

**THE ARRIVING ON TIME PROBLEM: A DISCRETE ALGORITHM THAT ENSURES CONVERGENCE**

**Yu Nie<sup>1</sup>**

PhD Candidate  
Department of Civil and Environmental Engineering  
University of California  
Davis, California 95616  
Tel: 530 754 6429  
Email: ynie@ucdavis.edu

**Yueyue Fan**

Assistant Professor  
Department of Civil and Environmental Engineering  
University of California  
Davis, CA 95616  
Email: yyfan@ucdavis.edu

Submitted to ADB30 – Committee of Transportation Network Modeling

Word count: 4559 words + 3 tables + 6 figures = 6559

---

<sup>1</sup> Corresponding author.

### **Abstract**

Finding optimal paths in stochastic networks is an important topic in many science and engineering fields. To cope with uncertainty, various performance measurements have been introduced, including expected travel time, reliability, value at risk, expectation with chance constraint, etc. Literature has shown that adaptive strategies that incorporate both anticipation and real-time information are more efficient than a pre-planned strategy in a stochastic environment. Most adaptive optimal path algorithms focus on least expected travel time. Recently, we have proposed an adaptive path finding strategy, named as the SOTA algorithm, to maximize the travel time reliability for any given time threshold. However, the originally proposed SOTA algorithm is based on the classic successive approximation. Whether this algorithm converges within finite number of successive approximation steps is an open question. In this paper, we propose a discrete SOTA algorithm that runs in an optimal polynomial time, and thus improves the computational efficiency significantly. Numerical examples and future extensions are also provided.

**Keywords:** stochastic networks, adaptive routing, reliability.

## 1. Introduction

Finding optimal paths in a given network is one of the most fundamental problems in network studies and has broad applications in physical and social sciences and engineering, particularly in the fields of computer science, operations research, and transportation engineering. Our main focus in this paper is on finding an adaptive optimal routing strategy that provides the maximum reliability for desired time thresholds. This paper, built on our previous work on this topic, provides a new algorithm to improve the computational efficiency and an extended formulation to incorporate more realistic modeling assumptions.

In a stochastic environment, different performance criteria may result in different optimal routing strategies. Most existing work (e.g., [1]) defines optimality based on least expected time (LET). An optimal path can be found by setting each random link travel time to its expected value and solving an equivalent deterministic problem. LET problems have been studied extensively, and have been extended to situations where time-varying link travel times (e.g., [2, 3, 4]) and/or correlation between link travel times (e.g., [5, 6]) are considered. However, an LET path may be risky. That is, following an LET path may result in a high chance of encountering an extremely large travel time. To address this issue, Frank [7] and Mirchandani [8] considered the path that maximizes the probability of realizing a travel time less than a predefined value as the optimal one. Sigal *et al.* [9] suggested that a path is optimal if its probability of being shorter than all other paths is the largest. However, all these formulations require enumerating paths and evaluating multiple integrals, thus can not be easily implemented for large-scale problems. As an alternative, Loui [10] introduced a general utility function to describe the path choice preference and defined the optimality based on maximum utility. It has been shown that such utility-based models can be solved efficiently only if an affine or exponential utility function is employed [10, 11]. In view of this limitation, Murthy *et al.* [12] proposed to approximate concave non-linear utility functions with a piecewise affine function and showed that efficient algorithms exist for such approximations.

Literature on general stochastic optimal control [13, 14] has shown that adaptive strategies, which incorporate both anticipation and real-time information, may generate different results from a pre-planned optimal strategy in a stochastic environment. Hall [2] introduced the concept of adaptive decision making to time-variant LET problems and showed that adaptive strategies are more efficient than simply following a pre-planned optimal path in dynamic and stochastic networks. Recently, Fan *et al.* [15] proposed an adaptive optimal path algorithm for the case where optimality is defined as maximum travel

time reliability. This stochastic on time arrival problem (SOTA) was formulated using dynamic programming and solved based on classic successive approximation. However, the method of successive approximation does not guarantee finite convergence when applied to SOTA because an optimal routing strategy may contain loops<sup>2</sup> (an example is given in Section 4.1).

This paper presents a discrete SOTA algorithm that ensures finite convergence, and more importantly, runs in a polynomial time. We shall show that, given a finite time budget  $T$ , the solution of SOTA can be found by scanning a discrete probability expansion network (DPEN), which is constructed from the original stochastic network (see Section 3 for details). Furthermore, when the scanning order is properly chosen, any node of DPEN will be visited no more than once before optimality is achieved. This new algorithm is computationally efficient and easy to implement for cases where probability density functions are either continuous or discrete.

The next section summarizes the mathematical formulation and the successive approximation algorithm for continuous and discrete SOTA. Section 3 presents the algorithm of increasing order of time budget (IOTB), and discusses its convergence and complexity. Numerical examples are provided in Section 4 to illustrate the efficiency of the proposed algorithm. Section 5 provides a possible extension of the SOTA formulation to include correlation of travel times among adjacent links. The last section concludes this study.

## 2. SOTA Formulation and Successive Approximation Algorithm

Consider a directed network  $G(N,A)$ , where  $N$  and  $A$  denote the sets of nodes and links, and  $n$  and  $m$  are number of nodes and links, respectively. Each directed link  $ij \in A$  is associated with an *independent* random travel time (cost)  $t_{ij}$  whose probability density function is denoted by  $p_{ij}(t)$ . The problem of stochastic on time arrival (SOTA) is to find an optimal routing strategy that maximizes the probability of arriving at the destination node  $S$ , starting from any node  $i$  and within any given time budget  $t \in [0, T]$ .

Let  $u_i(t)$  denote the maximum probability of arriving at the destination  $S$  from node  $i$  given time budget  $t$ . If a traveller at node  $i$  chooses to visit node  $j$  next, the probability that the traveller spends a time in the interval  $[\omega, \omega \pm d\omega]$  on link  $ij$  is  $p_{ij}(\omega)d\omega$  by definition. The time budget at node  $j$  is thus  $t - \omega$ . Based on Bellman's principle of optimality [13], no matter which node  $j$  the traveller elects to visit, s/he must follow the optimal routing strategy

---

<sup>2</sup> The authors are grateful to Prof. Travis Waller at University of Texas, Austin for discussion on this issue.

travelling from that node  $j$  to the destination within the remaining time  $t-\omega$ . SOTA thus can be formulated as the following system of nonlinear convolution integral equations:

$$u_i(t) = \max_{j \neq i} \int_0^t p_{ij}(\omega) u_j(t-\omega) d\omega, \forall i \in N, i \neq S, 0 \leq t \leq T, \quad (1)$$

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

The functions  $u_i(t)$  and the sequence of next nodes to visit  $q_i(t)$  are unknowns to be determined for each node  $i$  at any time  $t$ . Note that

$$q_i(t) = \arg \max_{j \neq i} \left\{ \int_0^t p_{ij}(\omega) u_j(t-\omega) d\omega \right\}, \forall i \in N, i \neq S, 0 \leq t \leq T. \quad (3)$$

The unknowns  $u_i(t)$  and  $q_i(t)$  can be obtained by solving the Equation system (1) – (2) simultaneously, which is done off line.

FIGURE 1 explains how the computed optimal routing policy can be used as a lookup table to guide real-time vehicle routing. The optimal choice for the next node can be retrieved from the table, given the traveller's location (node  $i$ ) and the time budget for completing the trip from that location ( $t$ ). Each time a traveller traverses a link, the *experienced* travel time on that link is deducted from the time budget. The updated time budget will then be used to fetch the routing strategy at the new location. Travellers should continue in this manner until the final destination node is reached. Note that  $t_{12}$  and  $t_{24}$  shown in FIGURE 1 are not the expected values. Rather, these are the actual times spent on links  $1 \rightarrow 2$  and  $2 \rightarrow 4$ . These times become known once these links are traversed.

The procedure of using successive approximation to solve the nonlinear equation system [15] is summarized below.

[SA Algorithm]:

**Step0:** Initialization. Set  $k = 0$ ,  $u_i^k(t) = 0, \forall i \in N, i \neq S, t \geq 0$  and  $u_S^k(t) = 1, t \geq 0$

**Step1:** Set  $k=k+1$ ,  $u_i^k(t) = \max_{j \neq i} \int_0^t p_{ij}(\omega) u_j^{k-1}(t-\omega) d\omega, \forall i \in N, i \neq S, u_S^k = 1, t \geq 0$ .

**Step2:** Convergence test, if  $\max(|u_i^k - u_i^{k-1}|, \forall i, t) = 0$  stop; otherwise go to **Step1**.

It has been shown in our previous work that the above algorithm will lead to convergence. However, whether or not it converges in a finite number of steps remains an unresolved problem. Note that following an optimal strategy may result in loops in stochastic networks. SOTA thus belongs to the general class of infinite-horizon stochastic Markovian decision processes. Although the existence and uniqueness of solutions to an infinite-horizon

Markovian process are known [16], how to design numerical algorithms with finite convergence is an open question. Next, we will show that such an algorithm exists for a discrete SOTA problem.

First, let time be considered in a discrete manner, i.e.,  $t = 0, \delta, 2\delta, \dots, L\delta$ , where  $\delta$  is unit time and  $L\delta = T$  is the upper bound of the time budget of interest. Correspondingly, the probability of taking time between  $h$  and  $h+\delta$  to traverse link  $ij$  is denoted by  $P_{ij}(h) = \int_h^{h+\delta} p_{ij}(h)dh$ . Throughout the paper we shall assume that traversal time  $h$  on any link can only take values that are (positive) integer multiples of the unit time  $\delta$ . This is to ensure that any  $h$  or  $t-h$  is a defined point in the discrete time space. The assumption simplifies the evaluation of the convolution integral in Equation (1), while the introduced errors can be controlled by reducing  $\delta$ . For completeness we further assume that  $p_{ij}(h) = 0$  if  $h > T$  or  $h \leq 0$ .

In discrete case, the convolution integral in Equation (1) can be replaced by a finite sum. The formulation of the discrete SOTA is

$$u_i(t) = \max_{j \neq i} \sum_{h=0}^t P_{ij}(h) u_j(t-h), \forall i \in N, i \neq S, t = 0, \delta, \dots, L\delta \quad (4)$$

$$u_s(t) = 1, t = 0, \delta, \dots, L\delta \quad (5)$$

Successive approximation algorithm for the discrete problem is summarized as below.

[Discrete SA algorithm]

**Step 0:** Initialization. Set  $u_i^0(t) = 0, q_i^0(t) = -1, \forall i \in N, i \neq S, t = 0, \delta, \dots, L\delta$ .

Set  $u_s^0(t) = 1, q_s^0(t) = -1, t = 0, \delta, \dots, L\delta, k = 1$

**Step 1:** for  $i = 1$  to  $n, i \neq S$  do

for  $t = \delta$  to  $L\delta$  do

for all  $j$  where link  $ij$  is an existing link

$$\{ r_t = \sum_{h=0}^t P_{ij}(h) u_j^{k-1}(t-h)$$

$$\text{if } r_t > u_i^{k-1}(t) \quad u_i^k(t) = r_t, q_i^k(t) = j$$

$$\text{else} \quad u_i^k(t) = u_i^{k-1}(t), q_i^k(t) = q_i^{k-1}(t) \}$$

**Step 2:** Convergence test. If  $\max(|u_i^k(t) - u_i^{k-1}(t)|, \forall i, t) < \varepsilon$ , stop; otherwise set  $k=k+1$  and go to **Step 1**.

In each iteration, SA algorithm visits each link  $L$  times and the total running time per link is  $1 + 2 + \dots + L \approx 0.5L^2$ . Thus, the complexity of the algorithm is approximately  $0.5KmL^2$ , where  $m$  is the number of links and  $K$  is the number of iterations when optimality is attained. However, note that the upper bound of  $K$  is unknown (theoretically it may go to infinity).

### 3. A Finite Convergence Algorithm

In this section, we present a modified SOTA algorithm that converges to optimality within finite steps. This is achieved by exploiting the special structure of the discrete problem. To manifest the structure, we introduce a discrete probability expansion network (DPEN). A DPEN, denoted as  $G_d$ , has a set of nodes  $N_d$  and a set of links  $A_d$  given as below:

$$N_d = \{[i, t] \mid i \in N, t = 0, \delta, \dots, L\delta\} \quad (6)$$

$$A_d = \{([i, t], [j, t-h]) \mid (i, j) \in A, P_{ij}(h) > 0, h = 0, \delta, \dots, L\delta, t = 0, \delta, \dots, L\delta\} \quad (7)$$

where  $L\delta = T$  is the upper bound of the time budget of interest.

The concept of DPEN is demonstrated in FIGURE 2. The physical network has 3 nodes and 2 links, with node 3 being the destination and  $T = 10\delta$ . A discrete set of probabilities of link travel times is also given in FIGURE 2.

As shown in FIGURE 2, each horizontal layer of the DPEN is a copy of the original network that corresponds to a given time budget. A node  $[i, t]$  in DPEN represents a state at which a routing decision needs to be made at  $i$  when remaining time budget is  $t$ . A link between two nodes, say  $[i, t]$  and  $[j, t-h]$ , denotes a transition between the two states. Precisely, this means that an amount of time  $h$  is consumed in traversing link  $ij$  and the remaining time budget decreases from  $t$  to  $t-h$ . We emphasize that the cost associated with each link in DPEN is  $P_{ij}(h)$  (i.e., the probability that the traversal time on  $ij$  is  $h$ ) rather than the travel time.

With DPEN constructed, solving the discrete SOTA is equivalent to setting  $u_i(t)$  for each node  $[i, t]$  such that Equations (4) – (5) are satisfied. Obviously, the setting order greatly affects computational efficiency. We introduce an algorithm that sets  $u_i(t)$  following an increasing order of time budget (IOTB).

#### [IOTB]

**Step 0:** Initialization. Set  $u_i(t) = 0, q_i(t) = -1, \forall i \in N, i \neq S, t = 0, \delta, \dots, L\delta$ .

$$\text{Set } u_S(t) = 1, q_S(t) = -1, t = 0, \delta, \dots, L\delta$$

**Step 1:** for  $t = \delta$  to  $L\delta$  do

for all link  $ij \in A$ , do

$$\left\{ \begin{aligned} r_t &= \sum_{h=\delta}^t P_{ij}(h)u_j(t-h) \\ \text{if } r_t > u_i(t), \text{ set } u_i(t) &= r_t, q_i(t) = j \end{aligned} \right\}$$

**Proposition 1** *Algorithm IOTB identifies an optimal solution of the discrete SOTA (Equations (4-5)) in finite steps.*

Proof: IOTB updates  $u_i(t)$  following the order of increasing time budget ( $t$  increases from 0 to  $T$ ), i.e., from the bottom to the top row in a DPEN (see FIGURE 2). Note that the optimal solution is known at the bottom row of the DPEN. That is,  $u_i(0) = 0, \forall i \in N, i \neq S, u_S(0) = 1$  since nobody can arrive at the destination within zero time from nodes other than the destination itself. This corresponds to **Step 0** in IOTB. The quantity  $u_i(t)$  only depends on the costs of its outgoing links ( $P_{ij}(h)$ ) and the optimal probabilities associated with less time budgets for the remaining process ( $u_j(t-h)$ ). In a DPEN, this means that the optimal probability at any node can only be affected by nodes lying below it. Thus, following the order of increasing time budget, any node  $(i,t)$  in DPEN will be scanned only once and the label  $u_i(t)$  will be permanently set. As such, the optimal solution will be identified within finite steps.  $\square$

The above convergence proof shows that algorithm IOTB visits each link in the DPEN only once for each discrete time budget  $t$ , and the major operation involved in each visit is to compute the convolution integral. Thus, the complexity of IOTB is  $0.5mL^2$ , which equals to the computational overhead of a single iteration in SA algorithm.

### Remarks

1. IOTB converges within finite steps, and more importantly, provides an optimal time complexity. Actually, no algorithm can identify an optimal SOTA solution without scanning DPEN at least once.
2. DPEN is introduced to illustrate the structure of the discrete problem and to prove the finite convergence of the proposed algorithm. Nevertheless, no additional overhead is required to construct DPEN explicitly in implementation.
3. When applying IOTB to solve a continuous SOTA problem (i.e., the probability density function is continuous),  $P_{ij}(h)$  needs to be calculated *a priori*. Different numerical approaches can be used to evaluate  $P_{ij}(h)$ , such as rectangular rule, trapezoid rule, Simpson's rule, etc. Advanced approaches such as Simpson's rule can provide accurate discrete probabilities, but they normally consume more computational resource. The

computing requirement of discrete probabilities  $P_{ij}(h)$  is not included in the above complexity analysis.

4. The computational overhead of discrete algorithms increases rapidly with the size of  $L$ , which is determined by the total time budget  $T$  and the unit time  $\delta$ . Fortunately, a prohibitively large  $L$  rarely arises in real-world transportation applications. Even for an urban network with thousands of nodes and links, the longest travel time between any origin and destination would probably not exceed two or three hours. Moreover, few travelers would switch from one path to another just to save a few seconds. Therefore, a unit time of 15 or 30 seconds might be good enough for practical purpose. As such,  $L$  is usually less than 1,000 in practice.
5. In the study of time-dependent shortest path (TDSP) problem, a space-time expansion network (STEN) has a similar structure of DPEN that allows efficient numerical implementations. The IOTB algorithm discussed here was inspired by the decreasing order of time (DOT) algorithm for TDSP [17].

#### 4. Numerical Studies

We first use a small example to demonstrate that following an optimal routing strategy may result in loops in stochastic networks. We then validate IOTB algorithm by comparing it with SA algorithm and explore some properties of the SOTA optimal solution. Finally, large examples are tested to highlight the computational efficiency of the proposed algorithm. Both SA and IOTB algorithms were coded in MS-VC++ and tested on a Windows-XP server with a 3.06 GHz CPU and 2 Gb memory.

##### 4.1 A Small Example Demonstrating Loops Resulted from An Optimal Strategy

Consider the three-node network in FIGURE 3. The possible times for traversing each link and the associated discrete probabilities are given. The destination is node 3.

TABLE 1 reports the analytical SOTA solution ( $T=10$ ), which can be easily obtained by manual calculation. For brevity we only examine the solution at node 1 with time budget 10. If a traveler spends 1 minute on link 1-2 (the probability is 0.5), the remaining time budget at node 2 is 9 thus s/he should visit node 3 next to maximize the on-time arrival probability. However, if the traveler happens to spend 6 minutes on link 1-2, node 3 is no longer a favorable choice at node 2 (the on-time arrival probability is only 0.1). A better chance ( $0.2 = 0.4 \times 0.5$ ) is obtained if the traveler *returns* to node 1 and then finds ways to go to node 3.

Two key points are illustrated in this example. First, an optimal routing strategy may suggest cycling (1->2->1->3 in this example) for some realizations of travel times. The possibility of cycling causes an issue of convergence in the classic successive approximation. Second, no optimal path can be identified prior to the actual routing process. Decisions are made step by step as a traveler moves from one node to another until the destination is reached.

#### 4.2 A Continuous Problem

The example is taken from Fan [15]. The network contains five nodes and 18 links, as shown in FIGURE 4. Link travel times are assumed to follow the Gamma distribution, whose probability density function takes the form of

$$p(t, \gamma, \alpha) = \frac{\alpha^\gamma e^{-\alpha t} t^{\gamma-1}}{\Gamma(\gamma)} \quad (8)$$

in which  $\Gamma(\gamma)$  is the Gamma function of  $\gamma$ . The mean  $\mu$  and the variance  $\sigma^2$  are  $\mu = \gamma/\alpha$ , and  $\sigma^2 = \gamma/\alpha^2$ , respectively.

To discretize the problem, we first uniformly divide the range  $[0, T]$  into  $L$  intervals, and then calculate  $P_{ij}(h)$  using the Simpson's rule. In reported results,  $T = 4$  and the unit time  $\delta$  is set to be 0.001.

TABLE 2 presents the optimal solutions produced by SA and IOTB algorithms. The maximum probabilities of on-time arrival and the associated optimal routing policy are reported for two starting nodes and eight time budgets. SA algorithm converged after 4 iterations and spent about 1.6 seconds CPU time, while IOTB algorithm consumed about 0.7 seconds CPU time.

An immediate observation from TABLE 2 is that IOTB and SA produced identical results, which validates the correctness of IOTB algorithm. TABLE 2 shows that the probability of arriving at destination node 5 on time decreases monotonically for both origins as the time budget is reduced. Furthermore, the optimal strategies change with the time budget. At node 1, for example, the optimal successor node changes from node 3 to node 2 as the time budget available for an on-time arrival is reduced to 0.5247. Even though the minimum expected travel time of link (2, 5) is bigger than that of link (3, 5), there is higher chance of on-time arrival by taking path (1, 2, 5) due to the higher variance of travel time over that path. This clearly illustrates that different definitions of optimality, such as reliability vs. expectation, can lead to different optimal routing decisions.

FIGURE 5 shows how optimal probabilities at origin node 1 given by IOTB change as the time resolution  $\delta$  becomes finer and finer. Each curve in the figure corresponds to a given time budget. As shown, all curves tend to become flat as  $\delta$  diminishes, meaning that the associated optimal probabilities are converging. This implies that the discretization scheme is stable so that finer time resolution always result in better approximation.

#### 4.3 Large Examples

We proceed to validate the computational efficiency of IOTB using a set of randomly generated grid networks (see TABLE 3). In this experiment we fix time budget  $T = 50$  and  $\delta = 0.05$ . Travel times on any link  $ij$  are assumed to follow a Gamma distribution, with arbitrarily assigned parameters. Trapezoid rule is used to evaluate discrete probabilities  $P_{ij}(h)$ .

The performance of the two algorithms is compared in TABLE 3 for five networks of various sizes. Note that the computational advantage of IOTB magnifies as network size increases. For the network with 2,598 nodes and 9,898 links, for example, IOTB is about 17 times faster than SA. The convergence curves of SA in the five network are given in FIGURE 6 for comparison purpose, where “convergence indicator” corresponds to  $\max(|u_i^k(t) - u_i^{k-1}(t)|, \forall i, t)$ .

### 5. Extension of SOTA to Stochastic Networks with Correlated Link Travel Times

The SOTA formulation given in Equations 1-2 assumes independent random link travel times. Inspired by recent work on shortest path problems with correlated link times [5, 6], here we give a possible extension of the SOTA formulation to incorporate the consideration of correlated link travel times. Suppose link travel times follow certain type of probability distribution under normal condition, and some other type of distribution under extreme condition (e.g., caused by bad weather, natural disasters or terrorist attack). The correlations between the states of adjacent links are taken into account by introducing conditional probabilities,  $\alpha_{ij}^{sr}$ , meaning the probability of having link  $ij$  in state  $r$  if the link leading to node  $i$  is in state  $s$ . For the time being, let us consider only two possible states, either normal condition ( $s = 1$ ) or extreme condition ( $s = 2$ ). Generalization of more possible link states is fairly straightforward.

Using the same reasoning for Equations (1-2), the SOTA problem with consideration of correlated link times can be formulated as

$$u_i(t, s) = \max_{j \in J_i} \left\{ \int_0^t p_{ij}^s(\omega) \sum_{r=1}^2 [\alpha_{ij}^{sr} u_j(t - \omega, r)] d\omega \right\}, \quad i = 1, 2, \dots, N-1; s = 1, 2; t \geq 0, \quad (9)$$

$$u_N(t, s) = 1, \quad (10)$$

where

$p_{ij}^s(\omega)d\omega$  = the probability of traversing the direct link  $ij$  within time  $\omega$  and  $\omega+d\omega$ , if the link leading to node  $i$  is in state  $s$ ; and  
 $u_i(t, s)$  = the probability that, starting from node  $i$  and given that the link leading to node  $i$  is in state  $s$ , the traveler arrives at node  $N$  by time  $t$  when an optimal sequence of choices is made,  $i = 1, 2, \dots, N$ ,  $0 \leq t < \infty$ .

Equations 9-10 have the same structure as Equations 1-2, except that the dimension of the system state vector is increased by one. This extension offers more modeling flexibility, but requires additional data support and more computing resource.

## 6. Conclusions and Discussions

We have presented a discrete SOTA algorithm guaranteeing finite convergence. Scanning the discrete probability expansion network (DPEN) in an increasing order of time budget, the algorithm IOTB offers an optimal polynomial complexity of running time ( $0.5mL^2$ ). Our numerical experiments on large grid networks indicate that IOTB always find optimal routing strategies at much less computational cost than successive approximation. However, the superiority of IOTB in terms of computational efficiency is not as significant as what the complexity analysis projects. SA seems to be able to converge within a fairly small number of iterations (roughly ranging from 1% to 15 % of the number of nodes). This suggests that an infinite-horizon routing process is unlikely to arise in large-size problems, in alignment with the argument of Bertsekas *et al.* [14]. Moreover, the average CPU time spent in a single SA iteration seems much less than that of IOTB, while our previous analysis suggests that they should be about same. A possible explanation is that our implementation of IOTB incurs some hidden costs not included in the complexity analysis. Further study on high-performance implementation of IOTB is needed.

Other than computational advantages, algorithm IOTB is appealing in the following respects. First, it can be used to solve both discrete and continuous SOTA problems. Second, it avoids intensive mathematical techniques that may introduce numerical instability, such as the inversion of Laplace transform in the originally proposed SOTA algorithm [15].

Implementation of IOTB algorithm does not require complicated data structure or intensive memory consumption.

The main focus of this paper is on the computational efficiency of IOTB algorithm. Applying IOTB to the design of efficient solution schemes for extended SOTA problems considering correlated and/or time-dependent link travel times is currently under study.

**References:**

1. Dijkstra, E. W. A Note on Two Problems on Connexion with Graphs. *Numerische Mathematik*, Vol. 1, 1959, pp. 269-271.
2. Hall, R. W. The Fastest Path through A Network with Random Time-Dependent Travel Time. *Transportation Science*, Vol. 20, 1986, pp. 182-188.
3. Miller-Hooks, E. D. and Mahmassani, H. S. Least Expected Time Paths in Stochastic, Time-varying Transportation Networks. *Transportation Science*, Vol. 34, 2002, pp. 198-215.
4. Dean, B. C. Algorithms for Minimum-Cost Path in Time-Dependent Networks with Waiting Policies. *Networks*, Vol. 44, 2004, pp. 41-46.
5. Waller, S. T. and Ziliaskopoulos, A. K. On the Online Shortest Path Problem with Limited Arc Cost Dependencies. *Networks*, Vol. 40, 2002, pp. 216-227.
6. Fan, Y., Kalaba, R. E. and Moore, J. E. Shortest Paths in Stochastic Networks with Correlated Link Costs. *Computers and Mathematics with Applications* (in press).
7. Frank, H. Shortest Paths in Probabilistic Graphs. *Operations Research*, Vol. 17, 1969, pp. 583-599.
8. Mirchandani, P. B. Shortest Distance and Reliability of Probabilistic Networks. *Computer and Operations Research*, Vol. 3, 1976, pp. 347-355.
9. Sigal, C. E., Pritsker, A. A. B and Solberg J. J. The Stochastic Shortest Route Problem. *Operations Research*, Vol. 28, 1980, pp. 1122-1129.
10. Loui, P. Optimal Paths in Graphs with Stochastic or Multidimensional Weights. *Communications of the ACM*, Vol. 26, 1983, pp. 670-676.
11. Eiger, A., Mirchandani, P., and Soroush, H. Path Preferences and Optimal Paths in Probabilistic Networks. *Transportation Science*, Vol. 19, 1985, pp. 75-84.
12. Murthy, I. A Relaxation-Based Pruning Technique for a Class of Stochastic Shortest Path Problems. *Transportation Science*, Vol. 30, 1996, pp. 220-236.
13. Bellman, R. E. *Dynamic Programming*. Dover Publications, New York, 2003.
14. Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996.
15. Fan, Y., Kalaba R. and Moore J. Arriving on Time, *Journal of Optimization Theory and Applications* (in press).
16. Ohtsubo, Y. Minimizing Risk Models in Stochastic Shortest Path Problems. *Mathematical Methods of Operations Research*, 2003, Vol. 57, 79-88.

17. Chabini, I. Discrete Dynamic Shortest Path Problems in Transportation Applications. *Transportation Research Record* 1645, 1998, pp. 170–175.

**LIST OF TABLES**

TABLE 1	Analytical solutions for the 3-node network
TABLE 2	Optimal solution for the five-node network (destination 5)
TABLE 3	Performance comparisons of two algorithms on grid networks

TABLE 1 Analytical solutions for the 3-node network

Node 2			Node 1		
Time budget	Optimal probability	Optimal policy	Time budget	Optimal probability	Optimal Policy
1	0.0	N/A	1	0.0	N/A
2	0.0	N/A	2	0.4	3
3	0.0	N/A	3	0.4	3
4	0.2	1	4	0.4	3
5	0.2	1	5	0.4	3
6	1.0	3	6	0.4	3
7	1.0	3	7	0.5	2
8	1.0	3	8	0.5	2
9	1.0	3	9	0.5	2
10	1.0	3	10	0.6	2

TABLE 2 Optimal solution for the five-node network (destination node 5)

Time Budget	Origin 1				Origin 2			
	$u_1(t)$		$p_1(t)$		$u_2(t)$		$p_2(t)$	
	SA	IOTB	SA	IOTB	SA	IOTB	SA	IOTB
3.9193	0.9844	0.9844	3	3	0.9844	0.9844	3	3
2.2861	0.8343	0.8343	3	3	0.8983	0.8983	5	5
1.4387	0.5482	0.5482	3	3	0.7626	0.7626	5	5
0.8958	0.2664	0.2664	3	3	0.5914	0.5914	5	5
0.5247	0.0974	0.0974	2	2	0.4079	0.4079	5	5
0.2708	0.0304	0.0304	2	2	0.2366	0.2366	5	5
0.1072	0.0053	0.0053	2	2	0.1015	0.1015	5	5
0.0201	0.0002	0.0002	2	2	0.0198	0.0198	5	5

TABLE 3 Performance comparisons of two algorithms on grid networks

Network	Nodes	Links	CPU Time (seconds)		SA/IOTB
			SA	IOTB	
1	120	380	7.4 (20 <sup>*</sup> )	1.9	3.9
2	440	1560	60.9 (40)	8.1	7.5
3	944	3524	175.7 (51)	18.4	9.5
4	1662	6302	521.4 (84)	33.1	15.8
5	2598	9898	883.6 (91)	51.7	17.1

\*-- The number of iterations required in SA algorithm.

## LIST OF FIGURES

- FIGURE 1 Using SOTA solution for real-time optimal routing
- FIGURE 2 An illustration of a discrete probability expansion network (DPEN)
- FIGURE 3 A three-node network with discrete probability of link travel times
- FIGURE 4 A five-node network with probabilistic link travel costs
- FIGURE 5 Optimal probabilities at origin 1 vs. time resolution for the five-node network
- FIGURE 6 Convergence curves of SA for grid networks

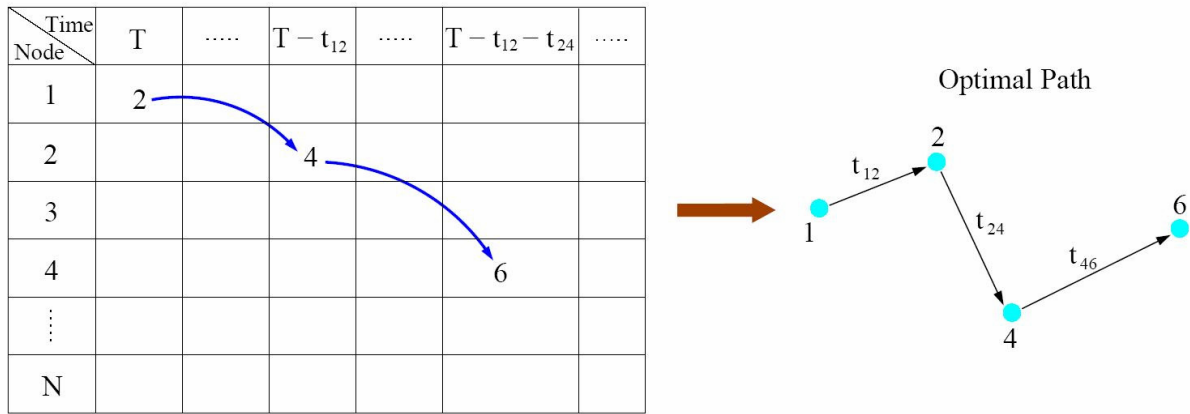


FIGURE 1 Using SOTA solution for real-time optimal routing

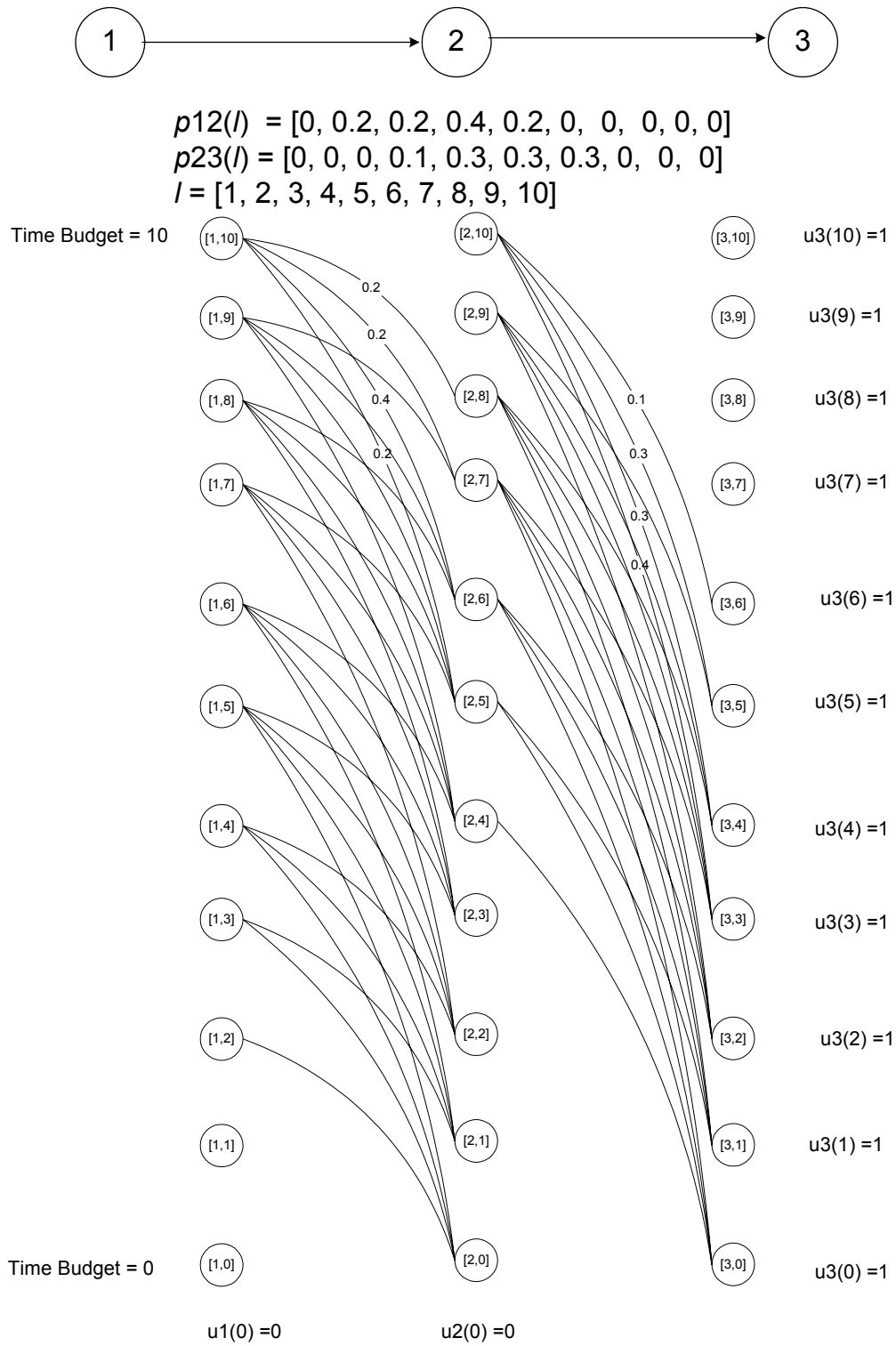
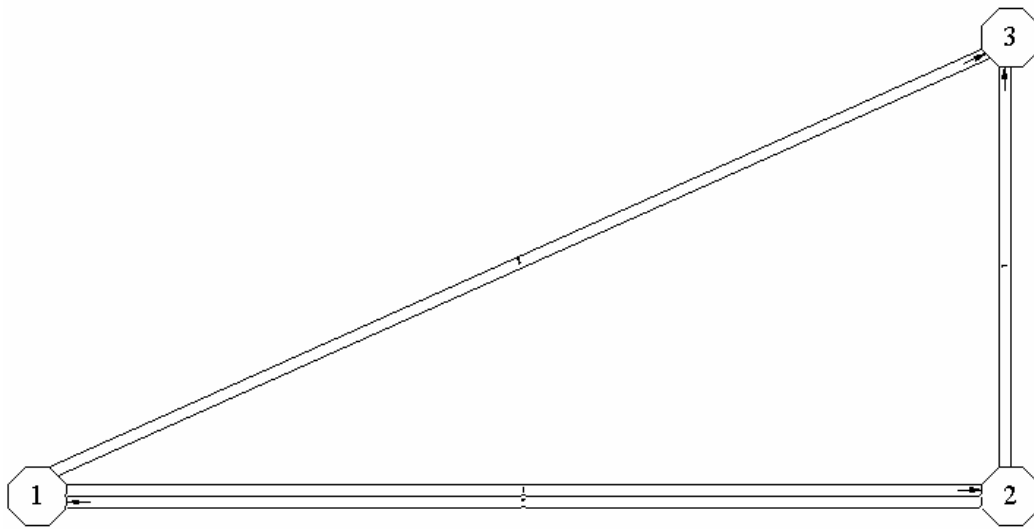


FIGURE 2 An illustration of a discrete probability expansion network (DPEN)



Link	Traversal time $t$ (minute) and the associated probability $p_{ij}(t)$	
1-2	$t=1, p_{ij}(1) = 0.5$	$t=6, p_{ij}(6) = 0.5$
2-1	$t=2, p_{ij}(2) = 0.5$	$t=4, p_{ij}(4) = 0.5$
2-3	$t=4, p_{ij}(4) = 0.1$	$t=6, p_{ij}(6) = 0.9$
1-3	$t=2, p_{ij}(2) = 0.4$	$t=12, p_{ij}(12) = 0.6$

FIGURE 3 A three-node network with discrete probability of link travel times

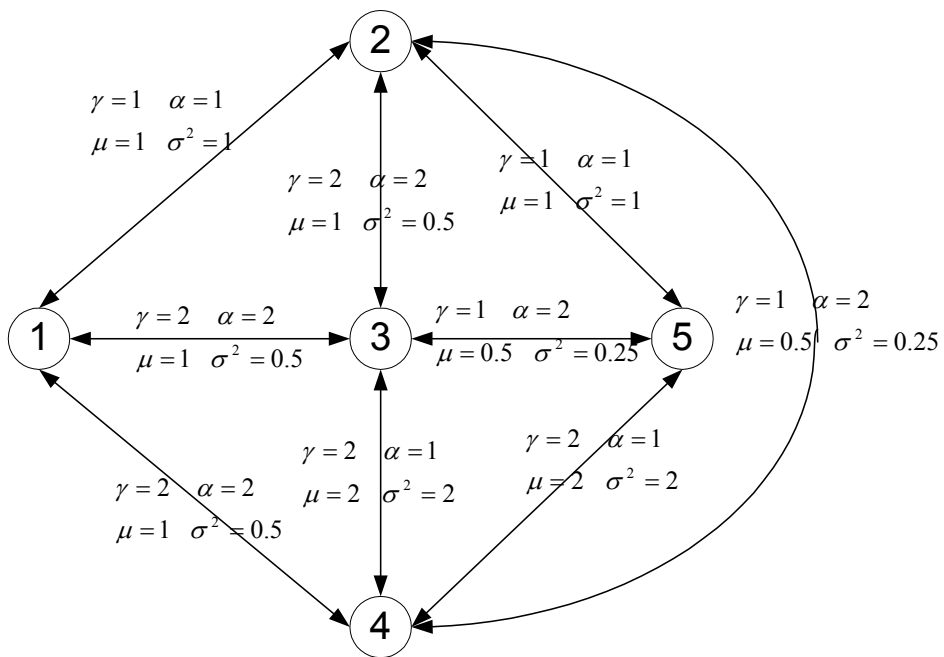


FIGURE 4 A five-node network with probabilistic link travel costs

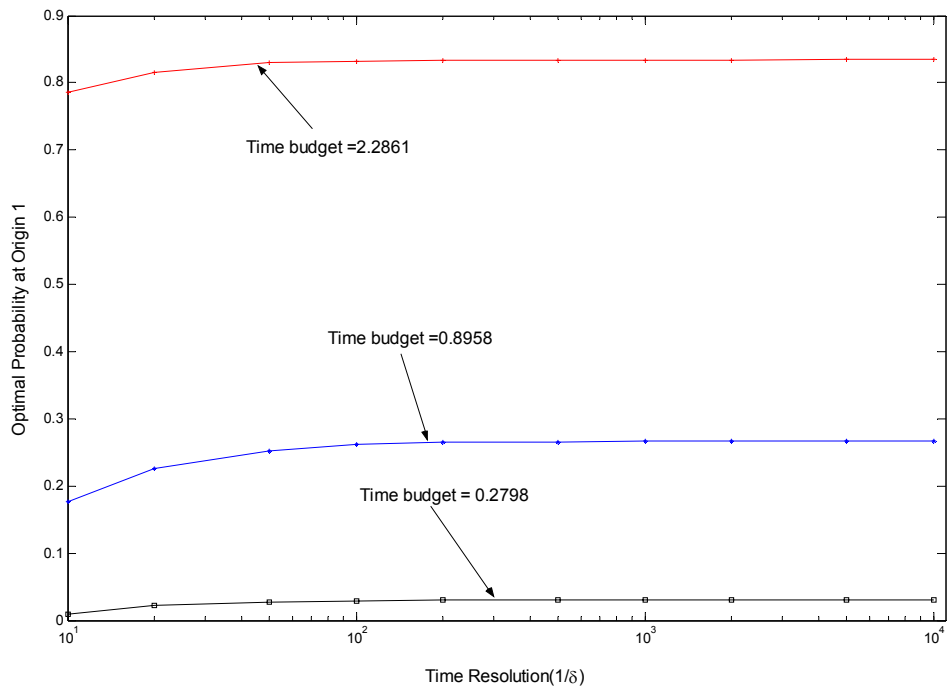


FIGURE 5 Optimal probabilities at origin 1 vs. time resolution for the five-node network

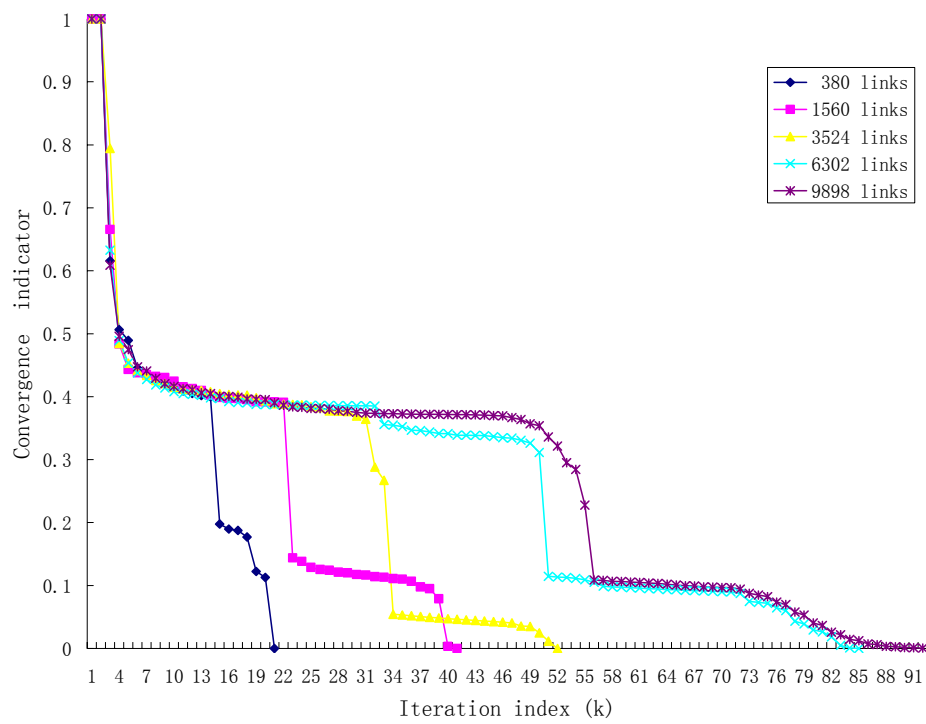


FIGURE 6 Convergence curves of SA for grid networks