



**HAL**  
open science

## An Algorithm for Robust Routing Strategies in Networks

Farida Manseur, Nadir Farhi, Habib Haj Salem, Jean Patrick Lebacque

► **To cite this version:**

Farida Manseur, Nadir Farhi, Habib Haj Salem, Jean Patrick Lebacque. An Algorithm for Robust Routing Strategies in Networks. *Journal of Traffic and Transportation Engineering*, 2017, 5, pp. 8-20. 10.17265/2328-2142/2017.01.002 . hal-01609970

**HAL Id: hal-01609970**

**<https://hal.science/hal-01609970v1>**

Submitted on 9 Oct 2017

**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.

# An Algorithm for Robust Routing Strategies in Networks

Farida Manseur, Nadir Farhi, Habib Haj-Salem and Jean Patrick-Lebacque

*University of Paris-Est, IFSTTAR/COSYS/GRETTIA, F-77447 Champs-sur Marne Cedex, France*

**Abstract:** In this article, the authors address the problem of optimal guidance of road network users. In the literature, several routing algorithms have been proposed under different approaches to solve this problem. Most adaptive algorithms for optimal path are based on the least expected travel time. Another approach has emerged named the SOTA (stochastic on time arrival). This approach based on the idea of Frank (1969), which aims to maximize the probability of arriving to a destination node parting from a given node in the network and with a given time budget. The authors' contribution consists here in extending this approach in order to introduce robustness towards path failure, in the guidance optimization. The authors 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. This new way of calculating travel time reliability guarantees a kind of robustness of the optimal guidance strategy.

**Key words:** Optimal guidance, travel-time reliability, robustness, optimization, traffic control.

## 1. Introduction

The most used way to plan a route in a road network is to compute the shortest path between a pair of origin/destination zones. Obtaining such optimal path in a given network is one of the most studied problems in network theory, with vast applications in many scientific and engineering fields, particularly in transportation engineering. Shortest path algorithms are widely available in the literature. In a stochastic framework, different performance criteria may lead to different optimal routing strategies. Most existing works define the optimality basing on LET (least expected time) as defined by Dijkstra [1]. Finding LET path when random link travel-times are independently distributed and does not vary over time is trivial. Fortunately, the LET problems have been studied broadly and have been extended to other cases.

A fundamental variant is the adaptive LET problem where travel-time on a link will become known and deterministic upon the arrival of its destination node [2-4]. Correlation between links' travel-times is investigated in Refs. [5, 6]. Another definition of

optimality in stochastic routing cited in the literature has to do with reliability, knowing that following LET path or policy can cause high risk to realize an extremely long travel-time. To resolve this problem, reliability-based stochastic routing has been widely studied. The authors of Refs. [7, 8] defined the optimal path as the one that maximize the probability of realizing a travel-time equal to or less than a predefined value. In Refs. [9, 10], the authors suggested as index of optimality, the probability that a path is shorter than all other paths. However, all these proposed formulations require enumerating paths and calculation multiple integrals, and thus they cannot be easily implemented in large-scale networks. As an alternative, the authors of Ref. [11] introduced the notion of a general utility function and proved that the problem of the maximum expected-utility can be reformulated as a class of bi-criteria shortest path problems (mean, variance) [9, 12]. Fan, Kalaba and Moore [5] defined optimality as maximum travel-time reliability and proposed an adaptive optimal path algorithm to solve this case. This is the SOTA (stochastic on-time arrival) problem. In this section, we briefly discuss the concept of the SOTA problem and evoke its two existing variants, for the convenience of readers and the continuity of our discussion in later sections. In the

---

**Corresponding author:** Farida Manseur, Ph.D. student, research fields: applied mathematics and transportation systems. E-mail: farida.manseur@ifsttar.fr.

SOTA problem, one seeks to maximize the probability of a time arrival at 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 identified for the SOTA problem. The first one consists on finding the most reliable fixed path to the destination. This first variant is designated as the shortest path problem on-time arrival reliability, or also the path-based SOTA problem as explored in Ref. [13]. The second one, which is referred as the policy-based SOTA problem, consists on calculating a routing policy such as, the selection of the next node, at each intersection, depends on the current state (remaining time budget). The policy-based SOTA problem, is solved in discrete-time. In Ref. [14], the authors presented a discrete SOTA algorithm that ensures finite convergence and runs very well in polynomial. 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 Ref. [15] for the policy-based SOTA problem. This algorithm is improved in Refs. [16, 17] to a dynamic programming algorithm and explored the speed-up techniques including zero-delay convolution [18], to solve the problem in pseudo-polynomial-time. Sabran, Samaranayake and Bayen [19] have shown how preprocessing 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 preprocessing methods that can be used for this problem, and prevents massive running time reductions in the deterministic case. Recently, Kobitzsch, Samaranayake and Schieferdecker [20] presented a novel approach to reduce the immense computational effort of stochastic routing based on existing techniques for alternative

routes.

In this paper, we focus on the reliable path problem that aims to find an adaptive optimal routing strategy, which takes into account the existence and the performance of alternative detours of the selected paths in the road network. That is to say, we assume that one or many links of the selected optimal path may fail during the travel, and that users may be sensitive to path changing. Some users may then prefer paths with efficient alternative detours, with respect to paths without or with less efficient detours. Our approach is based on the idea of Frank [8] considering that a reliable path from a given origin to a given destination maximizes the probability of realizing a travel time less than a given time budget. Thus, we propose an adaptation of this approach introducing robustness in the selection of the optimal path. We base here on the routing model of Samaranayake, Blandin and Bayen [16]. 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 through different possible routes. The model takes into account correlations between the travel times through the links of the network.

The remainder of this paper is organized as follows. In Section 2, the SOTA problem is summarized. The formulation of the problem and the description of our proposed algorithm for the robust guidance are provided in Section 3. Robust optimization and complexity analysis are given in Sections 4 and 5, respectively. Computational tests are conducted in Section 6. In Section 7, conclusions are drawn.

## 2. Summary of the SOTA Problem 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 Ref. [15].

We consider the graph  $G(N, A)$ , where  $N$  is the set of nodes, with  $|N| = n$ , and  $A$  is the set of arcs, with  $|A| = m$ . The set of successor and predecessor nodes are denoted by  $\Gamma^{+1}(i) = \{j | (i, j) \in A\}$  and  $\Gamma^{-1}(i) = \{k | (k, i) \in A\}$ , respectively. The SOTA problem is to find the best routing strategy from any starting node  $i$  ( $i = 1, 2, \dots, n$ ) maximizing the probability of arriving at the destination node, denoted  $d$ , within a desired time  $T$  (time budget).

Given a node  $i \in N$  and a time budget  $t$ ,  $u_i(t)$  denotes the maximum probability of arriving to the destination node  $d$  parting from origin node  $i$ , within a time budget  $t$ , under the optimal policy. The on-time arrival probability  $u_i(t)$  and the policy (optimal subsequent node), denoted by  $s_i(t)$  at node  $i$ , are written as follows:

$$u_i(t) = \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(w) u_j(t-w) dw \quad (1)$$

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

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

$$s_i(t) = \arg \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(w) u_j(t-w) dw \quad (3)$$

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

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

The meaning of Eq. (1) is that, knowing  $u_i(s), \forall s \in [0, t]$ , a traveler being at node  $i$ , and having a time budget  $t$ , should choose 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  $i$ . Eq. (2) tells simply that parting from node  $d$ , the maximum probability of arriving to the same node  $d$ , within any time budget is 1. Eq. (3) tells that the optimal successor node for the traveler being at node  $i$ , is given as the argument of the maximum taken in Eq. (1).

To solve the system of nonlinear Eqs. (1) and (2), the Picard method of successive approximation is one possible approach proposed in Ref. [5]. This fixed point method starts with initial approximations to the solution and refines these approximations by successive iterations. Then, the iterative relationships for successive approximations are given as follows:

$$u_i^{iter=0}(t) = 0 \quad (4)$$

$$\forall i \in \{1, 2, \dots, n\} \setminus \{d\}, \forall t \in [0, T]$$

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

$$u_i^{iter+1}(t) = \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(w) u_j^{iter}(t-w) dw \quad (6)$$

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

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

where, the superscript *iter* is the index iteration. 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.  $u_i^{iter}(t)$  belongs to  $[0, 1]$  since it is a probability, and is increasing with respect to the iteration index *iter*.

$$0 \leq u_i^{iter}(t) \leq u_i^{iter+1}(t) \leq 1, \forall iter \in N,$$

$$\forall i \in \{1, 2, \dots, n\}, \forall t \in [0, T]$$

The sequence  $\{u_i^{iter}(t)\}_{iter \geq 0}$  converges as *iter* increases and the solution reaches the optimal value when *iter* is equal to the number of links in the optimal path. A convergence proof is given in Eq. [15].

Eqs. (1) and (2) present the simple case of SOTA problem where the travel-times on the links of the network are uncorrelated. Different variants of the SOTA problem with correlation are proposed in the literature, e.g., Refs. [5, 16]. We will expose the model with correlation proposed in Ref. [16] for the continuation of our work in the coming sections.

Samaranayake, Blandin and Bayen [16] presented a simple extension of the SOTA model that considered correlation between the link and the upstream neighbors via which the link is reached. They proposed the following formulation:

$$u_{ki}(t, y) = \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(tt_{ij} = w | tt_{ki} = y) u_{ij}(t-w, w) dw \quad (8)$$

$$\begin{aligned} \forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), j \in \Gamma^{+1}(i), 0 \leq t \leq T, 0 \leq y \\ \leq T - t \\ u_{kd}(t, y) = 1, \forall k \in \Gamma^{-1}(d) \quad (9) \\ 0 \leq t \leq T, 0 \leq y \leq T - t \end{aligned}$$

$$s_{ki}(t, y) = \arg \max_{j \in \Gamma^{+1}(i)} \int_0^t p_{ij}(tt_{ij} = w \mid tt_{ki} = y) u_{ij}(t - w, w) dw \quad (10)$$

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

where,  $tt_{ij}$  denotes the travel time on link  $(i, j)$ , and  $p_{ij}(tt_{ij} \mid tt_{ki})$  is the probability distribution function (pdf) of  $tt_{ij}$ , conditioned on  $tt_{ki}$ . The pdf  $p_{ij}(\cdot)$  is assumed to be known and can for example be obtained 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  $t - w$ , parting from node  $j$ , conditioned that the travel time on link  $(i, j)$  is  $w$ .

In our previous articles [21, 22], we focused on the routing model presented in Ref. [16] (Eqs. (8)-(10)) and we followed the same idea in term of reliability of the travel-time of the selected path, but by introducing a robustness guarantee in the selected path. The work done in this paper is summarized in the next section [21].

### 3. Robust Guidance

In this section, we base on the routing model presented in Ref. [16] (Eqs. (8)-(10)) 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 using different possible routes. The model takes into account correlations between the travel times through the links.

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. 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, 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 maximum 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, 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 maximum probability  $u_{ki}(t, y)$  to reach the destination node, where we take into account the case where the selected path fails before the users who selected it reach the destination node; for which case, alternative neighboring paths are used.  $u_{ki}(t, y)$  denotes, as above, the maximum probability to reach the destination node, departing from node  $i$ , in a time-budget  $t$ , and knowing that the user comes from node  $k$  upstream of  $i$ , and that the realized travel time from  $k$  to  $i$  is  $y$ . However, the mathematical definition of  $u_{ki}(t, y)$  is different from Eqs. (8)-(10), and is given below.

#### 3.1 Calculation of the Probabilities $u_{ki}(t, y)$

We introduce here a modification on Eqs. (1)-(3) in order to take into account the existence and the performance of alternative paths in the calculated optimal routing strategy. The idea here is to replace the maximum operator in Eq. (1) by a weighted mean over a chosen number of successor nodes. Instead of calculating  $u_{ki}(t, y)$  basing on the successor node giving the optimum value of  $u_{ki}(t, y)$ , we propose here to consider also other successor nodes of  $i$ , and

we rather calculate  $u_{ki}(t, y)$  basing on a weighted mean over a number of successor nodes of  $i$ . Let us consider the following notation:

$$\begin{aligned} A_{kij}(t, y) &= \int_0^t p_{kij}(tt_{ij} = w \mid tt_{ki} \\ &= y)u_{ij}(t - w, y)dw \\ \forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), j \in \Gamma^{+1}(i) \\ \forall 0 \leq t \leq T, 0 \leq y \leq T - t \end{aligned}$$

We then sort, for a given  $(k, i)$  pair, the quantities  $A_{kij}(t, y)$  on  $j$ , in a decreasing order, and denote them  $B_{kip}(t, y)$  in that order, i.e.,

$$B_{ki1}(t, y) \geq B_{ki2}(t, y) \geq \dots \geq B_{ki\lambda}(t, y)$$

We then have

$$\begin{aligned} A_{ki}(t, y) &:= \{A_{kij}(t, y), \forall j \in \Gamma^{+1}(i)\} = \{B_{kip}(t, y) \\ &\in A_{ki}(t, y), B_{ki1}(t, y) \geq B_{ki2}(t, y) \\ &\geq \dots \geq B_{ki\lambda}(t, y)\} \end{aligned}$$

We then rewrite the maximum 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:

$$u_{ki}(t, y) = \sum_{p=1}^m \Psi_p B_{kip}(t, y) \quad (11)$$

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

$$u_{kd}(t, y) = 1 \quad (12)$$

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

where,  $m$  is a parameter giving the number of successor nodes taken into account in the sum of Eq. (11),  $\Psi_p$  are weighting coefficients satisfying

$$\sum_{p=1}^{\lambda} \Psi_p = 1$$

and where, in case where  $m > |\Gamma^{+1}(i)|$ ,  $B_{kip}(t, y) = 0$  for  $p > |\Gamma^{+1}(i)|$ . In this case,  $m$  is the number of successors to be taken into account in the network, independent of  $i$ . Indeed, one can distinguish the following two cases :

(1) Case 1:  $m \leq |\Gamma^{+1}(i)|$ , in which case, no more than the number of successors of  $i$  are considered in the sum of Eq. (11);

(2) Case 2:  $m > |\Gamma^{+1}(i)|$ , in which case, we have  $\sum_{j \in \Gamma^{+1}(i)} \Psi_j < \sum_{p=1}^m \Psi_p = 1$ .

In Case 2 above, nodes  $i$  that have a small number of successors are penalized; they get low values

$u_{ki}(t, y)$ . Therefore, paths passing through these nodes, i.e., paths with small number of alternatives or detours shall have low probabilities to be selected as optimal paths. One way to choose  $m$  would be to take the maximum over the cardinals of the sets  $\Gamma^{+1}(i)$  of successors of all the nodes of the network:

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

where,  $|\cdot|$  denotes the cardinal of a set.

In order that Eq. (11) will have a meaning,  $\Psi_p$  have to be chosen such that  $\Psi_1 \geq \Psi_2 \geq \dots \geq \Psi_m$ . That is to say,  $\Psi_p$  decrease as  $B_{kip}(t, y)$  decrease with respect to  $p$ . This dependence of  $\Psi_p$  on  $A_{kip}(t, y)$  makes the model nontrivial. Indeed, instead of taken the maximum over  $A_{kip}(t, y)$ , with respect to successors  $p$  of  $i$ , as in Eq. (1), we take a weighted mean in Eq. (11), where the weights are in the same order as the one of the quantities  $A_{kip}(t, y)$ . Therefore, we need to first sort the quantities  $A_{kip}(t, y)$ , before applying the mean operator. So the model Eq. (11) needs more operations than the model Eq. (1). Finally, let us notice that if  $m = 1$ , or if  $m > 1$  and  $\Psi_p = 0, \forall p \geq 2$ , then Eqs. (1) and (2) coincide with Eqs. (11) and (12). Therefore, the model Eqs. (11) and (12) are a kind of an extension of the model Eqs. (1) and (2).

### 3.2 Calculation of Successor Nodes

In the calculus of  $u_{ki}(t, y)$  by Eq. (11), instead of maximizing the quantities  $A_{kij}(t, y)$ , we propose to take a weighted mean of these quantities, with weights  $\Psi_p, p = 1, 2, \dots, m$ . The optimal guidance strategy is then determined by the sequence of successor nodes  $s_{ki}(t, y)$  as shown in the following equation:

$$s_{ki}(t, y) = \arg \max_{j \in \Gamma^{+1}(i)} (A_{kij}(t, y)) \quad (13)$$

$s_{ki}(t, y)$  denotes here the optimal successor node of node  $i$  for a user to reach the destination node, knowing that the user comes from node  $k$  upstream of  $i$ , and that the realized travel time on link  $(k, i)$  is  $y$ . By taking a mean in Eq. (11) rather than the maximum (as in Eq. (1)), we do not only take into account the path maximizing the probabilities  $u_{ki}(t, y)$ , but we

also take into account the existence and the performance of alternative deviations at each node. We notice here that although Eq. (13) resembles to Eq. (3), the resulted successor nodes from the two formulas are not necessarily the same, since the quantities  $u_i(t)$  in Eq. (3) and  $u_{ki}(t, y)$  in Eq. (13) are calculated differently.

#### 4. How to Fix the Parameter $\psi$

As mentioned above, in order that the model Eqs. (11) and (12) has a meaning,  $\psi_p$  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  $\psi_1$  and  $\psi_2$ , such that  $\psi_1 + \psi_2 = 1$ . In order to simplify the notations, we simply denote  $\psi = \psi_1$ , and  $\psi_2$  is given by  $\psi_2 = 1 - \psi$ . Therefore, we have only one parameter  $\psi$  for the robustness, such that the case  $\psi = 1$  corresponds to the case where robustness is not taken into account; and that the obtained routing strategy is as robust as the parameter  $\psi$  is small. We notice here that  $\psi$  should satisfy  $1/2 \leq \psi \leq 1$ , since we have  $\psi_1 \geq \psi_2$ .

Given a travel time budget  $\tau$  and a desired travel time reliability  $p$  (expressed as the probability that the destination will be reached on the time budget  $\tau$ );

given an interval  $\Psi$  to which the parameter  $\psi$  belongs (for example  $\Psi = (1/2, 1]$ ), optimal weighting coefficient  $\psi^*$  is determined as given by Eqs. (14) and (15) below.

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

$$\begin{aligned} T_{kiy}(p, \psi) &= u_{ki}^{-1}(\underline{t}, y, \psi) \\ &= \min(t, t \geq 0, u_{ki}(t, y, \psi) \geq p) \end{aligned} \quad (14)$$

The notation  $u_{ki}^{-1}(\underline{t}, y, \psi)$  means that we take the pseudo-inverse of the non-decreasing map  $t \mapsto u_{ki}(t, y, \psi)$  (Fig. 1).

Then, the desired travel time budget being fixed, we calculate the optimal robustness parameter  $\psi^*$  needed to satisfy the constraint of travel time budget:

$$\begin{aligned} \psi^*(\tau, p) &= T_{kiy}^{-1}(p, \underline{\psi}) \\ &= \min(\psi, \psi \in [1/2, 1], T_{kiy}(p, \psi) \leq \tau) \end{aligned} \quad (15)$$

The notation  $T_{kiy}^{-1}(p, \underline{\psi})$  means that we take the pseudo-inverse of the non-increasing map  $\psi \mapsto T_{kiy}(p, \psi)$  (Fig. 2).

Therefore, the calculus of  $\psi^*(\tau, p)$  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  $\psi$ , the optimal time budget  $T_{kiy}(p, \psi)$  needed to satisfy the desired travel time reliability  $p$ ;

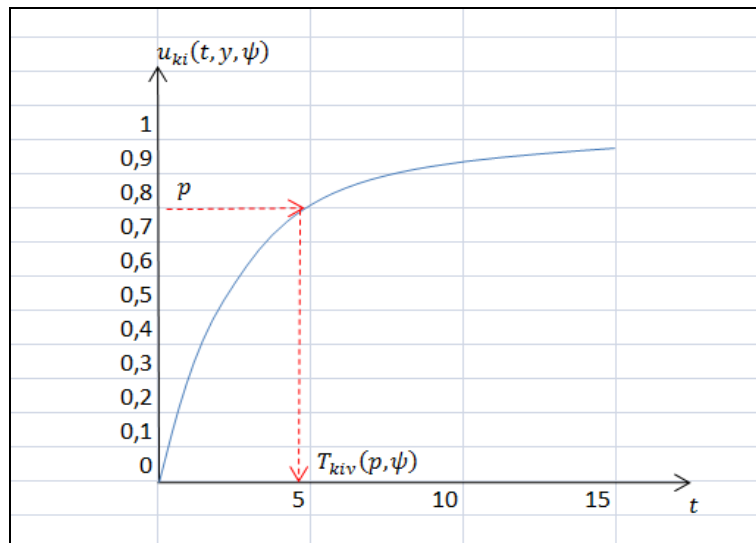


Fig. 1 Calculation of the minimum time budget  $T_{kiy}(p, \psi)$ .

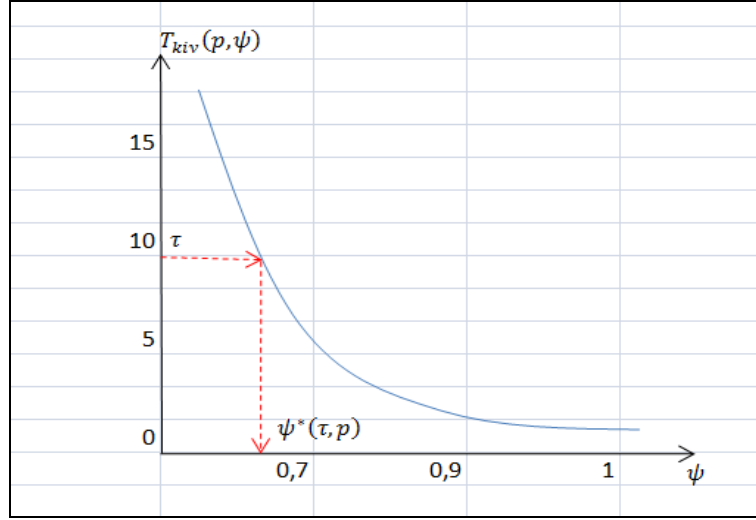


Fig. 2 Calculation of the optimal robustness parameter  $\psi^*(\tau, p)$ .

and a second time (inversing  $T_{kiv}(p, \psi)$ ) on the variable  $\psi$  in order to calculate the optimal robustness parameter  $\psi^*$  needed to satisfy the constraint on the travel time budget.

The optimization of robustness given by Eqs. (14) and (15) is general, in the sense that it includes the case where one only likes to optimize travel time reliability, and not path-failure robustness. In this case, one can just let  $\psi$  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, Eq. (15) will fix systematically  $\psi^*$  to the value  $\psi^* = 1$ . In the 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.

## 5. Analysis of the Complexity of the Algorithm

To study the complexity of our algorithm, as given in Table 1, the discrete form of our algorithm. We consider the time  $t$  in its discrete form such as  $t = 0, \delta, 2\delta, 3\delta, \dots, L\delta$ , where,  $\delta$  is the time unit. We assume that the time budget  $T$  is a multiple of  $\delta$  and write  $T = L\delta$ . If the distribution is continuous, the probability of taking time between  $h$  and  $h + \delta$  to cross the node  $i$  to node  $j$  can be written as follow:

$$p_{ij}(h) = \int_h^{h+\delta} p_{ij}(h)dh$$

If the distribution is discrete, the convolution integral can be replaced by a finite sum. Then the discrete form of our algorithm is summarized in Table 1.

**Proposition 1.** The complexity of Algorithm 1 is  $O(m(T/\delta)^2 + m \log m)$ .

**Proof.** The functions  $A_{kij}(\cdot)$  and  $p_{kij}(\cdot)$  are vectors of length  $L$ . Each link travel-time distribution in the network is of length  $T/\delta$ , and the discretized probability mass function is computed in time  $O(T/\delta)$  for each link. As there are  $m$  links, then the total time is  $O(mT/\delta)$ . In Step 0, there are  $iter$  vectors to initialize, and each vector is of length  $T/\delta$ . Then the initialization is done in time  $O(iter T/\delta)$ . In Step 1, the algorithm calculates:

- the sum of the convolution product from  $h = 0$  to  $h = t$ , or more precisely, from  $t = 1$  to  $t = T/\delta$ . Then the time complexity of the summation for  $m$  link is  $O(m(T/\delta)^2)$ ;
- the sum on the weighting coefficients from  $p = 1$  to  $p = m$ . Then the time complexity of this summation is  $O(mT/\delta)$ ;
- the quantities  $B_{kip}^{iter}$  are sorted in a decreasing order, then the complexity of this sorting is  $m \log m$ .

Therefore, the total complexity to this algorithm is  $O(m(T/\delta)^2 + m \log m)$ .



**Table 1 Algorithm 1.**

<p><b>Step 0. Initialization</b>  <math>m = \max_{i \in N}  \Gamma^{+1}(i) </math> (maximum number of successor nodes to take into account)  <math>iter = 0</math> (iteration index)  <math>u_{ki}^{iter}(t, y) = 0, \forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), 0 \leq t \leq L\delta, 0 \leq y \leq T - t</math>  <math>u_{kd}^{iter}(t, y) = 1, \forall k \in \Gamma^{-1}(d), 0 \leq t \leq L\delta, 0 \leq y \leq T - t</math></p> <p><b>Step 1. Update</b>                  For <math>iter = 1, 2, \dots, L</math>, with <math>L = T/\delta</math>,  <math>\tau^{iter} = iter * \delta</math>  <math>u_{kd}^{iter}(t, y) = 1, \forall k \in \Gamma^{-1}(d), 0 \leq t \leq L\delta, 0 \leq y \leq T - t</math>  <math>u_{ki}^{iter}(t, y) = u_{ki}^{iter-1}(t, y), \forall i \in N, i \neq d, (i, j) \in A, t \in [0, \tau^k - \delta]</math>  <math>u_{ki}^{iter}(t, y) = \sum_{p=1}^m \psi_p B_{kip}^{iter}(t, y)</math> where <math>B_{kip}^{iter}(t, y)</math> is as in Section 3.1, see Ref. (11)  <math>\forall i \in N \setminus \{d\}, k \in \Gamma^{-1}(i), j \in \Gamma^{+1}(i), t \in [\tau^k - \delta + 1, \tau^k], 0 \leq y \leq T - t</math></p>
---

## 5. Experimental Results

we use here the Sioux Falls network to test our algorithm (Fig. 3).

This network is simplified to Sioux Falls test networks that contains only 24 zones, 24 nodes and 76 links (Fig. 4).

We assume that link travel times on the network of Fig. 1 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 Ref. [23]. In the network of Fig. 4, the maximum number of successors over all the nodes is equal to 5. Therefore, according to Section 3, we take here  $m = 5$ . Then, we have five weighting coefficients  $\psi_1, \psi_2, \psi_3, \psi_4$ , and  $\psi_5$ . For simplification, we assume here that  $\psi_3 = \psi_4 = \psi_5 = 0$ , and denote  $\psi = \psi_1$  and then we have  $\psi_2 = 1 - \psi$ . To reach the destination node 24 parting from node 1, we have 3,856 elementary paths. We apply Algorithm 1 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:

Travel-times on two successor links of the network follow a bi-variate Gamma probability distribution, with given average travel time vector and variance-covariance matrix. We take different values of the average travel time for all the links, so that, the path consisting of links (2-7-37-39) have large values compared to other paths on the network. For every pair

of two successive links, we take the same covariance matrix (variance = 3, covariance = 1.5). We assume that we are at Node 1 and that we have spent a time  $y = 3$  time units on the upstream link, and we seek to reach destination Node 24. The maximum time budget we consider here is 60 time units.

We distinguish two scenarios. In the first scenario, we keep the number of links for each node, as shown in the network of Fig. 4. In the second scenario, we penalize (in term of robustness) the passage by Node 2 by removing link 3. Therefore, Node 2 will have only one successor node. The simulations results are given by the next figures.

Figs. 5 and 6 represent the simulation results given by the first scenario.

Figs. 7 and 8 represent the simulation results given by the second scenario.

Figs. 5 and 7 show the maximum probabilities  $u_{1,3}(t, 3)$  in function of time budget  $t$ , and for different values of  $\Psi$ . Figs. 6 and 8 give the optimal successor nodes  $s_{1,3}(t, 3)$  in function of time budget  $t$ , and for different values of  $\Psi$ .

The case  $\Psi = 1$  (dashed line) corresponds to the model Eqs. (1) and (2), while the cases  $\psi < 1$  (solid line) correspond to the model Eqs. (11) and (12). We see from Fig. 3 that the maximum probability  $u_{1,3}(t, 3)$  decreases as the values of  $\psi$  decrease. That is to say, for lower values of  $\psi$ , lower maximum probabilities  $u_{1,3}(t, 3)$  are obtained. This is because we replaced a maximum operator in Eqs. (1) and (2) by a mean value in Eqs. (11) and (12). Indeed, a user taking



Fig. 3 Sioux Falls network.

Source: image from OpenStreetMap.

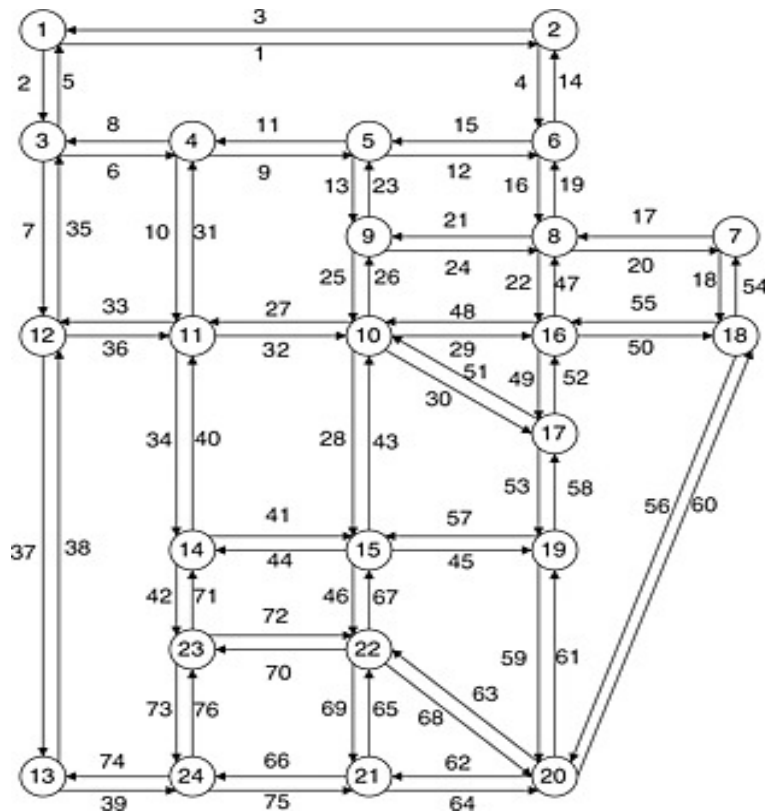


Fig. 4 Sioux Falls test network.

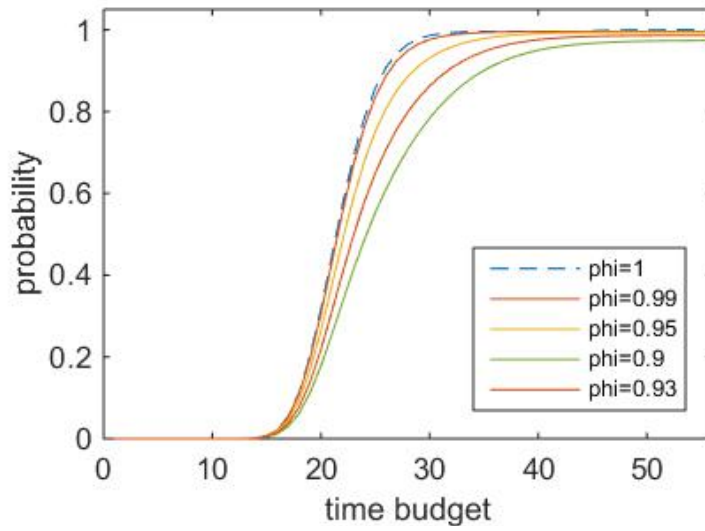


Fig. 5 The probability of arriving on time in function of the time budget, and for different values of  $\Psi$ .

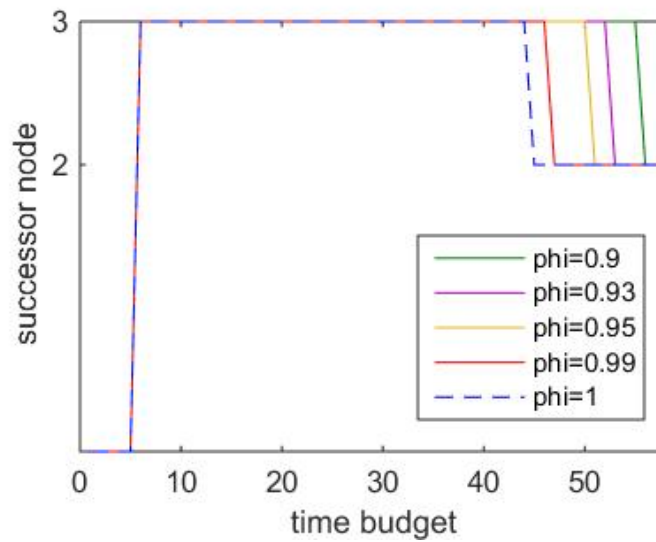


Fig. 6 The routing policy on Node 1 in function of the time budget, and for different values of  $\Psi$ .

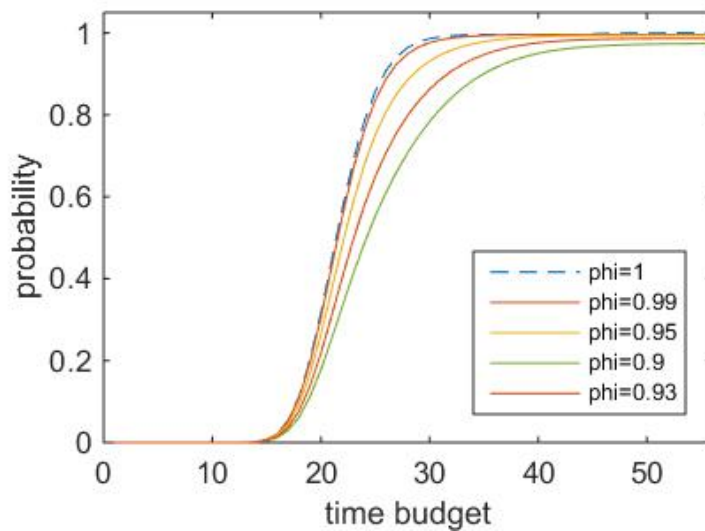


Fig. 7 The probability of arriving on time in function of the time budget, and for different values of  $\Psi$ .

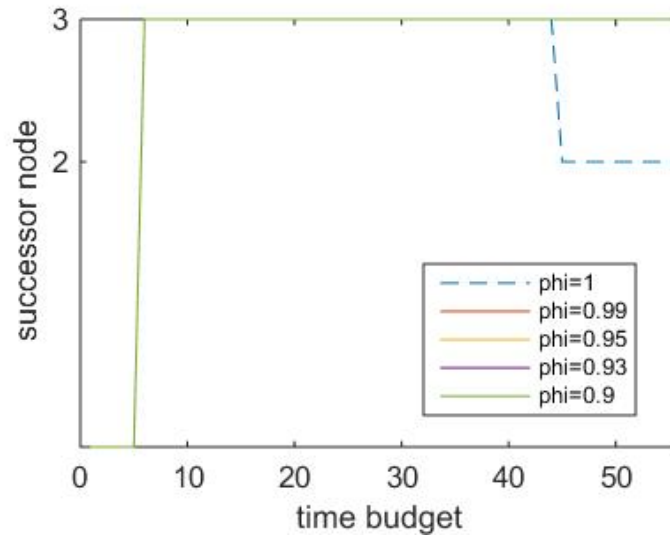


Fig. 8 The routing policy on node 1 in function of the time budget, and for different values of  $\Psi$ .

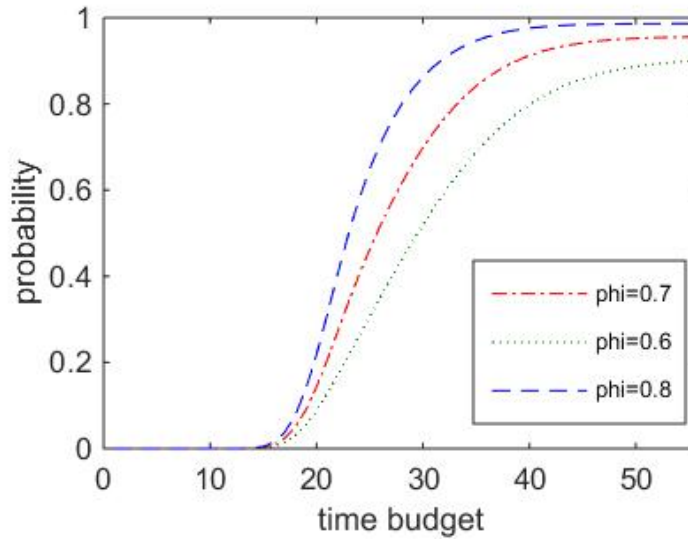


Fig. 9 The probability of arriving on time for  $0.5 < \Psi \leq 0.8$  (first scenario).

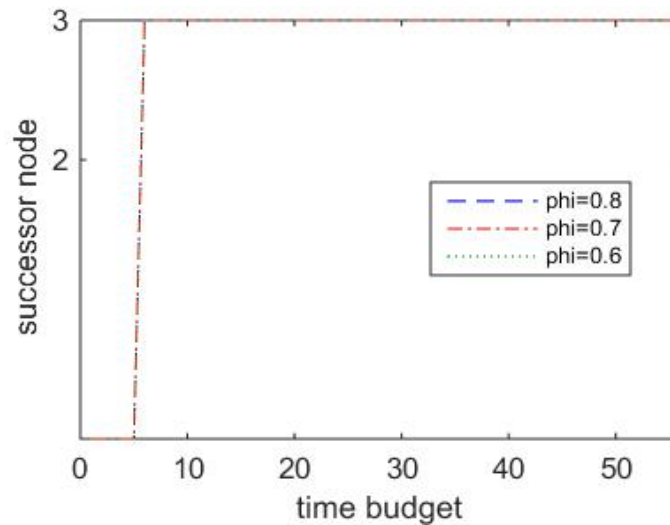


Fig. 10 The routing policy on node 1 for  $0.5 < \Psi \leq 0.8$  (first scenario).

a lower value of  $\psi$ , asks for more path-robustness or path-flexibility, and, in the counterpart, he loses in term of travel-time reliability.

The difference  $u_{1,3}(t, 3, \psi_1) - u_{1,3}(t, 3, \psi_2)$  can then be interpreted as the price of path-robustness corresponding to a measure of it, given by the difference  $\psi_1 - \psi_2$ .

In the first scenario (Figs. 5 and 6), for  $\psi = 1$  (dashed blue line), the optimal successor node is Node 3, for all the time budgets up to 44 time units. Then for a time budget bigger than 44, the optimal successor node is Node 2. Since we do not take into account path-robustness here ( $\psi = 1$ ), it is trivial that paths passing from Node 2 are better than those passing by Node 3, as we put large values of the average travel-time on links (2-7-37-39). For  $0.8 \leq \psi < 1$  (solid line), we can clearly see that, we obtain the same strategy as in the case  $\psi = 1$ . However, for  $0.5 < \psi \leq 0.8$ , the optimal policy changes, and Node 3 becomes the optimal successor node (Figs. 9 and 10). That means that, Node 2 which has a lower number of successors comparing to the number of successors of Node 3 is penalized, i.e., it gets low values  $u_{ki}(t, y)$ . Therefore, paths that pass through Node 2, i.e., paths with small number of alternatives or detours, have low probability to be selected as optimal paths.

In the second scenario (Node 2 has only one successor node), the optimal policy obtained by the two cases is clearly different from each other. For  $\psi = 1$  (dashed line, Figs. 7 and 8), we obtain the same optimal policy as in the first scenario. However, for  $\psi < 1$  (solid line, Figs. 7 and 8), the optimal policy changes, and Node 3 is the better optimal successor node for Node 1. That means that, Node 2 which has only one successor node is penalized, and paths that pass through that node, i.e., paths with small number of alternatives or detours, have low probability to be selected as optimal paths.

From these results, we can conclude that:

- If a user maximizes the travel time reliability of the paths, without taken into account their robustness,

then he should choose the path that passes from Node 2, because it is the one maximizing the probability of reaching the destination Node 24 in the considered time budget;

- If a user seeks a guarantee in terms of robustness even with a loss in terms of travel time reliability, then he should choose the path passing from Node 3.

## 7. Conclusions

This article considers the optimal guidance problem of users in road networks, and proposes a new robust adaptive strategy. We base on an existing routing model, which is a SOTA like algorithm, which we extend 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). Finally, we illustrated our algorithm on a small network, with some academic scenarios.

## References

- [1] Dijkstra, E. W. 1959. "A Note on Two Problems in Connection with Graphs." *Numerical Mathematics* 1: 269-71.
- [2] Provan, J. S. 2003. "A Polynomial-Time Algorithm to Find Shortest Paths with Recourse." *Networks* 41 (2): 115-25.
- [3] Cheung, C., and Lam, W. 1998. "Pedestrian Route Choices between Escalator and Stairway in MTR Stations." *Journal of Transportation Engineering* 124 (3): 277-85.
- [4] Gao, S., and Chabini, I. 2006. "Optimal Routing Policy Problem in Stochastic Time-Dependent Networks." *Transportation Research Part B* 40: 93-122.
- [5] Fan, Y., Kalaba, R., and Moore, J. E. 2005. "Arriving on Time." *Journal of Optimization Theory and Applications* 127 (3): 497-513.
- [6] Waller, S. T., and Ziliaskopoulos, A. K. 2002. "On the Online Shortest Path Problem with Limited Arc Cost Dependencies." *Networks* 40 (4): 216-27.

- [7] Mirchandani, P. B. 1976. "Shortest Distance and Reliability of Probabilistic Networks." *Computers and Operations Research* 3 (4): 347-55.
- [8] Frank, H. 1969. "Shortest Paths in Probabilistic Graphs." *Operations Research* 17: 583-99.
- [9] Sen, S., Pillai, R., Joshi, S., and Rathi, A. 2001. "A Mean-Variance Model for Route Guidance in Advanced Traveler Information Systems." *Transportation Science* 35 (1): 37-49.
- [10] Sigal, C. E., Alan, A., Pritsker, B., and Solberg, J. J. 1980. "The Stochastic Shortest Route Problem." *Operations Research* 28 (5): 1122-9.
- [11] Loui, R. P. 1983. "Optimal Paths in Graphs with Stochastic or Multidimensional Weights." *Communications of the ACM* 26 (9): 670-6.
- [12] Murthy, I., and Sarkar, S. 1996. "A Relaxation-Based Pruning Technique for a Class of Stochastic Shortest Path Problems." *Transportation Science* 30 (3): 220-36.
- [13] Nie, Y., and Wu, X. 2009. "Shortest Path Problem Considering On-Time Arrival Probability." *Transportation Research Part B: Methodological* 43 (6): 597-613.
- [14] Nie, Y., and Fan, Y. 1964. "Arriving-on-Time Problem: Discrete Algorithm That Ensures Convergence." *Transportation Research Record* 1964: 193-200.
- [15] Fan, Y., and Nie, Y. 2006. "Optimal Routing for Maximizing Travel Time Reliability." *Networks and Spatial Economics* 6 (34): 333-44.
- [16] Samaranayake, S., Blandin, S., and Bayen, A. 2012. "A Tractable Class of Algorithms for Reliable Routing in Stochastic Networks." *Transportation Research Part C: Emerging Technologies* 20 (1): 199-217.
- [17] Samaranayake, S., Blandin, S., and Bayen, A. 2012. "Speedup Techniques for the Stochastic On-Time Arrival Problem." In *OASICs—Open Access Series in Informatics*, volume 25.
- [18] Dean, B. C. 2010. "Speeding up Stochastic Dynamic Programming with Zero-Delay Convolution." *Algorithmic Operations Research* 5 (2): 96-104.
- [19] Sabran, G., Samaranayake, S., and Bayen, A. 2014. "Precomputation Techniques for the Stochastic On-Time Arrival Problem." In *ALENEX, SIAM*, 138-46.
- [20] Kobitzsch, M., Samaranayake, S., and Schieferdecker, D. 2014. "Pruning Techniques for the Stochastic On-Time Arrival Problem: An Experimental Study." *Computer Sciences, Data Structures and Algorithm*.
- [21] Manseur, F., Farhi, N., Haj-Salem, H., and Lebacque, J. P. 2016. "Robust Adaptive Strategies for the Guidance of Users in Road Networks." *Transportation Research Procedia*. arXiv:1606.09111.
- [22] Farhi, N., Haj.Salem, H., and Lebacque, J. P. 2014. "On the Robust Guidance of Users in Road Traffic Networks." In *AIP Conference Proceedings* 1648 (1): 530004.
- [23] Smith, O. E., Adelfang, S. I., and Tubbs, J. D. 1982. *A Bivariate Gamma Probability Distribution with Application to Gust Modeling*. Nasa Technical Memorandum-82483.