} = (r_{iT} + r_{jT})/2$). To limit the number of speed values v_{ijT} to estimate, the set of arcs A is partitioned into subsets $(A_c)_{1 \leq c \leq C}$. That is, the travel speed during period T on an arc (i, j) that belongs to a subset (or category) A_c is $v_{ijT} = v_{cT}$, where v_{cT} is the travel speed associated with category A_c and time period T . Consequently, the number of parameters in the model is considerably reduced, especially if C is small. Partitioning the set of arcs into subsets is a reasonable assumption for urban transportation networks since links (routes) can usually be classified into categories based on their physical characteristics (e.g., width, one/two ways, etc...), and their geographical location.

3.2.1 Changing travel speeds

The main point in our model is that we do not assume a constant speed over the entire length of a link. Rather, the speed changes when the boundary between two consecutive time periods is crossed.

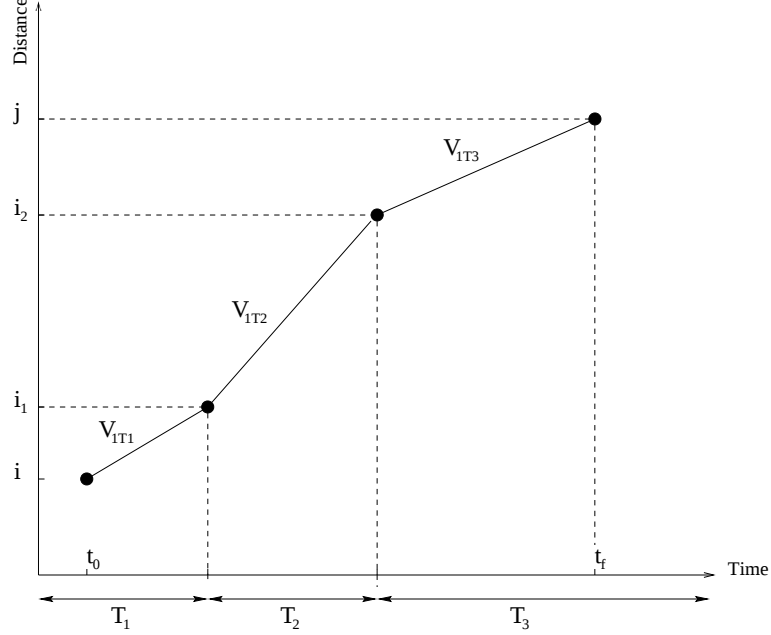


Figure 3: Changing travel speed over time.

This is illustrated in figure 3 where a route for one vehicle is considered. The horizon is "discretized" into three time periods $(T_j)_{1 \leq j \leq 3}$, with a different speed associated with each period. The vehicle leaves service point i at time $t_0 \in T_1$ and travels at speed v_{1T_1} until it reaches point i_1 at the boundary between periods T_1 and T_2 . From there, the vehicle travels at a faster speed v_{1T_2} until it reaches point i_2 at the boundary between T_2 and T_3 . Finally, it travels at speed v_{1T_3} for the remainder of its trip to reach point j .

3.2.2 Travel time calculation

Figure 4 describes the procedure for calculating the travel time between any pair of nodes i and j .

- | |
|--|
| <ol style="list-style-type: none"> 1. set t to t_0,
 set d to d_{ij},
 set t' to $t + (d/v_{cT_k})$. 2. while $(t' > \bar{t}_k)$ do <ol style="list-style-type: none"> 2.1 $d \leftarrow d - v_{cT_k}(\bar{t}_k - t)$. 2.2 $t \leftarrow \bar{t}_k$ 2.3 $t' \leftarrow t + (d/v_{cT_{k+1}})$. 2.4 $k \leftarrow k + 1$. 3. return $(t' - t_0)$. |
|--|

Figure 4: Travel time calculation procedure.

We suppose that the vehicle leaves node i at $t_0 \in T_k =]\underline{t}_k, \bar{t}_k]$ and that link (i, j) belongs to category c , $1 \leq c \leq C$. It is assumed that d_{ij} is the distance between i and j , and v_{cT_k} is the travel speed associated with category c and time period T_k . Also, t denotes the current time and t' denotes the arrival time.

3.2.3 Characteristics of the travel time function

In our model, the travel speed v_{cT} is a step function of the time of the day. Therefore, the travel time is a piecewise continuous function over time that is simple and easy to evaluate beside being a "natural" way to estimate travel times in real-world conditions. Figure 5 gives an example of the travel speed function and its associated travel time function for a link of length 1.

We are aware that travel speeds also change continuously over time, however using step functions to compute travel speeds is a more reasonable assumption than for travel times. That is, when the boundary between two consecutive time periods is crossed, the speed will typically change much faster than the travel time on a link.

Due to the travel speed changes, the travel time function proposed in this model satisfies the FIFO property. That is, leaving a node earlier guarantees that one will arrive earlier at destination. Consider the previous example given in figure 5.

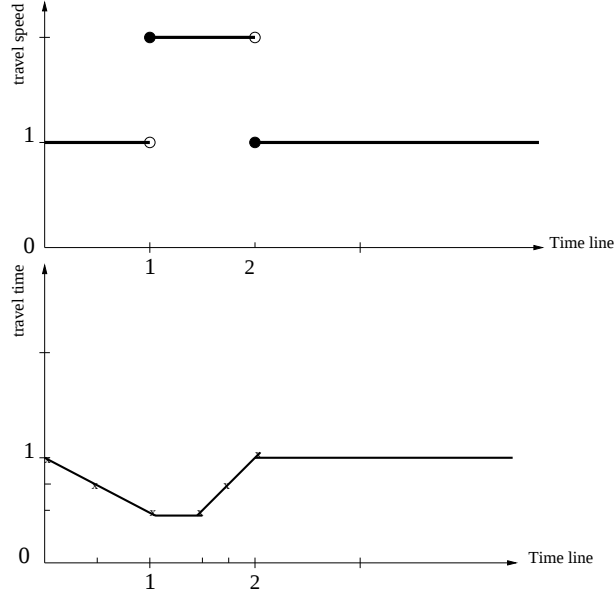


Figure 5: An example of travel speed and travel time functions.

A vehicle that is ready to leave the link before $t_1 = 1$ has no incentive to wait even if its speed will increase afterward. Actually, during this waiting time, the vehicle could have used the available speed to get closer to its destination until the time of speed change. At this point, the faster speed could be used to terminate the trip. Hence, the FIFO property precludes our model from inducing useless waiting times.

This approach has been implemented within a parallel tabu search developed by Taillard et al. (97) for the fixed travel time version of the problem. The next section will briefly introduce the original algorithm and then, explain how it was modified to cope with time-dependent travel times.

4 A parallel tabu search algorithm

4.1 The original algorithm

The algorithm developed in Taillard et al. (97) is a parallel tabu search heuristic with an adaptive memory. Tabu search is an iterative local search technique that

starts from some initial solution. At each iteration, a neighborhood is generated around the current solution and the best solution in this neighborhood becomes the new current solution (even if it does not provide an improvement). By allowing a degradation of the objective, it is possible to escape from bad local optima, as opposed to pure descent methods. The interested reader will find more details about this approach in Glover and Laguna (97).

The algorithm developed in Taillard et al. (97) can be summarized as follows :

- Construct I different initial solutions with a stochastic insertion heuristic (where the choice of the next customer to be inserted is randomized).
- Apply tabu search to each solution and store the resulting routes in an adaptive memory.
- While a stopping criterion is not met do :
 - Use the routes stored in the adaptive memory to construct an initial solution.
 - Decompose the problem into subproblems obtained through a geographical, distance-based, decomposition procedure (which partitions the service area into sectors by sweeping a ray, with the central depot as the pivot, over the routes).
 - Apply tabu search to each subproblem.
 - Add the routes of the resulting solution in the adaptive memory.

The objective is to minimize a weighted sum of total distance traveled and total lateness over all customers. The procedure for generating the neighborhood of the current solution is the CROSS exchange. Basically, two segments of variable lengths are taken from two different routes and swapped. Figure 6 illustrates this procedure.

To speed up the algorithm, a parallel implementation on a network of workstations was developed (Taillard et al. (97)). The parallelization of the procedure was achieved at two levels :

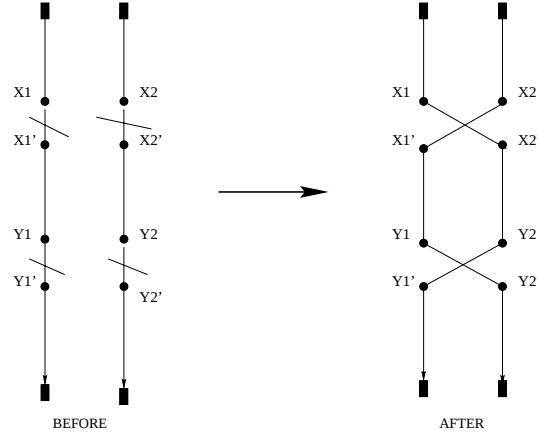


Figure 6: A CROSS exchange.

- (1) Different tabu search threads run in parallel, each of them starting from a different initial solution.
- (2) Within each search thread, many tabu searches run independently on the subproblems obtained through the decomposition procedure.

4.2 Implementing the time-dependent model

Implementing the time-dependent model mostly impacts the evaluation of a new solution after a CROSS exchange. In the following, we explain how the original algorithm was modified to cope with time-dependency.

4.2.1 Objective function

To evaluate a given solution, the total distance traveled is replaced by the total travel time in the objective function.

4.2.2 Neighborhood evaluation

The evaluation of a move, as implemented in the original algorithm cannot be directly applied to our problem. In the following, the original evaluation process is briefly described, then its adaptation to the time-dependent problem is discussed in detail.

The original procedure

As mentioned before, the original algorithm uses CROSS exchanges to generate the neighborhood. The evaluation of such moves is based on the difference between the value of the neighboring solution and the value of the current solution. Namely, $\delta f_1 = \delta d + \alpha * \delta l$, where δd and δl are the modification to the total distance and the modification to the total lateness of the solution, respectively, and α is a constant parameter. The evaluation of δd is done in constant time by simply subtracting the total length of the edges that are removed from the solution and by adding the total length of the edges that are introduced into the solution. The modification to the total lateness is the sum of the modifications to the total lateness incurred by both routes involved in the CROSS exchange. Unlike δd , this modification cannot be evaluated exactly in constant time because any additional lateness at a given customer location must be propagated along the route. To reduce the complexity of the calculation, an approximate evaluation procedure is used to evaluate each neighboring solution. Then, the M best solutions according to this approximation are evaluated exactly and the best solution is selected.

To illustrate the approximate evaluation procedure, the contribution δl of the new route servicing customers X_1, X'_2, Y_2 and Y'_1 in figure 6 is considered. The first part of this contribution is evaluated exactly in constant time by propagating the lateness at customer X'_2 along the route segment $X'_2 - Y_2$ (which is limited to at most 7 customers, see Taillard et al. (97)). The second part of δl is assessed using an approximation function associated with customer Y'_1 . To construct the approximation function of a given customer i , the start of service b_i is artificially delayed by a number of δb_i values, and the impact of each δb_i on the total lateness of the route δl_i is evaluated exactly. A piecewise linear function is then produced by interpolation between these points. The interested reader will find further details in Taillard et al. (97).

Adaptation to the time-dependent model

In the proposed time-dependent model, the difference between the value of the neighboring solution and the current solution is $\delta f_2 = \delta tr + \alpha * \delta l$, where δtr is the modification to the total travel time and δl is the modification to total lateness.

(a) Lateness

Due to the time-dependency component, the exact evaluation of lateness is more computationally expensive than in the original problem. Thereby, using an approximate evaluation procedure is even more important in this case. In the following, the adaptation of the original approximation procedure is illustrated for the new route servicing customers X_1, X'_2, Y_2 and Y'_1 in figure 6.

- (i) The contribution of the route segment $X'_2 - Y_2$ to the total lateness is assessed exactly as in the original algorithm. However, the time-dependent travel time calculation described in subsection 3.2 is used. This provides the value $\delta b_{Y'_1}$ at Y'_1 .
- (ii) The contribution of the remainder of the route is evaluated approximately as in the original procedure. That is, an approximate function is associated with each customer i , based on an exact evaluation of δl_i for a few values of δb_i , using the time-dependent travel time function. Now, suppose that we search for the value of $\delta l_{Y'_1}$ induced by $\delta b_{Y'_1} = z$ with $z_j \leq z \leq z_{j+1}$, where z_j and z_{j+1} are two consecutive values for which $\delta l_{Y'_1}$ is known exactly (i.e., these values have been used to construct the linear interpolation). In figure 7, we assume that $\delta l(z)$, δl_j and δl_{j+1} are associated with z , z_j and z_{j+1} respectively.

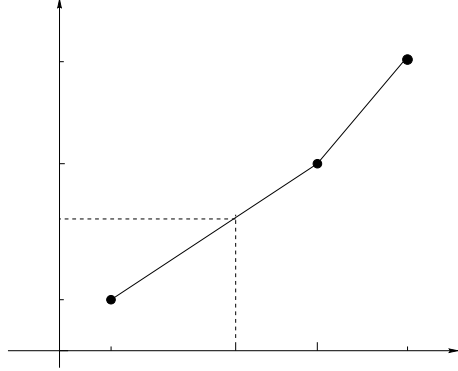


Figure 7: Approximate evaluation of lateness.

The interpolation is such that $\delta l_j \leq \delta l(z) \leq \delta l_{j+1}$, which makes sense since the travel time function satisfies the FIFO rule (that is, the exact variation of lateness $\delta l(z)^*$ should also satisfy $\delta l_j \leq \delta l(z)^* \leq \delta l_{j+1}$).

(b) Travel time

As opposed to the variation in total distance used in the original algorithm, assessing the modification to the total travel time cannot be achieved easily. To illustrate this, consider the two routes in figure 6. Subtracting the total length of the removed edges and adding the total length of the new edges, was sufficient to evaluate the total distance variation. However, in the time-dependent context, it may well happen that the total travel time over route segment $X'_1 - Y_1$ or (and) over route segment $X'_2 - Y_2$ will change after executing the move (i.e., the total travel time over segment $X'_1 - Y_1$ before the move depends on the arrival time to X'_1 from X_1 ; after the move, it depends on the arrival time to X'_1 from X_2 , which may be different). To overcome this difficulty, an approximate evaluation is used, similar to the one presented in (a).

At the end, the M best moves according to the approximation, are evaluated exactly using the time-dependent travel time function, and the best exact move is executed to obtain the new current solution. M is a parameter that needs to be adjusted, since the better the approximation is, the lower the value of M needs to be.

To examine the potential savings that may be obtained through the use of the proposed time-dependent approach, several tests were conducted. The next section reports computational results obtained with the new algorithm.

5 Computational results

In this section, we report experimental results obtained with the time-dependent algorithm. First, we describe the test problems and then we report numerical results.

5.1 Test problems

Our model was validated on Solomon's 100-customer Euclidean problems (see Solomon(87)). In these problems, customer locations are generated within a $[0, 100]^2$ square. Six different sets of problems are defined, namely C_1 , C_2 , R_1 ,

R_2 , RC_1 and RC_2 . The customers are uniformly distributed in the problems of type R , clustered in groups in the problems of type C and mixed in the problems of type RC . In the problems of type 1, only a few customers can be serviced on each route due to a narrow time window at the depot, as opposed to problems of type 2 where each route may have many customers. The travel times were calculated using a 3x3 time-dependent travel speed matrix $(v_{cT})_{1 \leq c \leq 3, 1 \leq T \leq 3}$, where each row corresponds to a category of arc and each column to a time period. Within the scheduling horizon, the first and third periods stand for the morning and evening rush hours, respectively. The second period corresponds to the middle of the day, when the traffic density is lower.

<i>Scenario1</i>			
<i>T</i>			
	0.54	0.81	0.54
<i>C</i>	0.81	1.22	0.81
	1.22	1.82	1.22
<i>Scenario2</i>			
<i>T</i>			
	0.33	0.67	0.33
<i>C</i>	0.67	1.33	0.67
	1.33	2.67	1.33
<i>Scenario3</i>			
<i>T</i>			
	0.12	0.46	0.12
<i>C</i>	0.46	1.92	0.46
	0.96	3.84	0.96

C = Category of arc.

T = Time period.

Table 1: *Travel speed matrices in scenarios 1, 2, 3*

Entries of the travel speed matrix were adjusted to create three different types of scenarios. For each scenario, the travel speeds in the morning and evening rush hours were obtained by dividing the travel speeds in the middle of the day by a factor α . In scenarios 1, 2 and 3, α was set to 1.5, 2 and 4 respectively. Hence,

scenario 3 is the one with the highest degree of time-dependency, while scenario 1 is the one with the lowest. The travel speed matrices for the three scenarios are reported in table 1. The average speed in each matrix is approximately 1, so the "average" difficulty is the same as in Solomon's original problems.

5.2 Experiments

The experiments reported in this section were performed on a network of 9 SUN UltraSparc-III workstations (300 MHz). As mentioned previously, the objective is to minimize the sum of total travel time and total lateness over all customers. In these experiments, the fleet size was set to the number of routes in the best solution reported in the literature for each problem. The values of the parameters were kept as in the original algorithm (see Gendreau et al. (99)). With respect to parameter M , which corresponds to the number of best moves (according to the approximation) which are evaluated exactly, several values were tested.

For each scenario, each problem is solved assuming time-dependent travel speeds, and then assuming constant speeds (i.e., for each category of arcs, the average speed is taken over the three time periods). The two solutions are then compared, using the time-dependent travel speed matrix, to evaluate what is gained by explicitly considering variations in travel times over the day rather than using the approximation represented by the average.

5.2.1 Preliminary tests

Scenario 1 was first considered for the preliminary tests. The intent was to find the best value for parameter M . The preliminary experiments were conducted over a small sample of problems selected in each of the six classes C_1, C_2, R_1, R_2, RC_1 , and RC_2 . Several values had to be tested for every problem in the sample. Thereby, a significant amount of computation time is required if the size of the sample is too large. On the other hand, this size has to be large enough to get a good calibration. In our experiments, the size of the sample was set to 4. Table 2 reports results obtained with the time-dependent travel speed matrix of scenario 1, and the values 4, 8, 16, 24, 32 and 40 for parameter M . The four numbers in each

entry are the average travel time, lateness, objective value and running time (in minutes) respectively, for each problem class. In these experiments, the algorithm stops after 50 restarts from the adaptive memory (see subsection 4.1). As expected, increasing the value of M increases the running time and decreases the objective value. However, the resulting increase in running time is small when compared to the total running time. In fact, since the neighborhood of a given solution is large, the algorithm spends much more time at evaluating the neighboring solutions, even approximatively, than it does at evaluating a few solutions exactly. Table 2 was used to determine the value for M that achieves the best trade-off between execution time and solution quality. We observed that $M = 32$ and 40 lead to approximatively the same results, while the execution time for $M = 32$ is slightly smaller. Hence, the value 32 was retained for further testing.

To quantify the effort spent by the proposed time-dependent model when compared with the original algorithm, we computed the amount of time consumed by one iteration of each algorithm. Preliminary results have shown that the time-dependent model leads to a very small increase in computational time (about 15.86 seconds against 15.29 seconds for problems of classes C_1 , R_1 and RC_1 , and about 89.78 seconds against 86.36 seconds for problems of classes C_2 , R_2 and RC_2).

5.2.2 Numerical results

Table 3 compares, for scenarios 1, 2 and 3, the solutions obtained with time-dependent travel speeds and the solutions obtained with constant speeds over all problems in each category. The four numbers in each entry are the fraction of infeasible solutions, the average travel time, lateness and objective value, respectively, for each problem class. The row "Overall" contains averages taken over the entire set of 56 test problems. It is worth noting that a solution obtained with constant speeds may well be infeasible in the time-dependent context (i.e., the upper bound of the time window at the depot may well be exceeded). This is what the first number in each entry indicates. The results show that a significant number of solutions obtained with constant speeds are infeasible in the time-dependent context and that this number increases with the degree of time-dependency. Also,

the use of time-dependent travel speeds considerably improves the objective value. This improvement is observed under the three scenarios, for all problem classes. More precisely :

Scenario 1: An improvement is observed in all classes of problems (apart from R_2). The reduction ranges from 1.0% to 5.0%. In the case of R_2 , the two models lead to approximatively the same results.

Scenario 2: As expected, the model performs better under scenario 2 where the degree of time-dependency is higher. In fact, the results show an improvement in all problem classes. The reduction ranges from 2.0% to 12.5%.

Scenario 3: In this scenario, the results are the most impressive. This is not surprising since the degree of time dependency is the highest. An improvement is observed in all problem classes. The reduction in the objective value ranges from 9.2% to 18.0%.

As we can see, the results obtained with time-dependent travel speeds are (almost) systematically better than those obtained with constant speeds. This is not really surprising, given that the average speed is a gross approximation of real conditions. Furthermore, this approximation gets worse when the degree of time dependency increases. It should also be noted that all numbers in Table 3 are averages taken over a number of problems (from 8 to 12 problems, depending on the problem class). These averages reflect the natural tendency of time-dependent solutions to be of better quality, although this is not necessary the case on specific instances within a problem class. The next section will now describe the dynamic version of the problem, where similar trends are observed.

6 Dynamic problem

The proposed model was also tested in a dynamic environment. In this section, the dynamic time-dependent vehicle routing problem is first introduced. Then, we explain how the algorithm of Taillard et al. (97) for the static problem with fixed travel times was adapted to the dynamic case (Gendreau et al. (99)). We then describe how we modified the later algorithm to cope with time-dependent travel

times in a dynamic environment. Finally, computational results are reported.

6.1 Problem definition

As opposed to the static problem, the number of service requests are not known completely ahead of time, but are rather dynamically revealed as time goes by.

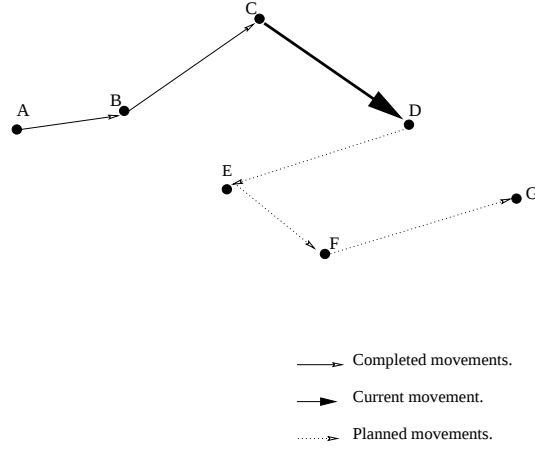


Figure 8: A vehicle route in a dynamic setting.

As illustrated in figure 8, in a dynamic environment, a vehicle route can be divided into three parts at any instant t :

- completed movements which form the part of the route already executed. Thereby, this part cannot be modified anymore;
- current movement of the vehicle toward its current destination;
- planned movements which constitute the portion of the route not yet executed by the vehicle (planned route).

Whenever a new request unfolds at instant t , the problem is to assign this request to a particular vehicle and incorporate it into its planned route at minimum cost.

6.2 Original dynamic algorithm (Gendreau et al. (99))

In a dynamic setting, at any instant t , a solution is a set of planned routes, each beginning with the current destination of the associated vehicle. The major

modification for adapting the static algorithm to the dynamic case was related to the management of new incoming events. In Gendreau et al. (99), a new event may be of two types: the occurrence of a new service request, or the completion of service at a customer. This latter event is related to the fact that the driver has no prior knowledge of his planned route. Consequently, he must be informed of his next destination as soon as he has completed the service at his current location. Whenever any of these two types of events occurs, the tabu search threads are interrupted. Then, after an appropriate update of the adaptive memory to reflect the current state of the world, the search is restarted with new solutions constructed from the updated memory. The tabu search is thus used to improve the current (static) solution between the occurrence of new events.

6.3 Time-dependent algorithm in a dynamic environment

In addition to the modifications already discussed in subsection 4.2, another modification is required in the dynamic case. In the original algorithm, a least commitment strategy is considered. That is, if there is some time flexibility at the vehicle's next destination, the vehicle waits at its current location rather than its next destination. The intent is to allow the vehicle to service a new request that may appear in the vicinity of its current location. Hence, one has to determine a value for the departure time that allows the vehicle to reach its next customer no earlier than the lower bound of its time window.

When time-dependency is taken into account, the adjustment of the vehicle departure time is more complicated, because the travel time between a given pair of locations depends on the departure time from the origin. Hence, one has to take into account the change that occurs in travel speed when the boundary between two consecutive time periods is crossed. This problem is solved through a backward recursive procedure (in contrast with the forward procedure used to compute the travel times between customers).

In figure 9, a route for one vehicle is considered. The horizon is "discretized" into two time periods $(T_k)_{1 \leq k \leq 2}$. In this figure, i is the current customer and j is the next customer to be serviced, with arc (i, j) belonging to category 1.

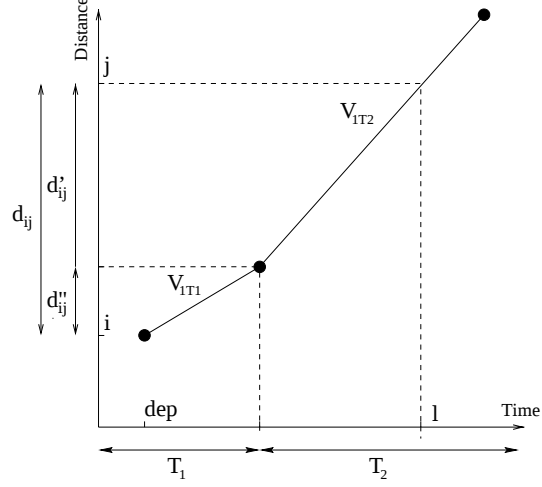


Figure 9: Finding the departure time for a vehicle in a dynamic environment.

We also suppose that the time of departure dep belongs to time period $T_1 =]\underline{t}_1, \bar{t}_1]$ and that l , the lower bound of the time window at customer j , belongs to time period $T_2 =]\underline{t}_2, \bar{t}_2]$, with $\bar{t}_1 = \underline{t}_2$. With the least commitment strategy, the departure time dep from node i is calculated as follows. First, we evaluate the distance traveled during time period T_2 which is $d'_{ij} = v_{IT_2}(l - \underline{t}_2)$. Since the remaining distance is $d''_{ij} = d_{ij} - d'_{ij}$, we obtain $dep = \bar{t}_1 - d''_{ij}/v_{IT_1}$.

A pseudo-code for this backward recursive procedure is provided in figure 10. For a given vehicle, it calculates the departure time from its current customer i to arrive at the lower bound l of the time window associated with its next destination j , where $l \in T_k =]\underline{t}_k, \bar{t}_k]$. In this figure, t is a time variable which is updated at each iteration and dep corresponds to the departure time. It is also assumed that arc (i, j) belongs to category c , $1 \leq c \leq C$.

The modified algorithm was used to assess the proposed time-dependent model in a dynamic context. Details about the experiments are reported in the next subsection.

```

1.  set  $t$  to  $l$ ,
    set  $d$  to  $d_{ij}$ ,
    set  $dep$  to  $l - (d/v_{cT_k})$ .
2.  while ( $dep < \underline{t}_k$ ) do
    2.1  $d \leftarrow d - v_{cT_k}(t - \underline{t}_k)$ .
    2.2  $t \leftarrow \underline{t}_k$ 
    2.3  $dep \leftarrow t - (d/v_{cT_{k-1}})$ .
    2.4  $k \leftarrow k - 1$ .
3.  return ( $dep$ ).

```

Figure 10: Departure time calculation procedure in a dynamic environment.

6.4 Computational results

6.4.1 Simulation framework

Simulations were performed to validate our model in a dynamic setting. As in the static case, data were taken from Solomon's 100-customer Euclidean problems (Solomon (87)). The same six classes of problems were considered, namely C_1 , C_2 , R_1 , R_2 , RC_1 and RC_2 . The set of requests was divided into two subsets. The first subset contains requests that are assumed to be known at the start of the day (50% of the entire requests in our implementation). The second subset contains requests that are received in real-time. Using minutes as time units for time-related data in Solomon's files, the time horizon was set to 15 minutes. This leads to about 3 requests per minute. The interested reader will find more details about the simulator in Gendreau et al. (99).

6.4.2 Numerical results

The experiments were conducted within the same framework used in the static case (see section 5). Hence, the three scenarios previously considered were examined in the dynamic setting, using the same parameter values than those used in the static environment. In the dynamic context, some requests may be rejected when they occur because no feasible insertion place is found in the current routes.

This phenomenon is rather marginal, but a few solutions may contain only 98 or 99 customers (rather than 100 customers). Table 4 compares, for scenarios 1, 2 and 3, the solutions obtained with time-dependent travel speeds to those obtained with constant speeds. The four numbers in each entry are the fraction of infeasible solutions, the average travel time, lateness and objective value, respectively, for each problem class. The row “Overall” contains averages taken over the entire set of 56 test problems. In the following, we summarize the main findings for each scenario.

Scenario 1: For all problem classes, the time-dependent model leads to substantial improvements to the objective value, ranging from 4.0% to 13.2%. A total of 18 solutions produced with constant speeds are now infeasible in the time-dependent context.

Scenario 2: As expected, the time-dependent model leads to larger improvements in the objective value with regard to scenario 1, given the higher degree of time-dependency. The improvement now ranges from 21% to 75%, depending on the problem class. Furthermore, 32 solutions obtained with constant speeds are infeasible (i.e., more than half of the solutions).

Scenario 3: The degradation of the model based on constant speeds appears clearly in this case. The time-dependent model provides improvements ranging from 62.5% to 78.0%, depending on the problem class, and almost all solutions (except four) obtained with constant speeds are infeasible in the time-dependent context.

7 Conclusion

Time-dependent vehicle routing is still a very challenging area that needs to be explored, since it provides a more accurate way to model real problems. This paper proposed a time-dependent model for a vehicle routing problem with time windows, based on time-dependent travel speeds, which satisfies the FIFO assumption. Characteristics of the model were addressed and discussed. Then, experiments were performed to evaluate the model in a static and a dynamic

environment. The results show that the time-dependent model provides very significant improvements over the model with fixed travel times, thus indicating the usefulness of additional information about the problem. Future work will now be aimed at trying to exploit probabilistic information about the future to make better dispatching decisions.

<i>Problem set</i>	$M = 4$	$M = 8$	$M = 16$	$M = 24$	$M = 32$	$M = 40$
C_1 4 <i>problems</i>	905.188 ^a	856.708	831.53	829.965	827.225	833.62
	31.765 ^b	27.645	25.272	27.687	25.272	25.272
	936.952 ^c	884.352	856.802	857.655	852.498	858.635
	27.833 ^d	28.039	28.962	29.284	29.304	29.508
R_1 4 <i>problems</i>	1023.62	997.88	980.392	972.38	973.195	963.225
	59.432	51.59	46.91	47.537	37.767	52.007
	1083.05	1049.47	1027.30	1019.92	1010.96	1015.23
	23.579	23.65	24.0625	24.632	24.95	24.929
RC_1 4 <i>problems</i>	1090.54	1064.38	1039.40	1036.74	1030.57	1030.68
	49.517	28.002	40.975	38.935	37.225	40.272
	1140.06	1092.38	1080.37	1075.08	1067.79	1070.95
	22.602	22.333	22.983	23.404	23.45	23.579
C_2 4 <i>problems</i>	836.25	758.518	785.675	763.92	759.705	737.045
	25.34	35.037	35.702	34.667	18.562	39.90
	861.59	793.555	821.378	798.588	778.268	776.945
	11.033	11.03	12.201	12.218	12.65	12.921
R_2 4 <i>problems</i>	996.258	1020.79	969.792	940.088	943.03	941.18
	68.365	42.262	52.767	22.342	20.62	16.725
	1064.62	1063.05	1022.56	962.43	963.65	957.905
	15.891	19.412	23.596	28.271	29.27	29.133
RC_2 4 <i>problems</i>	1162.63	1155.73	1087.93	1067.97	1015.51	1035.16
	73.742	39.86	10.247	22.437	29.47	9.362
	1236.37	1195.59	1098.18	1090.41	1044.98	1044.522
	10.912	12.558	16.783	16.515	16.867	16.903

^aTravel time.

^bLateness.

^cObjective value.

^dRunning time (in minutes).

Table 2: Searching for the best value for parameter M

<i>Problem set</i>	<i>Scenario1</i>		<i>Scenario2</i>		<i>Scenario3</i>	
	<i>Time – dep. speed</i>	<i>Const. speed</i>	<i>Time – dep. speed</i>	<i>Const. speed</i>	<i>Time – dep. speed</i>	<i>Const. speed</i>
C_1 9 <i>problems</i>	0 ^a	0	0	0	0	0.222
	818.57 ^b	836.878	836.787	857.241	984.556	1044.23
	21.514 ^c	15.764	112.553	109.45	399.28	637.99
	840.084 ^d	852.642	949.34	966.691	1383.84	1682.22
R_1 12 <i>problems</i>	0	0	0	0.167	0	0.417
	935.188	956.213	778.801	816.498	710.076	798.768
	26.461	25.655	36.395	33.692	48.907	51.53
	961.648	981.868	815.197	850.19	758.983	850.298
RC_1 8 <i>problems</i>	0	0	0	0.125	0	0.125
	1038.44	1063.02	873.741	908.692	825.97	900.741
	30.927	32.776	26.756	27.634	22.396	33.524
	1069.36	1095.79	900.498	936.33	848.366	934.265
C_2 8 <i>problems</i>	0	0	0	0.25	0	0.75
	659.221	699.714	764.091	810.00	1250.28	1119.10
	13.361	6.507	80.852	154.67	564.28	1038.26
	672.582	706.221	844.944	964.676	1814.56	2157.36
R_2 11 <i>problems</i>	0	0	0	0.091	0	0.364
	939.255	934.775	861.763	978.285	782.594	873.669
	10.698	9.18	8.100	16.581	18.465	40.546
	949.954	943.955	869.864	994.866	801.059	914.215
RC_2 8 <i>problems</i>	0	0.125	0	0	0	0.5
	1078.17	1081.59	988.124	1032.44	979.758	1034.29
	28.998	35.212	22.352	6.039	23.917	63.425
	1107.17	1116.803	1010.48	1038.48	1003.68	1097.71
Overall 56 <i>problems</i>	0	0.018	0	0.107	0	0.393
	919.612	929.349	859.806	897.917	900.687	946.905
	21.645	20.476	61.623	54.939	165.504	283.713
	941.257	949.825	921.430	952.855	1066.193	1230.618

^aFraction of infeasible solutions.

^bTravel time.

^cLateness.

^dObjective value.

Table 3: Comparison of time-dependent and constant speeds under the three scenarios in a static setting (with $M = 32$)

<i>Problem set</i>	<i>Scenario1</i>		<i>Scenario2</i>		<i>Scenario3</i>	
	<i>Time – dep. speed</i>	<i>Const. speed</i>	<i>Time – dep. speed</i>	<i>Const. speed</i>	<i>Time – dep. speed</i>	<i>Const. speed</i>
C_1 9 <i>problems</i>	0 ^a	0	0	0.222	0	1.0
	895.718 ^b	947.766	949.783	1080.75	1208.54	2058.99
	40.21 ^c	74.135	129.74	367.248	473.072	3190.72
	935.928 ^d	1021.90	1142.52	1447.99	1681.61	5249.71
R_1 12 <i>problems</i>	0	0.583	0	0.833	0	1.0
	1069.36	1199.90	931.956	1271.13	875.012	1506.57
	92.531	126.793	62.461	395.985	496.492	1490.15
	1161.89	1326.69	994.418	1667.12	924.661	2996.72
RC_1 8 <i>problems</i>	0	0.625	0	1.0	0	1.0
	1168.49	1288.60	1023.82	1364.41	974.794	1485.78
	132.084	210.266	69.048	595.28	47.77	1238.44
	1300.58	1498.87	1092.87	1959.69	1022.56	2724.22
C_2 8 <i>problems</i>	0	0.25	0	0.75	0	1.0
	712.542	696.881	884.101	751.332	1103.11	1184.45
	15.751	61.495	44.102	250.248	516.088	1660.43
	728.294	758.376	928.204	1001.58	1619.19	2844.87
R_2 11 <i>problems</i>	0	0.273	0	0.273	0	0.818
	1027.61	1086.55	971.037	1242.67	973.65	1520.63
	16.396	74.951	12.954	480.605	23.637	2934.54
	1044.00	1161.50	983.992	1723.27	997.287	4455.17
RC_2 8 <i>problems</i>	0	0.125	0	0.375	0	0.75
	1196.53	1253.62	1068.67	1342.52	1010.20	1783.80
	24.68	40.628	16.956	209.471	31.931	1779.71
	1221.21	1294.25	1085.62	1552.00	1042.14	3563.51
Overall 56 <i>problems</i>	0	0.321	0	0.571	0	0.929
	1014.60	1085.599	968.315	1184.210	1014.14	1588.731
	54.156	98.434	55.366	388.995	272.17	2076.908
	1068.76	1184.033	1033.80	1573.205	1286.31	3665.639

^aFraction of infeasible solutions.

^bTravel time.

^cLateness.

^dObjective value.

Table 4: Comparison of time-dependent and constant speeds under the three scenarios in a dynamic setting (with $M = 32$)

References

- [1] **M.O.Ball, T.L. Magnanti, C.L. Monna and G.L. Nemhauser (eds)**, *Network Routing*, Handbooks in Operations Research and Management Science, Volume 8, North-Holland: Amsterdam, 1995.
- [2] **G.G. Brown, C. J. Ellis, G. Lenn, W. Graves, David Ronen**, "*Real Time, Wide Area Dispatch of Mobil Tank Trucks*" , Interfaces **17**, 107-120 (1987).
- [3] **I. Chabini**, "*Fastest Routes in Temporal Networks Revisited*", Presented at *Optimization days*, Montreal, May 1996.
- [4] **I. Chabini**, "*A New Algorithm for Shortest Paths in Discrete Dynamic Networks*", Proc. 8th IFAC Symposium on Transportation Systems, Chania, Greece, 551-556 (1997).
- [5] **K.L. Cooke and E. Halsey**, "*The Shortest Route Through a Network with Time-Dependent Internodal Transit Times*", Journal of Math. Anal. Appl. **14**, 492-498 (1966).
- [6] **A. De Palma, P. Hansen, M. Labbé**, "*Commuters' Paths with Penalties for Early or Late Arrival Time*", Transportation Science **24 (4)**, 276-286 (1990).
- [7] **S.E. Dreyfus**, "*An Appraisal of Some Shortest Path Algorithms*", Operations Research **17**, 395-412 (1969).
- [8] **M.L. Fisher, A.J. Greenfield, R. Jaikuman and J.T. Lester**, "*A Computerized Vehicle Routing Application*", Interfaces **12 (4)**, 42-52 (1982).
- [9] **L.R. Ford Jr. and D.R. Fulkerson**, "*Constructing Maximal Dynamic Flows from Static Flows*", Operations Research **6**, 419-433 (1958).
- [10] **K.R. Fox, B. Gavish, S.C. Graves**, "*An n -Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem*", Operations Research **28 (4)**, 1018-1021 (1980).

- [11] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard**, "*Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*", *Transportation Science* **33** (4), 381-390 (1999).
- [12] **F. Glover and M. Laguna**, *Tabu Search*, Kluwer, 1997.
- [13] **G. Hadley**, "*Nonlinear and Dynamic Programming*", Addison-Wesley Publishing Co, Inc., Reading, Mass., 1964.
- [14] **R.W. Hall**, "*The Fastest Path through a Network with Random Time-Dependent Travel Times*", *Transportation Science* **20** (3), 182-188 (1986).
- [15] **J. Halpern**, "*The Shortest-Route with Time-Dependent Length of Edges and Limited Delay Possibilities in Nodes*", *Zeitschrift fuer Operations Research* **21**, 117-124 (1977).
- [16] **M.D. Hickman, D.H. Bernstein**, "*Transit Service and Path Choice Models in Stochastic and Time-Dependent Networks*", *Transportation Science* **31** (2), 129-146 (1997).
- [17] **M.D. Hickman and N.H.M. Wilson**, "*Passenger Travel Time and Path Choice Implications of Real-Time Transit Information*", *Transpn. Res.-C* **3** (4), 211-226 (1995).
- [18] **A.V. Hill and W.C. Benton**, "*Modeling Intra-City Time-Dependent Travel Speeds for Vehicle Scheduling Problems*", *J. Opl. Res. Soc.* **43** (4), 343-351 (1992).
- [19] **A. Hill, V. Mabert and D. Montgomery**, "*A Decision Support System for the Courier Vehicle Scheduling Problem*", *Omega Int. J. of Mgmt Sci.* **16**, 333-345 (1988).
- [20] **C. Malandraki**, "*Time Dependent Vehicle Routing Problems: Formulations, Solution Algorithms and Computations Experiments*", Ph.D. dissertation, Northwestern University, Evanston, III., 1989.

- [21] **C. Malandraki, M.S. Daskin**, "*Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms*", *Transportation Science* **26 (3)**, 185-200 (1992).
- [22] **C. Malandraki, R.B. Dial**, "*A Restricted Dynamic Programming Heuristic Algorithm for the Time Dependent Traveling Salesman Problem*", *Eur. J. Oper. Res.* **90**, 45-55 (1996).
- [23] **P.H.J. Marguier and A. Ceder**, "*Passenger Waiting Strategies for Overlapping Bus Routes*", *Transportation Science* **18 (3)**, 207-230 (1984).
- [24] **C.E. Miller, A.W. Tucker and R.A. Zemlin**, "*Integer Programming Formulation on Traveling Salesman Problems*", *J. Assoc. Comput. Mach.* **7**, 326-329 (1960).
- [25] **K. Nachtigall**, "*Time-Dependent Shortest-Path Problems with Applications to Railway Networks*", *European Journal of Operational Research* **83**, 154-166 (1995).
- [26] **A. Orda and R. Rom**, "*Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length*", *J. ACM* **37**, 607-625 (1990).
- [27] **J.C. Picard, M. Queyranne**, "*The Time-Dependent Traveling Salesman Problem and its Application to the Tardiness Problem in One-Machine Scheduling*", *Operations Research* **26 (1)**, 86-110 (1978).
- [28] **H.N. Psaraftis, J.N. Tsitsiklis**, "*Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs*", *Operations Research* **41 (1)**, 91-101 (1993).
- [29] **J.M. Rousseau, and S. Roy**, "*RAO Répartition Assistée par Ordinateur: la Description du Prototype*", publication CRT-564, Centre de recherche sur les transports, Université de Montréal, Canada, 1988.
- [30] **Y. Shen and J.Y. Potvin**, "*A Computer Assistant for Vehicle Dispatching with Learning Capabilities*", *Operations Research* **61**, 189-211 (1995).

- [31] **M.M. Solomon**, "*Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints*", *Operations Research* **35**, 254-265 (1987).
- [32] **É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin**, "*A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*", *Transportation Science* **31**, 170-186 (1997).
- [33] **A.K. Ziliaskopoulos**, "*Optimum Path Algorithms on Multidimensional Networks: Analysis, Design, Implementation and Computational Experience*", Ph.D. Dissertation, University of Texas at Austin, 1994.
- [34] **A.K. Ziliaskopoulos and H.S. Mahmassani**, "*Time-Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications*", *Transportation Research Record* **1408**, 94-100 (1993).

List of Figures

1	Travel time function on a link.	7
2	Travel speed function at a node.	8
3	Changing travel speed over time.	10
4	Travel time calculation procedure.	11
5	An example of travel speed and travel time functions.	12
6	A CROSS exchange.	14
7	Approximate evaluation of lateness.	16
8	A vehicle route in a dynamic setting.	22
9	Finding the departure time for a vehicle in a dynamic environment.	24
10	Departure time calculation procedure in a dynamic environment.	25

List of Tables

1	<i>Travel speed matrices in scenarios 1, 2, 3</i>	18
2	<i>Searching for the best value for parameter M</i>	28
3	<i>Comparison of time-dependent and constant speeds under the three scenarios in a static setting (with $M = 32$)</i>	29
4	<i>Comparison of time-dependent and constant speeds under the three scenarios in a dynamic setting (with $M = 32$)</i>	30