Robust Adaptive Routing Under Uncertainty

Arthur Flajolet
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, flajolet@mit.edu
Ecole Polytechnique, Route de Saclay, 91120 Palaiseau, France

Sébastien Blandin
IBM Research Collaboratory, 9 Changi Business Park Central 1, Singapore 486048, Singapore, sblandin@sg.ibm.com

Patrick Jaillet
Department of Electrical Engineering and Computer Science, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, jaillet@mit.edu

We consider the problem of finding an optimal history-dependent routing strategy on a directed graph weighted by stochastic arc costs when the decision maker is constrained by a travel-time budget and the objective is to optimize the expected value of a function of the budget overrun. Leveraging recent results related to the problem of maximizing the probability of termination within budget, we first propose a general formulation and solution method able to handle not only uncertainty but also tail risks. We then extend this general formulation to the robust setting when the available knowledge on arc cost probability distributions is restricted to a given subset of their moments. This robust version takes the form of a continuous dynamic programming formulation with an inner generalized moment problem. We propose a general purpose algorithm to solve a discretization scheme together with a streamlined procedure in the case of scarce information limited to lower-order statistics. To illustrate the benefits of a robust policy, we run numerical experiments with field data from the Singapore road network.

1. Introduction

1.1. Motivation

Stochastic Shortest Path (SSP) problems have emerged as natural extensions to the classical shortest path problem when arc costs are uncertain and modeled as outcomes of random variables. In particular, we consider in this paper the class of adaptive SSPs where we optimize over all history-dependent strategies, in the same sense as for Markov Decision Processes (MDPs), as opposed to just deterministic paths. In stark contrast to the deterministic setting, adaptive strategies may significantly outperform a priori paths depending on the actual criterion considered. In that respect, adaptive SSP frameworks share common properties with MDPs in the sense that optimal solutions are often characterized by dynamic programming equations involving expected values (e.g. Bertsekas and Tsitsiklis (1991)). Yet, computing the expected value of a function of a random variable generally requires a full description of its probability distribution, and this can be hard to obtain accurately due to errors and sparsity of measurements. In practice, only approximations of real arc cost probability distributions...
are available and the optimal strategy for the approximated arc cost probability distributions may be suboptimal with respect to the real arc cost probability distributions.

One of the most common applications of SSPs consists of the problem of routing vehicles in transportation networks. Providing driving itineraries is a challenging task as suppliers have to cope simultaneously with limited knowledge on random fluctuations in traffic congestion (e.g. caused by traffic incidents, variability of travel demand) and users’ desire to arrive on time. These considerations led to the definition of the Stochastic On-Time Arrival (SOTA) problem, an adaptive SSP problem with the objective of maximizing the probability of on-time arrival, and formulated using dynamic programming in Nie and Fan (2006). The algorithm proposed in Samaranayake et al. (2011) to solve this problem requires the knowledge of the complete arc travel-time distributions. Yet, in practice, these distributions tend to be estimated from samples which are sparse and not error-free.

In recent years, optimization methods have been developed to estimate lower and upper bounds on expected values when the underlying distribution is only known through some of its statistics or from a collection of samples. Approaches based on moments, generalizing the pioneer work of Chebyshev and referred to as Generalized Moment Problem (GMP), have led to the development of a fruitful theory yielding in particular closed-form bounds. See Kemperman (1987), Karlin and Shapley (1953), Isii (1960) and Smith (1995) for geometrical insights and Shapiro (2001) for duality results. Numerical bounds have been derived from these closed-form bounds, notably using tools drawn from linear, e.g. tailored dual simplex algorithm in Prékopa (1990), and semidefinite programming in Bertsimas and Popescu (2005) and Vandenberghe et al. (2007).

In the case of limited knowledge on arc cost probability distributions, we propose to bring GMP techniques to bear on adaptive SSP problems to help mitigate the impact of the lack of information by relying instead on lower-order statistics that are usually easier to estimate accurately. The resulting policy inherits the “robustness to error measurements property” induced by the approach, making it more reliable for the decision maker. Prior work, e.g. Prékopa (1990) and Bertsimas and Popescu (2005), has focused on developing efficient procedures to solve any single stage moment problem from scratch. Yet, in the course of solving a dynamic programming formulation, closely related problems (e.g. estimating an expected value $E[f(t,X)]$ where the random variable $X$ is fixed but $t$ varies depending on the state) arise consecutively and making the most of previous computations becomes crucial to achieve computational tractability. For this reason, existing GMP algorithms cannot be readily used as tractability is a major concern.

1.2. Related Work

Extending the shortest path problem by assigning random as opposed to deterministic costs to arcs requires some additional modeling assumptions. Over the years, many formulations have been proposed which differ along three main features:
• The specific objective function to optimize: in the presence of uncertainty, the most natural approach is to minimize the total expected costs, see Bertsekas and Tsitsiklis (1991), and Miller-Hooks and Mahmassani (2000) for time-dependent random costs. However, this approach is oblivious to risk. In an attempt to take that factor into account, Loui (1983) proposed earlier to rely on utility functions of moments (e.g. mean costs and variances) involving an inherent trade-off, and considered multi-objective criteria. However, Bellman’s principle of optimality no longer holds for arcs weighted by multidimensional costs, giving rise to computational hardness. A different approach consists of introducing a budget, set by the user corresponding to the maximum total cost he is willing to pay to reach his terminal node. Such approaches have been considered along several different directions. Research efforts have considered either minimizing the probability of budget overrun (see Frank (1969), Nikolova et al. (2006b), and also Xu and Mannor (2011) for probabilistic goal MDPs), minimizing more general functions of the budget shortfall as in Nikolova et al. (2006a), minimizing refined satisficing measures in order to guarantee good performances with respect to several other objectives as in Jaillet et al. (2013), and constraining the probability of over-spending while optimizing the expected costs as in Xu et al. (2012).

• The admissible set of strategies over which we are free to optimize: incorporating uncertainty may make history-dependent strategies more attractive depending on the chosen performance index. This is the case of the SOTA problem where two types of strategy were explored. On one hand, an a-priori formulation which consists in finding a path before taking any actions, see Nikolova et al. (2006b) and Nie and Wu (2009). On the other hand, an adaptive formulation which allows to refine the strategy on the way based on the remaining budget, see Nie and Fan (2006) and Samaranayake et al. (2011).

• The knowledge on the random arc costs taken as an input: it can range from the full knowledge of the probability distributions to having access to only a few samples drawn from them. In practical settings, the problem of estimating accurately some statistics (e.g. mean cost and variance) seems more reasonable than retrieving the full probability distribution. For instance, Jaillet et al. (2013) consider lower-order statistics (minimum, average and maximum costs) and make use of closed formed bounds derived in the GMP theory. In the context of MDPs, these considerations were extensively investigated, see Nilim and Ghaoui (2003), Satia and Lave (1973) and White III and Eldeib (1994) to cite a few. In Nilim and Ghaoui (2005), transition probability distributions are restricted to lie close to the empirical probability distributions in terms of entropy.

We give an overview of prior formulations in Table 1.
1.3. Contributions

In this paper, we extend the adaptive SSP framework to:

1. Allow for a large family of objective functions. Specifically, the objective can be the expected value of any asymptotically increasing concave function of the budget shortfall, provided this function satisfies a specific decaying property;

2. Relax the requirement of complete knowledge on arc cost probability distributions and substitute it with a subset of their moments together with lower and upper bounds on the arc costs, i.e. interval data.

The remainder of the paper is organized as follows. In Section 2, we first review the relevant prior work on the adaptive SSP and then extend the formulation to a large class of objective functions. We also illustrate, via an example, the possible shortcomings of this formulation when there is only limited knowledge on arc cost probability distributions. This motivates the introduction of an adaptive robust formulation based on moments which takes the form of a continuous dynamic programming formulation with a GMP as inner problem. In Section 3, we introduce a discretization scheme to solve the dynamic programming formulation which we show to converge toward the solution to the continuous problem. Then, we use known duality-based theoretical results on the GMP to develop a general purpose algorithm solving the problem for any given number of moments. In Section 3.3, in an effort to make the approach tractable for real-time applications, we propose a streamlined procedure, designed when the available knowledge is restricted to the minimum, mean and maximum costs on each arc. This effort is pursued for any number of moments in Section 3.4. In Section 4, we present results of numerical experiments run with both synthetic and real data on the Singapore road network.

| Author(s) | Objective function | Strategy | Uncertainty description | Approach |
|-----------|--------------------|----------|-------------------------|----------|
| Loui (1983) | utility function | a priori | moments | dominated paths |
| Nikolova et al. (2006b) | probability of budget overrun | a priori | normal distributions | convex optimization |
| Nie and Fan (2006) | probability of budget overrun | adaptive | distributions | dynamic programming |
| Samaranayake et al. (2011) | probability of budget overrun | adaptive | distributions | dynamic programming |
| Nilim and Ghaoui (2005) | expected cost | adaptive | samples | dynamic programming |
| Jaillet et al. (2013) | lateness index | a priori | distributions or moments | iterative procedure |
| Adulyasak and Jaillet (2014) | worst-case cost | a priori | intervals or discrete scenarios | integer programming |
2. Problem Formulation

In this section, we review results on the adaptive SOTA problem. Then, we extend the formulation to allow a wide variety of criteria. We also provide an example illustrating the limitations of the approach when little is known about arc cost probability distributions. This motivates the definition of a robust counterpart based on moments in Section 2.4.

2.1. Adaptive Formulation to the Stochastic On-Time Arrival Problem

Let \( G = (V, A) \) be a finite directed graph where each arc \((i, j) \in A\) is assigned a cost \( c_{ij}\). Arc costs are collectively defined as independent non-negative random variables, each with its own probability density function \( p_{ij}(\cdot) \) (for clarity of the exposition, \( c_{ij}\) is assumed to be a continuous random variable but this is by no means a limitation). The cost of an arc is assumed to correspond to an independent realization of its corresponding random variable each time it is crossed.

We consider a user traveling through \( G \) from \( s \in V \) to \( d \in V \). The user is constrained by a time budget \( T \) and wants to maximize the probability of reaching the destination \( d \) within his budget. We optimize over the restricted set of history-dependent strategies which correspond to Markov policies (those which can be described by local decisions based on current location and remaining budget). This problem is called the adaptive SOTA problem and was introduced in Nie and Fan (2006). In this paper, the authors calculate the optimal policy by computing \( u_i(t) \), the maximum probability of reaching the destination \( d \) within budget when leaving \( i \in V \) with remaining budget \( t \). The optimal policy is the solution to the following continuous dynamic programming formulation.

\[
\forall i \in V, i \neq d, \forall t \in [0, T]
\]

\[
u_i(t) = \max_{j \in V(i)} \int_0^t p_{ij}(\omega) \cdot u_j(t - \omega) d\omega
\]

\[
u_i(t) = 1_{t \geq 0}, \forall t
\]

where \( V(i) = \{ j \in V \mid (i, j) \in A \} \) refers to the set of immediate successors of \( i \) in \( G \) and \( u_i(t) = 0 \) for \( t < 0, i \in V \).

The definition of \( u_i(t) \) follows from a node-based Bellman principle for optimally selecting the next node to visit. At each node \( i \in V \), and for each potential remaining budget \( t \), the decision maker should pick the outgoing edge \((i, j)\) that gives the maximum probability of reaching the destination within remaining budget if acting optimally thereafter. Thus, \( u_s(T) \) is the optimal value of the quantity of interest to the user. Note that the inner problem arising in (1) can be expressed as the problem of computing the expected value of a function of \( c_{ij} \):

\[
\max_{j \in V(i)} \int_0^t p_{ij}(\omega) u_j(t - \omega) d\omega = \max_{j \in V(i)} \mathbb{E}(u_j(t - c_{ij}))
\]
If each of the random costs $c_{ij}$ has a positive minimum realizable value $\delta_{ij}^{\text{inf}} > 0$, Samaranayake et al. (2011) notice that the functions $u_i(\cdot)$ can be computed by a label-setting algorithm in increments of size $\delta_{ij}^{\text{inf}}$ without resorting to value iteration. Specifically, since $p_{ij}(\omega) = 0$ for $\omega \leq \delta_{ij}^{\text{inf}}$, $u_i(t)$ will not depend on the values taken by $u_j(\cdot)$ on $[t - \delta_{ij}^{\text{inf}}, t]$. After the following straightforward initializing step: $\forall i \neq d, u_i(t) = 0$ for $t \leq 0$, they compute $(u_i(\cdot))_{i \in V}$ block by block: first $(u_i(\cdot)_{[0, \delta_{ij}^{\text{inf}}]}), i \in V$, then $(u_i(\cdot)_{[0, 2\delta_{ij}^{\text{inf}}]}), i \in V$, and so on to eventually derive $(u_i(\cdot)_{[0, T]}), i \in V$.

2.2. Extension to a more general class of objectives

There are several potentially significant limitations to the problem formulated in (1). First, it looks for policies which maximize the probability of arriving within a given budget, but this does not allow for a comprehensive comparative study between optimal policies. Indeed, consider two equally good policies with 20% chance of achieving the goal within budget, there is no guarantee on the quality of these policies in the remaining 80% scenarios, and they may have completely different behaviors with respect to the expected budget shortfall. Another, perhaps more serious, drawback is the possibility of having the traveler trapped in some state $i$ without any further instructions. Indeed, no further directions are computed as soon as the remaining budget falls below zero (which may occur if $u_i(T) < 1$, since for $t \leq 0$ $u_i(t)$ is identically zero).

To address these issues, we observe that an optimal Markov strategy maximizing the expected value of a function $f(\cdot)$ of the budget shortfall (i.e., of $T - X$ where $X$ refers to the random cost of the selected strategy) can be obtained as the solution to the following modified mathematical optimization.

$$u_i(t) = \max_{j \in V(i)} \int_{0}^{\infty} p_{ij}(\omega) \cdot u_j(t - \omega) d\omega$$

$$u_d(t) = f(t), \ \forall t$$

(3)

There are two main differences with (1). First, the integral is extended over $[0, \infty)$ to account for the possibility of being on our way to $d$ with negative remaining budget. Second, $u_d(\cdot)$ is defined more generally as the function $f(\cdot)$. Examples include $f(t) = t \cdot 1_{t \leq 0}$ which corresponds to minimizing the expected budget gap. Although (2) can still be solved incrementally in the same fashion as (1), the initializing step gets tricky if $f(\cdot)$ does not have a one-sided compact support of the type $[a, \infty)$ with $-\infty < a$ (the initializing step is simply $\forall i \neq d, u_i(t) = 0$ for $t \leq a$ when $f(\cdot)$ has support $[a, \infty]$). Indeed, observe that no closed form formula is available for $u_i(-t)$ for $t \to \infty$ while we may need to compute this value as the random cost associated with an optimal strategy need not be bounded as it might contain infinitely many loops. For this last point, note that even when $f(\cdot)$ has a one-sided compact support, the optimal strategy may contain a loop. For instance, this is the case when the objective is the probability of arriving on-time, see Samaranayake et al. (2011), and the example
If the initial budget is $T = 8$ and the objective function is $f(t) = t \cdot 1_{t \leq 0}$, the user will first be routed to $a$. If he experiences $c_{sa} = 5$, then the optimal strategy is to head back to $s$.

Figure 1 If the initial budget is $T = 8$ and the objective function is $f(t) = t \cdot 1_{t \leq 0}$, the user will first be routed to $a$. If he experiences $c_{sa} = 5$, then the optimal strategy is to head back to $s$.

described therein can be adapted for other natural objective functions, e.g. when the objective is to minimize the expected budget shortfall, see Figure 1. But there may be infinitely many, as opposed to finitely many, loops for more general objective functions, e.g. when $f(t) = -t \cdot 1_{t \leq 0}$ which corresponds to maximizing the expected budget overrun and an associated optimal policy includes infinitely many cycles. That is to say, depending on $f(\cdot)$, a user traveling through $\mathcal{V}$ following the optimal strategy may get in state $i$ with an arbitrarily large negative budget. In Theorem 1 we show that the optimal strategy gives the least expected cost path when the remaining budget is below a threshold, provided that $f(\cdot)$ satisfies some asymptotic conditions. This enables us to properly initialize the functions $(u_i(\cdot))_i$ and thereby solve (3) using the label-setting algorithm described in Section 2.1. To prove the claim, we also need the following technical requirement which we will assume throughout this paper. It states that each arc has a finite maximum and a positive minimum realizable cost:

Assumption 1. $\forall (i,j) \in \mathcal{A}$, $p_{ij}(\cdot)$ has compact support included in $[\delta_{ij}^\inf, \delta_{ij}^\sup]$ with $\delta_{ij}^\inf > 0$ and $\delta_{ij}^\sup < \infty$. Thus $\delta_{ij}^\inf = \min_{(i,j) \in \mathcal{A}} \delta_{ij}^\inf > 0$ and $\delta_{ij}^\sup = \max_{(i,j) \in \mathcal{A}} \delta_{ij}^\sup < \infty$.

Theorem 1. Under Assumption 1 and assuming there exists $T_1$ such that the objective function of the budget shortfall $f(\cdot)$ is increasing, concave and $C^2$ on $(-\infty, -T_1)$, and satisfies the following asymptotic decay requirement:

$$\forall \alpha \geq 0 \lim_{t \to -\infty} \frac{\inf_{[t-\alpha,t]} f''}{f'(t)} = 0$$

then (3) can be solved with a label-setting algorithm.

The proof is deferred to the appendix, Section B.1.

Observe that any polynomial function satisfies the asymptotic decay requirement. Examples of valid objectives include minimization of the budget shortfall $f(t) = t \cdot 1_{t \leq 0}$, and minimization of the squared budget shortfall $f(t) = -t^2 \cdot 1_{t \leq 0}$. If $f(\cdot)$ is increasing but does not meet both the decaying and concavity requirements, following an optimal policy may lead to spending an arbitrarily large
budget with positive probability. As we highlight in Example 1, this property is intrinsically tied to risk aversion. For this reason, it is not clear how to solve (3) without resorting to value iteration when the assumptions on \( f(\cdot) \) do not hold.

**Example 1.** Consider the simple directed graph of Figure 2a and the objective function \( f(\cdot) \) illustrated in Figure 2b. \( f(\cdot) \) is defined piecewise, alternating between concavity and convexity on intervals of size \( 2T^* \) and the same pattern is repeated every \( 2T^* \). This means that, for this particular objective, the attitude towards risk keeps fluctuating as the budget decreases, from being risk-averse when \( f(\cdot) \) is locally concave to being risk-seeking when \( f(\cdot) \) is locally convex. Observe that, in addition to not being concave, \( f(\cdot) \) does not satisfy the decaying property as \( f'(\cdot) \) is \( 2T^* \)-periodic, implying that \( t \to \inf_{t \in [0,T^*]} f''(t) \) is also \( 2T^* \)-periodic and so cannot converge as \( t \to \infty \). Now consider \( \delta^{\inf} << 1, \epsilon << 1 \) and \( T^* > 3 \) and consider finding a strategy to get to \( d \) starting from \( s \) with initial budget \( T \).

Going straight to \( d \) incurs an expected objective value of \( f(T - 2) < \frac{1}{2} f(T - 1) + \frac{1}{2} f(T - 3) \) and we can make this gap arbitrarily large by properly defining \( f(\cdot) \). Therefore, by taking \( \epsilon \) and \( \delta^{\inf} \) small enough, going to \( a \) first is optimal. With probability \( \epsilon > 0 \), we arrive at \( a \) with a remaining budget of \( T - T^* \). Afterwards, the situation is reversed as we are willing to take as little risk as possible and the corresponding optimal solution is to go back to \( s \). With probability \( \epsilon \), we arrive at \( s \) with a budget of \( T - 2T^* \) and we are back in the initial situation, showing the existence of infinite cycling.

In the sequel, we focus on the case \( f(t) = 1_{t \geq 0} \), i.e. the objective is to minimize the probability of budget overrun, to simplify the presentation but results hold for the class of functions defined in Theorem 1 unless otherwise stated.

### 2.3. Sensitivity to error measurements

In order to identify an optimal policy, we need to evaluate for each node \( i \in V \) and for all \( 0 \leq t \leq T \) the expected values in (2). However, when arc cost probability distributions are not given, we are
unable to carry out this calculation. In practice, we often have only access to some limited realizations of the random variables $c_{ij}$. With this restricted amount of data, we might only be able to assess accurately some statistics of $p_{ij}(\cdot)$, which raises the question of how to select the cost probability distributions input into (1). Example 2 illustrates that using arbitrarily probability distributions in the formulation may produce misleading optimal policies, i.e. optimal with respect to the chosen probability distributions but not the real ones.

**Example 2.** Consider the graph illustrated in Figure 3(a) and assume that $(m_{ij})$, the set of average arc costs, is the only data at hand. Two different options for the cost distributions consistent with this a priori information are represented in Figure 3a and 3b. The solution of problem (1) with the distributions of Figure 3a is:

$$u_s(t) = \begin{cases} 
0 & \text{if } t < m^{\text{LET}} \\
1 & \text{if } t \geq m^{\text{LET}}
\end{cases}$$

where $m^{\text{LET}}$ denotes the overall least expected cost required to get to $d$ starting from $s$ ($m^{\text{LET}} = 6$ in our case) and the optimal policy is to follow the least expected path irrespective of the initial budget $T$. The solution of problem (1) with distributions of Figure 3b is:

$$u_s(t) = \begin{cases} 
0 & \text{if } 0 \leq t < 1 \\
\frac{2}{3} & \text{if } 1 \leq t < 5 \\
\frac{4}{3} & \text{if } 5 \leq t < 13 \\
1 & \text{if } 13 \leq t
\end{cases}$$

Consider the results in Table 2 giving the optimal policies for a budget $T = 6$. If the cost distributions in Figure 3b were the real ones but had been estimated to be those of Figure 3a, then the obtained policy would clearly be non-optimal.

Estimating accurately the cost probability distributions with samples drawn from realizations usually requires a large sample size. Hence, the data input for problem (1) is subject to errors. In light of Example 2, this may cause calculated policies to be non-optimal for the real cost probability distributions. On the other hand, the central limit theorem shows that lower-order statistics (e.g.
average costs) and static quantities (e.g. maximum/minimum costs) can be estimated more accurately and thus are more reliable. Since we are working with probability distributions, a robust strategy relying only on a single quantity (e.g. maximum costs) may be too conservative. Hence, we propose to use moments and support information to address the issue of error measurements as discussed above.

### 2.4. Robust Formulation Based On Moments

Two types of statistics are supposed to be available on the otherwise unknown cost probability distributions: their support and a subset of their moments consisted of their first \( k \in \mathbb{N} \) moments. For each link \((i, j) \in \mathcal{A}\), \( p_{ij}(\cdot) \) refers to the actual cost probability distribution while \([\delta_{ij}^{\inf}, \delta_{ij}^{\sup}]\) and \((m_{ij}^n)_{n \leq k}\) respectively denote its support and the available moments (i.e. we have available \( \mathbb{E}_{p_{ij}}(X^n) = m_{ij}^n \forall n \leq k \)). This a priori knowledge enables the definition of arc-based uncertainty sets on the unknown cost probability distributions.

**Definition 1.** For each link \((i, j) \in \mathcal{A}\), the set of probability distributions compatible with the available information is defined as

\[
P_{ij}^k = \{ p \in \mathcal{P}([\delta_{ij}^{\inf}, \delta_{ij}^{\sup}]) : \mathbb{E}_p(X) = m_{ij}^1, \ldots, \mathbb{E}_p(X^k) = m_{ij}^k \}
\]

where \( \mathcal{P}([\delta_{ij}^{\inf}, \delta_{ij}^{\sup}]) \) is the set of probability measures on \([\delta_{ij}^{\inf}, \delta_{ij}^{\sup}]\).

In the rest of the paper, we assume that \( P_{ij}^k \) is not empty. The interested reader is referred to Karlin and Shapley (1953) for explicit conditions on the existence of such a measure. We draw upon this class of uncertainty sets to define a robust version of (1).

**Definition 2. Robust Adaptive Stochastic Shortest Path Problem with \( k \) moments**

The robust counterpart of level \( k \) to the adaptive on-time arrival problem is formulated as follows:

\[
\forall i \neq d \in \mathcal{V}, \forall t \leq T
\]

\[
\ddot{u}_i^k(t) = \max_{j \in \mathcal{V}(i)} \inf_{p_{ij} \in P_{ij}^k} \int_0^t p_{ij}(\omega) \cdot \ddot{u}_j^k(t - \omega) d\omega
\]

\[
\ddot{u}_d^k(t) = 1_{t \geq 0}, \forall t
\]

where \( \mathcal{V}(i) = \{ j \in \mathcal{V} \mid (i, j) \in \mathcal{A} \} \).

Being at node \( i \) with remaining budget \( t \), the optimal robust strategy consists of picking the outgoing edge that has the highest worst-case probability of budget overrun, where the worst-case probability
of budget overrun assumes that an adversary chooses, for each outgoing edge, the arc-cost probability distribution yielding the lowest probability of budget overrun. It is formulated using a continuous dynamic programming equation similarly to the nominal approach (1). The only difference lies in the nature of the underlying inner problem which accounts for most of the computational burden. As opposed to being a convolution product as in (1), it turns into an optimization problem in (4). Specifically, for each budget \( t \leq T \) and for each arc \((i, j)\), we need to solve:

\[
\inf_{p_{ij} \in P_{ij}} \int_0^t p_{ij}(\omega) \cdot \tilde{u}^k_j(t - \omega) d\omega = \inf_{p_{ij} \in P_{ij}} \mathbb{E}_{p_{ij}}(\tilde{u}^k_j(t - c_{ij})).
\]

This instance falls within the theory of the generalized moment problems (GMP) extending Chebyshev’s inequality as described in the next section. To solve these conic problems, duality theory plays a crucial role as the dual form tends to be easier. For instance, Bertsimas and Popescu (2005) propose to solve the dual with semidefinite programming when \( \phi(\cdot) \) is a piecewise polynomial function.

Before proceeding further, let us state a lemma which guarantees the existence of a unique solution to (4).

**Lemma 1.** Under Assumption 1, \( \tilde{u}^k_i(\cdot) \) is a well defined function which is also non-decreasing for all nodes \( i \) and for all \( k \in \mathbb{N} \).

The proof is deferred to the appendix, Section 3.2.

### 3. Solution Methodology

In this section, we develop the methodology needed to find a solution to (4). First, we focus on the inner problem of (4) and review results derived in Shapiro (2001) on the duality theory of the GMP. Next, in Section 3.2 we study (4) for any number \( k \) of moments and design a discretization scheme along with a general purpose algorithm for the discretized problem. In Section 3.3 we specialize this algorithm to the case \( k = 1 \). In Section 3.4 we go back to \( k > 1 \) moments and, using additional discretization, we propose an efficient algorithm for a restricted class of objectives.

#### 3.1. Duality Theory of the Generalized Moment Problem

Let us first define the general optimization problem.

**Definition 3. Problem \( \mathcal{P} \)**

Take \( k \in \mathbb{N} \) and consider \( \Phi_1(\cdot), \ldots, \Phi_k(\cdot), \phi(\cdot) \), \( k + 1 \) real-valued functions and \((b_1, \ldots, b_k) \in \mathbb{R}^k\). The generalized moment problem is:

\[
\sup_{\mu \in \mathcal{P}(\Omega)} \mathbb{E}_\mu(\phi(X)) \quad \text{subject to} \quad \mathbb{E}_\mu(\Phi_1(X)) = b_1 \\
\quad \vdots \\
\quad \mathbb{E}_\mu(\Phi_k(X)) = b_k
\]
where $\mathcal{P}(\Omega)$ denotes the set of probability measures on $\Omega$.

As the set of non-negative measures on $\Omega$ is a cone, we can define the dual problem of $\mathcal{P}$.

**Lemma 2. Problem $D$**

The dual problem of $\mathcal{P}$ is the following semi-infinite linear programming problem:

$$\sup_{(x_1, \ldots, x_{k+1}) \in \mathbb{R}^{k+1}} \sum_{p=1}^{k} x_p \cdot b_p + x_{k+1}$$

subject to

$$\sum_{p=1}^{k} x_p \cdot \Phi_p(\omega) + x_{k+1} \leq \phi(\omega), \ \forall \omega \in \Omega$$

(6)

**Proof** This follows from standard conic duality theory. For a detailed discussion, see Shapiro (2001) Section 3. □

The following no-duality-gap proposition is a key result since the dual problem tends to be easier to solve.

**Proposition 1.** If the set $\Omega$ is compact, $\phi$ is a lower semi-continuous function, $\Phi_1(\cdot), \ldots, \Phi_k(\cdot)$ are continuous functions, and $\mathcal{P}$ admits a feasible solution, then the duality gap is zero and $\mathcal{P}$ admits an optimal solution.

**Proof** See Shapiro (2001) Corollary 3.1. □

### 3.2. Robust Approach for $k$ Moments

In this section, we highlight the computational difficulty of solving (11) and introduce a discretization scheme along with a general purpose algorithm.

**Computational Issue.** One of the major challenges in the design of algorithms to solve (11) is the lack of regularity of the functions $(\hat{u}_i^*(\cdot))$, induced by the dynamic programming equation. Indeed, semidefinite programming methods developed to solve GMPs require $\hat{u}_i^*(\cdot)$ to be “smooth”. Precisely, it has to be lower semi-continuous in order to rely on the dual form, as discussed in Section 3.1 and piecewise polynomial to formulate the dual as a semidefinite program, see Bertsimas and Popescu (2005). Yet, the maximum of lower semi-continuous functions needs not be lower semi-continuous itself. Therefore, $\hat{u}_i^*(\cdot)$, recursively defined as a maximum of finitely many functions, is not even guaranteed to satisfy this requirement, let alone being piecewise polynomial. To circumvent these problems, we discretize (11) by imposing that $(\hat{u}_i^k(\cdot))_i$ be piecewise lower semi-continuous linear functions.

**Discretization Scheme.** To numerically approximate $\hat{u}_i^k(\cdot)$, the interval $[0, T]$ is discretized into $L = \frac{T}{\Delta t}$ time steps of width $\Delta t$. As in Samaranayake et al. (2011), the discretization length is supposed to be small enough, namely $\Delta t < \delta_{inf}$. For each node $i \in \mathcal{V}$, $\hat{u}_i^k(\cdot)$ is approximated by a lower semi-continuous step function collectively defined in the next definition and referred to as $\hat{u}_i^{k, \Delta t}(\cdot)$. 
**Definition 4.** As initial conditions, we set \( \bar{u}_k^{i,0} = 0 \) on \( (-\infty, \Delta t) \), \( \forall i \in V - \{d\} \) (recall that \( \delta^{\text{inf}} > \Delta t \)), and \( \bar{u}_d^{k,0}(t) = 1 \) \( t > 0 \). At step \( n \in \{1, \cdots, L-1\} \), we have defined \( \bar{u}_k^{i,0}(\cdot) \) on \( (-\infty, n \Delta t) \), \( \forall i \in V \) and we compute:

\[
\bar{v}^{n+1} = \max_{j \in V(i)} \inf_{p_{ij} \in \mathcal{P}_{ij}} \mathbb{E}_{p_{ij}}(\bar{u}_j^{k,0}(n \Delta t - c_{ij}))
\]  

This enables us to define \( \bar{u}_k^{i,0}(t) \) for all \( t \in (n \Delta t, (n+1) \Delta t) \) by setting \( \bar{u}_k^{i,0}(t) = \bar{v}^{n+1} \).

We have the following properties on \( \bar{u}_k^{i,0}(\cdot) \):

**Lemma 3.** \( \forall i \in V, \bar{u}_k^{i,0}(\cdot) \) is well defined on \( (-\infty, T] \) and is a piecewise constant function which is also lower semi-continuous and non-decreasing.

The proof is deferred to the appendix, Section B.2.

In Theorem 2, we demonstrate the convergence of this “discrete” approximation.

**Theorem 2.** Under Assumption 1, the solution to the discretization scheme of Definition 4 converges increasingly to the lower semi-continuous regularization of the solution to the continuous formulation \( (4) \) when \( \Delta t \to 0 \). Specifically:

\[
\forall i \in V, \forall t \leq T, \quad \bar{u}_k^{i,0}(t) \uparrow_{\Delta t \to 0} \hat{u}_k^i(t-)
\]

where \( \hat{u}_k^i(t-) \) refers to the left one-sided limit of \( \hat{u}_k^i(\cdot) \) at \( t \).

The proof is deferred to the appendix, Section B.4.

As a by-product, we obtain that \( \bar{u}_k^{i,0}(\cdot) \) systematically yields a lower bound on \( \hat{u}_k^i(\cdot) \) irrespective of \( \Delta t \). Hence, \( \bar{u}_k^{i,0}(\cdot) \) is meaningful to the decision maker as a guarantee on the probability of completion within budget.

**General Purpose Algorithm.** We now propose an algorithm to solve the discretization scheme.

**Theorem 3.** Under Assumption 1, the discretization scheme of Definition 4 can be solved in a single iteration algorithm for any number of moments \( k \).

**Proof** The discretization scheme defined in Definition 4 is similar to the one introduced in Samaranayake et al. (2011) to solve the dynamic programming formulation \( (1) \). Provided that arcs all have positive minimum weights and that the discretization length is small enough, \( \delta^{\text{inf}} > \Delta t > 0 \), the authors show that the solution to the discrete counterpart of \( (1) \) can be computed in a single iteration (i.e. without resorting to value iteration). The only difference with \( (1) \) lies in the nature of the underlying inner problem which is a convolution product in \( (1) \) and turns into a GMP in \( (4) \). As a result, it is sufficient to show that the inner problems can be solved numerically. The latter correspond to GMPs with finitely many constraints, one for each of the first \( k \) moments, on a probability measure with prescribed compact support. Additionally, the function used as an argument of the expected value
(i.e. $\phi$ in (5)) is a lower semi-continuous step function with finitely many discontinuity points by Lemma 3. Bertsimas and Popescu (2005) prove that this optimization problem can be solved using semidefinite programming. □

Theorem 3 provides a general procedure based on semidefinite programming. However, such an algorithm is likely to be intractable. Indeed, at any step $n$ of the discretization scheme, the piecewise constant functions $\hat{u}^k_{\Delta t} (\cdot)$ have up to $\delta^\sup_{ij} - \delta^\inf_{ij}$ discontinuity points on $[n \cdot \Delta t - \delta^\sup_{ij}, n \cdot \Delta t - \delta^\inf_{ij}]$. Following Bertsimas and Popescu (2005), the induced dual problem can be cast as a semidefinite program with $\sim k \cdot \delta^\sup_{ij} - \delta^\inf_{ij}$ variables and constraints. As $\Delta t$ is required to be small ($\delta^\inf_{ij} > \Delta t$), a significant number of SDPs need to be solved.

3.3. Case $k = 1$

In this section, we characterize optimal solutions to the inner optimization problems of (4) in the case $k = 1$. This allows us to develop a fast algorithm tailored for this particular case. In Section 3.3.1, we first show that solving the inner problem of (7) amounts to finding the extreme points of a planar convex set. Second, in Section 3.3.2, we prove that we can solve (4) efficiently by maintaining the convex hull of a dynamic set of points and provide an algorithm. To simplify the presentation, we omit the index $k = 1$ and $\Delta t$ in $\hat{u}^k_{\Delta t}$ and assume that $\delta^\inf_{ij}$ and $\delta^\sup_{ij}$ are multiples of $\Delta t$. Hence, the approximation is denoted by $\hat{u}_i(\cdot)$ but should not be confused with the continuous solution to (4).

3.3.1. Solving the Inner Problem. Take a vertex $i \neq d$ and a vertex $j \in V(i)$, we need to solve, at each step $n$:

$$\sup_{\omega \in [\delta^\inf_{ij}, \delta^\sup_{ij}]} \mathbb{E}_\omega(\hat{u}_j(n \Delta t - X))$$

subject to

$$\mathbb{E}_\omega(X) = m_{ij}$$

First, remark that if $m_{ij} > n \cdot \Delta t$ then the optimal value is 0, attained by the Dirac measure supported on $\{m_{ij}\}$ ($\hat{u}_j(t)$ is zero for $t \leq 0$). This suggests that for budgets no larger than the least expected cost to get to the destination, the problem (4) with $k = 1$ does not provide any useful policy. Indeed, the probability of terminating with positive balance is 0, thereby no policy is retrieved. As a consequence, this approach can only be used for larger budgets. Nevertheless, note that this does not necessarily hold for different objectives $f(\cdot)$ as discussed in Section 2.2. We now assume $\delta^\sup_{ij} > \delta^\inf_{ij}$ otherwise the problem is trivial.

As $\hat{u}_j(\cdot)$ is lower semi-continuous, Proposition 1 shows that there exists an optimal solution to (8) and there is no duality gap with the dual, which is (see Lemma 2):

$$\sup_{(a,b) \in \mathbb{R}^2} a \cdot m_{ij} + b$$

subject to

$$a \cdot \omega + b \leq \hat{u}_j(n \Delta t - \omega), \quad \forall \omega \in [\delta^\inf_{ij}, \delta^\sup_{ij}]$$

(9)
In the next three paragraphs, we rewrite (9), describe optimal solutions to the new formulation, and provide an efficient procedure to compute the optimal value.

**Rewriting (9).** We start by changing the time direction \( \omega \rightarrow n \cdot \Delta t - \omega \) which corresponds to making the change of variables \((a, b) \rightarrow (-a, a \cdot n \Delta t + b)\):

\[
\sup_{(a,b) \in \mathbb{R}^2} \ a \cdot (n \cdot \Delta t - m_{ij}) + b \tag{10}
\]

subject to \( a \cdot \omega + b \leq \bar{u}_j(\omega), \ \forall \omega \in [n \cdot \Delta t - \bar{\delta}_{ij}^{\sup}, n \cdot \Delta t - \bar{\delta}_{ij}^{\inf}] \)

Since \( \bar{u}_j(\cdot) \) is a piecewise constant function with steps of width \( \Delta t \), only points \( \omega = k \cdot \Delta t, k \in \{n - \frac{\delta_{ij}^{\sup}}{\Delta t}, \cdots, n - \frac{\delta_{ij}^{\inf}}{\Delta t}\} \), are essential in the constraints of (10). Indeed, a straight line lies below \( \bar{u}_j(\cdot) \) on \([k \cdot \Delta t, (k + 1) \cdot \Delta t]\) if and only if both \((k \cdot \Delta t, \bar{u}_j(k \cdot \Delta t))\) and \(((k + 1) \cdot \Delta t, \bar{u}_j((k + 1) \cdot \Delta t))\) lie above this line. Building on this remark, (10) can be further simplified into a linear programming problem:

\[
\max_{(a,b) \in \mathbb{R}^2} \ a \cdot (n \cdot \Delta t - m_{ij}) + b \tag{11}
\]

subject to \( a \cdot k \cdot \Delta t + b \leq \bar{u}_j(k \cdot \Delta t), \ k = n - \frac{\delta_{ij}^{\sup}}{\Delta t}, \cdots, n - \frac{\delta_{ij}^{\inf}}{\Delta t} \)

**Optimal solutions to (11).** First we have the following lemma, ensuring the existence of an optimal solution to (11):

**Lemma 4.** The optimization problem (11) has a non-empty set of optimal solutions.

**Proof** (11) has finitely many constraints and thus is a linear programming problem. \((0,0)\) is a feasible solution and the optimization problem is bounded as \(a \cdot (n \cdot \Delta t - m_{ij}) + b \leq \bar{u}_j(n \cdot \Delta t - m_{ij}) \leq 1 < \infty\). As a result, there exists an optimal solution. \( \square \)

The next lemma provides insights on optimal solutions to (11).

**Lemma 5.** There exists an optimal solution to (11), \((a,b)\), such that the straight line \( \omega \rightarrow a \cdot \omega + b \) coincides with one of the straight lines joining two points of the form \((k \cdot \Delta t, \bar{u}_j(k \cdot \Delta t))\) with \( k \in \{n - \frac{\delta_{ij}^{\sup}}{\Delta t}, \cdots, n - \frac{\delta_{ij}^{\inf}}{\Delta t}\} \).

**Proof** Observe that the feasible region is pointed as the polyhedron described by the inequality constraints does not contain any line \((a \cdot n \cdot \Delta t - \frac{\delta_{ij}^{\inf}}{\Delta t} + b = 0\) and \(a \cdot n \cdot \Delta t - \frac{\delta_{ij}^{\sup}}{\Delta t} + b = 0\) implies \(a = b = 0\) since \(\delta_{ij}^{\inf} < \delta_{ij}^{\sup}\). Additionally, there exists an optimal solution to (11), therefore there also exists an optimal basic feasible solution for which two inequality constraints are binding. This uniquely determines a straight line connecting two points of the form \((k \cdot \Delta t, \bar{u}_j(k \cdot \Delta t))\) with \( k \in \{n - \frac{\delta_{ij}^{\sup}}{\Delta t}, \cdots, n - \frac{\delta_{ij}^{\inf}}{\Delta t}\} \). \( \square \)

**Finding an Optimal Basic Feasible Solution.** There is a one-to-one mapping between basic solutions to (11) and straight lines joining two time step points of \( \bar{u}_j(\cdot) \) on \([n \cdot \Delta t - \delta_{ij}^{\sup}, n \cdot \Delta t - \delta_{ij}^{\inf}]\). There are finitely many such straight lines (no more than \(\frac{1}{2} \cdot (\frac{\delta_{ij}^{\sup} - \delta_{ij}^{\inf}}{\Delta t})^2\) but not all of them are feasible since such a line might not lie below \( \bar{u}_j(\cdot) \) on the whole interval \([n \cdot \Delta t - \delta_{ij}^{\sup}, n \cdot \Delta t - \delta_{ij}^{\inf}]\). To find the admissible straight lines, we shall consider the upper convex hull generated by the discretization points, as defined below.
Let $\hat{C}_n$ be the convex set generated by the points of discontinuity of $\hat{u}_j(\cdot)$ on $[n \cdot \Delta t - \delta^\text{inf}_{ij}, n \cdot \Delta t - \delta^\text{sup}_{ij}]$. Let also $\hat{C}_n$ be its upper convex hull: $\hat{C}_n = \{y \in \mathbb{R}^2 \mid \exists x \in C_n \text{ such that } y \geq x\}$. $\hat{C}_n$ is a convex set and has a finite number of extreme points since they belong to $\{(k \cdot \Delta t, \hat{u}_j(k \cdot \Delta t)), k = n - \frac{\delta^\text{sup}_{ij}}{\Delta t}, \ldots, n - \frac{\delta^\text{inf}_{ij}}{\Delta t}\}$. We denote by $(\omega^n_i, \nu^n_i)_{0 \leq i \leq P_n \leq n}$ the extreme points of $\hat{C}_n$ listed in ascending order of the first component, where $P_n \leq \frac{\delta^\text{sup}_{ij} - \delta^\text{inf}_{ij}}{\Delta t}$.

Consider the example illustrated in Figure 4a. Not every point of discretization of $\hat{u}_j(\cdot)$ on $[n \cdot \Delta t - \delta^\text{sup}_{ij}, n \cdot \Delta t - \delta^\text{inf}_{ij}]$ is an extreme point of $\hat{C}_n$, but $(n \cdot \Delta t - \delta^\text{sup}_{ij}, \hat{u}_j(n \cdot \Delta t - \delta^\text{sup}_{ij}))$ and $(n \cdot \Delta t - \delta^\text{inf}_{ij}, \hat{u}_j(n \cdot \Delta t - \delta^\text{inf}_{ij}))$ both are, being respectively the leftmost and rightmost points. The next lemma provides a complete characterization of all the extreme points of $\hat{C}_n$.

**Lemma 6.** $(k \cdot \Delta t, \hat{u}_j(k \cdot \Delta t))$ is an extreme point of $\hat{C}_n$ if and only if there is no straight line joining two time step points $(r \cdot \Delta t, \hat{u}_j(r \cdot \Delta t))$ and $(q \cdot \Delta t, \hat{u}_j(q \cdot \Delta t))$ passing below $(k \cdot \Delta t, \hat{u}_j(k \cdot \Delta t))$ such that $n - \frac{\delta^\text{sup}_{ij}}{\Delta t} \leq r < k < q \leq n - \frac{\delta^\text{inf}_{ij}}{\Delta t}$.

**Proof** This characterization is straightforward since, by definition, $((q \cdot \Delta t, \hat{u}_j(q \cdot \Delta t)))_q$ are the only possible extreme points of $\hat{C}_n$.

Once we have identified the extreme points of $\hat{C}_n$, we can solve (11) in $O(\log(\frac{\delta^\text{sup}_{ij} - \delta^\text{inf}_{ij}}{\Delta t}))$ running time as shown in the following proposition.

**Proposition 2.** An optimal solution to (11) is given by the straight line joining the two consecutive extreme points of $\hat{C}_n$ satisfying $\omega^n_i \leq n \cdot \Delta t - m_{ij} \leq \omega^n_{i+1}$. Hence, it takes $O(\log(\frac{\delta^\text{sup}_{ij} - \delta^\text{inf}_{ij}}{\Delta t}))$ running time to solve (11) provided that extreme points are stored in a data structure allowing for binary search on the first coordinate.

The proof is deferred to the appendix, Section B.3.

As a result, solving (8) boils down to storing the extreme points of $\hat{C}_n$ in a data structure allowing for binary search on the first coordinate. So far, we have looked at (8) independently from one step
n to the next one. Yet, computing a solution to the discretized version of (4) requires to solve (8) at every step \( n = 1, \cdots, T \Delta t \) so that finding the set of extreme points from scratch every time is computationally demanding. However, optimization problems (8) arising at consecutive steps are very similar. This causes \( \hat{C}_n \) and \( \hat{C}_{n+1} \) to be closely related. In the next section, we make this statement precise using a class of problems known as Dynamic Convex Hull maintenance problems, and we provide an effective algorithm running in \( O(|A| \cdot T \Delta t \cdot \log(\frac{\delta_{\sup} - \delta_{\inf}}{\Delta t})) \) time. To put this result into perspective, we recall that the fastest known algorithm to solve the nominal problem runs in \( O(|A| \cdot T \Delta t \cdot \log(\frac{T}{\delta_{\inf}})) \), see Samaranayake et al. (2012).

### 3.3.2. Solving the Recursive Inner Problem.

To solve the global dynamic programming equation, functions \( \hat{u}_i(\cdot) \in V \) are computed in increments of size \( \delta_{\inf} \) along the same lines as for (1) in Section 2.1. This implies that, for a given arc \((i, j)\), values \( \hat{u}_j(k \cdot \Delta t) \) become sequentially available in ascending order of \( k \) as the algorithm runs. Therefore, a search for extreme points of \( \hat{C}_{n+1} \) begins upon identification of extreme points of \( \hat{C}_n \). Also observe that the interval of interest \([n \cdot \Delta t - \delta_{\sup}, n \cdot \Delta t - \delta_{\inf}] \) defining \( \hat{C}_n \) shifts to the right by one increment at each step. Thus, \( \hat{C}_n \) updates to \( \hat{C}_{n+1} \) by removing the leftmost extreme point \((n \cdot \Delta t - \delta_{\sup}, \hat{u}_j(n \cdot \Delta t - \delta_{\sup})) \) and adding \((n+1) \cdot \Delta t - \delta_{\inf}, \hat{u}_j((n+1) \cdot \Delta t - \delta_{\inf})) \) to the right, see Figure 4 for an illustration.

Maintaining the extreme points of a dynamically changing set is a well studied class of problems in computational geometry known as Dynamic Convex Hull problems. Specific instances from this class differ, depending on the operations to be performed on the set (e.g. insertions, deletions), on the queries to be answered on the extreme points, and on the dimensionality of the input data. Brodal and Jacob (2002) propose a data structure maintaining the convex hull of a finite set of \( N \) points in the plane under insertions and deletions in \( O(\log(N)) \) amortized time. Here, our problem has a very special structure: deletions occur at the left while insertions occur at the right and there is a constant number of points in our set. We rely on this property to come up with a simple algorithm to solve the discretized version of (4) in the case \( k = 1 \). It runs in \( O(|A| \cdot T \Delta t \cdot \log(\frac{\delta_{\sup} - \delta_{\inf}}{\Delta t})) \) worst-case, as opposed to amortized, running time with \( O(|A| \cdot \frac{\delta_{\sup} - \delta_{\inf}}{\Delta t}) \) space usage. Let us now provide the details about this algorithm.

**Adding a point to the right.** This is the easiest part. When deletions are disabled, the dynamic convex hull problem is referred to as Incremental Convex Hull, see Kallay (1984). This special case will prove useful as we never deal with deletions directly. Instead we do some bookkeeping on intermediate results derived when constructing the upper convex hull incrementally. We specialize the key result from Grunbaum et al. (1967) to our case in Proposition 3 assuming that no points are being removed from the upper convex hull. In this situation, points that were not extreme at step \( n \) cannot qualify as such at step \( n + 1 \) due to Lemma 6. On the other hand, some former extreme points might be discarded in this process as illustrated in Figure 11.
**Proposition 3.** If \((\omega^n_{i+1}, v^n_{i+1})\) lies above the straight line connecting \((\omega^n_i, v^n_i)\) and \(((n + 1) \cdot \Delta t, \tilde{u}_j((n + 1) \cdot \Delta t))\), then \((\omega^n_k, v^n_k)_{1 \leq k \leq n}\) are no longer extreme points at step \(n + 1\).

Conversely, if \((\omega^n_{i+1}, v^n_{i+1})\) lies strictly below the straight line connecting \((\omega^n_i, v^n_i)\) and \(((n + 1) \cdot \Delta t, \tilde{u}_j((n + 1) \cdot \Delta t))\), then \((\omega^n_k, v^n_k)_{0 \leq k \leq t+1}\) remain extreme points at step \(n + 1\).

The proof is deferred to the appendix, Section B.3.

Proposition 3 enables us to update the set of extreme points at a cost of \(O(\log(N))\), where \(N\) refers to the cardinality of our set, provided that we have the extreme points stored in a data structure which allows binary search on the first component. Consequently, extreme points of the upper convex hull of \(N\) points can be found in \(O(N \cdot \log(N))\) running time.

**Removing the leftmost point.** This part is tricky as discarding an extreme point might turn a formerly discarded point into an extreme point, see Figure 4b where this happens to be the case for the second leftmost point. This forces us to record some additional information on every point of our dynamic set. As it turns out, we can circumvent the difficulty induced by deletions by appropriately building and merging upper convex hulls of partial input data but this requires an efficient merging procedure. We state without proof a result derived in Overmars and Leeuwen (1981) that we use extensively in the design of the complete algorithm.

**Proposition 4.** Consider a set \(S\) of \(N\) points in the plane partitioned into two sets of points \(S_1\) and \(S_2\) such that for any two points \((x_1, y_1) \in S_1\) and \((x_2, y_2) \in S_2\), \(x_1 < x_2\). Suppose extreme points of \(\hat{\text{conv}}(S_1)\) and \(\hat{\text{conv}}(S_2)\) are stored in a concatenable queue (recall that \(\hat{C}\) refers to the upper convex hull of \(C\) for a set \(C\)). Then, we can obtain a concatenable queue containing the extreme points of \(\hat{\text{conv}}(S)\) in \(O(\log(N))\) time using \(O(N)\) space. Conversely, we can partition the resulting concatenable queue back into the two original concatenable queues in \(O(\log(N))\) time using \(O(N)\) space. For all these queues, the key used to store elements is the first component.

**Proof** See Overmars and Leeuwen (1981). □

Recall that a concatenable queue implements search, insertion, deletion, concatenation and partition in \(O(\log(N))\) computation time while using only \(O(N)\) data space where \(N\) is the number of points stored, see Aho et al. (1974). We are now ready to put forward the complete algorithm.

*Complete algorithm.* We use two concatenable queues \(Q_r\) and \(Q_l\), and a stack \(S\). We first describe the general mechanism broken down into three parts and then bring everything together to get the overall complexity.

At step \(n\), values \(\tilde{u}_j(k \cdot \Delta t), k = n - \frac{\delta_{\text{up}}}{\Delta t}, \ldots, n - \frac{\delta_{\text{inf}}}{\Delta t}\) are available. We incrementally build \(\hat{C}_n\) in descending order of the first coordinate, i.e. starting from the right, using \(Q_l\). This first procedure is symbolized by dash lines in Figure 5. Note that, by Proposition 3, a point \((k \cdot \Delta t, \tilde{u}_j(k \cdot \Delta t))\) can only be pruned once during this incremental procedure. We keep track of this information by pushing a list
Figure 5  Illustration of upper convex hulls’ incremental construction.

consisted of the newly discarded points into the stack \( S \) as soon as they no longer qualify as extreme points. This first part takes \( O(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t} \cdot \log(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t})) \) and we end up with the extreme points of \( \hat{C}_n \) stored in \( Q_l \).

For each of the following \( \tilde{n} = n + 1, \ldots, n + \frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t} \) steps, we begin by deleting the current leftmost point of \( Q_l \). Then, we remove the current topmost list of \( S \) and insert all the points contained in the list retrieved in \( Q_l \). Next, we append the newly available point, namely \( (\tilde{n} \cdot \Delta t, \bar{u}_j(\tilde{n} \cdot \Delta t)) \), to \( Q_r \) as described in Proposition 3. Finally we merge \( Q_l \) and \( Q_r \) to derive \( \hat{C}_{\tilde{n}} \) and find by binary search the optimal solution to (11), see Proposition 2. Next, we split \( \hat{C}_{\tilde{n}} \) up back into \( Q_l \) and \( Q_r \) for the next step. Each of the latter operations takes \( O(\log(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t})) \) computation time and is repeated at most \( \frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t} \) times. Overall, the cost of this second part is \( O(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t} \cdot \log(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t})) \).

At step \( n + \frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t} + 1 \), we restart from scratch and iterate this procedure. Each “loop” takes \( O(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t} \cdot \log(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t})) \) computation time and has to be run \( \lceil \frac{T}{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}} \rceil \) times. Hence the global algorithm runs in \( O(|A| \cdot \frac{T}{\Delta t} \cdot \log(\frac{\delta_{ij}^{\text{sup}} - \delta_{ij}^{\text{inf}}}{\Delta t})) \).

3.3.3. Example. To conclude this section, we come back to Example 2 in order to illustrate the performance of the robust approach. We compare, for the budget \( T = 12 \), the policy solution to (1) with probability distributions illustrated in Figures 6a and 6b taken as inputs and the solution to the robust approach (4) in the case \( k = 1 \) with the knowledge of Figure 6c taken as an input. We

![Figure 6](image-url)  Optimal \( s \rightarrow d \) policies in dash red lines for the budget \( T = 12 \). Comparison of the standard (1) and robust (4) SOTA formulations.
suppose that Figure 6b depicts the real cost uncertainty and that the transition probabilities have been mis-evaluated to those of Figure 6a and we solve the nominal approach (1) with this altered information. Note that, as mentioned in Section 2.3, average arc costs match in Figures 6a, 6b, and 6c. Solving the robust approach in this case is simple as there is a single node for which we need to solve the optimization problem (9), which can be done by hand. The results are gathered in Table 3. First remark that the nominal approach overestimates both the optimal and actual probability of termination within budget, as $\frac{3}{4}$ instead of $\frac{1}{4}$ and $\frac{2}{3}$ instead of $\frac{1}{3}$, respectively. Second, the robust approach guarantees a higher probability of reaching $d$ within budget than the actual probability of doing so when the policy derived from the nominal formulation is followed ($\frac{5}{7}$ against $\frac{2}{3}$).

3.4. Case $k > 1$

In the case $k = 1$, we are able to numerically solve the general discretization scheme of Definition 11 for any valid objective, as identified in Theorem 11 in time almost linear in $\frac{T \Delta t}{\Delta t} (O(|A| \cdot \frac{T \Delta t}{\Delta t} \cdot \log(\frac{T}{\Delta t}))$ to be precise). For $k \geq 2$, this corresponds to solving a large number of semidefinite programs as pointed out in Theorem 3 and the resulting procedure is likely to be intractable. To mitigate the computational burden, we propose to also discretize the support of the unknown random variables $(c_{ij})_{(i,j) \in A}$. Although convergence to the continuous solution guaranteed by Theorem 2 may no longer hold, this enables us to cast (7) as a linear programming problem. Remark that these two approaches end up being equivalent in the case $k = 1$ as a byproduct of (11) (the linear programs with finitely or infinitely many constraints turn out to be equivalent). Building on the results of Prékopa (1990), we design an efficient procedure of this simplified approach for a restricted class of objectives.

Consider an arc $(i,j)$, and assume that $k \in \mathbb{N}$ moments, $m^1_{ij} = \mathbb{E}(c_{ij}), \ldots$, and $m^k_{ij} = \mathbb{E}(c^k_{ij})$ are available. To simplify the presentation, we omit the superscripts $k$ and $\Delta t$ in $u^k_{ij} \Delta t$ and set $\Delta t = 1$.

**Discrete linear formulation.** At step $n$, we propose to restrict the uncertainty set $P^k_{ij}$ to probability distributions with discrete support, specifically $\text{supp}(p) \subset [\delta^\text{inf}_{ij}, \delta^\text{sup}_{ij}] \cap \mathbb{N}$. This turns (7) into a linear program:

| Table 3 |
| --- |
| Compared performance of the standard and robust approaches. The actual uncertainty is depicted in Figure 6b. |
| Policy | Probability outputted | Actual probability |
| Standard SOTA on Figure 6b | s → b → d | $\frac{3}{4}$ | $\frac{3}{4}$ |
| Standard SOTA on Figure 6a | s → a → d | 1 | $\frac{2}{3}$ |
| Robust SOTA on Figure 6c | s → b → d | $\frac{7}{9}$ | $\frac{7}{9}$ |
\[ \inf \quad u_j(n - \delta_{ij}^{\text{inf}}) \cdot p(\delta_{ij}^{\text{inf}}) + \cdots + u_j(n - \delta_{ij}^{\text{sup}}) \cdot p(\delta_{ij}^{\text{sup}}) \]

subject to

\[ p(\delta_{ij}^{\text{inf}}) + \cdots + p(\delta_{ij}^{\text{sup}}) = 1 \]

\[ \delta_{ij}^{\text{inf}} \cdot p(\delta_{ij}^{\text{inf}}) + \cdots + \delta_{ij}^{\text{sup}} \cdot p(\delta_{ij}^{\text{sup}}) = m_{ij}^1 \]

\[ \vdots \]

\[ (\delta_{ij}^{\text{inf}})^k \cdot p(\delta_{ij}^{\text{inf}}) + \cdots + (\delta_{ij}^{\text{sup}})^k \cdot p(\delta_{ij}^{\text{sup}}) = m_{ij}^k \]

\[ p(\delta_{ij}^{\text{inf}}) \geq 0, \cdots, p(\delta_{ij}^{\text{sup}}) \geq 0 \]  \hspace{1cm} (12)

(12) is a special case of the general class of discrete moment problems investigated in Prékopa (1990) of the following form:

\[ \inf \quad f_0 \cdot p(z_0) + \cdots + f_n \cdot p(z_n) \]

subject to

\[ p(z_0) + \cdots + p(z_n) = 1 \]

\[ z_0 \cdot p(z_0) + \cdots + z_n \cdot p(z_n) = m^1 \]

\[ \vdots \]

\[ z_0^k \cdot p(z_0) + \cdots + z_n^k \cdot p(z_n) = m^k \]

\[ p(z_0) \geq 0, \cdots, p(z_n) \geq 0 \]  \hspace{1cm} (13)

where \{z_0, \cdots, z_n\} and \(m^1, \cdots, m^k\) denote the support and the first \(k\) moments of the unknown probability mass function \(p\), respectively. \(f : n \to f_n\) is a given discrete function, hence the objective function corresponds precisely to \(E_p(f(X))\). Prékopa (1990) investigates optimality conditions and shows that the set of all dual feasible basis can be described in closed form provided \(f(\cdot)\) is either \((k+1)\)th order convex or concave.

**Definition 6.** Prékopa (1990)

For a function \(f(\cdot)\) defined on the discrete set \{z_0, \cdots, z_n\}, the first order divided differences of \(f(\cdot)\) are defined by

\[ [z_i, z_{i+1}]f = \frac{f(z_{i+1}) - f(z_i)}{z_{i+1} - z_i}, \quad i = 0, \cdots, n - 1 \]

The \(m\)th order divided differences of \(f\) are recursively defined by:

\[ [z_i, \cdots, z_{i+m}]f = \frac{[z_{i+1}, \cdots, z_{i+m}]f - [z_i, \cdots, z_{i+m-1}]f}{z_{i+m} - z_i}, \quad m \geq 2 \]

\(f\) is said to be \(m\)th order convex (resp. concave) if all its \(m\)th order divided differences are positive (resp. negative).

The following result due to Prékopa (1990) will be useful for our purpose.
Proposition 5. Prékopa (1990)
There exists a probability mass function, \((p_0^*, \cdots, p_n^*)\), satisfying the moment constraints of (13) such that \((p_0^*, \cdots, p_n^*)\) is an optimal solution to (13) for any \((k+1)\)th order convex or concave function \(f(\cdot)\).

Algorithm for \((k+1)\)th order convex/concave objective. The main result of this section is that, quite surprisingly, the property of being \((k+1)\)th order convex/concave carries through from \(u_j(\cdot)\) to the optimal value of (12) viewed as a function of \(n\). This is a consequence of the strong property stated in Proposition 5: the optimal solution to (13) does not depend on the function \(f(\cdot)\) and in particular we get the same optimal solution for any translate of \(f(\cdot)\), \(f(\cdot - n)\) for any \(n\), as long as \(f(\cdot)\) is globally higher order convex/concave.

Proposition 6. Define \(g\) as the discrete function mapping \(n \in \{0, \cdots, T\}\) to the value of the linear program (12). If \(\tilde{u}_j(\cdot)\) is \((k+1)\)th convex (resp. concave) on \(\{l_1 - \delta^{\sup}, \cdots, l_2 - \delta^{\inf}\}\) with \(l_2 > l_1 \in \mathbb{N}\), then so is \(g\) on \(\{l_1, \cdots, l_2\}\).

The proof is deferred to the appendix, Section B.3.

With Proposition 6 if the initial objective \(f(\cdot)\) was \((k+1)\)th order convex/concave then we would expect any intermediate function \(\tilde{u}_i(\cdot)\) to have the same property. This would imply that we can solve the robust approach in \(O(|A| \cdot \frac{T}{\Delta t} \cdot (\log^2 \frac{T}{\Delta t}) - \log^2 \frac{\delta^{\inf}}{2\Delta t}))\) running time (complexity of the nominal problem). Indeed, an optimal solution to (12) could be derived in closed form and would remain the same at every step \(n\) as a result of Proposition 5. Hence (4) could be solved in the same fashion as the nominal problem. However, the complete dynamic programming equation is

\[
\tilde{u}_i(n) = \max_{j \in V(i)} \inf_{p \in \mathcal{P}(\tilde{u}_j(n - X))} \mathbb{E}_p(\tilde{u}_j(n - X))
\]

and the maximum over \(V(i)\) may cause \(\tilde{u}_i(\cdot)\) not to be \((k+1)\)th order convex/concave. In the vicinity of a breakpoint, there is no guarantee that the property will be carried through to \(\tilde{u}_i(\cdot)\) even if \(\tilde{u}_j(\cdot), j \in V(i)\) all are \((k+1)\)th order convex/concave. Nevertheless, this is a local property, precisely we need \(\tilde{u}_j(\cdot)\) to be \((k+1)\)th order convex/concave on \(\{n - \delta^{\sup}, \cdots, n - \delta^{\inf}\}\) at step \(n\). Hence, if there are few breakpoints, it is reasonable to believe that, in most cases, the property will hold. For this reason, we propose the following algorithm to solve (13). At step \(n\),

1. Check if \(\tilde{u}_j(\cdot)\) is \((k+1)\)th order convex/concave on \(\{n - \delta^{\sup}, \cdots, n - \delta^{\inf}\}\). This can be done in constant time by additional bookkeeping as we are checking this property at every step;

2. If the property holds, solve (12) for all \(j \in V(i)\) using the optimal solutions derived in a preprocessing step. Then, take the maximum value;
3. Otherwise solve (12) via the primal simplex algorithm. Observe that the constraints remain the same throughout the algorithm. This enables us to efficiently warm start the linear program with the solution found at the previous step.

In special cases, this algorithm has the same complexity as the nominal one developed in Samaranayake et al. (2012). We believe that this algorithm is particularly fast in practice when the objective function is higher order convex/concave.

**Proposition 7.** If there is a single path from \( s \) to \( d \) in \( G \) and the objective \( f(\cdot) \) is \((k+1)\)th order convex, then the algorithm runs in \( O(|A| \cdot \frac{T}{\Delta t} \cdot (\log^2 (\frac{T}{\Delta t}) - \log^2 (\frac{\delta_{inf}}{\Delta t}))) \) computation time. Precisely, we can compute the worst-case objective in time \( O(|A| \cdot \frac{T}{\Delta t} \cdot (\log^2 (\frac{T}{\Delta t}) - \log^2 (\frac{\delta_{inf}}{\Delta t}))) \) (only the bound is relevant here as there is a single itinerary). Additionally, this bound is tight in the sense that there exists a set of probability distributions \((p_{ij})_{(i,j) \in A}\) such that the worst-case objective retrieved corresponds to \( u_s(T) \) derived from (3) with \((p_{ij})_{(i,j) \in A}\) plugged in.

**Proof** In this case there is no breakpoint as each relevant node \( i \) has only one successor and we can apply Lemma 6. The fact that the bound derived is tight comes from Proposition 5. \( \square \)

A typical example of objective function that satisfies both the assumptions of Theorem 1 and is also 3th order concave is \( f(t) = -|t|^3 \). This corresponds to a willingness to arrive exactly on-time, penalizing both late and early arrivals.

**Fast heuristic to solve (4)** Proposition 6 suggests the following heuristic procedure to solve (4). Instead of solving to optimality, we evaluate an upper bound given by the feasible probability distribution identified in Proposition 6. The latter is link-dependent but remains the same at every step \( n \). In general, we are not guaranteed to solve to optimality but this procedure is fast and correct in the cases highlighted in Proposition 7.

### 4. Numerical experiments

In this section, we compare, on a real-world application with field data from the Singapore road network, the performance of the nominal and robust approaches as routing procedures when traffic measurements are scarce and uncertain.

We point out that, in general, (4) could be too robust in the sense that there may exist a strategy such that, for any probability distributions in the uncertainty set, \((p_{ij})_{(i,j) \in A} \in \prod_{(i,j) \in A} P^{k}_{ij}\), the probability of arriving on time for this strategy under \((p_{ij})_{(i,j) \in A}\) is strictly greater than the worst-case probability of on-time arrival computed by solving (4). The reason is that, for a given link \((i,j) \in A\), the probability distribution solution to the inner problem is a function of the remaining time budget at \( i \). Yet one can reach this intermediate location with various remaining budgets on the way to \( d \). Hence, we can be too robust since we assume the worst for every potential intermediate time of arrival to an intermediate location allowing the probability distribution on each link to depend on
the current time \( t \). As a consequence, we need to further investigate the practical performance of optimal robust strategies computed from (4). To benchmark the performance of the robust approach, we propose a realistic framework where both the nominal and robust approaches could be run and for which it is up to the user to pick one.

### 4.1. Framework

**Data.** We work on a network composed of the main roads of Singapore with 20221 arcs and 11018 nodes for a total length of 1131 kilometers of roads. To evaluate the performance of the robust approach, we need a source and a destination which we choose to be respectively \( s = \text{"Woodlands avenue 2"} \) and \( d = \text{"Mandai link"} \), see Figure 7. These locations are good candidates since there are at least three reasonable alternative paths to get from \( s \) to \( d \) with similar travel times. Therefore, the best driving itinerary depends on the actual traffic conditions. The data consists of a 15-day recording of GPS probe vehicle speed samples coming from a combined fleet of over 15,000 taxis. Features of each recording include current location, speed and status (free, waiting for a customer, occupied).

**Method of performance evaluation.** Consider the following real-world situation. A user has to find an itinerary to get from \( s \) to \( d \) with a given budget \( T \) (the deadline) and objective function \( f(\cdot) \), which we will take to be the probability of on-time arrival, with the additional difficulty that only a few vehicle speed samples are available.

To model this real-world situation, we consider that the samples of vehicle speed measurement available in our dataset represent the real traffic conditions characterized by the travel-time distributions \( p_{ij}^{\text{real}} \)'s, which we can easily compute. In our setting, the \( p_{ij}^{\text{real}} \)'s are not available, instead only a fraction of the full set of samples, say \( \lambda \in [0, 1] \), is available. Based on this limited data, the challenge is to select an itinerary with a probability of on-time arrival with respect to the real traffic conditions \( p_{ij}^{\text{real}} \)'s as high as possible. We propose to use the methods listed in Table 4 to choose such an itinerary. For each of these methods, the process goes as follows:
| Method                          | Travel-time parameters to estimate from samples | Approach                      |
|--------------------------------|------------------------------------------------|-------------------------------|
| adaptive robust $k = 1$       | $\delta_{ij}^{\text{inf}}, \delta_{ij}^{\text{sup}}, m_{ij}^1$ | $\text{LET}$ with $k = 1$   |
| adaptive robust $k = 2$       | $\delta_{ij}^{\text{inf}}, \delta_{ij}^{\text{sup}}, m_{ij}^1, m_{ij}^2$ | $\text{LET}$ with $k = 2$   |
| adaptive robust $k = 2$, heuristic | $\delta_{ij}^{\text{inf}}, \delta_{ij}^{\text{sup}}, m_{ij}^1, m_{ij}^2$ | $\text{LET}$ with $k = 2$   |
| adaptive nominal              | empirical distributions $p_{ij}$               | $\text{LET}$                 |
| LET                            | $m_{ij}^1$                                     | Standard shortest path       |

1. Estimate the arc-based travel-time parameters required to run the method using the fraction of data available.
2. Run the corresponding algorithm to find an itinerary, depending on the chosen method.
3. Compute the probability of on-time arrival of this itinerary for the real traffic conditions ($\lambda = 1$).

The result obtained depends on both $\lambda$ and the available samples as there are many ways to pick a fraction $\lambda$ out of the entire dataset. Hence, for each $\lambda$ in a set $\Lambda$, we randomly pick $\lambda \cdot N_{ij}$ samples for each arc $(i, j)$, where $N_{ij}$ is the number of samples collected in the entire dataset for that particular arc. For each $\lambda \in \Lambda$, and for each method, we store the calculated probability of on-time arrival. We repeat this procedure 100 times.

A few remarks are in order. We choose $\Lambda = \{0.0002, 0.001, 0.005\}$, this corresponds to an average number of samples per arc of $[1.4, 5.5, 25.1]$ respectively (we take at least one sample per arc). Arcs are 50-meter long, hence we set $\Delta t = 0.01$ second to get a good accuracy. This parameter has a significant impact on the running time and it could also be optimized. For any of the methods listed in Table 4, we extend the policy obtained by the least expected path (LET) when no policy is provided. This can occur if the probability of on-time arrival, either estimated or guaranteed, is zero, see the discussion in Section 2.2. The method labeled “adaptive robust $k = 2$” solves $\text{LET}$ with $k = 2$ to optimality while “adaptive robust $k = 2$, heuristic” is not guaranteed to solve $\text{LET}$ to optimality.

For more details on these two approaches, we refer to Section 3.4. To solve the “adaptive nominal” method, we use the Fast Fourier Transform mechanism proposed in Samaranavake et al. (2011). We include the LET approach as it is a reasonably robust approach, although not tailored to the objective function considered, and because it is very fast to solve.

4.2. Results

Probability of on-time arrival. The results are plotted in Figure 8, 9, and 10. Each of these figures corresponds to one of the fraction $\lambda \in \Lambda$ so as to see the impact of an increasing knowledge. The time budget is “normalized”: 0 (resp. 1) corresponds to the minimum (resp. maximum) amount of time it takes to reach $d$ from $s$. For each $\lambda$, for each method in Table 4, for each time budget $T$, and for each
of the 100 simulations, we compute the actual probability of on-time arrival of the corresponding strategy. The average (resp. 5 % worst-case) probability of on-time arrival over the simulations is plotted on the figures labeled “a” (resp. “b”). The 5 % worst-case measure, which corresponds to the average over the 5 simulations out 100 that yield the lowest probability of arriving on-time, is particularly relevant as commuters opting for this objective function would expect the approach to have good results even under bad scenarios.

Runtime. We plot the average runtime for each of the method as a function of the time budget in Figure 11.

Conclusions. As can be observed on the figures, the adaptive nominal method is not competitive when only a few samples are available. To be specific, LET and the adaptive robust \( k = 1 \) strategies outperform the adaptive nominal method when there are very few measurements, see Figure 8, while both the adaptive robust \( k = 2 \) and the heuristic outperform it for more samples, see Figures 9 and
Furthermore, the itinerary retrieved by the nominal approach may have a probability of on-time arrival up to two times lower than the optimal strategy. However, the performance of the nominal method improves as more samples get available, as expected.

For very few measurements, the adaptive robust \( k = 1 \) method outperforms the other methods. Strictly speaking, the adaptive robust \( k = 2 \) method performs better than all other methods for more samples (although observe that the heuristic method is twenty times as fast as the latter for similar performances). Our interpretation is that relying on quantities, either moments or distributions, that cannot be accurately estimated may be misleading even for robust strategies. On the other hand, failure to capture the increasing knowledge on the actual travel-time probability distributions (e.g. by
estimating more moments) as the amount of available data increases may lead to poor performances, as shown in Figure 10 for the adaptive robust $k = 1$ strategy.

As a final interpretation of the results, it looks like that, on traffic networks, path providers need not be very robust whenever there are usually only a few reasonable itineraries between an origin and a destination. The set of reasonable policies then consists of a few paths and slight variations around these.

**Acknowledgments**

This research is supported in part by the National Research Foundation (NRF) Singapore through the Singapore MIT Alliance for Research and Technology (SMART) and its Future Urban Mobility (FM) Interdisciplinary Research Group. The authors would like to thank Chong Yang Goh from the Massachusetts Institute of Technology for his help in preprocessing the data. We also thank Laurent El Ghaoui for insightful discussions.

**References**

Adulyasak, Y., P. Jaillet. 2014. Models and algorithms for stochastic and robust vehicle routing with deadlines. Working paper.

Aho, A. V., J. E. Hopcroft, J. D. Ullman. 1974. *Design and Analysis of Computer Algorithms*, vol. 1. Addison-Wesley, Boston, MA.

Bertsekas, D. P., J. Tsitsiklis. 1991. An analysis of stochastic shortest path problems. *Math. Oper. Res.* 16(3) 580–595.

Bertsimas, D., I. Popescu. 2005. Optimal inequalities in probability theory: A convex optimization approach. *SIAM J. Optim.* 15(3) 780–804.

Brodal, G. S., R. Jacob. 2002. Dynamic planar convex hull. *Proc. 43rd IEEE Annual Symp. Foundations of Comput. Sci.* IEEE, 617–626.

Frank, H. 1969. Shortest paths in probabilistic graphs. *Oper. Res.* 17(4) 583–599.

Gabrel, V., C. Murat, L. Wu. 2013. New models for the robust shortest path problem: complexity, resolution and generalization. *Annals of Oper. Res.* 207 97–120.

Grunbaum, B., V. Klee, M. A. Perles, G. C. Shephard. 1967. *Convex polytopes*, vol. 2. John Wiley, New York.

Isii, K. 1960. The extrema of probability determined by generalized moments (i) bounded random variables. *Ann. Inst. Statist. Math.* 12(2) 119–134.

Jaillet, P., J. Qi, M. Sim. 2013. Routing optimization with deadlines under uncertainty. Working paper.

Kallay, M. 1984. The complexity of incremental convex hull algorithms in $\mathbb{R}^d$. *Information Processing Letters* 19(4) 197.
Karlin, S., L. S. Shapley. 1953. Geometry of moment spaces. *Memoirs AMS* 12.

Kemperman, J. H. B. 1987. Geometry of the moment problem. *Proc. Symposia Appl. Math.*, vol. 37. 16–53.

Loui, R. P. 1983. Optimal paths in graphs with stochastic or multidimensional weights. *Comm. ACM* 26(9) 670–676.

Miller-Hooks, E. D., H. S. Mahmassani. 2000. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Sci.* 34(2) 198–215.

Nie, Y., Y. Fan. 2006. Arriving-on-time problem: discrete algorithm that ensures convergence. *Transportation Res. Record* 1964 193–200.

Nie, Y., X. Wu. 2009. Shortest path problem considering on-time arrival probability. *Transportation Res. B* 43(6) 597–613.

Nikolova, E., M. Brand, D. R. Karger. 2006a. Optimal route planning under uncertainty. *Proc. Internat. Conf. Automated Planning Scheduling*. 131–140.

Nikolova, E., J. A. Kelner, M. Brand, M. Mitzenmacher. 2006b. Stochastic shortest paths via quasi-convex maximization. *Proc. 14th Annual Eur. Symp. Algorithms (ESA’06)*, vol. 14. Springer Berlin Heidelberg, 552–563.

Nilim, A., L. El Ghaoui. 2005. Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.* 53(5) 780–798.

Overmars, M. H., J. Van Leeuwen. 1981. Maintenance of configurations in the plane. *J. Comput. and System Sci.* 23(2) 166–204.

Prékopa, A. 1990. Sharp bounds on probabilities using linear programming. *Oper. Res.* 38(2) 227–239.

Samaranayake, S., S. Blandin, A. Bayen. 2011. A tractable class of algorithms for reliable routing in stochastic networks. *19th Internat. Sympos. Transportation Traffic Theory (ISTTT 2011)* 17 341–363.

Samaranayake, S., S. Blandin, A. Bayen. 2012. Speedup techniques for the stochastic on-time arrival problem. *12th Workshop Algorithmic Approaches Transportation Model. Optim. Systems (ATMOS 2012)*, vol. 25. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 83–96.

Satia, J. K., R. E. Lave. 1973. Markovian decision processes with uncertain transition probabilities. *Oper. Res.* 21(3) 728–740.

Shapiro, A. 2001. On duality theory of conic linear problems. *Semi-Infinite Programming*, vol. 57. Kluwer Academic Publishers, 135–165.

Smith, J. E. 1995. Generalized Chebyschev inequalities: theory and applications in decision analysis. *Oper. Res.* 43(5) 807–825.

Vandenberghhe, L., S. Boyd, K. Comanor. 2007. Generalized Chebyschev bounds via semidefinite programming. *SIAM Rev.* 49(1) 52–64.
White III, C. C., H. K. Eldeib. 1994. Markov decision processes with imprecise transition probabilities. *Oper. Res.* 42(4) 739–749.

Xu, H., C. Caramanis, S. Mannor. 2012. Optimization under probabilistic envelope constraints. *Oper. Res.* 60(3) 682–699.

Xu, H., S. Mannor. 2011. Probabilistic goal Markov decision processes. *Proc. 22th Internat. Joint Conf. Artificial Intelligence (AAAI).* AAAI Press, 2046–2052.
APPENDICES

Appendix A: Extensions

Speed-up technique for $k = 1$. From a theoretical standpoint, the algorithm designed in Section 3.3.2 to keep track of the dynamic convex hull has time complexity similar to the nominal problem. However, it can be slow in practice as the convex hull has to be recomputed from scratch many times. To get a rough order of magnitude take $\Delta t = 0.01s$, $\delta_{ij}^{\text{inf}} = 1$, $\delta_{ij}^{\text{sup}} = 1s$ and $T = 600s$. With these parameters, we have to rebuild the upper convex hull of a 100-point set from scratch 600 times per arc. A simple faster heuristic consists in using a single list as opposed to two concatenable queues. Specifically, at each step, we continue to append the newly available point at the right but prune the leftmost point only if the second leftmost point has its $x$-coordinate no greater than $n\Delta t - \delta_{ij}^{\text{sup}}$. Hence we solve a more constrained version of (10) for which $\omega$ ranges in a larger interval $I_n$, $\min(I_n) \leq n \cdot \Delta t - \delta_{ij}^{\text{sup}}$ and $\max(I_n) = n \cdot \Delta t - \delta_{ij}^{\text{inf}}$. In particular, the bound derived is still a valid minimum guarantee. Observe that if $\bar{u}_j(\cdot)$ is affine on $I_n$ the bound is the same.

Appendix B: Omitted Proofs

B.1. Proof of Theorem 1

Proof of Theorem 1. Define for all nodes $i \in V$ and budgets $t \leq T$, the random variable $X_i^t$ as the cost of reaching $d$ starting from $i$ with an initial budget $t$ if making decisions in accordance with the optimal policy solution to (1), which we refer to as $f$-policy. Similarly, let $Y_i^t$ be the random cost associated with the least expected path which simply consists of following the shortest $i \to d$ path on the basis of expected costs. We show that these two strategies are essentially identical for large budget overrun. Consider $T_1$ large enough such that $f(\cdot)$ is increasing, concave and $C^2$ on $(-\infty, -T_1)$. For $i \in V$ and $t < -T_1$, we have:

$$f(t - X_i^t) = f(t) - f'(t) \cdot X_i^t + \frac{1}{2} \cdot f''(\xi_i^t) \cdot (X_i^t)^2,$$

where $\xi_i^t \in [t - X_i^t, t]$; and

$$f(t - Y_i^t) = f(t) - f'(t) \cdot Y_i^t + \frac{1}{2} \cdot f''(\xi_i^t) \cdot (Y_i^t)^2,$$

where $\xi_i^t \in [t - Y_i^t, t]$. By optimality, we obtain $E[f(t - X_i^t)] \geq E[f(t - Y_i^t)]$. Expanding and rearranging yields:

$$-f'(t) \cdot (E[X_i^t] - E[Y_i^t]) \geq \frac{1}{2} \cdot (E[-f''(\xi_i^t) \cdot (X_i^t)^2] + E[f''(\xi_i^t) \cdot (Y_i^t)^2]).$$

Concavity of $f(\cdot)$ implies that $f''(\xi_i^t) \cdot (X_i^t)^2 \leq 0$ almost surely. Since $f(\cdot)(-\infty, -T_1)$ is increasing, we obtain $0 \leq E[X_i^t] - E[Y_i^t] \leq \frac{E[-f''(\xi_i^t) \cdot (X_i^t)^2]}{2f'(t)}$. As $Y_i^t$ is the cost of a path, Assumption 1 implies $0 \leq Y_i^t \leq \alpha = \sum_{(i,j) \in A} \delta_{ij}^{\text{sup}} < \infty$ and as a byproduct $\xi_i^t \in [t - \alpha, t]$. It follows that:

$$0 \leq E[X_i^t] - E[Y_i^t] \leq \frac{E[(Y_i^t)^2]}{2f'(t)} \cdot \inf_{[t - \alpha, t]} f''.$$
Consequently, the above inequality imposes $f$-policy cost, the optimal strategy among all history-dependent rules is to follow the least expected path. When leaving node $i$ with remaining budget $t \leq \min(-T_1, -T_2)$, the decision maker picks an outgoing edge $(i, j)$. Even though the overall policy can be fairly complicated (history-dependent), the first action is deterministic and incurs an expected cost of $m_{ij}$. When the objective is to minimize the average cost, the optimal strategy among all history-dependent rules is to follow the least expected path.

Consequently, the above inequality imposes $f$-policy to choose the same outgoing edge as if we were following the least expected $i \to d$ path. This remark applies to all nodes and for all smaller budgets.

An immediate consequence is that for $t \leq \min(-T_1, -T_2)$, $f$-policy does not include any cycles and $X_i^t$ is bounded by $\alpha$. As $f$-policy corresponds to the least expected path, this implies an ordering of the nodes to initialize the collection of functions $(u_i(\cdot))_{i \in V}$ on $[\min(-T_1, -T_2) - \delta_{\sup}, \min(-T_1, -T_2)]$ (the starting node being $d$). Specifically, we build a spanning tree of $G$, denoted by $\tilde{G}$, as follows: for each node $i \in V$ we find the least expected cost path to go from $i$ to $d$ and we add to $\tilde{G}$ an arc joining $i$ and the next node to visit along this path. By the Bellman principle of optimality, $\tilde{G}$ is a spanning tree with root $d$. It is now equivalent to solve (3) on $\tilde{G}$ and $\tilde{G}$ for $\ell \leq \min(-T_1, -T_2)$ and the ordering is given by the heights of the nodes in $\tilde{G}$ (ties are broken arbitrarily). Functions $u_i(\cdot)$ can subsequently be computed using the incremental procedure outlined in Section 2.4.

**B.2. Proof of results involved in the construction of the general robust formulation**

We prove Lemma 1 and 3 that ensure existence and uniqueness of a solution to the continuous robust formulation (4) and its discretized counterpart. These results are proved by induction and the base case turns out to be straightforward for the objective considered (minimizing the probability of budget overrun). These results still hold for the objective functions identified in Theorem 1 but the base case is more involved as we have to use the tree constructed in the proof of Theorem 1. We can adapt the proof as follows: observe that for any number of moments $k \in \mathbb{N}$, we have access to the arc mean costs $(m_{ij})$ which is all we need to identify the optimal strategy. Therefore the optimal robust policy and the optimal nominal policy match for $t \leq \min(-T_1, -T_2)$ (as defined in the proof of Theorem 1). We can use the ordering given by the tree constructed in the proof to initialize the induction by proving that the properties pass along from a node to its children in the tree (the root being $d$ and $u_d(t) = f(t)$ which satisfies the properties). Hence we circumvent the difficulty caused by the potential existence of cycles.

*Proof of Lemma 1* The base case is straightforward as $\tilde{u}_i^k(\cdot)$ is well defined and non-decreasing on $(-\infty, \delta_{\inf})$, $\forall i \in V$ (recall that $\tilde{u}_i^k(t) = 0$ for $t \leq \delta_{\inf}$ and $i \in V - \{d\}$). Suppose that $\tilde{u}_i^k(\cdot)$ is well defined and non-decreasing on $(-\infty, n \cdot \delta_{\inf})$, $\forall i \in V$, with $n \in \mathbb{N}$ and consider $i \in V$. Remark that for
all $j \in V(i)$ and for all $p_{ij} \in P^k_{ij}$, $p_{ij}$ has compact support contained in $[\delta_{\inf}, \delta_{\sup}]$ which implies that $t \to \mathbb{E}_{p_{ij}}[\hat{u}_j^k(t - c_{ij})]$ is well defined on $(-\infty, (n+1) \cdot \delta_{\inf}]$. As a consequence and since $P^k_{ij}$ is not empty, $t \to \inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[\hat{u}_j^k(t - c_{ij})]$ is well defined on $(-\infty, (n+1) \cdot \delta_{\inf}]$. This holds true for all $j \in V(i)$, thus $\hat{u}_j^k(\cdot)$ is also well defined on $(-\infty, (n+1) \cdot \delta_{\inf}]$. Now take $t, \bar{t} \in [0, (n+1) \cdot \delta_{\inf}]$ with $t \geq \bar{t}$, $j \in V(i)$ and $p_{ij} \in P^k_{ij}$. As $\hat{u}_j^k(\cdot)$ is non-decreasing on $(-\infty, n \cdot \delta_{\inf}]$, we have:

$$\mathbb{E}_{p_{ij}}[\hat{u}_j^k(t - c_{ij})] \geq \mathbb{E}_{p_{ij}}[\hat{u}_j^k(\bar{t} - c_{ij})].$$

Taking the infimum over $p_{ij}$ and the maximum over $j$ yields $\hat{u}_j^k(t) \geq \hat{u}_j^k(\bar{t})$. Therefore, we conclude that $\hat{u}_j^k(\cdot)$ is non-decreasing on $(-\infty, (n+1) \cdot \delta_{\inf}]$. Iterating over $n$ we get the claim.  

**Proof of Lemma 3** By construction, $\hat{u}_i^{k,\Delta t}(\cdot)$ is a lower semi-continuous piecewise constant function. The base case is straightforward as $\hat{u}_i^{k,\Delta t}(\cdot)$ is non-decreasing on $(-\infty, \Delta t]$, $\forall i \in V$ (recall that $\hat{u}_i^{k,\Delta t}(t) = 0$ for $t \leq \Delta t$ and for $i \in V \setminus \{d\}$). Suppose that for some $n \in \mathbb{N}$, $\hat{u}_i^{k,\Delta t}(\cdot)$ is non-decreasing on $(-\infty, n \cdot \Delta t]$, $\forall i \in V$. Consider $i \in V$. We have:

$$\mathbb{E}_{p_{ij}}[\hat{u}_i^{k,\Delta t}(n \Delta t - c_{ij})] \geq \mathbb{E}_{p_{ij}}[\hat{u}_i^{k,\Delta t}((n - 1) \Delta t - c_{ij})],$$

$\forall j \in V(i), \forall p_{ij} \in P^k_{ij}$. Thereby, we get:

$$\inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[\hat{u}_i^{k,\Delta t}(n \Delta t - c_{ij})] \geq \inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[\hat{u}_i^{k,\Delta t}((n - 1) \Delta t - c_{ij})].$$

This further implies:

$$\hat{v}_i^{n+1} = \max_{j \in V(i)} \inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[\hat{u}_i^{k,\Delta t}(n \Delta t - c_{ij})] \geq \max_{j \in V(i)} \inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[\hat{u}_j^{k,\Delta t}((n - 1) \Delta t - c_{ij})] = \hat{v}_i^n.$$

This precisely means that $\hat{u}_i^{k,\Delta t}(\cdot)$ is non-decreasing on $(-\infty, (n+1) \cdot \Delta t]$. We conclude by induction on $n$.  

**B.3. Proof of results from Section 3.3 and 3.4**

**Proof of Proposition 2** We first prove that the optimal value is given by a straight line joining two consecutive extreme points. Lines joining two non-consecutive extreme points do not form feasible solutions to (11) since such a straight line passes above any extreme point located (first coordinate-wise) in between these two extreme points (otherwise they would not qualify as extreme points according to Lemma 5). Hence, straight lines joining two consecutive extreme points are the only basic feasible solutions to (11). 

We now prove that the optimal straight line is the one joining the two consecutive extreme points satisfying $\omega^n_i \leq n \cdot \Delta t - m_{ij} \leq \omega^n_{i+1}$. Take $l$ such that $\omega^n_l \leq n \cdot \Delta t - m_{ij} \leq \omega^n_{l+1}$. Note that the objective value is the value of the straight line at $n \cdot \Delta t - m_{ij}$. As straight lines joining two consecutive extreme points constitute feasible solutions, they all remain below the straight line $((\omega^n_l, v^n_l), (\omega^n_{l+1}, v^n_{l+1}))$ on the interval $[\omega^n_l, \omega^n_{l+1}]$ and thus have lower objective values than $((\omega^n_i, v^n_i), (\omega^n_{i+1}, v^n_{i+1}))$.  

Proof of Proposition 6 We begin by proving the first claim. Let us use the shorthand $\Omega_{n+1} = (n+1) \cdot \Delta t$ and $V_{n+1} = \bar{u}_j(\Omega_{n+1})$. If $(\omega_t^n, v_t^n)$ lies above $((\omega_t^n, v_t^n), \Omega_{n+1}, V_{n+1})$, then using Lemma 6 $(\omega_t^n, v_t^n)$ is no longer an extreme point. Additionally, restating the hypothesis, $(\omega_t^n, v_t^n)$ necessarily lies below $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$. This means that $((\omega_t^n, v_t^n), (\omega_t^{n+1}, v_t^{n+1}))$ is located above $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$ on $[\omega_t^n, \Omega_{n+1}]$ (since these two lines intersect at $(\omega_t^n, v_t^n)$). Yet $(\omega_t^{n+1}, v_t^{n+1})$ lies above the line $((\omega_t^n, v_t^n), (\omega_t^{n+1}, v_t^{n+1}))$, otherwise $(\omega_t^{n+1}, v_t^{n+1})$ would not qualify as extreme point of $\hat{C}_n$ by Lemma 6. Hence $(\omega_t^{n+1}, v_t^{n+1})$ lies above $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$. This is the exact same property as stated in the hypothesis but for the points $(\omega_t^n, v_t^n)$ and $(\omega_t^{n+1}, v_t^{n+1})$. Proceeding recursively, we can prove the first claim.

We now establish the second statement. Suppose $(\omega_t^{n+1}, v_t^{n+1})$ lies strictly below $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$. According to Lemma 6 in order to show that $(\omega_t^n, v_t^n)$ remains an extreme point, it is sufficient to prove that all straight lines of the form $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$ pass strictly above $(\omega_t^n, v_t^n)$, $(\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1})$ lies strictly above $(\omega_t^n, v_t^n)$. This means that $(\omega_t^n, v_t^n)$ is located strictly above $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$ on $[0, \omega_t^n]$ (since these two lines cross at $(\omega_t^n, v_t^n)$). Yet $(\omega_t^n, v_t^n)$ lies strictly above the line $((\omega_t^n, v_t^n), (\omega_t^{n+1}, v_t^{n+1}))$, since $(\omega_t^n, v_t^n)$ is an extreme point of $\hat{C}_n$. This implies that $((\omega_t^{n-1}, v_t^{n-1}), (\Omega_{n+1}, V_{n+1}))$ is located strictly above $(\omega_t^n, v_t^n)$ and thus passes strictly above $(\omega_t^n, v_t^n)$. We can iterate this process to show that all straight lines of the form $((\omega_t^n, v_t^n), (\Omega_{n+1}, V_{n+1}))$ pass strictly above $(\omega_t^n, v_t^n)$ which shows that $(\omega_t^n, v_t^n)$ remains an extreme point. To prove that all points $((\omega_t^n, v_t^n))_{0 \leq k \leq l}$ also remain extreme points, we can proceed recursively along the same lines as in the proof of the first claim. □

Proof of Proposition 7 Using Proposition 6 there exists a probability mass function $p^*$ such that $p^*$ is an optimal solution to (10) for any cost function $(k+1)$th order convex/concave on $\{\delta_{\text{inf}}, \cdots, \delta_{\text{sup}}\}$. Denote by $Z$ a discrete random variable with support $\{\delta_{\text{inf}}, \cdots, \delta_{\text{sup}}\}$ and probability mass function given by $p^*$. For $n \in \{l_1, \cdots, l_2\}$, $\bar{u}_j(n - \cdot)$ is $(k + 1)$th order convex/concave on $\{\delta_{\text{inf}}, \cdots, \delta_{\text{sup}}\}$, we obtain:

$$g(n) = E(\bar{u}_j(n - Z)).$$

By induction, we get that for any consecutive set of integers $z_i, \cdots, z_{i+k+1}$ in $\{l_1, \cdots, l_2\}$:

$$[z_i, \cdots, z_{i+k+1}] g = E([z_i - Z, \cdots, z_{i+k+1} - Z] \bar{u}_j).$$

This yields the claim as $Z$ has discrete support in $\{\delta_{\text{inf}}, \cdots, \delta_{\text{sup}}\}$ and since $\bar{u}_j(\cdot)$ is $(k + 1)$th order convex (resp. concave) on $\{l_1 - \delta_{\text{sup}}, \cdots, l_2 - \delta_{\text{inf}}\}$. □
B.4. Proof of Theorem 2

The proof is divided into three steps. In Lemma 7 we give an upper bound on the solution to the discretization scheme which holds regardless of the mesh size. The main difficulty in proving general convergence lies in the fact that sequences \((\tilde{u}_i^{k, \Delta t_p}(t))\) are not necessarily monotonic. However, this sequence is non-decreasing for regular mesh size sequences such as \(\Delta t_p = \frac{1}{p}\). Hence we first demonstrate convergence in that particular case in Lemma 8 and rely on this last result to prove Theorem 2. Some of the results are proved by induction. As discussed in Section B.2 the base case may be more involved for other objective functions but we can adapt the proof along the same lines as in Section B.2.

**Lemma 7.** For any positive discretization sequence \((\Delta t_p)_{p \in \mathbb{N}}:\)

\[
\forall p \in \mathbb{N}, \forall i \in \mathcal{V}, \forall t \leq T \quad \tilde{u}_i^{k, \Delta t_p}(t) \leq \tilde{u}_i^k(t^-)
\]
as long as \(\forall p \in \mathbb{N}, \Delta t_p < \delta^{\text{inf}}\).

**Proof** For the base case, observe that \(\tilde{u}_i^{k, \Delta t_p}(t) = 0 = \tilde{u}_i^k(t), \forall i \in \mathcal{V}\) and for \(t \in (-\infty, \Delta t_p]\) (recall that \(\Delta t_p < \delta^{\text{inf}}\)). Fix an integer \(p \in \mathbb{N}\) and suppose that, \(\forall i \in \mathcal{V}, \tilde{u}_i^{k, \Delta t_p}(\cdot) \leq \tilde{u}_i^k(\cdot)\) on \((-\infty, n \cdot \Delta t_p]\) for some \(n \in \mathbb{N}\). Consider \(i \in \mathcal{V}\) and \(\tilde{t} \in (n \cdot \Delta t_p, (n + 1) \cdot \Delta t_p]\). As \(\tilde{u}_i^k(\cdot)\) is non-decreasing (see Lemma 7), we have:

\[
\tilde{u}_i^k(\tilde{t}) \geq \tilde{u}_i^k(n \cdot \Delta t_p) = \max_{j \in V(i)} \inf_{P_{ij} \in \mathcal{P}} \mathbb{E}_{P_{ij}}[\tilde{u}_j^k(n \cdot \Delta t_p - c_{ij})].
\]

Using the induction hypothesis (recall that \(\delta^{\text{inf}} > \Delta t_p\)), we derive:

\[
\max_{j \in V(i)} \inf_{P_{ij} \in \mathcal{P}} \mathbb{E}_{P_{ij}}[\tilde{u}_j^k(n \cdot \Delta t_p - c_{ij})] \geq \max_{j \in V(i)} \inf_{P_{ij} \in \mathcal{P}} \mathbb{E}_{P_{ij}}[\tilde{u}_j^{k, \Delta t_p}(n \cdot \Delta t_p - c_{ij})] = \tilde{u}_i^{n+1}.
\]

Yet, by definition, \(\tilde{u}_i^{n+1} = \tilde{u}_i^{k, \Delta t_p}(\tilde{t})\). Hence, \(\tilde{u}_i^{k, \Delta t_p}(\tilde{t}) \leq \tilde{u}_i^k(\tilde{t})\). By iterating over \(n\), we get:

\[
\tilde{u}_i^{k, \Delta t_p}(t) \leq \tilde{u}_i^k(t), \forall t \leq T.
\]

Taking the left one-sided limit at any \(t\), we conclude:

\[
\tilde{u}_i^{k, \Delta t_p}(t) \leq \tilde{u}_i^k(t^-), \forall t \leq T,
\]
as \(\tilde{u}_i^{k, \Delta t_p}(\cdot)\) is lower semi-continuous by construction. We proceed by induction on \(p\) to establish the statement. \(\square\)

**Lemma 8.** For the regular mesh \((\Delta t_p = \frac{1}{p})_{p \in \mathbb{N}}\) and for any \(t \leq T\), the sequence \((\tilde{u}_i^{k, \Delta t_p}(t))_{p \in \mathbb{N}}\) is non-decreasing and converges to \(\tilde{u}_i^k(t^-)\).
Proof We break down the proof into three steps. First, for any \( t \leq T \), \( (\bar{u}_i^{k,\Delta_T}(t))_p \) is shown to be non-decreasing, hence \( (\bar{u}_i^{k,\Delta_T}(\cdot))_p \) converges pointwise to a limit \( f_i(\cdot) \). Next, we establish that for any \( t \leq T \) and \( \epsilon > 0 \), \( f_i(t) \geq \bar{u}_i^k(t-\epsilon) \). This will enable us to squeeze \( f_i(t) \) to finally derive \( f_i(t) = \bar{u}_i^k(t^-) \).

First Claim: \( \forall t \leq T \), \( (\bar{u}_i^{k,\Delta_T}(t))_p \) is non-decreasing. Take \( p \in \mathbb{N} \) large enough such that \( \frac{1}{2p+T} < \delta \inf \).
The base case is straightforward: \( \bar{u}_i^{k,\Delta_T}(t) = 0 = \bar{u}_i^{k,\Delta_{T+1}}(t) \) for \( t \leq \frac{1}{2p+T}, \forall i \in \mathcal{V} - \{d\} \). Suppose that \( \forall i \in \mathcal{V}, \bar{u}_i^{k,\frac{1}{2p+T}}(\cdot) \leq \bar{u}_i^{k,\frac{1}{2p+T+1}}(\cdot) \) on \(( -\infty, n \cdot \frac{1}{2p+T} \) for some \( n \in \mathbb{N} \). Consider \( i \in \mathcal{V} \) and \( t \in (n \cdot \frac{1}{2p+T}, (n+1) \cdot \frac{1}{2p+T+1}) \). By construction, we have:

\[
\bar{u}_i^{k,\frac{1}{2p+T}}(t) = \max_{j \in \mathcal{V}(i)} \inf_{P_{ij}^k} \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\frac{1}{2p+T}}(n \cdot \frac{1}{2p+T+1} - c_{ij})].
\]

Using the hypothesis, we get:

\[
\bar{u}_i^{k,\frac{1}{2p+T}}(t) \geq \max_{j \in \mathcal{V}(i)} \inf_{P_{ij}^k} \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\frac{1}{2p+T}}(n \cdot \frac{1}{2p+T+1} - c_{ij})].
\]

Yet, regardless of the parity of \( n \):

\[
\max_{j \in \mathcal{V}(i)} \inf_{P_{ij}^k} \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\frac{1}{2p+T}}(n \cdot \frac{1}{2p+T+1} - c_{ij})] \geq \bar{u}_i^{k,\frac{1}{2p+T}}(t),
\]

with equality if \( n \) is even. To conclude, we have \( \bar{u}_i^{k,\frac{1}{2p+T}}(t) \geq \bar{u}_i^{k,\frac{1}{2p+T}}(t) \). By iterating over \( n \), we get:

\[
\forall t \leq T, \bar{u}_i^{k,\frac{1}{2p+T}}(t) \geq \bar{u}_i^{k,\frac{1}{2p+T}}(t).
\]

Bringing together Lemma [2] and the last point, for any \( t \leq T \), the sequence \( (\bar{u}_i^{k,\Delta_T}(t))_p \) is non-decreasing and bounded above by \( \bar{u}_i^k(t^-) \), hence it has a limit \( f_i(t) \). Second Claim: \( \forall t \leq T \) and \( \forall \epsilon > 0 \), \( f_i(t) \geq \bar{u}_i^k(t^-) \). Consider \( \epsilon > 0 \) and a large enough \( p \) such that \( \epsilon > 1 / 2p \) which implies \( \Delta_T \cdot \frac{1}{\Delta_T p} \geq t - \epsilon \).

We have \( \bar{u}_i^{k,\Delta_T} (\cdot) \) is non-decreasing):

\[
\bar{u}_i^{k,\Delta_T}(t) = \max_{j \in \mathcal{V}(i)} \inf_{P_{ij}^k} \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\Delta_T}(\Delta_T p \cdot \frac{t}{\Delta_T p} - c_{ij})] \geq \max_{j \in \mathcal{V}(i)} \inf_{P_{ij}^k} \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\Delta_T}(t - \epsilon - c_{ij})].
\]

Take \( j \in \mathcal{V}(i) \). Since \( \bar{u}_j^{k,\Delta_T}(\cdot) \) is lower semi-continuous, the infimum in the previous inequality is attained for some \( P_{ij}^k \) which gives:

\[
\bar{u}_i^{k,\Delta_T}(t) \geq \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\Delta_T}(t - \epsilon - c_{ij})].
\]

As the sequence \( (\bar{u}_j^{k,\Delta_T}(t - \epsilon - \omega))_p \) is non-decreasing for any \( \omega \) (first claim), we have, for any \( m \leq p \):

\[
\bar{u}_i^{k,\Delta_T}(t) \geq \mathbb{E}_{P_{ij}^k} [\bar{u}_j^{k,\Delta_T m}(t - \epsilon - c_{ij})].
\]
Observe that \( p^k_i \in P^k_{ij}, \forall p \in \mathbb{N} \). Since \( P^k_{ij} \) is a compact set for the weak topology, there exists a subsequence of \( (p^k_{ij})_p \) converging weakly in \( P^k_{ij} \) to some probability measure \( p_{ij} \). Without loss of generality, we continue to refer to this subsequence as \( (p^k_{ij})_p \). We can now take the limit inferior \( p \to \infty \) in the previous inequality which yields:

\[
f_i(t) \geq \liminf_{p \to \infty} \mathbb{E}_{p^k_{ij}}[\hat{u}^{k, \Delta t_m}_{ij}(t - \epsilon - c_{ij})].
\]

Since \( \hat{u}^{k, \Delta t_m}_{ij}(\cdot) \) is lower semi-continuous, we derive (we refer to standard results in convergence of measure in the weak topology):

\[
\liminf_{p \to \infty} \mathbb{E}_{p^k_{ij}}[\hat{u}^{k, \Delta t_m}_{ij}(t - \epsilon - c_{ij})] \geq \mathbb{E}_{p_{ij}}[\hat{u}^{k, \Delta t_m}_{ij}(t - \epsilon - c_{ij})].
\]

To take the limit \( m \to \infty \), note that \( \hat{u}^{k, \Delta t_m}_{ij}(\cdot) \) is bounded above by 1 and pointwise converges to \( f_j(\cdot) \), the dominated convergence theorem can then be applied:

\[
\lim_{m \to \infty} \mathbb{E}_{p_{ij}}[\hat{u}^{k, \Delta t_m}_{ij}(t - \epsilon - c_{ij})] = \mathbb{E}_{p_{ij}}[f_j(t - \epsilon - c_{ij})].
\]

This yields \( f_i(t) \geq \mathbb{E}_{p_{ij}}[f_j(t - \epsilon - c_{ij})] \), which further implies \( f_i(t) \geq \inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[f_j(t - \epsilon - c_{ij})] \). As the last inequality holds for any \( j \in V(i) \), we finally obtain:

\[
f_i(t) \geq \max_{j \in V(i)} \inf_{p_{ij} \in P^k_{ij}} \mathbb{E}_{p_{ij}}[f_j(t - \epsilon - c_{ij})].
\]

Observe that \( \hat{u}^k_i(\cdot) \) is defined by a similar relation in (4). Denoting \( L = \left \lfloor \frac{T}{\tau_{\text{inst}}} \right \rfloor \), we can show by interval increments of size \( \delta_{\text{inst}} \) that \( f_i(t) \geq u^k_i(t - L \cdot \epsilon), \forall t \leq T \), and this holds \( \forall \epsilon > 0 \). The latter result can be reformulated as:

\[
\forall \epsilon > 0, \forall t \leq T, f_i(t) \geq \hat{u}^k_i(t - \epsilon)
\]

We now move on to the third claim: \( \forall t \leq T, f_i(t) = \hat{u}^k_i(t^-) \). Using Lemma 7 and the inequality above, we are able to squeeze \( f_i(\cdot) \):

\[
\forall t \leq T, \forall \epsilon > 0 \quad \hat{u}^k_i(t^-) \leq f_i(t) \leq \hat{u}^k_i(t^-).
\]

Taking the limit \( \epsilon \to 0^+ \) for a given \( t \leq T \) yields \( f_i(t) = \hat{u}^k_i(t^-) \). □

We are now ready to establish Theorem 2.

Proof of Theorem 2 In contrast to the particular case handled by Lemma 8, our approximation of \( \hat{u}^k_i(t) \) may not improve as \( p \) increases. For that reason, there is no straightforward comparison between \((\hat{u}^{k, \Delta t_p}_i(t))_p\) and \((\hat{u}^{k, \Delta t_p'}_i(t))_p\). However, for a given \( t \leq T, \epsilon > 0 \) and a large enough \( p \), \((\hat{u}^{k, \Delta t_p}_i(t))_p\) can be shown to be lower bounded by a subsequence of \((\hat{u}^{k, \Delta t_p'}_i(t - \epsilon))_p\). This is how we proceed to establish convergence.
Consider $t \leq T$, $\epsilon > 0$ and $p \in \mathbb{N}$. Define $\sigma(p) \in \mathbb{N}$ as the unique integer satisfying $\frac{1}{2^{\sigma(p)} - 1} < \Delta t_p \leq \frac{1}{2\sigma(p)}$. Since $\lim_{p \to \infty} \Delta t_p = 0$, we necessarily have $\lim_{p \to \infty} \sigma(p) = \infty$. Remark that $\dot{u}_i^k \frac{1}{2^{\sigma(p)}}(\cdot)$ has steps of size $\frac{1}{2^{\sigma(p)}} \geq \Delta t_p$, i.e. $\dot{u}_i^k \Delta t_p(\cdot)$ is expected to be a tighter approximation of $\dot{u}_i^k(\cdot)$ than $\dot{u}_i^k \frac{1}{2^{\sigma(p)}}(\cdot)$ is. However time steps do not overlap (multiples of either $\Delta t_p$ or $\frac{1}{2p}$) making the two sequences impossible to compare. Nevertheless, they differ by no more than $\Delta t_p$. Thus, if $p$ is large enough so that $\Delta t_p < \epsilon$, for each update needed to calculate $\dot{u}_i^k \frac{1}{2^{\sigma(p)}}(t - \epsilon)$ there is a corresponding update for a larger budget to compute $\dot{u}_i^k \Delta t_p(t)$. As a consequence, the sequence $(\dot{u}_i^k \Delta t_p(t))_p$ forms a lower bound on the sequence of interest $(\dot{u}_i^k(t - \epsilon))_p$. Using in addition Lemma 7, we are able to squeeze $(\dot{u}_i^k \Delta t_p(t))_p$: 

$$
\dot{u}_i^k \frac{1}{2^{\sigma(p)}}(t - \epsilon) \leq \dot{u}_i^k \Delta t_p(t) \leq \dot{u}_i^k(t^-),
$$

$\forall t \leq T, \forall \epsilon > 0$ and for $p$ large enough. Yet, Lemma 8 shows that:

$$
\lim_{p \to \infty} \dot{u}_i^k \frac{1}{2^{\sigma(p)}}(t - \epsilon) = \dot{u}_i^k((t - \epsilon)^-).
$$

The squeeze theorem cannot be readily used here to back up the claim as the two bounds do not match. However, since the sequence $(\dot{u}_i^k \Delta t_p(t))_p$ is bounded for any $t$, one can extract a subsequence converging to a limit $f_i(t)$. By taking the limit $p \to \infty$ for this subsequence in the previous inequality, we obtain $\dot{u}_i^k((t - \epsilon)^-) \leq f_i(t) \leq \dot{u}_i^k(t^-)$. Taking the limit $\epsilon \to 0$, we derive $f_i(t) = \dot{u}_i^k(t^-)$. This shows that for any $t$, $\dot{u}_i^k(t^-)$ is the unique limit point of $(\dot{u}_i^k \Delta t_p(t))_p$. This sequence is known to lie in a compact set, thus $(\dot{u}_i^k \Delta t_p(t))_p$ converges to $\dot{u}_i^k(t^-)$ for any $t$. $\square$