



Robust routing, its price, and the tradeoff between routing robustness and travel time reliability in road networks

Farida Manseur, Nadir Farhi, Cyril Nguyen van Phu, Habib Haj Salem, Jean Patrick Lebacque

► To cite this version:

Farida Manseur, Nadir Farhi, Cyril Nguyen van Phu, Habib Haj Salem, Jean Patrick Lebacque. Robust routing, its price, and the tradeoff between routing robustness and travel time reliability in road networks. *European Journal of Operational Research*, 2020, 285 (1), pp.159-171. 10.1016/j.ejor.2018.10.053 . hal-01959493

HAL Id: hal-01959493

<https://hal.science/hal-01959493>

Submitted on 18 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust routing, its price, and the tradeoff between routing robustness and travel time reliability in road networks

Farida Manseur, Nadir Farhi¹, Cyril Nguyen Van Phu, Habib Haj-Salem, Jean-Patrick Lebacque

Université Paris-Est, Ifsttar, Cosys, Grettia, F-77447 Marne-la-Vallée, France.

Abstract

We propose in this article an adaptive algorithm for optimal and robust guidance for the users of the road networks. The algorithm is based on the Stochastic On Time Arrival (SOTA) family of routing algorithms, which is appropriate for taking into account the variability of travel times through the road networks. The SOTA approach permits the derivation of the maximum cumulative probability distribution of the time arrival toward a given destination in the network. Those distributions allow the selection of the most reliable origin-destination paths under given travel time budgets. We investigate here the introduction of robustness against link and path failures in the criterion of the guidance strategy selection. Our algorithm takes into account the reliability of itinerary travel times, since it is based on a SOTA approach. In addition, the algorithm takes into account itinerary robustness, by favoring itineraries with possible and reliable alternative diversions, in case of link failures, with respect to itineraries without or with less reliable alternatives. We first analyze the algorithm in its static version, without considering the traffic dynamics, and show some interesting properties. We then combine the robust guidance algorithm with a dynamic traffic model by using the traffic simulator SUMO (Simulation of Urban Mobility), and illustrate its effectiveness in some dynamic scenarios.

Keywords: Optimal routing, Robust guidance, Travel time reliability, Optimization, Traffic control.

¹Corresponding author. E-mail: nadir.farhi@ifsttar.fr

Introduction

The most common way to select a route in a given network consists in determining the shortest path from the origin to the destination. Shortest path problems are intensively studied in the network theory and operations research, with various applications in many scientific and engineering fields, particularly in transportation engineering. Shortest path algorithms are widely available in the literature. In case of having perfect information on the link travel-times, a shortest path problem may be solved for example by Dijkstra algorithm [7], or more generally by value iteration [23]. If we have a known dependence on the link travel-times, an optimal strategy may be a sequence of links, i.e. a path, because no additional information can be acquired therein. In case of a stochastic shortest path problem, link travel-times are random variables, and available information is given in the form of travel-time probability distributions. Several formulations have been proposed in the literature, associating one or many criteria for the optimal path selection. We cite for example, -minimizing the expected travel-time [15, 28], -maximizing the expected utility [19, 29], -maximizing the probability of arriving at the destination on time [9, 36], -minimizing the expected travel-time while ensuring a pre-specified probability of arriving by a given deadline [33, 3, 32, 31, 34], and minimizing the sum of expected travel-time [24, 33, 4, 40]. The stochastic shortest path problem solutions are either a priori paths or adaptive strategies. In the a priori case, the path is determined based on the information available at the starting time only. In the adaptive case, the solution consists in a routing policy that adapts the path according to information obtained during the travel. In a stochastic framework, different performance criteria may lead to different optimal routing strategies.

Among from several formulations of the stochastic shortest path problem, the basic one is most likely the Least Expected Time (LET). This formulation is based on the minimization of the expected travel-time. Determining a LET path when random link travel-times are independently distributed and constant over time, is trivial. Fortunately, LET problems have been studied broadly and have been extended to other cases. Hall [18] studied the LET problem in a stochastic time-dependent network and considered that the travel-times on outgoing links from a node are conditional on the arrival time to the node. The authors of [15, 16, 17] addressed the LET problem by including real-time information about the travel-time. In [15], Fu has shown that the relationship between route planning and information can be charac-

terized in three schemes. The first one is the *non-adaptive routing* rule which is made at the beginning of the task. In this category, a complete fixed path is identified on the basis of a priori or historical travel-time information. In general, this complete path is computed before a trip starts and no re-route adaptive diversion is taken into account, either because of lack of real-time information, or of an unavailability of route guidance system. The second category is the *open-loop adaptive routing*. It is similar to the non-adaptive routing in the sense that a complete path must be computed. However, in this case, the remaining path by non-adaptive routing is recalculated every time new data become available. The third class is the *closed-loop adaptive routing*. Contrary to the open-loop category, the closed-loop one takes into account the future availability of travel-time information, and thus, specifies at each node how to react to the new information.

The LET formulation does not take into account travel time variations. In other words, LET solutions are risk-neutral and do not depend on the uncertainties associated to the link travel-times. Utility functions [14, 30] can be used to model attitude towards risk in stochastic shortest path problems; see for example [19, 29]. Other heuristic approaches for the risk aversion in such problems can be found for example in [33, 4]. However, most of these approaches result in extremely complex formulations in relation to the objective function that does not accumulate over the links, in which case, dynamic programming cannot be applied [25].

For the stochastic path planning optimization, Frank [13] introduced the objective of maximizing the probability to reach the destination node under a given time deadline. Since the work of Frank, the on-time arrival problems have been explored by many researchers [9, 11, 24, 36]. In [9], Fan et al. defined optimality as the maximization of the travel-time reliability, and proposed an adaptive optimal path algorithm to solve this optimization problem, called the Stochastic On-Time Arrival (SOTA) problem. In this section, for the convenience of readers and for the continuity of our discussion in the next sections, we give a brief review and a discussion on the concept of the SOTA formulation including two existing variants.

In the SOTA formulation, one seeks to maximize the probability of the time arrival at a given destination, departing from a given origin, with a given travel-time budget. The travel-time across every link is a random variable with some arbitrary probability distribution. Two primary variants can be distinguished in the SOTA model. The first one consists in finding

the most reliable non adaptive path to the destination. It is designed as the shortest path problem on-time arrival reliability, or as the path-based SOTA problem; see [32]. The second variant is referred as the policy-based SOTA problem, which consists in calculating a routing policy such as the selection of the next node at each intersection, depending on the current state (remaining time budget).

The policy-based SOTA problem is solved in discrete-time. In [10], the authors presented a discrete SOTA algorithm that ensures finite convergence and runs very well in polynomial time. Solving the SOTA problem in discrete-time allows computing product convolution of arbitrary distributions. The computation of the policy requires a subsequent maximization step. Unfortunately, this step mixes distributions and prevents finding an analytical solution in continuous-time. A successive approximation method is proposed in [11] for solving the policy-based SOTA problem. This algorithm is improved in [36, 37] to a dynamic programming algorithm, and the speed-up techniques including zero-delay convolution [6] have been explored to solve the problem in pseudo-polynomial-time. The authors of [35] have shown how pre-processing methods can be used to further reduce the computation time of the SOTA problem. Unfortunately, the structure of the SOTA problem formulation limits the types of pre-processing methods that can be used for this problem, and prevents massive running time reductions in the deterministic case. Recently, the authors of [22] presented a novel approach to reduce the immense computational effort of stochastic routing based on existing techniques for alternative routes.

In this paper an extension of the SOTA algorithm reported in [36] is presented. This extension permits to include robustness in the criteria of the routing optimization. Instead of considering a unique objective of maximizing the travel-time reliability, we propose to consider also the optimization of the robustness of the selected paths and routing policies against link failures. Link failures refer here to a effective cuts of links caused for example by accidents, works, etc. excluding slowdowns caused by traffic congestions. Finally, lets notice that the approach we propose here is rather an individual user guidance than a traffic assignment. The failure of a link is taken into account in the calculus of a new path (in case of off-line routing) and of a new routing strategy (in case of on-line routing) for one user and for one origin-destination pair. The effect of the link failure on the paths of other origin-destination pairs and on the flows on those paths are not taken into account here. The remainder of this

paper is organized as follows.

In section 1, we give a short review on the SOTA problem in both cases of uncorrelated and correlated link travel-times. In section 2, we consider the problem of robust guidance in case of uncorrelated link travel-times. We begin this section by extending the stochastic on-time arrival (SOTA) algorithm proposed in [36]. We incorporate here the alternative routing choices at nodes by considering eventual link failures. A user being at a given node of the network has a number of routing possibilities for the next routing step. Each routing possibility corresponds to a successor node of the considered node. The probability for the user to reach the destination varies depending on the selected successor node. The existing SOTA model takes the maximum over all the probabilities of reaching the destination node in a given time budget. The routing model we propose here takes a weighted mean, instead of a maximum, of those probabilities. As explained below in section 2, the weights of the mean operator depend on the probabilities to reach the destination node from every successor node. Therefore, the model is not trivial. We provide a new definition of robust-optimality and explain how a user can measure the robustness, and evaluate the quality of the solution. Finally, we give a discretization scheme of the model for the case of uncorrelated link travel times, and analyze the complexity of the associated algorithm.

In section 3, we provide the robust guidance model in the case of correlated link travel-times. We present an extension of our model in order to consider eventual correlations of the travel times associated to neighboring links. In section 4, we propose a parameter optimization of the robust routing algorithm, and show how to choose the robustness parameter in function of the desired travel-time reliability and of the available travel-time budget.

In section 5, we conduct numerical calculus and compare two scenarios on a well-known road network (the Sioux Falls network) in the static case, where the traffic dynamics are not taken into account. By that, we assess the sensitivity of the proposed approach to changes in the key parameters. In section 6, we evaluate our routing approach in the context of dynamic guidance, where the traffic dynamics are taken into account through the microscopic simulation tool SUMO. Finally we draw some conclusions and perspectives for future research.

1. A short review on the stochastic on time arrival (SOTA) formulation

In this section, we briefly summarize the original SOTA problem for the convenience of readers and for the continuity of our discussion in the next sections. The details of the approach summarized here are available for example in [11, 27, 26, 12].

A road network is represented by a graph with arcs and nodes corresponding respectively to the links and the junctions of the road network. We denote the graph by $G(N, A)$, where N is the set of nodes, with $|N| = n$, and A is the set of arcs, with $|A| = a$. The set of successor and predecessor nodes of a given node i are denoted by $\Gamma^+(i) = \{j, (i, j) \in A\}$ and $\Gamma^-(i) = \{k, (k, i) \in A\}$ respectively. The travel times through the links of the network are assumed here to be stochastic. Probability distributions of the travel times are then associated to the links of the network. The SOTA problem consists in finding the best routing strategy from any starting node $i, (i = 1, 2, \dots, n)$, that maximizes the probability of arriving to a given destination node, denoted d , within a time budget t .

Given a node $i \in N$ and a time budget t , $u_i(t)$ denotes the maximum probability of arriving to the destination node d departing from node i , within a time budget t , and under the optimal routing policy. The latter, denoted by $s_i(t), i \in N$, gives the optimal subsequent node of node i . The probability distribution functions (pdf) of the travel times on links (i, j) are denoted $p_{ij}(\cdot)$. They are assumed to be known and can for example be obtained using historical data or real-time traffic information. The maximum probability $u_i(t)$ and the optimal successor node $s_i(t)$ are written as follows.

$$u_i(t) = \max_{j \in \Gamma^+(i)} \int_0^t p_{ij}(w) u_j(t-w) dw, \forall i \in N \setminus \{d\}, j \in \Gamma^+(i), 0 \leq t \leq T \quad (1)$$

$$u_d(t) = 1, 0 \leq t \leq T \quad (2)$$

$$s_i(t) = \arg \max_{j \in \Gamma^+(i)} \int_0^t p_{ij}(w) u_j(t-w) dw, \forall i \in N \setminus \{d\}, j \in \Gamma^+(i), 0 \leq t \leq T \quad (3)$$

where T is the maximum time budget.

Formula (1) expresses the fact that a traveler, being at node i , having a time budget t , and knowing $u_p(t), \forall p \in \{1, 2, \dots, n\}, \forall s \in [0, t]$, should go through the link (i, j) that maximizes the probability of arriving within time t to the destination node d , with respect to all the possible successor nodes j of i . Formula (2) tells simply that departing from node d , the maximum probability of arriving to the same node d , within any time budget is 1. Formula (3) tells that

the optimal successor node for the traveler being at node i , is given as the argument of the maximum taken in formula (1).

One of the possible approaches to solve the system of nonlinear equations (1)-(2) is the Picard method of successive approximation proposed in [9]. This fixed point method starts with initial approximations of the solution and refines these approximations by successive iterations. The iterative relationships for successive approximations are given as follows.

$$u_i^{iter=0}(t) = 0, \forall i \in N \setminus \{d\}, \forall t \in [0, T] \quad (4)$$

$$u_d^{iter=0}(t) = 1, 0 \leq t \leq T \quad (5)$$

$$u_i^{iter+1}(t) = \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(w) u_j^{iter}(t-w) dw, \forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), 0 \leq t \leq T \quad (6)$$

$$u_d^{iter+1}(t) = 1, 0 \leq t \leq T, \quad (7)$$

where the superscript $iter$ is the iteration index. The function $u_i^{iter}(t)$ represents the probability of reaching the destination node d if optimal choices are made, and using a path with no more than $iter$ intermediate nodes. It is easy to show that $u_i^{iter}(t)$ belongs to $[0, 1]$ and is increasing with respect to the iteration index $iter$.

$$0 \leq u_i^{iter}(t) \leq u_i^{iter+1}(t) \leq 1, \quad \forall iter \in \mathbb{N}, \forall i \in N, \forall t \in [0, t] \quad (8)$$

The sequence $\{u_i^{iter}(t)\}_{iter \geq 0}$ converges as $iter$ increases and it reaches its limit when $iter$ is equal to the number of links in the optimal path; see [11] for more details.

Formulas (1)-(2) present the simple case of the SOTA problem where the travel-times on the links of the network are uncorrelated. Different variants of the SOTA problem with correlations are proposed in the literature, see for example [9, 36]. We expose below the model with travel-time correlations proposed in [36].

In [36], the authors presented a simple extension of the SOTA model that considered correlations between any two consecutive links. Let us denote by $u_{ki}(t, y)$ the maximum probability for a user to arrive to destination node d within a time t , departing from node i , conditioned that the user comes from node k , predecessor of i , and that the realized travel time on link (k, i) is y . The maximum probabilities $u_{ki}(t, y)$ are written as follows.

$$u_{ki}(t, y) = \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(w|y) u_{ij}(t-w, w) dw, \quad (9)$$

$$\forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), j \in \Gamma^{+1}(i), 0 \leq t \leq T, 0 \leq y \leq T-t,$$

$$u_{kd}(t, y) = 1, \forall k \in \Gamma^{-1}(d), 0 \leq t \leq T, 0 \leq y \leq T-t, \quad (10)$$

$$s_{ki}(t, y) = \arg \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(w|y) u_{ij}(t-w, w) dw, \quad (11)$$

$$\forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), j \in \Gamma^{+1}(i), 0 \leq t \leq T, 0 \leq y \leq T-t,$$

where $p_{ij}(w|y)$ denotes the probability distribution function (pdf) of the travel time on link (i, j) conditioned on the travel time on link (k, i) . The pdf $p_{ij}(\cdot)$ is assumed to be known and can be obtained for example, using historical data or real-time traffic information. $u_{ij}(t-w, w)$ is the maximum probability of arriving to destination node d within time $t-w$, departing from node j , conditioned that the travel time on link (i, j) is w .

In the next sections, we will base on the SOTA formulation described above and propose an extension of the routing algorithm including a robustness criterion for the routing strategy, against link and path failures.

2. Robust guidance with uncorrelated link travel-times

In a stochastic framework, robustness is generally defined as the probability that the system of interest has the ability to resist to changes without adapting its initial stable configuration. Recently, path robustness in transport networks has emerged as an important topic. It has attracted many researchers to develop various indicators to assess the path robustness in road networks [20, 21]. In this paper we provide a new idea of incorporating robustness with route choice using the case that link failure may occur. As mentioned above, link failures refer here to a effective cuts of links caused for example by accidents, works, etc. excluding slowdowns caused by traffic congestions. We notice here that for a user who calculates an off-line routing path, if a link belonging to that path fails, then the link failure can be considered as a path failure for that user, since he needs to calculate another path. We consider here that a routing strategy is robust if it minimizes the deterioration of its maximum value calculated before

the depart at the origin, against eventual reconfigurations of the network that may be due to accidents, works, etc. The value of a strategy is maximized with respect to the average travel time and its reliability associated to the routing strategy. Consequently, a robust routing strategy would resist to network reconfiguration due to link failures. We notice here that the connectivity of the network is important for the robustness of routing strategies. Indeed, a path passing through nodes having many successor nodes would be more robust than a path passing through nodes having one or a low number of successor nodes.

The impact of a link failure on the traffic, and by that on the routing, depends on the importance of the link for the network topology. Many indices, called generally centrality indices (the average degree, the shortest path betweenness, the random walk betweenness, etc.), have been defined in the literature in order to measure the importance or the role of links in a given network; see for example [38, 1, 2, 5].

2.1. Continuous time formulation of the robust guidance algorithm

In this section we base on the routing model presented in [11] (formulas (1)-(2) above) where from the probability distributions of travel times through the links of the network, users evaluate their maximum probability to reach their destination in given time budgets, and passing through different possible routes. We propose here an extension of this approach in order to take into account the existence and the performance of alternative detours of the selected paths, in the calculus of the guidance strategy. This first extension assumes that the travel times through the links of the network are uncorrelated. We take into account the fact that one or many links of the selected optimal path may fail during the travel. We then consider that users may be sensitive to path changing. That is to say that they may prefer paths with efficient alternative detours, with respect to paths without, or with less efficient detours, even with a loss in the average travel time, and/or in its reliability. In order to take into account such behaviors, we propose a model that includes the existence as well as the performance of detours for selected paths, in the calculus of the travel time reliability (i.e. the probability of reaching a destination node). This new way of calculating travel time reliability guarantees a kind of robustness of the guidance strategies. That is to say that the travel time reliability associated to the obtained optimal guidance strategy is not likely to change, however associated adaptive paths change during the travel. The variation of the travel time reliability, with respect to a network structure changing, is thus improved. For

that, we propose to calculate for each node i the probability $u_i(t)$ to reach the destination node d , where we take into account the case where the selected path fails before the users who selected it reach the destination node d ; for which case, alternative neighboring paths are used. $u_i(t)$ denotes, as above, the probability to reach the destination node d , departing from node i , in a time-budget t . The mathematical definition of $u_i(t)$ is different from formulas (1)-(2). It is given below.

The probabilities $u_i(t)$

We introduce here a modification on the model (1)-(3) in order to take into account the existence and the performance of alternative paths in the calculated optimal routing strategies. In our model, the word *optimal* refers to the robustness of the strategy. The idea here is to replace the maximum operator in equation (1) by a weighted mean over a chosen number of successor nodes. Instead of calculating $u_i(t)$ basing on the successor node giving the maximum value of $u_i(t)$, we propose here to consider also other successor nodes of i , and we rather calculate $u_i(t)$ basing on a weighted mean over a number of successor nodes of i . Let us consider the following notation.

$$A_{ij}(t) = \int_0^t p_{ij}(w) u_j(t-w) dw, \forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), \forall 0 \leq t \leq T. \quad (12)$$

We denote by $A_i(t)$ the vector $A_i(t) = (A_{i1}(t), A_{i2}(t), \dots, A_{in_i}(t))$, where n_i is the number of successor nodes of node i in the graph. We then define n maps $S_i, i = 1, \dots, n$ as follows.

$$\begin{aligned} S_i : \mathbb{R}^{n_i} &\rightarrow \mathbb{R}^{n_i} \\ A_i(t) &\mapsto S_i(A_i(t)), \end{aligned} \quad , \forall i \in \{1, 2, \dots, n\}.$$

where $S_i(A_i(t))$ is the vector whose components are the same as those of $A_i(t)$, but sorted in a decreasing order. $S_{ij}(A_i(t))$ denotes here the j^{th} component $(S_i(A_i(t)))_j$ of vector $S_i(A_i(t))$.

We then rewrite the probability for a user to reach the destination node d from node i in a time budget t , as follows.

$$u_i(t) = \sum_{p=1}^m \psi_p S_{ip}(A_i(t)), \forall i \neq d, 0 \leq t \leq T, \quad (13)$$

$$u_d(t) = 1, 0 \leq t \leq T, \quad (14)$$

where m is a parameter giving the number of successor nodes taken into account in the sum of equation (13), and ψ_p are non increasing weighting coefficients satisfying

$$\psi_p \geq 0, \forall p \in \{1, 2, \dots, m\}, \sum_{p=1}^m \psi_p = 1, \text{ and } \psi_1 \geq \psi_2 \geq \dots \geq \psi_m.$$

The following two cases are distinguished.

1. $m \leq |\Gamma^{+1}(i)|$, in which case, the robust routing calculus (13) takes into account the best m successor nodes of i , without considering eventual other successor nodes.
2. $m > |\Gamma^{+1}(i)|$, in which case, passing through node i is penalized by the robust routing calculus (13) with respect to passing through nodes with higher number of successors (than $|\Gamma^{+1}(i)|$), up to m .

In case 2 above, nodes i with small numbers of successors are penalized; they get low values $u_i(t)$. Therefore, paths passing through these nodes i.e. paths with low numbers of alternatives or detours shall have low probabilities to be selected as optimal paths. One way to choose m can be to take the maximum over the cardinals of the sets $\Gamma^{+1}(i)$ of successors of all the nodes i of the network.

$$m = \max_{i \in N} |\Gamma^{+1}(i)|,$$

where $|\cdot|$ denotes the cardinal of a set. In the examples we give below, we simply take $m = 2$.

In order that formula (13) will have a meaning, ψ_p have to be chosen such that $\psi_1 \geq \psi_2 \geq \dots \geq \psi_m$. That is to say that ψ_p decrease as $S_{ip}(A_i(t))$ decrease with respect to p . This dependence of ψ_p on $A_{ip}(t)$ makes the model non-trivial. Indeed, instead of taking the maximum over $A_{ip}(t)$, with respect to successors p of i , as in formula (1), we take a weighted mean in formula (13), where the weights are in the same order as the one of the quantities $S_{ip}(A_i(t))$. Therefore, we need to, first, sort the quantities $A_{ip}(t)$, before applying the mean operator. So the model (13) needs more operations than the model (1). Finally, let us notice that if $m = 1$, or if $m > 1$ and $\psi_p = 0, \forall p \geq 2$, then the model (1)-(2) coincides with the model (13)-(14). Therefore, the model (13)-(14) extends the model (1)-(2).

The successor nodes

The optimal guidance strategy is determined by the sequence of successor nodes $s_i(t)$ as follows.

$$s_i(t) = \arg \max_{j \in \Gamma^{+1}(i)} (A_{ij}(t)), i \in N. \quad (15)$$

$s_i(t)$ denotes here the optimal successor node of node i for a user to reach the destination node d . By taking a mean in formula (13) rather than the maximum (as in formula (1)), we take into account the existence and the performance of alternative deviations at every intermediate node from i to d . We notice here that although formula (15) resembles to formula (3), the resulted successor nodes from the two formulas are not necessarily the same, since the maximized quantities in both formulas are calculated differently. We also notice that the model (13)-(14) is a dynamic programming-like model which propagates information on the routing, from the destination node to the origin one, as done by the SOTA model (1)-(2). The extension consists here in the fact that the existing SOTA model (1)-(2) propagates only information on the path travel time and its reliability, while the proposed model (13)-(14) propagates also information on the robustness of the routing, in addition to information on the path travel time and its reliability. The robust routing approach we propose in this article does not take loops as a matter of robust decision. The robustness of the calculated routing strategies is offline. However, users can apply the robust routing in an adaptive way, by launching a new robust routing calculus whenever they arrive to a new node, based on the information available at the arrival time to the node.

Algorithm 1 below gives a continuous time scheme for the model (13)-(15). In Algorithm 1, we fix the time unit $\delta = tt_{min} - \varepsilon$, where tt_{min} is the minimum realizable travel-time across the network, and ε is a short time. We then have $\forall i, j \in N, \forall w \in [0, \delta], p_{ij}(w) = 0$.

Algorithm 1 (Continuous time formulation of the robust guidance algorithm)

Step 0. Initialization

- Fix m (max. number of successor nodes); e.g. $m = 2$ or $m = \max_{i \in N} |\Gamma^{+1}(i)|$.
- Fix $\psi_p, p = 1, \dots, m$ such that $\sum_{p=1}^m \psi_p = 1$ and $\psi_1 \geq \psi_2 \geq \dots \geq \psi_m$.
- Fix the time unit δ .
- Fix the total time budget T as a multiple of δ : $T = L\delta$.

- Fix $iter = 0$ (iteration index).
- $u_i^{iter}(t) = 0, \forall i \in N \setminus \{d\}, 0 \leq t \leq T, \quad u_d^{iter}(t) = 1, 0 \leq t \leq T$

Step 1. Update For $iter = 1, 2, \dots, L = T/\delta$.

- $\tau^{iter} = iter * \delta$
- $u_i^{iter}(t) = u_i^{iter-1}(t), \forall i \in N, i \neq d, (i, j) \in A, t \in (0, \tau^{iter} - \delta]$.
- Calculate $A_{ip}^{iter-1}(t)$, for every $i, p, t \in (\tau^{iter} - \delta, \tau^{iter}]$, by (12) and using u_j^{iter-1} for u_j .
- For every $i, t \in (\tau^{iter} - \delta, \tau^{iter}]$ apply S_i to sort $A_i^{iter-1}(t)$ in a decreasing order.
- $u_i^{iter}(t) = \sum_{p=1}^m \psi_p S_{ip}(A_i^{iter-1}(t)), \quad \forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), t \in (\tau^{iter} - \delta, \tau^{iter}]$.
- $s_i^{iter}(t) = \arg \max_{j \in \Gamma^{+1}(i)} A_{ij}^{iter-1}(t), \quad \forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), t \in (\tau^{iter} - \delta, \tau^{iter}]$.

Definition 2.1. (*Robust-optimality*) We say that u is robust-optimal, with respect to parameters $m, \psi_p, p = 1, 2, \dots, m$ and time-budget T , if it is the unique possible routing strategy for the time budget T , or if u is robust-optimal with respect to parameters $m, \psi_p, p = 1, 2, \dots, m$ and time-budget $T - \delta$, and u satisfies

$$u_i(t) = \sum_{p=1}^m \psi_p S_{ip}(A_i(t)), \forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), t \in (t - \delta, T].$$

Robustness of a routing strategy has a price in term of travel time reliability and a price in term of travel time budget, as explained in section 2.4 below.

Proposition 2.1. For a total time budget $T = \delta L$, the solution obtained at iteration L of Step 1 of Algorithm 1 is robust-optimal.

Proof. By induction on L .

- For $L = 0$, the total time budget is zero, and thus only Step 0 is performed by the algorithm, which consequently terminates with a u satisfying $u_i(0) = 0, \forall i \in N \setminus \{d\}$ and $u_d(0) = 1$. This solution is robust-optimal since it is the unique possible solution for the time budget zero.
- Assume that for $T = \delta L$, the solution given at iteration L of Step 1 of Algorithm 1 is robust-optimal. Then with a time budget $T = (L + 1)\delta$, and for every time $t \in [0, \delta L]$, the algorithm keeps the same robust-optimal solution $u_i^{L+1}(t) = u_i^L(t), \forall i \in N$. For times $t \in (\delta L, \delta(L + 1)]$, $u_i^{L+1}(t)$ is robust-optimal by definition of the robust-optimality (Definition 2.1 above). \square

The meaning of Proposition 2.1 is that the continuous scheme of Algorithm 1 can be solved in a single update i.e. without resorting to value iteration. It is similar to the one introduced in [36]. The unique difference lies in the calculus of $u_i(t)$, using here a weighted mean instead of the maximum operator.

2.2. Discrete-time scheme of the robust guidance algorithm

To numerically approximate $u_i(t)$, we discretize the interval $[0, T]$ into $L = T/\Delta t$ time steps of length Δt . As in [36], the discretization length is supposed to satisfy $\Delta t \leq \delta$. The discretization is then given in Algorithm 2 below.

The probability distributions of the link travel times are discrete here. Therefore, the calculus of $A_{ij}(x)$ for $x \in \{0, 1, \dots, L\}$ is done as follows (replacing (12)).

$$A_{ij}(x) = \sum_{h=0}^x p_{ij}(tt_{ij} = h) u_j(x - h) \quad (16)$$

Complexity analysis

Proposition 2.2. *The complexity of Algorithm 2 is $O\left(a(T/\Delta t)^2 + (T/\Delta t)a \log a\right)$.*

Proof. The functions $A_{ij}(\cdot)$ and $p_{ij}(\cdot)$ are vectors of length L . Each link travel-time distribution in the network is of length $T/\Delta t$, and the discretized probability mass function is computed in time $O(T/\Delta t)$ for each link. As there are a links, then the total time is $O(aT/\Delta t)$. In step 0, there are $iter$ vectors to initialize for each node, and each vector is of length $T/\Delta t$. Then the initialization is done in time $O(iterT/\Delta t)$. In step 1, the algorithm progressively calculates:

- The sum of the convolution product $A_{ij}(x)$ from $x = 0$ to $x = T/\Delta t$. Then the time complexity of the summation for a link is $O\left(a(T/\Delta t)^2\right)$.
- The sum on the weighting coefficients from $p = 1$ to $p = m$. Then the time complexity of this summation is $O(aT/\Delta t)$.
- The quantities $A_{ij}(x)$ are sorted for each x . Then the complexity of this sorting is $(T/\Delta t)a \log a$.

Therefore, the total complexity to this algorithm is $O\left(a(T/\Delta t)^2 + (T/\Delta t)a \log a\right)$. \square

Algorithm 2 (Discrete time formulation of the robust guidance algorithm)

Step 0. Initialization

- Fix m (max. number of successor nodes); e.g. $m = 2$ or $m = \max_{i \in N} |\Gamma^{+1}(i)|$.
- Fix $\psi_p, p = 1, \dots, m$ such that $\sum_{p=1}^m \psi_p = 1$ and $\psi_1 \geq \psi_2 \geq \dots \geq \psi_m$.
- Fix the time unit $\Delta t \leq \delta$. We then have $\forall i, j \in N, \forall w \in [0, \Delta t], p_{ij}(w) = 0$.
- Fix the total time budget T as a multiple of Δt : $T = L\Delta t$.
- $u_i^{iter}(x) = 0, \forall i \in N \setminus \{d\}, x \in N, 0 \leq x \leq \frac{T}{\Delta t}, \quad u_d^{iter}(x) = 1, x \in N, 0 \leq x \leq \frac{T}{\Delta t}$.

Step 1. Update For $iter = 1, 2, \dots, L = T/\Delta t$.

- $\tau^{iter} = iter * \delta$
- $u_i^{iter}(x) = u_i^{iter-1}(x), \forall i \in N, i \neq d, (i, j) \in A, x \in \left(0, \frac{\tau^{iter}-\delta}{\Delta t}\right], x \in N$.
- Calculate $A_{ip}^{iter-1}(x)$ for every $i, p, x \in \left[\frac{\tau^{iter}-\delta}{\Delta t} + 1, \frac{\tau^{iter}}{\Delta t}\right], x \in \mathbb{N}$, by (16) and using u_j^{iter-1} for u_j .
- For every $i, x \in \left[\frac{\tau^{iter}-\delta}{\Delta t} + 1, \frac{\tau^{iter}}{\Delta t}\right], x \in N$ apply S_i to sort $A_i^{iter-1}(x)$ in a decreasing order.
- $u_i^{iter}(x) = \sum_{p=1}^m \psi_p S_{ip}(A_i^{iter-1}(x)),$
 $\forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), x \in \left[\frac{\tau^{iter}-\delta}{\Delta t} + 1, \frac{\tau^{iter}}{\Delta t}\right], x \in \mathbb{N}.$
- $s_i^{iter}(x) = \arg \max_{j \in \Gamma^{+1}(i)} (A_{ij}^{iter-1}(x)), \forall i \in N \setminus \{d\}, j \in \Gamma^{+1}(i), x \in \left[\frac{\tau^{iter}-\delta}{\Delta t} + 1, \frac{\tau^{iter}}{\Delta t}\right], x \in \mathbb{N}.$

2.3. An illustration example

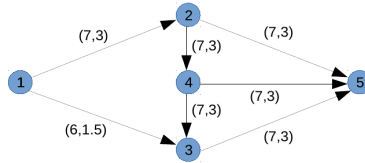


Figure 1: A small network for illustration. The numbers on the links give the average travel times through the links and their variances respectively.

Let us consider the network of Figure 1 including 5 nodes and 7 links. We illustrate here our approach of robust routing by applying Algorithm 2 to find a robust-optimal path from node 1 to node 5. We assume that the travel times on the links of the network are uncorrelated, and follow Gamma probability distributions. For all the links of the network, we assume an average travel time of 7 time units with a standard deviation of 3 time units, except for link

(1,3), for which we assume an average travel time of 6 time units and a standard deviation of 1.5 time units. To reach node 5 from node 1, four routes exist: route 1 (1-3-5), route 2 (1-2-4-5), route 3 (1-2-4-3-5) and route 4 (1-2-5). In term of minimum average travel time (i.e. applying the LET routing algorithm), it is easy to check that route 1 is the optimal one.

We apply Algorithm 2 (formulas (13)-(15)) for this example. We fix $m = 2$ (the maximum number of successor nodes in the network). We then have two weighting parameters ψ_1 and ψ_2 , with $\psi_1 + \psi_2 = 1$. To simplify, we denote $\psi = \psi_1$ and then $\psi_2 = 1 - \psi$. We vary ψ in $(1/2, 1]$. The results are given in Table 1 below.

Table 1: Robust-optimal solutions (u and s) to reach node 5 from node 1 for different value of ψ .

Time budget		15	16	17	18	19	20	21	22	23
$\psi = 1$	$s_1(t)$	3	3	3	3	3	3	3	3	3
	$u_1(t)$	0,9436	0,9742	0,9909	0,9968	0,9989	0,9990	0,9996	0,9999	1
$\psi = 0.9$	$s_1(t)$	3	2	2	2	2	2	2	2	2
	$u_1(t)$	0,8442	0,8772	0,9038	0,9283	0,9457	0,9587	0,9683	0,9751	0,9797
$\psi = 0.8$	$s_1(t)$	2	2	2	2	2	2	2	2	2
	$u_1(t)$	0,7472	0,7817	0,8208	0,8503	0,8744	0,8941	0,9096	0,9213	0,9300
$\psi = 0.7$	$s_1(t)$	2	2	2	2	2	2	2	2	2
	$u_1(t)$	0,6522	0,6921	0,7294	0,7600	0,7865	0,8092	0,8277	0,8423	0,8540

The results of Table 1 are obtained with the model (13)-(15). We notice that for $\psi = 1$, the model (13)-(15) is equivalent to the model (1)-(3). We can clearly see in Table 1 that, for any fixed time budget t , probabilities $u_1(t)$ are decreasing with ψ . For $\psi = 1$ the optimal route is the route passing through node 3 i.e. route 1, for all the considered time budgets. For $\psi < 1$, the robust-optimal route changes with the time budget. For example, for $\psi = 0.9$, the robust-optimal route is route 1 (the same as in the case $\psi = 1$) for time budgets less than 16 time units, while for time budgets higher than or equal to 16 time units, the robust-optimal policy changes and node 2 becomes the robust-optimal successor node of node 1. The robust-optimal route in this case is route 4. The selection of route 4 as the robust-optimal route instead of route 1 in case $\psi < 1$ is justified by the fact that, node 2 offers two routing options (go to node 5, or go to node 4). More options offer more alternatives in adaptive routing. Indeed, for a user who has chosen route 4, if link (2,5) fails once the user arrives to node 2, then he will still have two other alternative routes to reach the destination node 5: route 2 and route 3 by link (2,4). However, if the user chooses route 1, then if link (3,5) fails once the user arrives to node 3, then he will not have any other alternative route to reach the destination. Therefore, route 4 passing through node 2 is more robust than route 1 passing

through node 3, because node 2 has more successor nodes than node 3. The robust guidance algorithm favors routes passing by nodes with alternative detours. From these results we can conclude that:

- If a user prefers to maximize the travel time reliability of his routing strategy, without taking into account its robustness against link and path failures, then he should select route 1 because it is the one maximizing the probability of reaching the destination node 5 in the considered time budget.
- If a user seeks a guarantee in terms of robustness and if he has sufficient travel time budget and/or travel time reliability margins, then he should select route 4 because it is the one giving more alternative detours in case of link failure.

2.4. Price of robust-optimality

On the one hand, robustness of a routing strategy has a price in term of travel time reliability, in the sense that a user with a travel fixed time budget can improve the robustness of his routing strategy if he accepts to lose travel time reliability. On the other hand, robustness of a routing strategy has a price in term of travel time budget, in the sense that a user who requires a fixed level of travel time reliability can improve the robustness of his routing strategy if he accepts to extend his time budget. In order to illustrate this concept of price of robust-optimality, let us back to the example above. From Table 1, if a user likes to reach the destination node within a time budget of 16 time units, then he gets a travel time reliability of 0.9742 by passing by a non robust route (route 1 with $\psi = 1$), and a travel time reliability of 0.8772 by passing by a more robust route (route 4 with $\psi = 0.9$). Then, the value $0.097 = 0.9742 - 0.8772$ can be interpreted as the price of robustness (of passing from a robustness level corresponding to $\psi = 1$ to a robustness level corresponding to $\psi = 0.9$) to pay in term of travel time reliability; see Figure 2. On the other hand, if the user likes to reach the destination node with a travel time reliability of at least 0.9742, he can select a non robust route (route 1 with $\psi = 1$) with a time budget of 16 time units, or a more robust route (route 4 with $\psi = 0.9$) with a time budget of 22 time units (assuring a travel time reliability of 0.9751). Then, the value $6 = 22 - 16$ can be interpreted as the price of robustness (of passing from a robustness level corresponding to $\psi = 1$ to a robustness level corresponding to $\psi = 0.9$) to pay in term of travel time budge; see Figure 2.

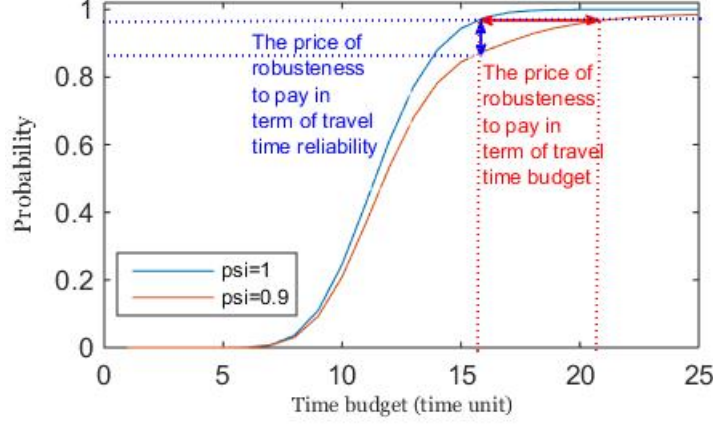


Figure 2: The price of robust-optimality.

3. Formulation of the algorithm in the case of correlated link travel-times

In this section we extend our algorithm to the case of correlated link travel-times. More precisely, we propose an extension of the model (13)-(15) that takes into account eventual correlation between any consecutive links in the network. For that, we propose to calculate for each node i the probability to reach the destination node d , where we take into account the case where the selected path fails before the users who selected it reach the destination node d ; for which case, alternative neighboring paths are used. Let us first notice that we follow here the approach of [36] recalled in section 1 (formulas (9)-(11)). Therefore, travel time correlations are considered on pairs of consecutive links of the road network. This is not the general case, but it is sufficient here since we treat in this article individual user guidance without calculating global traffic assignments in the network. $u_{ki}(t, y)$ denotes, as above, the probability to reach the destination node d , departing from node i , in a time-budget t , and knowing that the user comes from node k , a predecessor node of i , and that the realized travel time from k to i is y . The mathematical definition of $u_{ki}(t, y)$ is given below in (17) and (18). We proceed as in section 2. Let us consider the following notation.

$$A_{kij}(t, y) = \int_0^t p_{kij}(w|y) u_{ij}(t - w, y) dw,$$

$$\forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), j \in \Gamma^{+1}(i), \forall 0 \leq t \leq T, 0 \leq y \leq T - t,$$

where $p_{kij}(\cdot|\cdot)$ denotes the probability distribution function of the travel time on link (i, j) conditioned by the one on link (k, i) .

For given k, i, t and y , we sort the quantities $A_{kij}(t, y)$ in a decreasing order with respect to the index j as done above in section 2.1. We denote by $A_{ik}(t, y)$ the vector $A_{ki}(t, y) = (A_{ki1}(t, y), A_{ki2}(t, y), \dots, A_{kin_i}(t, y))$, where n_i is the number of successor nodes of node i in the network. We then define n maps $S_i, \forall i = 1, \dots, n$, as follows.

$$\begin{aligned} S_i : \mathbb{R}^{n_i} &\rightarrow \mathbb{R}^{n_i}, \forall i \in \{1, 2, \dots, n\}. \\ A_{ki}(t, y) &\mapsto S_i(A_{ki}(t, y)), \end{aligned}$$

where $S_i(A_{ki}(t, y))$ is the vector whose components are the same as those of $A_{ki}(t, y)$, but sorted in a decreasing order. $S_{ij}(A_{ki}(t, y))$ denotes here the j^{th} component $(S_i(A_{ki}(t, y)))_j$ of vector $S_i(A_{ki}(t, y))$.

We then rewrite the probability for a user to reach the destination node d from node i in a time budget t , knowing that the user comes from node k and that the travel time from k to i is y , as follows.

$$u_{ki}(t, y) = \sum_{p=1}^m \psi_p S_{ip}(A_{ki}(t, y)), \forall i \neq d, k \in \Gamma^{-1}(i), 0 \leq t \leq T, \quad (17)$$

$$u_{kd}(t, y) = 1, k \in \Gamma^{-1}(d), 0 \leq t \leq T, \quad (18)$$

where m is a parameter giving the number of successor nodes taken into account in the sum of formula (17), ψ_p are non increasing weighting coefficients satisfying

$$\psi_p \geq 0, \forall p \in \{1, 2, \dots, m\}, \sum_{p=1}^m \psi_p = 1, \text{ and } \psi_1 \geq \psi_2 \geq \dots \geq \psi_m.$$

The optimal guidance strategy is then determined by the sequence of successor nodes $s_{ki}(t, y)$ as given by the formula (19) below.

$$s_{ki}(t, y) = \arg \max_{j \in \Gamma^{+1}(i)} (A_{kij}(t, y)) \quad (19)$$

4. How to fix the parameters ψ_j

As mentioned above, in order that the model (17)-(19) has a meaning, ψ_p have to be chosen such that $\psi_1 \geq \psi_2 \geq \dots \geq \psi_m$. We will be interested here in the case where $m = 2$ (i.e. we only take into account the two best successor nodes of every node i). In this case, we have two weighting parameters ψ_1 and ψ_2 , such that $\psi_1 + \psi_2 = 1$. In order to simplify the notations, we simply denote $\psi = \psi_1$, and ψ_2 is given by $\psi_2 = 1 - \psi$. Therefore, we have only one parameter ψ for the robustness, such that the case $\psi = 1$ corresponds to the case where robustness is not taken into account; while the obtained routing strategy is as robust as the parameter ψ is low. We notice here that ψ should satisfy $1/2 \leq \psi \leq 1$, since we have $\psi_1 \geq \psi_2$.

We propose in this section a method to fix the parameter ψ in such a way that a user will be able to manage his travel time budget to favor travel time reliability with respect to robustness. More precisely, the user fixes a desired travel time reliability that he aims to reach, and by that, the routing algorithm will use all the available travel time budget in order to reach the desired level of travel time reliability; and the remaining travel time budget is used to improve the robustness of the routing strategy. The details are given below.

Given a travel time budget τ and a desired travel time reliability p (expressed as the probability that the destination will be reached on the time budget τ); given a time interval Ψ to which the parameter ψ belongs (for example $\Psi = (1/2, 1]$), the optimal weighting coefficient ψ^* is determined as follows (formulas (20)-(21) below).

The desired travel time reliability p being fixed, the algorithm first calculates the map $\psi \mapsto T_{kiy}(p, \psi)$ giving, for every value of ψ , the minimum time budget needed to satisfy the desired travel time reliability p .

$$T_{kiy}(p, \psi) = u_{ki}^{-1}(\underline{t}, y, \psi) := \min(t, t \geq 0, u_{ki}(t, y, \psi) \geq p) \quad (20)$$

The notation $u_{ki}^{-1}(\underline{t}, y, \psi)$ denotes the pseudo-inverse of the non-decreasing map $t \mapsto u_{ki}(t, y, \psi)$; see Figure 3 (left side).

Then, the desired travel time budget being fixed, the algorithm calculates the optimal robustness parameter ψ^* needed to satisfy the constraint on the travel time budget.

$$\psi_{kiy}^*(\tau, p) = T_{kiy}^{-1}(p, \underline{\psi}) := \min(\psi, \psi \in [1/2, 1], T_{kiy}(p, \psi) \leq \tau) \quad (21)$$

The notation $T_{kiy}^{-1}(p, \underline{\psi})$ denotes the pseudo-inverse of the non-increasing map $\psi \mapsto T_{kiy}(p, \psi)$; see Figure 3. (right-side).

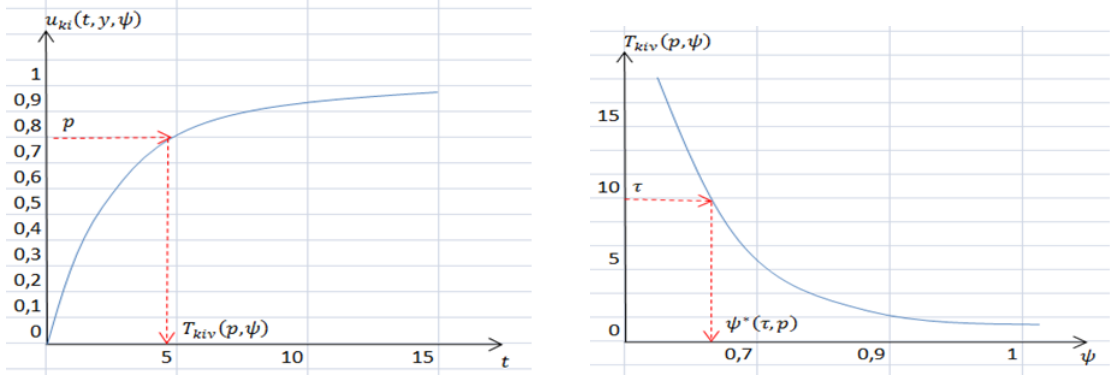


Figure 3: How to choose the robustness parameter ψ .

Therefore, the calculus of $\psi_{kiy}^*(\tau, p)$ consists in inverting the maximum cumulative probability distribution $u_{ki}(t, y, \psi)$ once on the variable t in order to derive, for every fixed value of robustness parameter ψ , the optimal time budget $T_{kiy}(p, \psi)$ needed to satisfy the desired travel time reliability p ; and a second time (inverting $T_{kiy}(p, \psi)$) on the variable ψ in order to calculate the optimal robustness parameter ψ^* needed to satisfy the constraint on the travel time budget.

The approach of fixing the parameter ψ , given by formulas (20)-(21), is general, in the sense that it includes the case where one only likes to optimize travel time reliability, and not robustness against path failure. In this case, one can just let ψ belonging to the singleton $\{1\}$, i.e. set $\psi = 1$. Moreover, if one tries to optimize robustness, but he does not have any margin on the time budget that permits this optimization, equation (21) will fix systematically ψ^* to the value $\psi^* = 1$. In other terms, any margin on the travel time budget is first used to optimize travel-time reliability, and after that, the remaining margin is used to optimize robustness.

The parameter optimization we propose in this section allows each user to optimize his travel time budget in order to feat his personal compromise between travel time reliability and robustness of the routing. With this parameterization, we achieve the theoretic development of our algorithm. We obtain thus a robust routing model in the form of a dynamic programming-like system, which we know that it admits a solution which can be derived by the Algorithms above (Algorithmes 1 and 2). In term of algorithmic complexity, it is clear that the case

where the travel time correlations are taken into account necessitates more computing time and more memory space than the case where travel time correlations are not taken into account. We finally notice that our algorithm retrieves the well-known min-max concept of robust optimization, since, as mentioned above, it is conceived in such a way that it *minimizes*, over link failure scenarios, the deterioration of the *maximum* probability of arrival.

5. Static routing in Sioux Falls network

We use here the well known Sioux Falls network to test the model (17)-(19). The network is simplified to 24 nodes and 76 links as illustrated in Figure 4 below. We assume that link travel times on the network are drawn from bi-variate Gamma distribution. More precisely, we assume that the joint probability distribution of two successive links of the network is a bi-variate Gamma. We base here on the bi-variate Gamma distribution from [32]. In the network of Figure 4, even though the maximum number of successors over all the nodes is equal to 5, we take here, for simplification, $m = 2$, with weighting coefficients ψ_1 and ψ_2 , and denote $\psi = \psi_1$ and then we have $\psi_2 = 1 - \psi$.

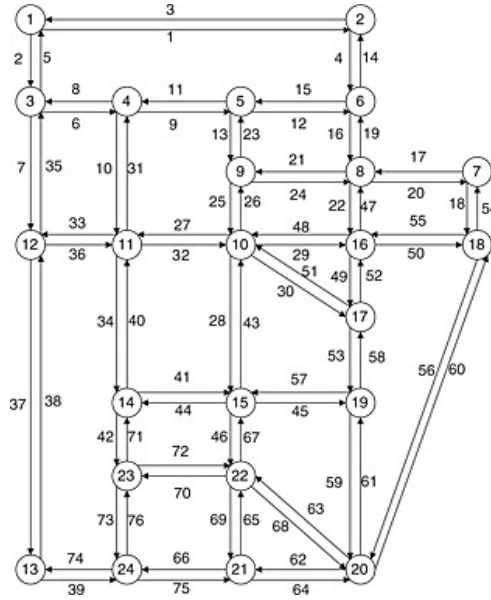


Figure 4: Sioux falls network. The nodes as well as the links are numbered in this figure.

To reach the destination node 10 departing from node 1, we have 2979 elementary paths. We apply the model (17)-(19) and derive the probabilities $u_{ki}(t, y)$ for all origin nodes i of the network. In order to illustrate our approach, let us consider the following parameters.

Travel-times on two successor links of the network follow a bi-variate Gamma probability distribution, with given average travel times tt_l for links l and variance-covariance matrix Cov , with $Cov_{ll} = 3, \forall l$ and $Cov_{ll'} = 1.5, \forall l \neq l'$. The average travel times we take here are shown in Figure 5.

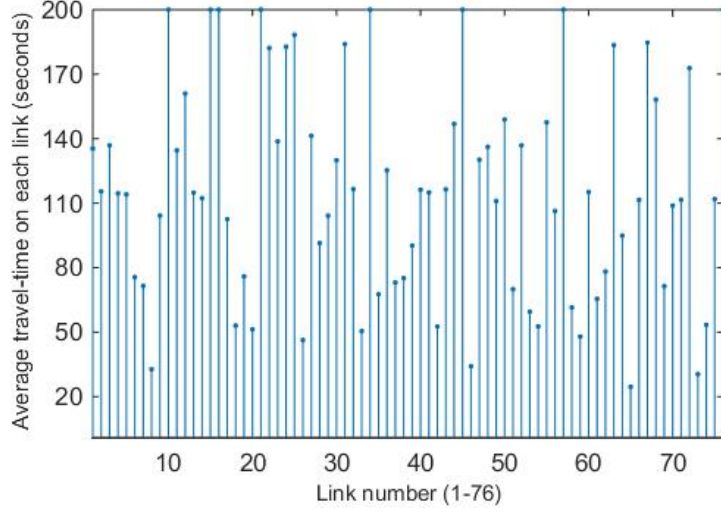


Figure 5: The average travel-time on each link (seconds).

As mentioned in section 2, our approach takes into account the existence and the performance of alternative detours of the selected paths, in the calculus of the guidance strategy. The proposed algorithm takes into account the fact that one or many links of the selected optimal path may fail during the travel. We consider that users may be sensitive to path changing. That is to say that they may prefer paths with efficient alternative detours, with respect to paths without, or with less efficient detours, even with a loss in the average travel time, and/or in its reliability. To illustrate our algorithm, we will be interested in what will happen at node 3 in two different scenarios presented in sub-sections 5.1 and 5.2.

In the following, we apply our algorithm for a user being at node 1 and going to node 10. We assume that the user has a time budget equal to 2400 seconds (40 minutes). In order to illustrate the algorithm, we consider two scenarios. In scenario 1, we use the network of Figure 4. We show that the algorithm gives the path passing by nodes (1,3,12,11,10) as the robust-optimal one in the two cases of $\psi = 1$ (without taking into account the robustness criterion) and $\psi \in (1/2, 1)$ (taking into account the robustness criterion). For scenario 2, we remove link 37 from the network of Figure 4. By removing link 37, node 12 will have only 2

successor nodes. Then, the user being at node 3 will have to chose between the two successor nodes: node 4 (with 3 successor nodes) and node 12 (with only 2 successor nodes). We will show that the algorithm gives the same optimal path as for scenario 1, in the case $\psi = 1$, but it gives another optimal path, the one passing by nodes (1,3,4,11,10), in the case $\psi \in (1/2, 1)$. Indeed, path (1,3,4,11,10) has more alternatives than path (1,3,12,11,10), since, in scenario 2, node 4 has more successor nodes than node 12.

5.1. Scenario 1

In this first scenario, we consider the network of Figure 4. Let us consider the probability $u_{13}(t, y)$ of reaching the destination node 10 from node 3, coming from node 1, and knowing that $tt_{13} = y$. We take here $y = 115$ seconds. In this first scenario, we have three routing actions from node 3 coming from node 1: go to successor node 12, which will give us three routing actions at the next step (go to successor node 11, or 13 or back to 3), or go to successor node 4, which will give us three routing actions at the next step (go to 5 or 11, or back to 3). The simulation results are given in Figure 6. The left side of Figure 6 shows the maximum probabilities $u_{13}(t, 115)$ in function of the time budget t , and for different values of ψ . The right side of Figure 6 gives the optimal successor nodes $s_{13}(t, 115)$ in function of the time budget t , and for different values of ψ .

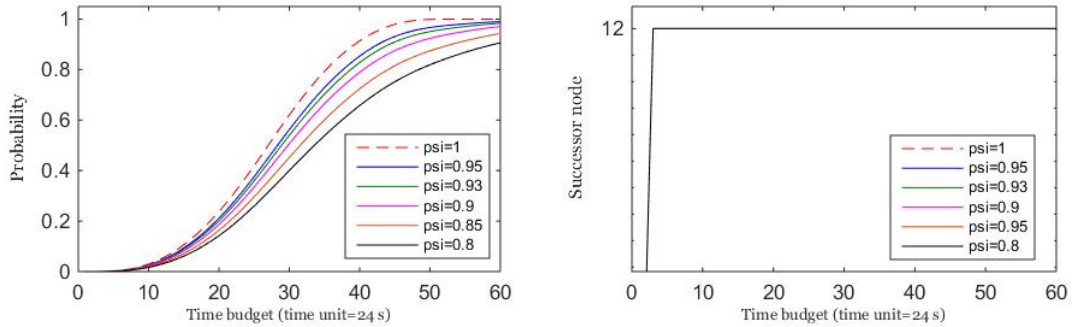


Figure 6: The robust-optimal probability of arriving on time (left side) and the robust-optimal routing policy at node 3, in function of the time budget, and for different values of ψ . - Scenario 1.

The case $\psi = 1$ (dashed line) corresponds to the model (9)-(11) taking into account travel time correlations, while the cases $\psi \in (1/2, 1)$ (solid lines) correspond to the model (17)-(19). We see from the left side of Figure 6 that, for any time budget t , the robust-optimal probabilities $u_{13}(t, 115)$ decrease as the values of ψ decreases. That is to say that for lower

values of ψ , lower values of $u_{13}(t, 115)$ are obtained. This is because we replaced a maximum operator in (9)-(11) by a mean value in (17)-(19). Indeed, a user taking a lower value of ψ , asks for more path-robustness or path-flexibility, and, in the counterpart, he loses in term of travel-time reliability. The difference $u_{13}(t, 115, \psi_1) - u_{13}(t, 115, \psi_2)$ can then be interpreted as the price of path-robustness corresponding to a measure of it, given by the difference $\psi_1 - \psi_2$. The right side of Figure 6 gives the robust-optimal routing policy in term of the robust-optimal successor nodes. We see from that figure that the optimal routing policy is the same for every value of ψ , and corresponds to a path passing by node 12 as the optimal successor node of node 3.

5.2. Scenario 2

This second scenario aims to illustrate our approach with respect to the robustness of the optimal strategy offered by our algorithm, in the case of path failure. The scenario is the following. We take the same network of Figure 4, but we remove link 37. By that, we penalize (in term of robustness) the passage by node 12, since we decrease the number of successor nodes of it. Therefore, node 12 will have only two, instead of three successor nodes. At node 3, coming from node 1, we have three routing actions: go to successor node 12, which will give us only two routing options at the next step (go to successor node 11 or back to 3), or go to successor node 4, which will give us three routing options at the next step (go to successor nodes 5 or 11, or back to 3). Indeed, if we chose node 12 as successor of node 3, then at node 12, if link 36 fails, we have to back to node 3. However, if we chose node 4 as successor of node 3, then at node 4, if one of the links 9 or 10 fails, we have the possibility to change and take a detour using the other link; see Figure 4. Therefore, passing by node 4 allows us more options than passing by node 12. We will see below that our algorithm is able to take into account such robustness criterion in the selection of the optimal routing strategy.

In scenario 2, the optimal policies obtained by the algorithm for the two cases of $\psi = 1$ and $\psi \in (1/2, 1)$ are clearly different. For the case $\psi = 1$, we obtain the same optimal policy as in scenario 1 (dashed line on the right side of Figure 7). However, with low values of ψ ($\psi = 0.95$, respectively $0.93, 0.9, 0.85, 0.8$), where path-robustness is considered, we see that, with a time budget greater than or equal to 57 time units (a time unit is 24 seconds here) (respectively 56, 55, 53 time units), the optimal successor node of node 3 is node 4 rather than node 12, even though paths passing through node 12 have lower average travel time comparing

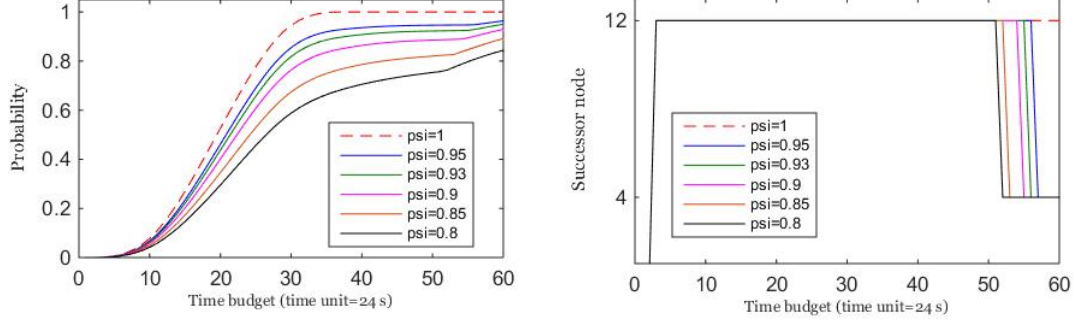


Figure 7: The robust-optimal probability of arriving on time (left side) and the robust-optimal routing policy at node 3, in function of the time budget, and for different values of ψ . - Scenario 2.

to those passing through node 4. For example, for $\psi = 0.95$, we see that with a time budget lower than 56 time units, the algorithm prefers paths passing through node 12 (see blue line on the right side of Figure 7). However, with a time budget higher than 56 time units, the optimal policy at node 3 changes, and node 4 becomes the robust-optimal successor node). That means that, node 12 which has only two successor nodes is penalized, i.e. it gets low values $u_{ki}(t, y)$. Therefore, paths that pass through that node i.e. paths with small number of alternatives or detours have low probability to be selected as robust-optimal paths.

6. Combination of the robust routing algorithm with a dynamic traffic model

In this section we combine the robust guidance algorithm with a dynamic traffic model in a closed loop, in order to see how the algorithm reacts dynamically to the state of traffic. We use here the microscopic traffic simulator SUMO (Simulation of Urban MObility) [8], with the traffic control interface Traci (Traffic control interface), where we implement our algorithm. We illustrate our results on the same network of Figure 4. We apply the guidance here only for the traffic demand departing from the zone around node 1 to the zone around node 10. The routing of all other origin-destination traffic demands is done by SUMO, by means of DUAROUTER algorithm. The application of our algorithm in parallel with another dynamic routing algorithm permits also to evaluate its efficiency in case where only some travelers are optimizing the robustness of the guidance against link failures.

The guidance is applied in its static form but periodically in time. The traffic is simulated by SUMO for a fixed period of time (10 minutes here). At the end of the time period, travel times on the links of the network are retrieved, and estimations are made on the average

travel times and on their variations. By that, we obtain an estimation of the bi-variate Gamma probability distribution of every couple of successive links in the network. With these travel time distributions, we apply the robust guidance algorithm for the traffic demand corresponding to the origin-destination pair (1,10). For the traffic demand corresponding to the other origin-destination pairs, the assignment is done by SUMO using DUAROUTER algorithm. The optimal and robust itineraries (obtained by our algorithm for the origin-destination pair (1,10), and by DUAROUTER for the other origin-destination pairs) are then used by SUMO to load the network and simulate the traffic for another time period. We iterate this procedure up to a given total simulation time.

6.1. Simulations results

In the simulation with SUMO we have used the network of Figure 4, and set node 1 as the origin, and node 10 as the destination. We considered a total simulation time of 40 minutes, with a time period of 10 minutes, at the end of which, we apply the guidance algorithm for the traffic demand from 1 to 10. The guidance algorithm is applied as follows. We assume that we are at node 1 and that we have spent a time y on the upstream link, and we seek to reach destination node 10. The maximum time budget we consider here is $T = 40$ minutes. The routes obtained at different times are given in Table 2, for $\psi = 1$ (without taking into account robustness criterion) and for $\psi = 0.9$ (with taking into account robustness criterion).

In AppendixA we give the 10th shortest paths from node 1 to node 10, for the four time periods of Table 2. The calculus is based on Yen's k-Shortest Path algorithm [39]. In AppendixB, we give the computation times as well as the occupied space memory for one run of robust routing calculus (at a given node and independent of the vehicular traffic simulation) in the case of correlated link travel times, and for varied number of links a , and varied number of time horizon units $T/\Delta t$. We notice that the implementation of the robust routing algorithm and of its combination with the vehicular traffic simulation are not optimized here. We intend to do the optimization of the implementations in our future works.

For $\psi = 1$, the optimal route obtained at 10 and 20 minutes is the one passing through the links [2-7-36-32]. At 30 minutes, the optimal strategy changes to [2-6-10-32]. We obtain the same route at 40 minutes. This optimal route is different from that obtained at 10 minutes. The optimal strategy obtained from node 3 changes, and node 4 becomes the optimal successor node, instead of node 12. Indeed, we observed that the link 7 is highly congested during the

Table 2: Optimal route for $\psi = 1$ and optimal robust route for $\psi = 0.9$.		
	$\psi = 1$	$\psi = 0.9$
Time periods (minutes)	Optimal route	Robust-optimal route
[0, 10]	$2 \rightarrow 7 \rightarrow 36 \rightarrow 32$	$2 \rightarrow 6 \rightarrow 10 \rightarrow 32$
[10, 20]	$2 \rightarrow 7 \rightarrow 36 \rightarrow 32$	$2 \rightarrow 6 \rightarrow 10 \rightarrow 32$
[20, 30]	$2 \rightarrow 6 \rightarrow 10 \rightarrow 32$	$2 \rightarrow 6 \rightarrow 10 \rightarrow 32$
[30, 40]	$2 \rightarrow 6 \rightarrow 10 \rightarrow 32$	$2 \rightarrow 6 \rightarrow 10 \rightarrow 32$

time periods [20,30] and [30,40] minutes. The guidance algorithm then proposes to rather pass by link 6, during these two periods of time.

For $\psi = 0.9$, where robustness is taken into account by the guidance algorithm, the robust-optimal route obtained during all the period times is [2-6-10-32]. We explain here why the algorithm proposes this robust-optimal route during the time periods [0,10] and [10,20] minutes, where the link 7 is not yet congested, and where the guidance algorithm with $\psi = 1$ proposes rather path [2-7-36-32]. From node 3 coming from node 1 we have three routing options: go to successor node 12, which will give us three routing actions at the next step (back to 3, go to 11, or 13), or go to successor node 4, which will give us also three routing actions at the next step (back to 3, go to 11, or 5), or back to node 1. It is clear that the option of backing to node 1 is not very interesting. Let us now compare the two options of going to node 4, and going to node 12. The two nodes 4 and 12 have two common successor nodes 3 and 11. The difference is in the third successor node (node 5 successor for node 4, and node 13 successor for node 12). Thus, we need to compare the two nodes 5 and 13. In terms of routing robustness, node 5 is better than node 13, since node 5 has three successor nodes, while node 13 has only two successor nodes. Therefore, it is better to pass by node 4 rather than passing by node 12. We recall here that the number of successor nodes is important for the robust guidance because, if we are at a given node with many successor nodes, we have more options in case of failures of any successor link. In this example, being at node 4 is better than being at node 12, because, if we are at node 4 and if link 10 fails, we need to choose between backing to node 3 or to go to node 5, which has three routing options (successor nodes), but if we are at node 12 and if link 36 fails, we have to choose between backing to node 3 (which option is also offered by node 4) or to go to node 13, which has only two routing options.

Let us now derive the price of robustness in term of travel time reliability and budget. We consider the first time period [0-10]. We follow section 2.4. Directly from Figure 8 and

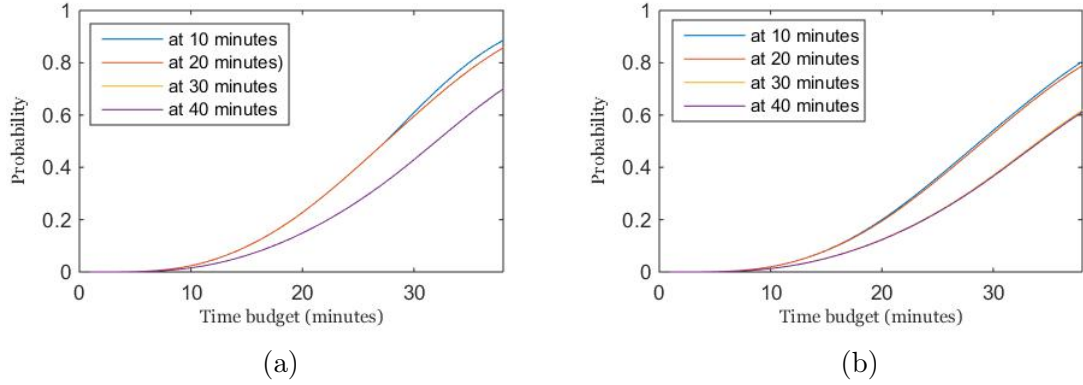


Figure 8: (a) The maximum probability of arriving on time at destination node 10 from node 1 coming from node 2, for $\psi = 1$. The optimal paths corresponding to the maximum cumulative probability distributions are given in Table 2. (b) The probability of arriving on time at destination node 10 from node 1 coming from node 2, for $\psi = 0.9$. The robust-optimal paths corresponding to the robust-optimal cumulative probability distributions are given in Table 2.

Table 2, we have the following.

- With a time budget of 40 minutes, a user can take the non robust path $2 \rightarrow 7 \rightarrow 36 \rightarrow 32$, with $\psi = 1$ and guarantees a travel time reliability of 0.9, or he can take the more robust path $2 \rightarrow 6 \rightarrow 10 \rightarrow 32$ with $\psi = 0.9$ and guarantees a travel time reliability of 0.8. Therefore, with a time budget of 40 minutes, the price of 0.1 point of robustness is 0.1 of travel time reliability.
- With a level of 0.8 of travel time reliability, a user can take the non robust path $2 \rightarrow 7 \rightarrow 36 \rightarrow 32$, with $\psi = 1$ and for that he needs 35 minutes as a time budget, or he can take the more robust path $2 \rightarrow 6 \rightarrow 10 \rightarrow 32$ with $\psi = 0.9$ and for that he needs 40 minutes as a time budget. Therefore, at a level of 0.8 of travel time reliability, the price of 0.1 point of robustness is 5 minutes of travel time budget.

Conclusion

We considered here the optimal guidance problem of users in road networks, for which we proposed a new robust adaptive strategy. We based on an existing SOTA routing model, which we extended in order to take into account robustness of routing strategies against path failure. In order to include the performance of alternative detours of the selected paths, we extended the concept of reliability by introducing a new reliability index. The improvement we made here allows the selection of an optimal path according to two criteria: the reliability

of the path in term of travel time and the robustness of the path in term of its flexibility (i.e. existence and performance of alternative detours). We have shown that the robustness of routing strategies has a price in term of time budget and a price in term of travel time reliability. In other terms, a user can increase the robustness of his routing strategy against path failures if he has a margin on his travel time budget or if he has a margin on his travel time reliability requirement. Finally, we illustrated our algorithm on the Sioux Falls network, with some scenarios in static and in dynamic traffic cases. Our future research in this direction is to optimize the implementation of our algorithm, in order to permit evaluations on large networks.

References

- [1] Borgatti Stephen P. *Centrality and network flow*. Social Networks, 27(1):55–71, 2005.
- [2] Borgatti Stephen P. and Everett Martin G. *A graph-theoretic perspective on centrality*. Social Networks, 28(4):466–484, 2006.
- [3] Chen A., Ji Z. *Path finding under uncertainty*. J. Adv. Transp. 39 (2005) 19–37.
- [4] Chen B.Y., Lam W.H., Sumalee A., Li Z. *Reliable shortest path finding in stochastic networks with spatial correlated link travel times*. Int. J. Geogr. Inf. Sci. 26 (2012).
- [5] Çolak Serdar, Lu Hilmi, Atlgan Ali Rana, *Vulnerability of Networks Against Critical Link Failures*. arXiv 1012.5961. 2010. url: <http://arxiv.org/abs/1012.5961>.
- [6] Dean B. C. *Speeding up stochastic dynamic programming with zero-delay convolution*. Algorithmic Operations Research, 5(2)(2010).
- [7] Dijkstra E. W. *A note on two problems in connection with graphs*. Numerical Mathematics. 1(1959) 269-271.
- [8] DLR, sumo.dlr.de
- [9] Fan Y., Kalaba R., Moore J. E. I. *Arriving on time*. J. Optim. Theory Appl. 127 (2005) 497–513 .doi:10.1007/s10957-005-7498-5.
- [10] Fan Y., Nie Y. *Arriving-on-time problem: discrete algorithm that ensures convergence*. Transp. Res. Record : J. Transp. Res. Board 1964 (2006) 193–200.

- [11] Fan Y., Nie Y. *Optimal routing for maximizing the travel time reliability*. Networks and Spatial Economics 6 (2006) 333–344. doi:10.1007/s11067-006-9287-6.
- [12] Farhi Nadir, Haj-Salem Habib, Lebacque Jean-Patrick. *On the robust guidance of users in road traffic networks*. AIP Conference Proceedings 1648 (1), 530004. 2015.
- [13] Frank H. *Shortest paths in probabilistic graphs*. Oper. Res. 17(1969).
- [14] French S., Insua D.R. *Statistical decision theory*. Kendall’s Library of Statistics, vol. 9. Wiley (2000).
- [15] Fu L. *An adaptive routing algorithm for in-vehicle route guidance systems with real-time information*. Transportation Research Part B, 2001.
- [16] Gao S., Chabini I. *Optimal routing policy problems in stochastic time-dependent networks*. Transp.Res. B Methodol. 40(2006) 93–122 . doi:10.1016/j.trb.2005.02.001.
- [17] Gao S. *Real-time traveler information for optimal adaptive routing in stochastic time-dependent net-works*. Transportation Research Part C, 21(2012).
- [18] Hall R.W. *The fastest path through a network with random time-dependent travel times*. Transplant. Sci. 20 (1986) 182–188 . doi:10.1287/trsc.20.3.182.
- [19] Huang H., Gao S. *Optimal paths in dynamic networks with dependent random link travel times*. Transp.Res. B Methodol. 46 (2012) 579–598 . doi:10.1016/j.trb.2012.01.005.
- [20] Karasan O. E., Pinar M. Ç., and Yaman H. *The robust shortest path problem with interval data*. Technical report , Bilkent University, 2001.
- [21] Kasperski A. and Zielinski P. *The robust shortest path problem in series-parallel multi digraphs with interval data*. Operations Research Letters 34 (2006) 69–76.
- [22] Kobitzsch M., Samaranayake S., Schieferdecker D. *Pruning techniques for the stochastic on-time arrival problem, An experimental study*. Computer sciences, data structures and algorithm, (2014).
- [23] LaValle S.M. *Planning algorithms*. Cambridge University Press (2006).

- [24] Lim S., Sommer C., Nikolova E., Rus D. *Practical route planning under delay uncertainty: Stochastic shortest path queries*. Proceed. of Robotics: Science and Systems, 2012.
- [25] Loui R. P. *Optimal paths in graphs with stochastic or multidimensional weights*. Commun. ACM 26 (1983) 670–676 . doi:10.1145/358172.358406.
- [26] Manseur Farida, Farhi Nadir, Haj-Salem Habib, Lebacque Jean-Patrick. *An algorithm for robust routing strategies in networks*. Journal of Traffic and Transportation Engineering, 2017.
- [27] Manseur Farida, Farhi Nadir, Haj-Salem Habib, Lebacque Jean-Patrick. *Robust adaptive strategies for the guidance of users in road networks*. Transportation Research Procedia, 2017.
- [28] Miller-Hooks E., Mahmassani H. S. *Least expected time paths in stochastic, time-varying transportation networks*. Transplant. Sci. 34(2000) 198–215.
- [29] Murthy I., Sarkar S. *Stochastic shortest path problems with piecewise-linear concave utility functions*. Manag. Sci. 44(1998) S125–S136. doi:10.1287/mnsc.44.11.S125
- [30] Neumann J., Morgenstern O. *Theory of games and economic behaviour*. Princeton University Press, Princeton (1947).
- [31] Nie Y.M., Wu X. Dillenburg J.F., Nelson P.C. *Reliable route guidance: A case study from Chicago*. Transportation Research part A, 46(2012).
- [32] Nie Y.M., Wu X. *Shortest path problem considering on-time arrival probability*. Transp. Res. B Methodol. 43(2009) 597–613. doi:10.1016/j.trb.2009.01.008.
- [33] Nikolova E. *Approximation algorithms for reliable stochastic combinatorial optimization*. In: Lecture Notes in Computer Science, 2010.
- [34] Pan Y., Sun L., Ge M. *Finding reliable shortest path in stochastic time-dependent network*. Procedia - Social and Behavioural Sciences, 96 (2013).
- [35] Sabran G., Samaranayake S., Bayen A. *Precomputation techniques for the stochastic on-time arrival problem*. In ALLENEX, SIAM(2014)138-146.

- [36] Samaranayake S., Blandin S., Bayen A. *A tractable class of algorithms for reliable routing in stochastic networks*. Transportation Research Part C, 2012.
- [37] Samaranayake S., Blandin S., Bayen A. *Speedup techniques for the stochastic on-time arrival problem*. In OASICS-OpenAccess Series in Informatics, volume 25. (2012).
- [38] Stephenson Karen and Zelen Marvin. *Rethinking centrality: Methods and examples*. Social Networks, 11(1):1 - 37, 1989.
- [39] Yen J. Y., *Finding the K shortest loopless paths in a network*, Management Science 17:712716, 1971.
- [40] Zockaie A., Nie Y., Mahmassani H. *Simulation-based method for finding minimum travel time budget paths in stochastic networks with correlated link times*. TRR 2014.

AppendixA. k - shortest paths for the OD 1-10 and four the four considered periods of time

We give here the 10 shortest paths from node 1 to node 10 four the time period [0-10].

10th Paths for time period [0-10]	Average travel time
2 - 7 - 36 - 32	515.47
2 - 6 - 10 -32	526.86
2 - 6 - 9 - 13 - 25	607.23
1 - 4 - 15 - 13 - 25	758.56
1 - 4 - 16 - 22 -48	767.52
2 - 7 - 37 - 39 - 75 - 64 - 61 - 58 - 51	805.47
1 - 4 - 16 - 20 - 18 - 55 - 48	827.33
1 - 4 - 16 - 22 - 49 - 51	836.40
2 - 7 - 36 - 34 - 41 - 43	842.91
2 - 7 - 37 - 39 - 75 - 65 - 67 - 43	847.56

We give here the 10 shortest paths from node 1 to node 10 four the time period [10-20].

10th Paths for time period [10-20]	Average travel time
2 - 6 - 10 - 32	552.97
2 - 7 - 36 -32	595.53
2 - 6 - 9 - 13 - 25	661.78
1 - 4 - 15 - 13 - 25	767.75
1 - 4 - 16 - 22 -49 - 51	835.75
1 - 4 - 16 - 21 - 25	850.58
1 - 4 - 16 - 22 - 48	850.58
2 - 7 - 37 - 39 - 75 - 64 - 61 - 58 - 51	852.02
1 - 4 - 15 - 11 - 10 - 32	860.58
1 - 4 - 16 - 20 - 18 - 56 - 61 - 58 - 51	899.44

We give here the 10 shortest paths from node 1 to node 10 four the time period [20-30].

10th Paths for time period [20-30]	Average travel time
2 - 6 - 10 - 32	633.31
2 - 7 - 36 -32	633.31
2 - 6 - 9 - 13 - 25	725.94
1 - 4 - 15 - 13 - 25	788.29
1 - 4 - 16 - 21 - 25	866.45
1 - 4 - 16 - 22 - 48	866.45
1 - 4 - 15 - 11 - 10 - 32	875.51
1 - 4 - 16 - 22 - 49 - 51	893.81
2 - 7 - 37 - 39 - 75 - 64 - 61 - 58 - 51	897.57
1 - 4 - 16 - 20 - 18 - 56 - 61 - 58 - 51	900.62

We give here the 10 shortest paths from node 1 to node 10 four the time period [30-40].

10th Paths for time period [30-40]	Average travel time
2 - 6 - 10 - 32	713.48
2 - 7 - 36 -32	713.48
1 - 4 - 15 - 13 - 25	816.96
2 - 6 - 9 - 13 - 25	848.41
1 - 4 - 16 - 21 - 25	896.17
1 - 4 - 16 - 22 - 48	896.17
1 - 4 - 16 - 20 - 18 - 56 - 61 - 58 - 51	925.76
1 - 4 - 15 - 11 - 10 - 32	949.77
1 - 4 - 16 - 22 - 49 - 51	964.44
2 - 7 - 37 - 39 - 75 - 64 - 61 - 58 - 51	992.25

AppendixB. Computation times and space memory

We give here the computation times and the occupied space memory for one run of robust routing calculus (at a given node and independent of the vehicular traffic simulation) in the case of correlated link travel times, and for varied number of links a , and varied number of time horizon units $T/\Delta t$.

Table B.3: Computation times in seconds (s.) and space memory in Megabytes (MB) for one run of robust routing calculus in the case of correlated link travel times, and for varied number of links a , and varied number of time horizon units $T/\Delta t$.

$a \backslash (T/\Delta t)$	35	40	45	50
20	363 s. 71.3 MB	480 s. 71.7 MB	611 s. 72.2 MB	743 s. 72.7 MB
30	876 s. 73.1 MB	1157 s. 73.9 MB	1424 s. 74.7 MB	1746 s. 75.6 MB
50	2455 s. 75.1 MB	3251 s. 76.7 MB	4182 s. 78.5 MB	5178 s. 80.4 MB
76	3367 s. 79.9 MB	4409 s. 82.7 MB	5646 s. 85.8 MB	6966 s. 89.2 MB

The implementation is done with Python 3 on an Intel (R) Xeron (R) CPU E5-2609 v3, 1.90 GHz, single thread. As mentioned above, the implementation of the robust routing algorithm and of its combination with the vehicular traffic simulation are not optimized here. We intend to do the optimization of the implementations in our future works.