View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Approximating the longest path length of a stochastic DAG by a normal distribution in linear time

Ei Ando [a,*], Toshio Nakata [b], Masafumi Yamashita [a,c]

[a] *Dept. Computer Sci. and Communication Eng., Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, 819-0395, Japan*
[b] *Dept. of Mathematics, Fukuoka University of Education, Akama-Bunkyomachi, Munakata, Fukuoka, 811-4192, Japan*
[c] *Institute of Systems, Information Technologies and Nanotechnologies, Fukuoka SRP Center Building 7F 2-1-22, Momochihama, Sawara-ku, Fukuoka, 814-0001, Japan*

ARTICLE INFO

Article history:
Received 3 September 2007
Received in revised form 7 January 2009
Accepted 12 January 2009
Available online 21 January 2009

Keywords:
Directed acyclic graph
Longest path problem
Stochastic edge weight
Normal distribution
Approximation

ABSTRACT

This paper presents a linear time algorithm for approximating, in the sense below, the longest path length of a given directed acyclic graph (DAG), where each edge length is given as a normally distributed random variable. Let $F(x)$ be the distribution function of the longest path length of the DAG. Our algorithm computes the mean and the variance of a normal distribution whose distribution function $\tilde{F}(x)$ satisfies $\tilde{F}(x) \leqslant F(x)$ as long as $F(x) \geqslant a$, given a constant a ($1/2 \leqslant a < 1$). In other words, it computes an upper bound $1 - \tilde{F}(x)$ on the tail probability $1 - F(x)$, provided $x \geqslant F^{-1}(a)$. To evaluate the accuracy of the approximation of $F(x)$ by $\tilde{F}(x)$, we first conduct two experiments using a standard benchmark set ITC'99 of logical circuits, since a typical application of the algorithm is the delay analysis of logical circuits. We also perform a worst case analysis to derive an upper bound on the difference $\tilde{F}^{-1}(a) - F^{-1}(a)$.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

1.1. The longest path problem for stochastic DAGs

Many practical problems are formulated as optimization problems on directed acyclic graphs (DAGs) with edge weights (lengths).[1] Typically the edges represent tasks, each of whose length represents the time necessary to complete it. The DAG represents the precedence relation among the tasks. Assuming that independent tasks can be executed simultaneously, the problem of computing the makespan of a task schedule minimizing the completion time is then reducible to the longest path problem for the DAG, which is solvable in linear time [10], although it is NP-complete for general directed graphs [5].

In cases where problems in real world are too complicated to know the lengths with certainty, we sometimes assign to the edge lengths random variables following suitable distributions, and try to obtain some quantities, e.g., the average of the longest path length, for this stochastic DAG.

PERT [11] and Critical Path Planning [9] are two classical approaches to the stochastic longest path problem. Critical Path Planning simply transforms the stochastic problem into the deterministic problem by taking some constant values as the lengths. PERT on the other hand tries to approximate the expected longest path length, whose accuracy is studied e.g., in [4].

* Corresponding author. Tel. and fax: +81 92 802 3637.
E-mail addresses: ando@tcslab.csce.kyushu-u.ac.jp (E. Ando), nakata@fukuoka-edu.ac.jp (T. Nakata), mak@csce.kyushu-u.ac.jp (M. Yamashita).

[1] In this paper, we consider the longest path problem, where the length of an edge is given by the weight of the edge. We thus prefer term "length" to "weight", as long as the longest path problem is considered.

1570-8667/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.jda.2009.01.001

Martin [12] presented an algorithm for symbolically computing the distribution of the longest path length, assuming that the density functions of the random variables are polynomials. Two key ingredients of his algorithm are a series and a parallel reductions. He symbolically computes the distribution of series edges or parallel edges and repeats to replace these edges in a network with an equivalent single edge, until the network is reduced to a single edge. For a series-parallel graph, he can successfully compute the distribution, although the distribution may be a piecewise polynomial function whose number of pieces is huge. In order to apply this approach to a DAG, he unfolds the DAG to a tree in such a way that each of distinct paths of the DAG from the source to the sink corresponds to an independent path in the tree. Since the number of distinct paths can be as large as an exponential in the order of DAG, his algorithm is inefficient.

Another approach is to consider a discrete version of the problem. We ask each of the random variables to take a discrete value and try to apply a combinatorial optimization technique. However, according to Ball et al. [1], treating arbitrary series of convolutions or max operations is known to be $\#P$-hard, and hence this approach seems to be applicable only to small instances.

In spite of this difficult situation, there are many applications to which *linear time* algorithms are inevitably required, even at the expense of accuracy, because their instances are huge. A typical of such applications is the delay analysis of logical circuits, whose ultimate (and usually unreachable) goal is to compute the distribution of circuit delay. The uncertainty of gate delay comes from manufacturing fluctuation, and a normal distribution is used to model it (see e.g., [2]). Given a success rate a (slightly less than 1.0), their practical goal is hence to estimate a value d such that the probability that the circuit delay is less than d is at least a. We of course want to take d as small as possible.

Difficulty in the calculation of the distribution of the delay mainly arises from parallel gates (which correspond to parallel edges in a DAG); the distribution of the delay of a circuit consisting of two parallel gates is, in general, not a normal distribution any more, even if the delay of each of the gates is independently normally distributed. To avoid this difficulty, Berkelaar [2] proposed a method to approximate the distribution of the delay of two parallel gates by a normal distribution with the same mean and variance as the delay of the two gates. Since his method tends to under-estimate the mean $+ 3\sigma$ point of delay, Hashimoto and Onodera [7] proposed a method to adjust the mean and the variance in order for the approximation to have the same mean $+ 3\sigma$ point as the delay. Tsukiyama [14] proposed a method to obtain the approximate mean and variance of the delay using Clark's method [3], taking into account the correlation among paths. We however note that all of these methods cannot remove the risk of under-estimation.

1.2. Our contributions

Let $F(x)$ be the distribution function of the longest path length of a DAG, where the length of each edge is given as a random variable following a normal distribution. We assume that the edge lengths are mutually independent. In this paper, we present an algorithm A-DAG that computes a normal distribution $N(\tilde{\mu}, \tilde{\sigma}^2)$ whose distribution function $\tilde{F}(x)$ satisfies that $\tilde{F}(x) \leqslant F(x)$ if $F(x) \geqslant a$, given a $(1/2 \leqslant a < 1)$. In the context of circuit delay analysis, since $F(x)$ is the probability that the circuit delay is less than or equal to x, the solution $x = d$ of equation $\tilde{F}(x) = a$ guarantees that the delay of a product is less than or equal to d with the probability at least a, where d can be easily calculated by referring to the table of standard normal distribution, since $\tilde{F}(x)$ is a normal distribution function.

We can restate our result as follows: The tail probability is the probability that a random variable deviates at least a given amount from the expectation. Many studies have been done to obtain upper bounds on it. The Chernoff bound on the sum of independent Poisson trials is an example (see, e.g., [13, Section 4.1]). In our setting, the Chernoff bound is an upper bound on the tail probability of the length of (a longest path of) a path graph, although the distribution of the length of each edge is Poisson. For any DAG and constant a $(1/2 \leqslant a < 1)$, our algorithm computes a normal distribution function $\tilde{F}(x)$ such that $1 - \tilde{F}(x)$ is an upper bound on the tail probability $1 - F(x)$ as long as $x \geqslant a$.

Let $F_i(x)$ $(i = 1, \dots, k)$ be the normal distribution function given by a pair of a mean μ_i and a variance σ_i^2. Then our algorithm uses, as a primitive operator, a function $\mathrm{Inv}(F_1, F_2, \dots, F_k; y)$ that returns a (unique) real number x_0 such that $F_1(x_0)F_2(x_0)\dots F_k(x_0) = y$ holds for any given real number y $(0 < y < 1)$. $\mathrm{Inv}(F_1, F_2, \dots, F_k; y)$ can be calculated by adopting a numerical method like the Newton's method or the binary search method referring to the table of the standard normal distribution. Provided that Inv returns a correct value in $O(k)$ time,[2] the time complexity of our algorithm is $O(|V| + |E|)$, where $G = (V, E)$ is the given DAG.

From the view of practice, a small upper bound on $F^{-1}(a)$ is looked for. Since the circuit delay analysis is a main application, we conduct experiments to observe the accuracy measured by $\mathrm{Err}(G, a) = \tilde{F}^{-1}(a) - F^{-1}(a)$ of $\tilde{F}^{-1}(a)$ for the large circuits in ITC'99 benchmark set. A-DAG shows sufficiently good performance except for a couple of instances. We formally derive an upper bound on Err, too.

1.3. Road map

To explain the ideas behind our liner time algorithm A-DAG, we start with a well-known algorithm D-DAG for computing the longest path length of a DAG $G = (V, E)$, in which a constant non-negative value c_{ji} is associated with each edge

[2] This assumption essentially assumes that the Newton's method or the binary search method terminates in a constant number of repetitions and produces the correct root up to the required level of accuracy. See Appendix A for a formal justification.

$(v_j, v_i) \in E$. D-DAG is based on the dynamic programming method, and computes, for each vertex $v_i \in V$ in a topologically sorted order, the longest path length l_i from the source to v_i.

```
Algorithm D-DAG($G$)
Topologically sort $V = \{v_1, v_2, \ldots, v_n\}$;
(Assume that $v_i < v_{i+1}$.)
$l_1 := 0$;
for $i = 2, \ldots, n$ do;
$l_i := \max_{v_j \in V_i}\{l_j + c_{ji}\}$, where $V_i$ is the set of parent vertices of $v_i$;
return $l_n$.
```

A-DAG tries to maintain "an approximate distribution C_i of the longest path length" from the source to vertex $v_i \in V_i$, instead of "the definite longest path length l_i" in D-DAG. For this, we re-define the operations *plus* and *max* in D-DAG for normal distributions in such a way that C_n becomes an approximation of the distribution of l_n. Since the sum of two normally distributed random variables also obey a normal distribution, the mean and the variance of which are the sums of the means and the variances of the two normal distributions, we can immediately replace operation *plus* with two *plus*s, one for mean and the other for variance.

Technical details of A-DAG arise from how to implement *max* operation. To this end, we introduce three subclasses of DAG called quasi-tree graphs (QTREE), racket graphs (RACKET) and parallel graphs (PARA), and transform the problem of computing C_i for a DAG finally into the problem of approximating, by a normal distribution, the distribution of the longest path length of a PARA, using first QTREE and then RACKET.

We however describe this transformation in the reverse direction. We explain 1) how to approximate the distribution of the longest path length of a PARA by a normal distribution (Section 3.1), 2) how to transform a RACKET into a PARA (Section 3.2), 3) how to transform a QTREE into a RACKET (Section 3.3), and finally 4) how to transform a DAG into a QTREE (Section 4), in this order.

The paper is organized as follows: After preparing basic notions in Section 2, Section 3 introduces three subclasses PARA, RACKET and QTREE, and describes the above transformations. Then we in Section 4 present our algorithm A-DAG. Section 5 is devoted to performance analysis of A-DAG. We first explain about experimental results using ITC'99 as instances, and then formally bound error Err. Finally we conclude the paper by giving some open problems in Section 6.

2. Preliminaries

Consider a DAG $G = (V, E)$ with a pair of a source and a sink. For each edge $e_i \in E$ of G, we associate a random variable X_i that represents the length of e_i, and assume that $X_i \sim N(\mu_i, \sigma_i^2)$, where $N(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2. We assume that the random variables are mutually independent. The (probability) density function of X_i is

$$f_i(x) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2\right),$$

and its distribution function is

$$F_i(x) = \int_{-\infty}^{x} f_i(t)\,dt.$$

$N(0,1)$ is called the standard normal distribution. Let $\Phi(x)$ and $\phi(x)$ be the distribution function and the density function of $N(0,1)$. Then we have

$$F_i(x) = \Phi\left(\frac{x-\mu_i}{\sigma_i}\right).$$

Since each edge length is given as a random variable, the length of a longest path from the source to the sink is also a random variable. Before starting the introduction of our approximation algorithm, we would like to explain how to calculate in theory the distribution of the longest path length of a series-parallel graph, to give readers an intuition about the difficulty of the problem.

Let e_1 and e_2 be two edges with edge lengths X_1 and X_2, respectively. We assume that X_1 and X_2 are mutually independent. Suppose that two edges e_1 and e_2 are connected in series. The length X of the longest path is then $X = X_1 + X_2$ and its distribution function $F(x)$ is

$$F(x) = P(X_1 + X_2 \leqslant x) = \int_{-\infty}^{\infty} f_1(x-t)F_2(t)\,dt.$$

If $X_i \sim N(\mu_i, \sigma_i^2)$ for $i = 1, 2$, then $X \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$, and we can easily calculate $F(x)$.

Suppose that e_1 and e_2 are connected in parallel. The length X of a longest path is $X = \max\{X_1, X_2\}$, whose distribution function $F(x)$ is given by

$$F(x) = P\bigl(\max\{X_1, X_2\} \leqslant x\bigr) = F_1(x)F_2(x).$$

Note that X is not normally distributed, even if both X_1 and X_2 follow normal distributions. Hence we cannot identify $F(x)$ in terms of a mean and a variance, unlike the case of series edges.

The distribution function of the longest path length of a series-parallel graph can be numerically calculated by repeatedly calculating the above two formulae (after generalizing them to more than 2 random variable cases), but it is by no means easy, since the longest path length of a sub-graph is not normally distributed. If the longest path length of parallel edges were to be normally distributed, the calculation for series-parallel graphs would be quite simple. This motivates our Theorem 3, which approximates the distribution of the longest path length of parallel edges by an appropriate normal distribution. However, as will be seen, Theorem 3 alone is not strong enough even for very simple series-parallel graphs. In the next section, we give key ideas behind our algorithm by developing approximation algorithms for three restricted classes, the parallel graphs, the racket graphs, and the quasi-tree graphs, of DAGs.

3. Key ideas

3.1. Parallel graphs

Let $F(x)$ and $\tilde{F}(x)$ be functions and let x_0 be a fixed value. If $F(x) \geqslant \tilde{F}(x)$ for any $x \geqslant x_0$, we say that $\tilde{F}(x)$ *approximates* $F(x)$ *with respect to* x_0. Our goal is to find a $\tilde{F}(x)$ that approximates $F(x)$ with respect to $F^{-1}(a)$, for a given a $(1/2 \leqslant a < 1)$.

We first consider a DAG $G = (V, E)$ consisting only of two parallel edges, i.e., $V = \{u, v\}$ and $E = \{e_1, e_2\}$ where $e_i = (u, v)$, $i = 1, 2$. As in Section 2, let X_1 and X_2 be random variables assigned to e_1 and e_2, respectively, where $X_i \sim N(\mu_i, \sigma_i^2)$ for $i = 1, 2$. We assume without loss of generality that $\sigma_1 \geqslant \sigma_2$. Let $f_i(x)$ and $F_i(x)$ be the density and the distribution functions of X_i $(i = 1, 2)$, respectively; then $F(x) = F_1(x)F_2(x)$ is the distribution function of $X = \max\{X_1, X_2\}$. Let $\tilde{f}(x)$ and $\tilde{F}(x)$ respectively be the density function and the distribution function of $N(\tilde{\mu}, \tilde{\sigma}^2)$.

Suppose that a real number x_0 is given. For $\tilde{\sigma} = \sigma_1$ and $\tilde{\mu} = x_0 - \sigma_1 \Phi^{-1}(F(x_0))$, $\tilde{F}(x)$ approximates $F(x)$ with respect to x_0. We show this fact in the following by investigating $H(x) = F(x) - \tilde{F}(x)$.

Lemma 1. *For any $N(\mu_2, \sigma_2^2)$, σ_1 that satisfies $\sigma_1 \geqslant \sigma_2$ and any real number x_0, there is a $\tilde{\mu}_2$ such that the distribution function $\tilde{F}_2(x)$ of $N(\tilde{\mu}_2, \sigma_1^2)$ approximates $F_2(x)$ with respect to x_0.*

Proof. First observe that equation $F_2(x) - \tilde{F}_2(x) = 0$ has a unique root x_0 if $\tilde{\mu}_2 = x_0 - \sigma_1 \Phi^{-1}(F_2(x_0))$. Then it is obvious to show that $\tilde{F}_2(x) \leqslant F_2(x)$ holds for $x \geqslant x_0$, since $\sigma_2 \leqslant \sigma_1$ by assumption. □

Since $\tilde{F}_2(x)$ approximates $F_2(x)$ with respect to x_0 by Lemma 1, $F_1(x)\tilde{F}_2(x)$ approximates $F_1(x)F_2(x)$ with respect to x_0. It is thus sufficient to show that $\tilde{F}(x)$ approximates $F_1(x)F_2(x)$ with respect to x_0, provided that $\sigma_2 = \sigma_1 = \sigma$. Let $\tilde{\mu} = x_0 - \sigma \Phi^{-1}(F(x_0))$ and $\tilde{\sigma} = \sigma$.

Lemma 2. $H(x_0) = 0$.

Proof. Since $\tilde{\mu} = x_0 - \sigma \Phi^{-1}(F(x_0))$, $\tilde{F}(x_0) = \Phi((x_0 - \tilde{\mu})/\sigma)$, which implies that $H(x_0) = 0$. □

Lemma 3. $\tilde{\mu} > \max\{\mu_1, \mu_2\}$.

Proof. We assume $\mu_1 \geqslant \mu_2$ without loss of generality and show $\tilde{\mu} > \mu_1$. Since $F_2(x) < 1$, $F(x) < F_1(x)$. Suppose $\tilde{\mu} \leqslant \mu_1$. Since $F_1(x)$ and $\tilde{F}(x)$ are normal distribution functions sharing variance σ^2, $F(x) < F_1(x) \leqslant \tilde{F}(x)$ holds, a contradiction by Lemma 2. □

In the following, we concentrate on showing that equation $H(x) = 0$ has no roots greater than x_0. Consider the derivative $h(x)$ of $H(x)$ and let

$$h(x) = H'(x) = f_1(x)F_2(x) + F_1(x)f_2(x) - \tilde{f}(x).$$

Lemma 4. *There exists a constant x^+ such that $h(x) < 0$ holds for all $x > x^+$.*

Proof. By definition,

$$h(x)=\tilde{f}(x)\Bigg(\underbrace{\frac{f_1(x)}{\tilde{f}(x)}F_2(x)}_{A(x)}+\underbrace{\frac{f_2(x)}{\tilde{f}(x)}F_1(x)}_{B(x)}-1\Bigg).$$

As for $A(x)$, since $\tilde{\sigma}=\sigma_1=\sigma$,

$$A(x)=\frac{f_1(x)}{\tilde{f}(x)}F_2(x)=\exp\bigl(-(c_1x-c_2)\bigr)F_2(x),$$

where $c_1=(\tilde{\mu}-\mu_1)/\sigma^2$ and $c_2=(\tilde{\mu}^2-{\mu_1}^2)/2\sigma^2$. Since $F_2(x)<1$ and $\tilde{\mu}>\mu_1$ by Lemma 3, $\lim_{x\to+\infty}A(x)=0$.

Since $\lim_{x\to+\infty}B(x)=0$ holds by a similar argument, there exists a constant x^+ such that $A(x)+B(x)<1$ for all $x>x^+$, which completes the proof, since $\tilde{f}(x)$ is positive. □

Corollary 1. *There is a constant w^+ such that $H(x)>0$ holds for all $x>w^+$.*

Proof. Observe that $\lim_{x\to+\infty}H(x)=0$. Thus the corollary holds by Lemma 4. □

We now start counting the number of roots of equation $h(x)=0$ by using the following technical lemma.

Lemma 5. *Let $g(x)$ and $G(x)$ be the probabilistic density and the distribution functions of a normal distribution $N(\mu,\sigma^2)$. Then $K(x)=G(x)/g(x)$ is positive strictly monotonically increasing.*

Proof. Clearly, $K(x)$ is positive. Consider the derivative $k(x)$ of $K(x)$. Then

$$k(x)=K'(x)=\frac{1}{g(x)}\underbrace{\left(g(x)+\left(\frac{x-\mu}{\sigma^2}\right)G(x)\right)}_{A(x)}.$$

Since

$$\frac{d}{dx}A(x)=\frac{d}{dx}\left(g(x)+\left(\frac{x-\mu}{\sigma^2}\right)G(x)\right)=\frac{1}{\sigma^2}G(x)>0,$$

$A(x)$ is monotonically increasing. Since

$$\lim_{x\to-\infty}xG(x)=\lim_{x\to-\infty}\frac{g(x)}{x^{-2}}=0$$

by l'Hôpital's rule, we have

$$\lim_{x\to-\infty}\left(g(x)+\left(\frac{x-\mu}{\sigma^2}\right)G(x)\right)=0,$$

which implies that $A(x)$ and hence $k(x)$ are positive. Thus $K(x)$ is strictly monotonically increasing. □

Next lemma is used to bound the number of extremal points of $H(x)$.

Lemma 6. *Equation $h(x)=0$ has at most two roots.*

Proof. Instead of $h(x)$, we analyze $L(x)$, where

$$L(x)=\ln\bigl(f_1(x)F_2(x)+F_1(x)f_2(x)\bigr)-\ln\tilde{f}(x).$$

Consider the derivative $L'(x)$ of $L(x)$ and equation $L'(x)=0$, which leads to

$$2-\frac{1}{\sigma^2}(\tilde{\mu}-\mu_2)\frac{F_1(x)}{f_1(x)}-\frac{1}{\sigma^2}(\tilde{\mu}-\mu_1)\frac{F_2(x)}{f_2(x)}=0. \tag{1}$$

Since $F_1(x)/f_1(x)$ and $F_2(x)/f_2(x)$ are positive monotonically increasing functions by Lemma 5 and $\tilde{\mu}>\max\{\mu_1,\mu_2\}$ (otherwise $\tilde{F}(x)$ does not intersect with $F_1(x)F_2(x)$), the left-hand side of (1) is strictly monotonically decreasing, and hence equation $L'(x)=0$ has at most one root for $x\geqslant x_0$. Thus $L(x)=0$ has at most two roots, and so does the equation $h(x)=0$. □

We are ready to show the following theorem.

Theorem 1. *Let x_0 be any real number. For $\tilde{\sigma} = \sigma_1$ and $\tilde{\mu} = x_0 - \sigma_1 \Phi^{-1}(F(x_0))$, $\tilde{F}(x)$ approximates $F(x)$ with respect to x_0.*

Proof. Equation $h(x) = 0$ has at most two roots by Lemma 6. Since $\lim_{x \to -\infty} H(x) = 0$, $H(x_0) = 0$, and $\lim_{x \to +\infty} H(x) = 0$, if $H(x) = 0$ had a root greater than x_0, equation $h(x) = 0$ would have at least three roots. Hence $H(x) \geqslant 0$ for all $x \geqslant x_0$. □

Given a real number y $(0 < y < 1)$, we introduce a function $\mathrm{Inv}(F_1, F_2, \ldots, F_k; y)$ that returns a (unique) real number x_0 such that $F_1(x_0)F_2(x_0)\ldots F_k(x_0) = y$ holds. We also use Inv to denote the inverse of a function and let $\mathrm{Inv}(F_1; y) = F_1^{-1}(y) = x_0$. In this paper, we assume that $\mathrm{Inv}(F_1, F_2, \ldots, F_k; y)$ returns x_0 in $O(k)$ time. (See Appendix A for justification.) Now we describe an algorithm 2-PARA that calculates $\tilde{F}(x)$ in terms of its mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$, given F_1, F_2 and a real number a $(0 < a < 1)$.

```
**Algorithm** 2-PARA($F_1, F_2, a$):
$x_0 := \mathrm{Inv}(F_1, F_2; a)$;
$\tilde{\sigma} := \max\{\sigma_1, \sigma_2\}$;
$\tilde{\mu} := x_0 - \tilde{\sigma}\,\mathrm{Inv}(\Phi; a)$;
**return** $(\tilde{\mu}, \tilde{\sigma}^2)$.
```

Theorem 2. *Algorithm* 2-PARA *correctly computes $\tilde{F}(x)$ that approximates $F(x) = F_1(x)F_2(x)$ with respect to $F^{-1}(a)$, in $O(1)$ time.*

Proof. Since $x_0 = F^{-1}(a)$, it is sufficient to check that $\tilde{\mu}$ and $\tilde{\sigma}$ satisfy the conditions of Theorem 1. Clearly $\tilde{\sigma}$ is correctly chosen. Since $\mathrm{Inv}(\Phi; a) = \Phi^{-1}(F(x_0))$, $\tilde{\mu}$ is correctly chosen as well. The time complexity $O(1)$ of 2-PARA is trivial by assumption on Inv. □

We next generalize 2-PARA to propose an Algorithm A-PARA that can treat a general parallel graph G with k parallel edges e_i $(1 \leqslant i \leqslant k)$, whose lengths are given by mutually independent random variables X_i $(1 \leqslant i \leqslant k)$ that obey normal distributions $N(\mu_i, \sigma_i^2)$ $(1 \leqslant i \leqslant k)$. Let $F_i(x)$ be the distribution function of $N(\mu_i, \sigma_i^2)$. Then $F(x) = \Pi_{1 \leqslant i \leqslant k} F_i(x)$ is the distribution function of the longest path length of G. We now present Algorithm A-PARA, which calculates a normal distribution $N(\tilde{\mu}, \tilde{\sigma}^2)$ whose distribution function $\tilde{F}(x)$ approximates $F(x)$ with respect to $F^{-1}(a)$.

```
**Algorithm** A-PARA($G, a$):
$x_0 := \mathrm{Inv}(F; a)$;
$\tilde{\sigma} := \max_{1 \leqslant i \leqslant k}\{\sigma_i\}$;
$\tilde{\mu} := x_0 - \tilde{\sigma}\,\mathrm{Inv}(\Phi; a)$;
**return** $(\tilde{\mu}, \tilde{\sigma}^2)$.
```

Theorem 3. *For a given real number a $(0 < a < 1)$,* A-PARA *returns a normal distribution function $\tilde{F}(x)$ that approximates $F(x)$ with respect to $F^{-1}(a)$.* A-PARA *runs in $O(k)$ time.*

Proof. We first show that the time complexity of A-PARA is $O(k)$. Since $F(x) = \Pi_{1 \leqslant i \leqslant k} F_i(x)$, x_0 is calculated in $O(k)$ time by the assumption on Inv. The calculation of $\tilde{\sigma}$ needs $O(k)$ time, and that of $\tilde{\mu}$ needs $O(1)$ time.

In order to show the theorem, we show a slightly stronger claim by induction on k: $\tilde{F}(x)$ approximates $F(x)$ with respect to $F^{-1}(a)$ and $\tilde{F}(x_0) = F(x_0) = a$ holds. When $k = 1$, $\tilde{\mu} = \mu$ and $\tilde{\sigma} = \sigma$ hold, and thus $\tilde{F}(x) = F(x)$. When $k = 2$, Theorem 2 guarantees the first part of the claim. To observe that $\tilde{F}(x_0) = F(x_0) = a$, we calculate

$$\begin{aligned}
\tilde{F}(x_0) &= \Phi\left(\frac{x_0 - \tilde{\mu}}{\tilde{\sigma}}\right) \\
&= \Phi\left(\frac{x_0 - (x_0 - \tilde{\sigma}\Phi^{-1}(F(x_0)))}{\tilde{\sigma}}\right) \\
&= F(x_0).
\end{aligned}$$

We thus concentrate on the induction step.

Let G^- be the parallel graph constructed from G by removing the k-th edge. Then the distribution function of the longest path length of G^- is given by $H(x) = \prod_{1 \leqslant i \leqslant k-1} F_i(x)$. By induction hypothesis, for $x_0 = F^{-1}(a)$, the output $\tilde{H}(x)$ of A-PARA($G^-, H(x_0)$) approximates $H(x)$ with respect to x_0 $(= H^{-1}(H(x_0)))$ and $\tilde{H}(x_0) = H(x_0)$ holds, since $0 < H(x_0) < 1$.

Let $K(x)$ be the output of 2-PARA($\tilde{H}, F_k, \tilde{H}(x_0)F_k(x_0)$). Then $K(x)$ is an approximation of $F(x) = \Pi_{1 \leqslant i \leqslant k} F_i(x) = H(x)F_k(x)$ with respect to $F^{-1}(a)$ since $\tilde{H}(x_0)F_k(x_0) = H(x_0)F_k(x_0) = F(x_0) = a$ and $K(x) \leqslant \tilde{H}(x)F_k(x) \leqslant H(x)F_k(x) = F(x)$ for all $x \geqslant x_0$. For 2-PARA (or the case of $k = 2$), we know that $K(x_0) = \tilde{H}(x_0)F_k(x_0) = F(x_0)$. Thus all what we need is to show $\tilde{F}(x) = K(x)$.

First, $\tilde{F}(x)$ and $K(x)$ share variance $\tilde{\sigma} = \max_{1 \leqslant i \leqslant k}\{\sigma_i\}$. Since $\tilde{F}(x_0) = F(x_0)$ by the same argument as in case $k = 2$, we have $\tilde{F}(x_0) = K(x_0)$. Thus $\tilde{F}(x) = K(x)$ holds. □

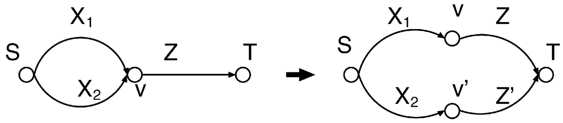

Fig. 1. A racket graph and its transformation to a parallel graph.

3.2. *Racket graphs*

Since the longest path length of a parallel graph can be approximated by A-PARA, we proceed to a slightly more complex class of DAGs, which will be referred to as *racket graphs*. A simple racket graph consists only of three vertices and three edges (see Fig. 1 (left)). In the figure, X_i's and Z are mutually independent random variables associated with the edges. Let $Z \sim N(\mu_Z, \sigma_Z^2)$. By Theorem 1, we calculate a normal distribution $N(\tilde{\mu}, \tilde{\sigma}^2)$ that approximates $\max\{X_1, X_2\}$ with respect to some given x_0. Then we optimistically hope that $N(\tilde{\mu} + \mu_Z, \tilde{\sigma}^2 + \sigma_Z^2)$ would correctly approximate the distribution of $\max\{X_1, X_2\} + Z$ with respect to x_0, which is *not* always correct. For example, consider a case where $X_1 \sim N(0,1)$, $X_2 \sim N(0,0)$ and $Z \sim N(0,1)$; A-PARA returns $\tilde{\mu} = 0$ and $\tilde{\sigma} = 1$ in this example, if parameter a is greater than 1/2. Now the distribution function of $N(\tilde{\mu} + \mu_Z, \tilde{\sigma}^2 + \sigma_Z^2) = N(0,2)$ is greater than the distribution function of $\max\{X_1, X_2\} + Z$ for any x, which does not meet our definition of approximation. Theorem 3 alone is not sufficient to process even such a simple DAG.

To avoid this problem, we consider a graph shown in Fig. 1 (right), which is obtained from Fig. 1 (left) by duplicating vertex v and the outgoing edge. Random variables X_1, X_2 and Z follow the same normal distributions as those in Fig. 1 (left), and are assumed to be mutually independent. New random variable Z' follows the same distribution as Z, and is assumed to be mutually independent with X_1, X_2 and Z.

Let $F_a(x)$ (resp. $F_b(x)$) be the distribution function of the longest path length of graph in Fig. 1 (left) (resp. Fig. 1 (right)). As Property 1 below claims in a general form, $F_a(x) \geqslant F_b(x)$ holds for any x, since Z and Z' follow the same distribution. Then $\tilde{F}(x)$ approximates $F_a(x)$ if $\tilde{F}(x)$ approximates $F_b(x)$ with respect to a given x_0. A good news is that Theorem 3 guarantees that A-PARA can compute such $\tilde{F}(x)$, since the lengths of two paths of graph in Fig. 1 (right) follow normal distributions $N(\mu_1 + \mu_Z, \sigma_1^2 + \sigma_Z^2)$ and $N(\mu_2 + \mu_Z, \sigma_2^2 + \sigma_Z^2)$, respectively.

Consider a generalization of the graph in Fig. 1 (left) such that there are k edges between S and v. Those graphs are referred to as *racket graphs*. Let us call the parallel edges between S and v *head* and the edge between v and T *shaft*. As explained in above, we make use of a transformation from a racket graph to a parallel graphs. We introduce it in a more general form.

Let $G = (V, E)$ be a DAG; X_e, a random variable associated with each edge $e \in E$; and $N(\mu_e, \sigma_e^2)$, a normal distribution that X_e follows. By $P = \mathrm{Par}(G)$, we denote the parallel graph with $|\Pi|$ multiple edges, where Π is the set of paths π in G that connect source S and sink T. For each of edges e_π, a random variable Y_π is associated with, which follows a normal distribution $N(\mu_\pi, \sigma_\pi^2)$, where $\mu_\pi = \sum_{e \in \pi} \mu_e$ and $\sigma_\pi^2 = \sum_{e \in \pi} \sigma_e^2$. The following property seems to be well-known,[3] but we provide a proof for the convenience of readers.

Property 1. *Let $F(x)$ (resp. $F_P(x)$) be the distribution function of the longest path length of G (resp. $P = \mathrm{Par}(G)$). Then $F_P(x) \leqslant F(x)$ for all x.*

Proof. We only show the simplest case in which G is given in Fig. 1 (left). A general case can be shown by using a structural induction, with this simplest case as the base case. $P = \mathrm{Par}(G)$ is hence the parallel graph in Fig. 1 (right). We show

$$P\big(\max\{X_1 + Z, X_2 + Z'\} > x\big) \geqslant P\big(\max\{X_1 + Z, X_2 + Z\} > x\big).$$

Let F_{X_1} and F_{X_2} be the distribution functions of X_1 and X_2, respectively. By definition, the left-hand side is

$$L = \iint_{\{(x_1,x_2) \in \mathbf{R}^2\}} P\big(\max\{x_1 + Z, x_2 + Z'\} > x \mid X_1 = x_1,\ X_2 = x_2\big)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2),$$

which is reducible to

$$\iint_{\{(x_1,x_2) \in \mathbf{R}^2\}} P\big(\max\{x_1 + Z, x_2 + Z'\} > x\big)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2),$$

since X_1, X_2, Z and Z' are mutually independent.

By the same reason, the right-hand side is

[3] Although we deal with only normally distributed edge lengths, Property 1 holds for arbitrarily distributed edge lengths.

$$R = \iint\limits_{\{(x_1,x_2)\in \mathbf{R}^2\}} P\bigl(\max\{x_1+Z, x_2+Z\} > x\bigr)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2).$$

Let $E_1 = \{(x_1, x_2) \in \mathbf{R}^2\colon\ x_1 > x_2\}$ and $E_2 = \{(x_1, x_2) \in \mathbf{R}^2\colon\ x_1 \leqslant x_2\}$. Then $L = L_1 + L_2$ and $R = R_1 + R_2$, where

$$L_1 = \iint\limits_{E_1} P\bigl(\max\{x_1+Z, x_2+Z'\} > x\bigr)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2),$$

$$L_2 = \iint\limits_{E_2} P\bigl(\max\{x_1+Z, x_2+Z'\} > x\bigr)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2),$$

$$R_1 = \iint\limits_{E_1} P\bigl(\max\{x_1+Z, x_2+Z\} > x\bigr)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2),$$

and

$$R_2 = \iint\limits_{E_2} P\bigl(\max\{x_1+Z, x_2+Z\} > x\bigr)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2).$$

We have $L_1 \geqslant R_1$, since

$$R_1 = \iint\limits_{E_1} P(x_1+Z > x)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2)$$

and $P(x_1+Z > x) \leqslant P(\max\{x_1+Z, x_2+Z'\} > x)$. Then we have $L_2 \geqslant R_2$, since

$$R_2 = \iint\limits_{E_2} P(x_2+Z > x)\, dF_{X_1}(x_1)\, dF_{X_2}(x_2)$$

and $P(x_2+Z > x) = P(x_2+Z' > x)$. □

Suppose that a racket graph R and a real number a $(0 < a < 1)$ are given. We first construct a parallel graph $P = \mathrm{Par}(R)$ and next compute a normal distribution function $\tilde{F}(x)$ that approximates $F_P(x)$ by Algorithm A-PARA. Then $\tilde{F}(x)$ approximates $F_R(x)$ by Property 1. Let us call this algorithm A-RACKET.

```
**Algorithm** A-RACKET($R, a$):
$P := \mathrm{Par}(R)$;
Call A-PARA($P, a$), which returns $(\tilde{\mu}, \tilde{\sigma}^2)$;
**return** $(\tilde{\mu}, \tilde{\sigma}^2)$.
```

Theorem 4. *Let R and $F_R(x)$ be a racket graph and the distribution function of the longest path length of R, respectively. For any given real number a $(0 < a < 1)$,* A-RACKET *returns a normal distribution function $\tilde{F}(x)$ that approximates $F_R(x)$ with respect to $F_R^{-1}(a)$.* A-RACKET *runs in $O(m)$ time, where m is the size (i.e., the number of edges) of R.*

Proof. By Theorem 3, $\tilde{F}(x)$ approximates $F_P(x)$ with respect to $F_P^{-1}(a)$, where $F_P(x)$ is the distribution function of the longest path length of $P (= \mathrm{Par}(R))$. For all x such that $F_P(x) \geqslant a$, $\tilde{F}(x) \leqslant F_P(x) \leqslant F_R(x)$ by Property 1, which implies that $\tilde{F}(x) \leqslant F_R(x)$ for all x such that $F_R(x) \geqslant a$.

As for the time complexity, we can construct P in $O(m)$ time, and A-PARA(P, a) requires $O(m)$ time. □

Note that, by Property 1, the distribution function $F(x)$ of the longest path length of a DAG G can be approximated by applying A-PARA to $\mathrm{Par}(G)$. However the time complexity $O(k)$ of A-PARA in this case is as large as $\Omega(2^n)$ in the worst case, where n is the order of G, since k is the number of paths in G (and hence the number of edges of $\mathrm{Par}(G)$). Our Algorithm A-DAG proposed in Section 4 implements this idea in linear time using a dynamic programming technique.

3.3. Quasi-tree graphs

We discuss quasi-tree graphs in this subsection. Fig. 2 illustrates a quasi-tree graph. A *quasi-tree graph* consists of an in-tree with the root (hence sink) T and the source S, from which there is an edge to each of the leaves of the in-tree. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of a quasi-tree graph $G = (V, E)$, where $v_1 = S$ and $v_n = T$. A vertex v_i (resp. v_j) is called *a parent* (resp. *a child*) of vertex v_j (resp. v_i) if there is an edge (v_i, v_j) in E. We denote the set of all parents of a vertex v_i by V_i.

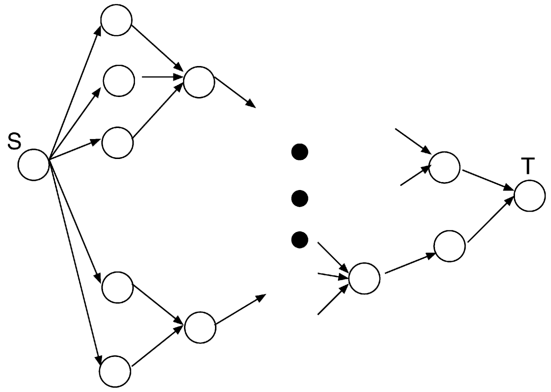

Fig. 2. A quasi-tree graph.

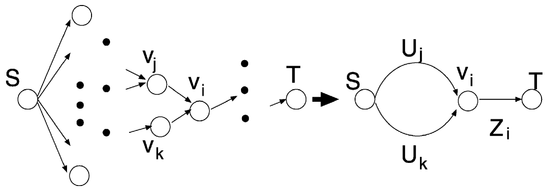

Fig. 3. Transformation of a general quasi-tree graph.

For a vertex v_i ($\neq S$), let G_i be the subgraph of G induced by the set of paths in G connecting S and T through v_i, and we denote by Y_i the length of a longest path of G_i. Let F_{Y_i} be the distribution function of Y_i. Since $G_n = G$, we are approximating F_{Y_n}. By W_{ij} we denote the longest path length between v_i and v_j in G. The path π_i from v_i to T in G_i is unique. We denote the length of π_i by Z_i. Then Z_i follows a normal distribution $N(\mu_{Z_i}, \sigma^2_{Z_i})$; i.e., $\mu_{Z_i} = \sum_{e\in\pi_i} \mu_e$ and $\sigma^2_{Z_i} = \sum_{e\in\pi_i} \sigma^2_e$, where $X_e \sim N(\mu_e, \sigma^2_e)$. By definition, we have $Y_i = W_{1i} + Z_i$. Algorithm A-QTREE computes for each i a normal distribution $D_i = N(\tilde{\mu}_i, \tilde{\sigma}^2_i)$ that approximates F_{Y_i}. To compute D_i, we use normal distribution $C_j = N(\tilde{\mu}_j - \mu_{Z_j}, \tilde{\sigma}^2_j - \sigma^2_{Z_j})$ $= N(\nu_j, \tau^2_j)$ for $v_j \in V_i$.

We explain how to calculate C_i and D_i. First topologically sort $V = \{v_1, v_2, \ldots, v_n\}$ and assume without loss of generality that $v_1 = S$, $v_n = T$ and $v_i < v_{i+1}$ for all $1 \leqslant i \leqslant n-1$. Hence $j < i$ if $v_j \in V_i$. We calculate C_i and D_i in the increasing order of i. For the base case, we define $C_1 = N(0,0)$, i.e., $\nu_1 = \tau^2_1 = 0$. (Note that the definition of C_1 is just for initializing $\nu_1 = \tau^2_1 = 0$.)

Let v_i ($i \geqslant 2$) be a vertex. We assume that C_j have already calculated for all $j < i$. The key of the algorithm is a transformation from G_i to a racket graph R_i (Fig. 3). As the head of R_i, we put $|V_i|$ parallel edges a_j ($v_j \in V_i$) between S and v. Let $e_j = (v_j, v_i)$ and assume that its length obeys $N(\mu_{e_j}, \sigma^2_{e_j})$. We assign to each edge a_j a random variable U_j that follows a normal distribution $N(\nu_j + \mu_{e_j}, \tau^2_j + \sigma^2_{e_j})$, and to the shaft a random variable Z_i that follows $N(\mu_{Z_i}, \sigma^2_{Z_i})$.

A-QTREE then calculates a normal distribution $D_i = N(\tilde{\mu}_i, \tilde{\sigma}^2_i)$, that approximates the distribution of the longest path length of R_i by executing A-RACKET, and put $C_i = N(\tilde{\mu}_i - \mu_{Z_i}, \tilde{\sigma}^2_i - \sigma^2_{Z_i})$. We give a description of A-QTREE in the following.

```
**Algorithm** A-QTREE($G, a$):
Topologically sort $V = \{v_1, v_2, \ldots, v_n\}$;
(Assume that $v_1 = S$, $v_n = T$ and $v_i < v_{i+1}$.)
Compute $\mu_{Z_i}$ and $\sigma^2_{Z_i}$ for all $v_i \in V$;
$C_1 := (0, 0)$;
**for** $i = 2, 3, \ldots, n$ **do**
    Construct $R_i$;
    Call A-RACKET($R_i, a$), which returns $(\tilde{\mu}, \tilde{\sigma}^2)$;
    $C_i := (\tilde{\mu} - \mu_{Z_i}, \tilde{\sigma}^2 - \sigma^2_{Z_i})$;
**return** $C_n = (\tilde{\mu}, \tilde{\sigma}^2)$.
```

Note that in A-QTREE we adopt a convention that $\mu_{Z_n} = \sigma^2_{Z_n} = 0$, since π_n is an empty path of length 0. That is, $D_n = C_n$.

Theorem 5. *Let G and $F(x)$ be a quasi-tree graph and the distribution function of its longest path length, respectively. For any given real number a $(0 < a < 1)$,* A-QTREE *returns a normal distribution function $\tilde{F}(x)$ that approximates $F(x)$ with respect to $F^{-1}(a)$ in $O(n)$ time, where n and m are respectively the order and the size of G.*

Proof. Let $\tilde{F}_{Y_i}(x)$ be the distribution function of $D_i = N(\tilde{\mu}_i, \tilde{\sigma}_i^2)$, where D_i is the output of A-RACKET(R_i, a) in A-QTREE. ($\tilde{F}_{Y_i}(x)$ approximates the distribution function $F_{Y_i}(x)$ the longest path length Y_i of G_i). By Theorem 4, $\tilde{F}_{Y_i}(x)$ approximates the distribution function of the longest path length of R_i. In the following, by induction on i, we show that $\tilde{F}_{Y_i}(x)$ also approximates $F_{Y_i}(x)$, which is the distribution function of the longest path length of G_i.

Consider v_2 as the base case (since $v_1 = S$). Since G is a quasi-tree, v_2 is a child of S and $|V_2| = 1$. Let $e = (v_1, v_2)$. Then A-RACKET returns $N(\mu_e + \mu_{Z_2}, \sigma_e^2 + \sigma_{Z_2}^2)$ as D_2, which is clearly the distribution that Y_2 follows.

Then we go on the induction step. By induction hypothesis, $\tilde{F}_{Y_j}(x)$ approximates $F_{Y_j}(x)$ with respect to $F_{Y_j}^{-1}(a)$ for each $v_j \in V_i$ (since $v_j \in V_i$ implies $j < i$). Since G_j is a subgraph of G_i, $F_{Y_j}(x) \geqslant F_{Y_i}(x)$ for any j, which implies that $F_{Y_j}^{-1}(a) \leqslant F_{Y_i}^{-1}(a)$. That is, $\tilde{F}_{Y_j}(x)$ approximates $F_{Y_j}(x)$ with respect to $F_{Y_i}^{-1}(a)$. By Property 1, $F_{Y_i}(x) \geqslant \prod_{v_j \in V_i} F_{Y_j}(x)$. By the definition of A-RACKET, we have $F_{Y_i}(x) \geqslant \prod_{v_j \in V_i} \tilde{F}_{Y_j}(x) \geqslant \tilde{F}_{Y_i}(x)$ for all $x \geqslant F_{Y_i}^{-1}(a)$, since $\prod_{v_j \in V_i} \tilde{F}_{Y_j}(x)$ is the distribution function of Par(R_i).

As for the time complexity, topological sort of V and calculations of μ_{Z_i} and $\sigma_{Z_i}^2$ for all $v_i \in V$ require $O(m+n)$ time. Construction of R_i and A-RACKET(R_i, a) requires $O(|V_i|)$ time. Hence the whole computation time of A-QTREE is $O(m+n)$. □

4. The algorithm

4.1. Algorithm A-DAG

This subsection introduces our algorithm A-DAG. We would like to explain the idea behind A-DAG first of all. For any DAG G, $F(x)$ can be approximated by applying A-PARA to Par(G). The time complexity $O(k)$ of A-PARA(Par$(G), a)$ however can be as large as $\Omega(2^n)$, where n is the order of G, since k is the number of paths in G, as pointed out in Section 3. Let $N(\tilde{\mu}_{\mathrm{Par}}, \tilde{\sigma}_{\mathrm{Par}}^2)$ be the output of A-PARA(Par$(G), a)$. A-DAG approximates $N(\tilde{\mu}_{\mathrm{Par}}, \tilde{\sigma}_{\mathrm{Par}}^2)$ in linear time using a dynamic programming technique. We denote the normal distribution that the length of edge e_π follows by $N(\mu_\pi, \sigma_\pi^2)$, for any edge e_π in Par(G). Let $\sigma_{\max}^2 = \max_\pi \sigma_\pi^2$. Then $\tilde{\sigma}_{\mathrm{Par}}^2 = \sigma_{\max}^2$ by the definition of A-PARA, and we can easily calculate $\sigma_{\max}^2$ by applying a linear time longest path algorithm to a DAG (with deterministic edge lengths). What A-DAG essentially needs to do is to estimate an upper bound $\tilde{\mu}$ on $\tilde{\mu}_{\mathrm{Par}}$. Then the distribution function of $N(\tilde{\mu}, \sigma_{\max}^2)$ approximates that of $N(\tilde{\mu}_{\mathrm{Par}}, \tilde{\sigma}_{\mathrm{Par}}^2)$ and hence $F(x)$.

A-QTREE realizes the same idea, and indeed the descriptions of A-QTREE and A-DAG are identical. A-DAG first topologically sorts $V = \{v_1, v_2, \ldots, v_n\}$ like A-QTREE. We assume without loss of generality that $v_1 = S$, $v_n = T$ and $v_i < v_{i+1}$ for any $1 \leqslant i \leqslant n-1$. As before, we denote the subgraph of G induced by the set of paths connecting v_1 $(= S)$ and v_n $(= T)$ through v_i by G_i, and let Y_i denote the longest path length of G_i. Like A-QTREE, A-DAG calculates a normal distribution D_i that approximates the distribution function of Y_i in the increasing order of i.

For any two vertices v_i and v_j in G such that v_i is reachable from v_j, we denote the subgraph of G induced by the set of paths in G connecting v_j and v_i by G_{ji}. We have seen that A-QTREE approximates the distribution function of $Y_i = W_{1i} + W_{in}$ for each $v_i \in V$ when G is a quasi-tree. (We used symbol Z_i instead of W_{in} in the last subsection.) Recall that in A-QTREE, we calculate a normal distribution C_i for all i. The essence of the correctness proof of A-Q-TREE is to show that $C_i = N(\tilde{\mu}_i - \mu_{Z_i}, \tilde{\sigma}_i^2 - \sigma_{Z_i}^2)$ gives a normal distribution function that can be used as an approximation of the distribution function of W_{1i}, although it is of course not the case in general. A-QTREE is not directly applicable for our purpose here however, since first, G_{in} is not a quasi-tree and second, W_{in} is not the length of a path, when G is a DAG. Nevertheless, we pursue this idea optimistically believing that the above claim on C_i holds even for a DAG, if we choose an appropriate path π_i in G_{in}. A-DAG chooses, as π_i, a path that has the maximum variance among the paths in G_{in}. We shall then show that this choice makes the claim true.

We denote the length of π_i by Z_i, and let $N(\mu_{Z_i}, \sigma_{Z_i}^2)$ be its distribution. That is, $\sigma_{Z_i} \geqslant \sigma_\pi$ for any path π in G_{in}, where $N(\mu_\pi, \sigma_\pi^2)$ is the distribution of the length of π. The description of A-DAG is exactly the same as A-QTREE, except for the difference of the definition of π_i. π_i is defined to be a path with the maximum variance in G_{in} in A-DAG, while in A-QTREE it is the unique path connecting v_i and T. Then C_n $(= D_n)$ is the approximation which we are looking for; that is, the distribution function of C_n approximates that of the longest path length of G with respect to $F^{-1}(a)$. Now we are ready to present our algorithm, which however is exactly the same as A-QTREE, since the computations of μ_{Z_i} and $\sigma_{Z_i}^2$ are implicit in this description.

```
Algorithm A-DAG($G, a$):
Topologically sort $V = \{v_1, v_2, \ldots, v_n\}$;
(Assume that $v_1 = S$, $v_n = T$ and $v_i < v_{i+1}$.)
Compute $\mu_{Z_i}$ and $\sigma^2_{Z_i}$ for all $v_i \in V$;
$C_1 := (0, 0)$;
for $i = 2, 3, \ldots, n$ do
    Construct $R_i$;
    Call A-RACKET($R_i, a$), which returns $(\tilde{\mu}, \tilde{\sigma}^2)$;
    $C_i := (\tilde{\mu} - \mu_{Z_i}, \tilde{\sigma}^2 - \sigma^2_{Z_i})$;
return $C_n = (\tilde{\mu}, \tilde{\sigma}^2)$.
```

Note that in A-DAG we adopt a convention that $\mu_{Z_n} = \sigma^2_{Z_n} = 0$, since π_n for v_n is an empty path. That is, $D_n = C_n = N(\tilde{\mu} - \mu_{Z_n}, \tilde{\sigma}^2 - \sigma^2_{Z_n})$.

4.2. Correctness of A-DAG

Algorithm A-DAG sorts $V = \{v_1, v_2, \ldots, v_n\}$ topologically and assumes that $v_1 = S$, $v_n = T$, and $v_i < v_{i+1}$ for all $1 \leqslant i \leqslant n-1$. We show that the distribution function $\tilde{F}(x)$ of C_n approximates the distribution function $F(x)$ of the longest path length of G with respect to $F^{-1}(a)$ $(1/2 \leqslant a < 1)$.

First of all, let us confirm that $\tilde{\sigma}^2 = \sigma^2_{\max}$, where $\sigma^2_{\max}$ is the maximum variance appeared as a variance of a normal distribution attached to an edge in Par(G). We show this fact in a more general form. For a vertex v_i, let π_i be the path connecting v_i and T that A-DAG chooses. Consider a graph B_i that is G_{1i} with a tail π_i. More formally, B_i is the subgraph of G induced by all paths π connecting S and T that share π_i as a suffix, i.e., π can be rewritten as $\pi'\pi_i$, where π' is a path connecting S and v_i. Let $P_i = \mathrm{Par}(B_i)$, and by $\sigma^2_{\max_i}$ we denote the maximum variance appeared as a variance of a normal distribution attached to an edge in P_i. Let $D_i = N(\tilde{\mu}_i, \tilde{\sigma}^2_i) = N(\nu_i + \mu_{Z_i}, \tau^2_i + \sigma^2_{Z_i})$ be the output of A-RACKET(R_i, a), where $C_i = N(\nu_i, \tau^2_i)$ and $Z_i \sim N(\mu_{Z_i}, \sigma^2_{Z_i})$. Considering the consistency, let us define $D_1 = N(\mu_{Z_1}, \sigma^2_{Z_1})$. We denote the set of all parents of v_i by V_i as before.

Lemma 7. $\tilde{\sigma}^2_i = \sigma^2_{\max_i}$.

Proof. The proof is by induction on i. The base case is trivial, since P_1 is a path π_1 and $\tilde{\sigma}^2_1 = \sigma^2_{Z_1}$.

Consider the induction step. By definition $\sigma^2_{\max_i} = \max_{v_j \in V_i}\{\sigma^2_{\max_j} - \sigma^2_{Z_j} + \sigma^2_{Z_{ji}}\}$, where $\sigma^2_{Z_{ji}}$ is the variance of the normal distribution that the length of path $(v_j, v_i)\pi_i$ follows. By induction hypothesis, we have $\tilde{\sigma}^2_j = \sigma^2_{\max_j}$ since $j < i$. That is, $\tau^2_j = \sigma^2_{\max_j} - \sigma^2_{Z_j}$. By the definition of R_i, the variance of the normal distribution assigned to the j-th edge a_j of the head is $\tau^2_j + \sigma^2_{e_{ji}}$, where $e_{ji} = (v_j, v_i)$, and since $\sigma^2_{Z_i}$ is assigned as the variance of its shaft, $\tilde{\sigma}^2_i = \max_{v_j \in V_i}\{\tau^2_j + \sigma^2_{Z_{ji}}\} = \sigma^2_{\max_i}$, by the definitions of Par and A-PARA. □

By Property 1, the distribution function of the longest path length of P_i is smaller than or equal to that of B_i. Let $F_i(x)$ be the distribution function of the longest path length of P_i, and $\tilde{F}_i(x)$ the distribution function of D_i. Since $B_n = G$, it is sufficient to show that $\tilde{F}_i(x)$ approximates $F_i(x)$ for all $1 \leqslant i \leqslant n$ with respect to $F_i^{-1}(a)$ for any $1/2 \leqslant a < 1$. We show this by an induction on i.

First consider the base case where $i = 1$. Since B_1 is actually a path π_1, $B_1 = P_1$ and hence clearly D_1 is the exact normal distribution that the length of π_1 obeys.

Let us proceed to the induction step. By induction hypothesis, $\tilde{F}_j(x)$ approximates $F_j(x)$ for all $1 \leqslant j \leqslant i-1$ with respect to $F_j^{-1}(a)$. For each of $v_j \in V_i$, let P_{ji} be the subgraph of P_i induced by the set of paths through v_j. Observe that $P_{ji} = \mathrm{Par}(B_{ji})$, where B_{ji} is the graph constructed from B_j by replacing π_j with $(v_j, v_i)\pi_i$.

As in the proof of Lemma 7, let Z_{ji} denote the length of path $(v_j, v_i)\pi_i$ and assume that $Z_{ji} \sim N(\mu_{Z_{ji}}, \sigma^2_{Z_{ji}})$. By the construction of R_i in A-DAG, Par(R_i) is a parallel graph with $|V_i|$ edges, where the length U_j of edge corresponding to v_j obeys $N(\nu_j + \mu_{Z_{ji}}, \tau^2_j + \sigma^2_{Z_{ji}})$, where $C_j = N(\nu_j, \tau^2_j)$. Let $A_i(x)$ be the distribution function of the longest path length of Par(R_i). Then by Theorem 3 $\tilde{F}_i(x)$ approximates $A_i(x)$ with respect to $A_i^{-1}(a)$.

Let $F_{ji}(x)$ be the distribution function of the longest path length of P_{ji}, and let $\tilde{F}_{ji}(x)$ be the distribution function that U_j follows. Consider the following Claim C.

Claim C. *$\tilde{F}_{ji}(x)$ approximates $F_{ji}(x)$ with respect to $F_{ji}^{-1}(a)$.*

Suppose that Claim C is correct. Since $F_i(x) = \prod_{1 \leqslant j \leqslant |V_i|} F_{ji}(x)$, we have $F_i(x) \leqslant F_{ji}(x)$ for all x, which implies that $F_i^{-1}(a) \geqslant F_{ji}^{-1}(a)$. Thus $\tilde{F}_{ji}(x)$ approximates $F_{ji}(x)$ with respect to $F_i^{-1}(a)$. By Claim C, with respect to $F_i^{-1}(a)$, we have

$F_i(x) \geqslant \prod_{1 \leqslant j \leqslant |V_i|} \tilde{F}_{ji}(x) = A_i(x)$. Hence $\tilde{F}_i(x)$, which is an approximation of $A_i(x)$, approximates $F_i(x)$ with respect to $F_i^{-1}(a)$, which concludes the correctness of A-DAG. The rest of this subsection is devoted to proving Claim C.

Let $N(\lambda_{ji}, \rho_{ji}^2)$ be the output of A-PARA(P_{ji}, a) and let $J_{ji}(x)$ be its distribution function. Since $J_{ji}(x)$ approximates $F_{ji}(x)$ with respect to $F_{ji}^{-1}(a)$, the proof completes if $J_{ji}(x) \geqslant \tilde{F}_{ji}(x)$ for all x. In the following, we show 1) $\lambda_{ji} \leqslant \nu_j + \mu_{Z_{ji}}$ and 2) $\rho_{ji}^2 = \tau_j^2 + \sigma_{Z_{ji}}^2$, which are sufficient to conclude $J_{ji}(x) \geqslant \tilde{F}_{ji}(x)$.

Since $\rho_{ji}^2 = \tau_j^2 + \sigma_{Z_{ji}}^2$ holds by Lemma 7, we concentrate on the proof of $\lambda_{ji} \leqslant \nu_j + \mu_{Z_{ji}}$. We would like to emphasize that the mean ν_j of P_j is not determined only from G_{1j}; it also depends on the distribution of Z_j, i.e., μ_{Z_j} and $\sigma_{Z_j}^2$, unlike the case of the variance. To investigate the effect of the distribution of Z_j on the mean, by $\nu(\mu, \sigma^2)$ let us denote the mean when the distribution of Z_j is $N(\mu, \sigma^2)$. More formally, let $P(\mu, \sigma^2)$ be the graph constructed from P_j by assigning normal distribution $N(\mu, \sigma^2)$ (instead of $N(\mu_{Z_j}, \sigma_{Z_j}^2)$) to path π_j as the distribution of the length of π_j, and let $N(\lambda(\mu, \sigma^2), \rho^2(\mu, \sigma^2))$ be the output of A-PARA$(P(\mu, \sigma^2), a)$. Then $\nu(\mu, \sigma^2)$ is defined to be $\lambda(\mu, \sigma^2) - \mu$. By definition, $P(\mu_{Z_j}, \sigma_{Z_j}^2) = P_j$, $P(\mu_{Z_{ji}}, \sigma_{Z_{ji}}^2) = P_{ji}$, $\nu(\mu_{Z_j}, \sigma_{Z_j}^2) = \nu_j$, and $\nu(\mu_{Z_{ji}}, \sigma_{Z_{ji}}^2) = \lambda_{ji} - \mu_{Z_{ji}}$. Now our goal is thus to derive $\nu(\mu_{Z_j}, \sigma_{Z_j}^2) \geqslant \nu(\mu_{Z_{ji}}, \sigma_{Z_{ji}}^2)$.

To this end, we show that 1) function ν is a constant function with respect to μ, i.e., $\nu(\mu, \sigma^2) = \nu(0, \sigma^2)$, and that 2) ν is monotonically increasing with respect to σ^2. It is obvious to observe that these two properties are sufficient for our purpose since $\sigma_{Z_j}^2 \geqslant \sigma_{Z_{ji}}^2$. The following three lemmas show these facts.

We define some symbols concerning $P(\mu, \sigma^2)$. The length of the ℓ-th edge e_ℓ of $P(\mu, \sigma^2)$ is denoted by U_ℓ and assume that $U_\ell \sim N(\xi_\ell + \mu, \chi_\ell^2 + \sigma^2)$. Let max be an ℓ that maximizes χ_ℓ^2. Let $L_\ell^{(\mu,\sigma^2)}(x)$ (resp. $H_\ell(x)$) be the distribution function of $N(\xi_\ell + \mu, \chi_\ell^2 + \sigma^2)$ (resp. $N(\xi_\ell, \chi_\ell^2)$). By definition $H_\ell(x) = L_\ell^{(0,0)}(x)$. In the following, we may omit (μ, σ^2) from $L_\ell^{(\mu,\sigma^2)}(x)$ if it is clear from the context. Finally, let $J_{(\mu,\sigma^2)}(x)$ be the distribution function of the longest path length in graph $P(\mu, \sigma^2)$.

Lemma 8. *For any μ and σ^2, $\nu(\mu, \sigma^2) = \nu(0, \sigma^2)$.*

Proof. By definition

$$J_{(\mu,\sigma^2)}(x) = \prod_\ell L_\ell^{(\mu,\sigma^2)}(x) = \prod_\ell \Phi\left(\frac{x - (\xi_\ell + \mu)}{\sqrt{\chi_\ell^2 + \sigma^2}}\right).$$

Thus $J_{(\mu,\sigma^2)}(x) = J_{(0,\sigma^2)}(x - \mu)$, and hence $J_{(\mu,\sigma^2)}^{-1}(a) = J_{(0,\sigma^2)}^{-1}(a) + \mu$. By the definition of A-PARA

$$\nu(\mu, \sigma^2) + \mu = J_{(\mu,\sigma^2)}^{-1}(a) - \sqrt{\chi_{\max}^2 + \sigma^2}\,\Phi^{-1}(a),$$

and hence

$$\nu(\mu, \sigma^2) + \mu = J_{(0,\sigma^2)}^{-1}(a) + \mu - \sqrt{\chi_{\max}^2 + \sigma^2}\,\Phi^{-1}(a) = \nu(0, \sigma^2) + \mu. \qquad \Box$$

Lemma 9. *For $1/2 \leqslant a < 1$, $\nu(0, \sigma^2) \geqslant \nu(0, 0)$.*

Proof. Assume with/out loss of generality that $\sigma^2 > 0$. Consider a function

$$K(\chi_\ell^2, \sigma^2) = \left(L_\ell^{(0,\sigma^2)}\right)^{-1}(a) - H_\ell^{-1}(a) = \left(\sqrt{\chi_\ell^2 + \sigma^2} - \chi_\ell\right)\Phi^{-1}(a)$$

for any ℓ. It is easy to see that $K(\chi_\ell^2, \sigma^2)$ is monotonically decreasing with respect to χ_ℓ^2, since $\Phi^{-1}(a) \geqslant 0$ for $1/2 \leqslant a < 1$. Hence for all ℓ

$$K(\chi_{\max}^2, \sigma^2) \leqslant K(\chi_\ell^2, \sigma^2).$$

Observe next that equation $L_\ell^{(0,\sigma^2)}(x + K(\chi_\ell^2, \sigma^2)) = H_\ell(x)$ has a root $x_0 = H_\ell^{-1}(a)$. Since two normal distribution functions with different variances intersect each other at a single point, x_0 is the unique one. Hence for all $x > H_\ell^{-1}(a)$,

$$L_\ell^{(0,\sigma^2)}\left(x + K(\chi_\ell^2, \sigma^2)\right) < H_\ell(x),$$

since the variance of $L_\ell^{(0,\sigma^2)}(x)$ is greater than that of $H_\ell(x)$. We thus have for all $x > \max_\ell H_\ell^{-1}(a)$,

$$\begin{aligned} J_{(0,\sigma^2)}\bigl(x+K\bigl(\chi^2_{\max},\sigma^2\bigr)\bigr) &= \prod_\ell L_\ell^{(0,\sigma^2)}\bigl(x+K\bigl(\chi^2_{\max},\sigma^2\bigr)\bigr) \\ &\leqslant \prod_\ell L_\ell^{(0,\sigma^2)}\bigl(x+K\bigl(\chi^2_\ell,\sigma^2\bigr)\bigr) \\ &< \prod_\ell H_\ell(x) = J_{(0,0)}(x). \end{aligned}$$

Since $J_{(0,0)}(x) \leqslant H_\ell(x)$ for all x and ℓ, $J^{-1}_{(0,0)}(a) \geqslant \max_\ell H_\ell^{-1}(a)$. By putting $x = J^{-1}_{(0,0)}(a)$,

$$J_{(0,\sigma^2)}\bigl(J^{-1}_{(0,0)}(a)+K\bigl(\chi^2_{\max},\sigma^2\bigr)\bigr) < J_{(0,0)}\bigl(J^{-1}_{(0,0)}(a)\bigr) = a,$$

which implies that

$$J^{-1}_{(0,0)}(a)+K\bigl(\chi^2_{\max},\sigma^2\bigr) < J^{-1}_{(0,\sigma^2)}(a),$$

since function $J^{-1}_{(0,\sigma^2)}(x)$ is strictly monotonically increasing. Since $K(\chi^2_{\max},\sigma^2) = (L^{(0,\sigma^2)}_{\max})^{-1}(a) - H^{-1}_{\max}(a)$, we have $J^{-1}_{(0,0)}(a) - H^{-1}_{\max}(a) < J^{-1}_{(0,\sigma^2)}(a) - (L^{(0,\sigma^2)}_{\max})^{-1}(a)$. By the definitions of $L^{(0,\sigma^2)}_{\max}(x)$ and $H_{\max}(x)$, for any $\sigma^2 > 0$,

$$\nu(0,0) = J^{-1}_{(0,0)}(a) - \bigl(H^{-1}_{\max}(a) - \xi_{\max}\bigr) < J^{-1}_{(0,\sigma^2)}(a) - \bigl(\bigl(L^{(0,\sigma^2)}_{\max}\bigr)^{-1}(a) - \xi_{\max}\bigr) = \nu\bigl(0,\sigma^2\bigr). \qquad \square$$

Lemma 10. *For any $1/2 \leqslant a < 1$, $\nu(0,\sigma^2)$ is monotonically increasing with respect to σ^2.*

Proof. We show that $\nu(0,\sigma_a^2) > \nu(0,\sigma_b^2)$ holds if $\sigma_a > \sigma_b^2 \geqslant 0$. Consider $P(\mu,\sigma^2)$. Recall that $U_\ell \sim N(\xi_\ell+\mu, \chi_\ell^2+\sigma^2)$. We modify $P(\mu,\sigma^2)$ and construct a new graph $P'(\mu,\sigma^2)$ by changing the distribution that U_ℓ follows to $N(\xi_\ell+\mu, (\chi_\ell^2+\sigma_b^2)+\sigma^2)$, and let us write the mean of the output of A-PARA($P'(\mu,\sigma^2)$) as $\nu'(\mu,\sigma^2)+\mu$. By definition, $P'(0,0) = P(0,\sigma_b^2)$ and $P'(0,\sigma_a^2-\sigma_b^2) = P(0,\sigma_a^2)$. By Lemma 9, $\nu(0,\sigma_b^2) = \nu'(0,0) < \nu'(0,\sigma_a^2-\sigma_b^2) = \nu(0,\sigma_a^2)$. □

Since the difference between A-DAG and A-QTREE from the view of the time complexity is that in A-DAG the calculation of $N(\mu_{Z_i},\sigma^2_{Z_i})$ becomes necessary. We now describe the way how to calculate μ_{Z_i} and $\sigma^2_{Z_i}$. Let G_σ be a graph (with edge lengths) having the same vertex and the edge sets as G. The graph G_σ however is not a stochastic network but a network with deterministic edge lengths: Recall that $N(\mu_e,\sigma_e^2)$ is the normal distribution that the edge length of e follows in G. The edge length of e in G_σ is σ_e^2. Then there is an algorithm to calculates for each vertex v_i the longest path length $\sigma^2_{Z_i}$ connecting v_i and T that runs in $O(n+m)$ time [10]. By modifying the algorithm slightly one can calculate μ_{Z_i} as well in the same time complexity. Thus the overall time complexity of A-DAG is $O(n+m)$. Now let us summarize as a theorem.

Theorem 6. *Let G and $F(x)$ be a* DAG *and the distribution function of the longest path length of G, respectively. For any given real number a such that $1/2 \leqslant a < 1$,* A-DAG *returns a normal distribution function $\tilde{F}(x)$ that approximates the distribution function $F(x)$ with respect to $F^{-1}(a)$ in $O(n+m)$ time, where n and m are respectively the order and the size of G.*

5. Performance analysis

This section studies the performance of A-DAG. We first demonstrate, for some values of a, the performance of A-DAG by using as instances logical circuits registered in a standard benchmark set ITC'99. The performance measures are the execution time and the error $\mathrm{Err}(G,a) = \tilde{F}^{-1}(a) - F^{-1}(a)$. In order to evaluate $F^{-1}(a)$, we used Monte Carlo simulations. We next perform a worst case analysis to derive an upper bound on Err.

5.1. Performance of A-DAG for ITC'99 benchmark set

Since we expect that A-DAG is typically used in circuit delay analysis, we conducted experiments by using, as instances, logical circuits registered in a standard benchmark set ITC'99 (see its home page [8] for details).

An n-input m-output gate is modeled by a directed edge e, whose length follows a normal distribution (see Fig. 4). In this experiment, for the ease of understanding, we assume that all gates follow the same delay distribution. We further neglect the wire delay, but it is easy to develop a DAG representation such that the wire delay is taken into account.

In order to estimate Err, we needed $F^{-1}(a)$. We conducted Monte Carlo simulation to this end. We repeated the simulation 10 000 times for each of the instances to obtain the circuit delay and drew $F(x)$. We then estimated $F^{-1}(a)$ by interpolation. Although the accuracy of the result depends on the number of repetitions, we assumed that correct $F^{-1}(a)$ is obtained by the simulations. Finally we used 3.4 GHz processor.

Table 1 summarizes the results for $a = 0.99$, which is a value frequently used in applications. The columns correspond to, from left to right, the circuit name, the circuit size (i.e., the number of gates), the circuit height (i.e., the maximum

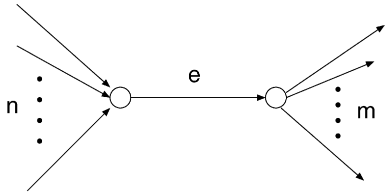

Fig. 4. Model of a gate.

Table 1
Performance of A-DAG for benchmark set ITC'99 (1/2).

Name	# of gates	Height h	Monte Carlo		A-DAG		
			$F^{-1}(a)$	CPU time (sec)	$\tilde{F}^{-1}(a)$	Err (%)	CPU time (sec)
b09	140	9	11.25	2.50	11.61	3.20	0.02
b09_opt	130	9	11.23	2.52	11.62	3.46	0.02
b10	172	12	13.84	2.83	13.88	0.31	0.02
b10_opt	146	12	13.77	3.06	13.89	0.87	0.02
b11	726	34	36.82	11.75	37.14	0.85	0.07
b11_opt	504	37	40.07	8.66	40.53	1.15	0.06
b12	944	19	21.49	18.61	21.89	1.85	0.12
b12_opt	874	19	21.52	15.74	21.81	1.31	0.11
b13	289	20	22.03	4.94	22.09	0.29	0.03
b13_opt	244	11	12.72	4.44	12.80	0.63	0.03
b14	9767	60	64.34	372.38	65.88	2.39	0.98
b14_opt	5347	41	44.73	210.86	45.90	2.61	0.65
b15	8367	63	67.43	322.50	69.16	2.55	0.96
b15_opt	7022	45	48.51	284.24	51.78	6.74	0.88
b17	30777	92	97.14	1172.01	100.87	3.84	3.34
b17_opt	22757	44	48.98	939.89	52.37	6.92	2.90
b18	111241	164	170.75	4226.41	185.77	8.80	11.72
b18_opt	69913	90	100.47	2850.47	116.23	15.68	8.96
b19	224624	168	175.52	8469.62	191.87	9.32	23.29
b20	19682	67	72.55	727.68	75.52	4.09	1.96
b20_opt	11957	73	77.03	480.73	78.05	1.33	1.41
b21	20027	68	72.77	743.31	75.32	3.50	1.98
b21_opt	12134	73	77.12	492.18	78.05	1.21	1.45
b22	29162	68	72.54	1083.96	74.81	3.13	2.92
b22_opt	17329	78	82.61	711.37	84.27	2.00	2.03

Table 2
Performance of A-DAG for several values of a.

a	Monte Carlo	A-DAG	Err (%)
0.99	64.34	65.88	2.39
0.90	63.03	66.09	4.85
0.80	62.53	66.56	6.45
0.70	62.15	67.06	7.89
0.60	61.85	67.59	9.28
0.50	61.57	68.17	10.72

number of gates in a path from the source to the sink), $F^{-1}(a)$ of the circuit, the total execution time for the 10 000 times Monte Carlo simulations, an approximation $\tilde{F}^{-1}(a)$ of $F^{-1}(a)$ (calculated from the output of A-DAG), the relative error Err (%) defined by $100 \times (\tilde{F}^{-1}(a) - F^{-1}(a))/F^{-1}(a)$, and the execution time of A-DAG. We assumed that the edge lengths follow a normal distribution $N(1.0, 0.04)$.

From the table, we make the following observations. First, A-DAG shows a good performance on average. The execution time is extremely small and A-DAG finishes in a few seconds for all except for circuits b18, b18_opt, and b19. The relative error is less than 5% for all except for circuits b15_opt, b17_opt, b18, b18_opt, and b19. From the execution times for b18 and b19, we can expect that the execution time of A-DAG is linear in the size, as Theorem 6 states. We can also observe that the relative error increases as the size increases. Since A-DAG approximates the longest path length for the whole DAG by repeatedly taking upper bounds on approximations for sub-DAGs, one can expect that the larger the circuit height, the larger the relative error would be, and indeed this guess will be formally shown to be correct, as long as the worst case analysis is concerned, in the next subsection. Fortunately however, this expected correlation is not apparent from the table; circuits b17, b20, b20_opt, b21, b21_opt, b22, b22_opt have heights larger than 60 but have relative errors less than b15_opt and b17_opt, whose heights are smaller than 50.

Table 3
Performance of A-DAG for several variances.

Name		0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16
b09	$F^{-1}(a)$	10.59	11.25	11.74	12.16	12.54	12.91	13.25	13.54
	$\tilde{F}^{-1}(a)$	10.85	11.61	12.20	12.70	13.14	13.54	13.90	14.25
	Err(%)	2.44	3.20	3.90	4.39	4.76	4.82	4.95	5.20
b09_opt	$F^{-1}(a)$	10.57	11.23	11.70	12.19	12.56	12.97	13.29	13.45
	$\tilde{F}^{-1}(a)$	10.85	11.62	12.21	12.71	13.15	13.55	13.91	14.26
	Err(%)	2.62	3.46	4.32	4.26	4.70	4.45	4.73	6.03
b18	$F^{-1}(a)$	168.59	170.75	172.56	174.05	175.65	177.05	178.51	179.88
	$\tilde{F}^{-1}(a)$	175.92	185.77	194.44	202.23	209.36	215.96	222.14	227.97
	Err(%)	4.35	8.80	12.68	16.19	19.19	21.98	24.44	26.74
b18_opt	$F^{-1}(a)$	97.14	100.47	103.22	105.53	107.66	109.65	111.52	113.48
	$\tilde{F}^{-1}(a)$	106.74	116.23	124.05	130.88	137.01	142.62	147.83	152.71
	Err(%)	9.87	15.68	20.18	24.02	27.26	30.07	32.56	34.57
b19	$F^{-1}(a)$	173.10	175.52	177.28	179.09	180.59	182.31	183.61	185.53
	$\tilde{F}^{-1}(a)$	181.36	191.87	201.02	209.21	216.68	223.60	230.08	236.19
	Err(%)	4.77	9.32	13.39	16.82	19.98	22.65	25.31	27.31

The performance of A-DAG would depend on the value of a. Table 2 shows $F^{-1}(a)$ and $\tilde{F}^{-1}(a)$ of circuit b_14 for several values of a, assuming that the edge lengths follow $N(1.0, 0.04)$. Since $F^{-1}(a)$ decreases as a decreases, A-DAG is suggested to use for a large a.

Table 3 shows the relation between the variance of edge lengths and the performance of A-DAG. The leftmost column corresponds to the circuit name, and rest columns to the estimated values of the functions on the right side of the circuit name assuming that all edge lengths follow $N(1.0, \sigma^2)$, where σ^2 is the value on the top of each column. The edge length variance varies from 0.02 to 0.16. The relative error increases as the variance increase.

Since we used Monte Carlo simulation to draw $F(x)$, one might consider that Monte Carlo simulation would be superior to A-DAG. Although we agree that Monte Carlo simulation is a powerful technique, A-DAG has several advantages that Monte Carlo simulation does not have. Firstly, A-DAG approximates $F(x)$ by a normal distribution function $\tilde{F}(x)$ in terms of its mean and variance; it is easy to apply the result to further formal analysis. Secondly, we formally guarantee that $\tilde{F}(x)$ is an approximation of $F(x)$ with respect to a. Thirdly, A-DAG is very fast. Table 1 shows that A-DAG is more than 300 times faster than Monte Carlo simulation. However, it could be argued that Monte Carlo simulation could be speeded up simply by reducing the number of repetitions. This may seriously sacrifice the accuracy, though.

5.2. An upper bound on Err

Let $N(\tilde{\mu}, \tilde{\sigma}^2)$ and $\tilde{F}(x)$ be the output of A-DAG (for G and a) and its distribution function, respectively. We derive an upper bound on $\mathrm{Err}(G, a) = \tilde{F}^{-1}(a) - F^{-1}(a)$. Let us start with estimates of the standard normal distribution function.

Lemma 11. $\Phi^{-1}(a^{1/d}) = \Theta(\sqrt{\ln d})$.

Proof. By *Mills' ratio* (see e.g. [6, p. 98]), we have

$$\left(\frac{1}{x} - \frac{1}{x^3}\right)\phi(x) < 1 - \Phi(x) < \frac{1}{x}\phi(x), \quad \text{for } x > 0.$$

It is then easy to see that there exists $\epsilon > 0$ such that

$$1 - e^{(-\frac{1}{2}+\epsilon)x^2} < 1 - \frac{\phi(x)}{x} < \Phi(x) < 1 - \left(\frac{1}{x} - \frac{1}{x^3}\right)\phi(x) < 1 - e^{(-\frac{1}{2}-\epsilon)x^2}$$

for x sufficiently large. The positive solution of the equation $1 - e^{(-\frac{1}{2}\pm\epsilon)x^2} = a^{1/d}$ is $x = \sqrt{\frac{-\ln(1-a^{1/d})}{1/2\pm\epsilon}}$. On the other hand, we have $1 - a^{1/d} \sim \frac{-\ln a}{d}$, since $\lim_{d\to\infty}(1 + \frac{\ln a}{d})^d = a$. Hence $\Phi^{-1}(a^{1/d}) = \Theta(\sqrt{\ln d})$. □

We next construct a path A connecting S and T starting from T backward. Initially, A is an empty path. Suppose that a path A between v_i and T has been constructed. Let $\tilde{W}_{1j}$ be a random variable that follows C_j. As in Section 4.2, the length of a path $(v_j, v_i)\pi_i$ is denoted by Z_{ji}. Put $\tilde{Y}_{ji} = \tilde{W}_{1j} + Z_{ji}$. Then $\tilde{Y}_{ji}$ follows a normal distribution $N(\mu_{\tilde{Y}_{ji}}, \sigma^2_{\tilde{Y}_{ji}})$. Now choose an edge (v_j, v_i) that maximizes the *mean* $\mu_{\tilde{Y}_{ji}}$ and extend A to $(v_j, v_i)A$.[4] Repeating this procedure finishes

[4] We maximize the mean $\mu_{\tilde{Y}_{ji}}$, not the variance $\sigma^2_{\tilde{Y}_{ji}}$.

at S and outputs a path A connecting S and T. Let $N(\mu_A, \sigma_A^2)$ be the distribution of the length of A and let $F_A(x)$ be its distribution function. Obviously, $F^{-1}(a) \geqslant F_A^{-1}(a)$, since $F(x) \leqslant F_A(x)$ (because A is a single path connecting S and T).

Each edge e_π in $\mathrm{Par}(G)$ corresponds to a path π in G, the length of e_π follows a normal distribution $N(\mu_\pi, \sigma_\pi^2)$. Let $N(\mu_{\max}, \sigma_{\max}^2)$ be the distribution of the maximum variance path. Hence $\sigma_{\max}^2 = \max_\pi \sigma_\pi^2$, but $\mu_{\max} \neq \max_\pi \mu_\pi$ in general. Let $\mu^* = \mu_A + (\sigma_A - \sigma_{\max})\Phi^{-1}(a)$, and we denote the distribution function of $N(\mu^*, \sigma_{\max}^2)$ by $H(x)$. Obviously $H^{-1}(a) = F_A^{-1}(a)$ holds. By definition,

$$\begin{aligned}\mathrm{Err}(G,a) &= \tilde{F}^{-1}(a) - F^{-1}(a)\\ &\leqslant \tilde{F}^{-1}(a) - F_A^{-1}(a)\\ &= \tilde{F}^{-1}(a) - H^{-1}(a).\end{aligned}$$

Since $\tilde{\sigma}^2 = \sigma_{\max}^2$ by Lemma 7, $\mathrm{Err}(G,a) \leqslant \tilde{F}^{-1}(a) - H^{-1}(a) = \tilde{\mu} - \mu^*$. The following theorem is obtained by estimating $\tilde{\mu} - \mu^*$.

Theorem 7. *Let h and d be the height and the maximum in-degree of G, respectively.*

$$\mathrm{Err}(G,a) = O\big(h\sigma_{\max}\sqrt{\ln d}\big).$$

Proof. A very rough idea behind the proof is the following: We first derive an upper bound δ_i on the contribution of the iteration of A-DAG for vertex v_i to $\mathrm{Err}(G,a)$, and then derive an upper bound on $\mathrm{Err}(G,a)$ by calculating $\sum_{v_i \in A} \delta_i$.

We use the symbols defined in the paragraphs after Lemma 11. Recall that D_i is a normal distribution that A-RACKET(R_i, a) returns in A-DAG and the distribution function of D_i is denoted by $\tilde{F}_i(x)$. Let $H_{ji}(x)$ be the distribution function of $N(\mu_{\tilde{Y}_{ji}}, \sigma^2_{\tilde{Y}_{ji}})$. Then

$$H_{ji}(x) \geqslant \Phi\left(\frac{x - \mu_{\max_i}}{\sigma_{\max}}\right),$$

where $\mu_{\max_i} = \max_{v_j \in V_i}\{\mu_{\tilde{Y}_{ji}}\}$. Hence for $x \geqslant \mu_{\max_i}$,

$$\tilde{F}_i(x) \geqslant \left\{\Phi\left(\frac{x - \mu_{\max_i}}{\sigma_{\max}}\right)\right\}^{d_i} \geqslant \left\{\Phi\left(\frac{x - \mu_{\max_i}}{\sigma_{\max}}\right)\right\}^{d},$$

which implies that

$$\tilde{F}_i^{-1}(a) \leqslant \mu_{\max_i} + \sigma_{\max}\Phi^{-1}\big(a^{1/d}\big),$$

where $d_i = |V_i|$ and $d = \max_{1 \leqslant i \leqslant n} d_i$ is the maximum in-degree of G. By g we denote a j that satisfies $\mu_{\max_i} = \mu_{\tilde{Y}_{ji}}$. We thus have

$$\begin{aligned}\tilde{F}_i^{-1}(a) - H_{gi}^{-1}(a) &\leqslant \big(\mu_{\tilde{Y}_{gi}} + \sigma_{\max}\Phi^{-1}\big(a^{(1/d)}\big)\big) - \big(\mu_{\tilde{Y}_{gi}} + \sigma_{\tilde{Y}_{gi}}\Phi^{-1}(a)\big)\\ &= \sigma_{\max}\Phi^{-1}\big(a^{(1/d)}\big) - \sigma_{\tilde{Y}_{gi}}\Phi^{-1}(a).\end{aligned}$$

Define δ_i by

$$\delta_i = \sigma_{\max}\Phi^{-1}\big(a^{(1/d)}\big) - \sigma_{\tilde{Y}_{gi}}\Phi^{-1}(a).$$

Since $\tilde{\sigma}_i = \max_{v_j \in V_i}\{\sigma_{\tilde{Y}_{ji}}\}$ by Lemma 7 and two normal distribution functions intersect at a single point, $\tilde{F}_i(x) \geqslant H_{gi}(x - \delta_i)$ for all $x < \tilde{F}_i^{-1}(a)$. Since $1/2 \leqslant a < 1$, $\tilde{F}_i(x_0) \geqslant H_{gi}(x_0 - \delta_i)$, where $x_0 = \tilde{F}_i^{-1}(1/2)$ is the mean of D_i. Hence the mean of $\tilde{F}_i(x)$ is not larger than the mean of $H_{gi}(x - \delta_i)$. Since ν_i and μ_{π_i} are the mean of C_i and the mean of the distribution of the length of π_i, we have $\nu_i + \mu_{\pi_i} \leqslant \nu_g + \mu_{gi} + \mu_{\pi_i} + \delta_i$ or $\nu_i \leqslant \nu_g + \mu_{gi} + \delta_i$, where μ_{gi} is the mean of the normal distribution that edge (v_g, v_i) follows. By the definition of path A, when A has been constructed between T and v_i, the procedure always extends A to v_g. Thus we have

$$\nu_n \leqslant \nu_1 + \mu_A + \sum_{v_i \in A} \delta_i.$$

Let us estimate $\tilde{\mu} - \mu^*$. By definition, $\nu_1 = 0$ and $\nu_n = \tilde{\mu}$. By using $\mu^* = \mu_A + (\sigma_A - \sigma_{\max})\Phi^{-1}(a)$, we have

$$\tilde{\mu} - \mu^* = (\sigma_{\max} - \sigma_A)\Phi^{-1}(a) + \sum_{v_i \in A} \delta_i.$$

Since $\delta_i \leqslant \sigma_{\max}\Phi^{-1}(a^{(1/d)})$ by definition, we finally have

$$\begin{aligned}\tilde{\mu} - \mu^* &\leqslant \sigma_{\max}\Phi^{-1}(a) + \sum_{v_i \in A} \delta_i \\ &\leqslant \sigma_{\max}\Phi^{-1}(a) + h\sigma_{\max}\Phi^{-1}\big(a^{(1/d)}\big),\end{aligned}$$

since $|A| \leqslant h$. Since $\Phi^{-1}(a) \leqslant \Phi^{-1}(a^{\frac{1}{d}})$, by Lemma 11, we conclude

$$\tilde{\mu} - \mu^* = O\big(h\sigma_{\max}\sqrt{\ln d}\big). \qquad \square$$

If the variances σ_e^2 for edges e are bounded from above by a constant, then $\sigma_{\max} = O(\sqrt{h})$. Thus we have the following corollary.

Corollary 2. *Provided that the variances σ_e^2 for edges e are bounded from above by a constant,*

$$\mathrm{Err}(G, a) = O\big(\sqrt{h^3 \ln d}\big).$$

Example 1. Consider a DAG G with height h in Fig. 5. The in-degree of every vertex except S is d; there are d parallel edges between two consecutive vertices. Let e_{ij} be the j-th outgoing edge of v_i. The length of edge e_{ij}, represented by a random variable X_{ij}, follows a normal distribution $N(\mu, \sigma^2)$. Following the execution of A-DAG for G, we can observe that $D_i = N(h\mu + (i-1)\alpha, h\sigma^2)$ for any $1 \leqslant i \leqslant h+1$, where $\alpha = \sigma(\sqrt{h}\Phi^{-1}(a^{(1/d)}) - \Phi^{-1}(a))$. For this example, by the central limit theorem, an asymptotically exact distribution of the longest path length is $N(h\mu_*, h\sigma_*^2)$, where μ_* and σ_*^2 are the mean and the variance of $\max_{1 \leqslant j \leqslant d}\{X_{ij}\}$. Thus $F^{-1}(a) = O(h)$ and hence $\mathrm{Err}(G, a) = \tilde{F}^{-1}(a) - F^{-1}(a) = h\mu + (h-1)\alpha + \sqrt{h}\sigma\Phi^{-1}(a) - O(h) = \Omega(\sqrt{h^3 \ln d})$ by Lemma 11.

Example 2. We next show an example of $k \times k$ square grid (Fig. 6). In this square grid, it is convenient to write $v(i, j)$ to refer to the vertex in the ith row from the top and in the j-th column from the left; hence $v(1, 1)$ is the source and $v(k, k)$ is the terminal. Similarly, instead of C_i and D_i for v_i, we have the notation of $C(i, j)$ and $D(i, j)$ for vertex $v(i, j)$. If $i \geqslant 2$ (resp. $j \geqslant 2$), each vertex $v(i, j)$ has an edge $(v(i-1, j), v(i, j))$ (resp. $(v(i, j-1), v(i, j))$). The edge lengths are given as mutually independent random variables that obey the same distribution $N(\mu, \sigma^2)$.

For each vertex $v(i, 1) \in V$ or $v(1, i)$, A-DAG computes $N(i\mu, i\sigma^2)$ for $C(i, 1)$ or $C(1, i)$. As for the other vertices, a path from $v(i, j)$ to the terminal is chosen as $\pi(i, j)$; since all paths have the same distribution, any single path is sufficient. For a vertex $v(i, j)$, A-DAG creates a racket graph that ends up with a parallel graph with two edges with length distributions $D(i-1, j)$ and $D(i, j-1)$. Then $D(i, j)$ is computed from $D(i-1, j)$ and $D(i, j-1)$ by calling A-PARA with a given parameter a.

One may wonder if there is any point to compute $C(i, j)$ in this example. Actually, we do not have to compute $C(i, j)$ if all paths between $v(i, j)$ and the terminal have the same mean and variance. $C(i, j)$ is computed to deal with paths with various means and variances.

After computing $D(k, k) = N(\tilde{\mu}, \tilde{\sigma}^2)$, we have the approximate distribution function $\tilde{F}(x) = \Phi((x - \tilde{\mu})/\tilde{\sigma})$. Since the height h of the graph is given as $h = 2k = O(\sqrt{n})$, the error $\tilde{F}^{-1}(a) - F^{-1}(a)$ is $O(n^{3/4})$ by Corollary 2.

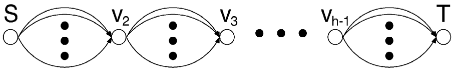

Fig. 5. A candidate for the worst case instance of A-DAG.

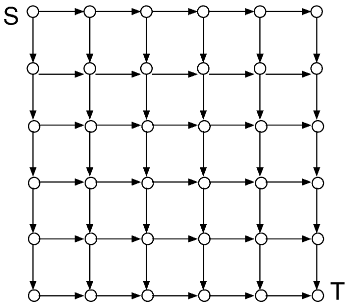

Fig. 6. An example of 6 × 6 square grid with acyclic orientation.

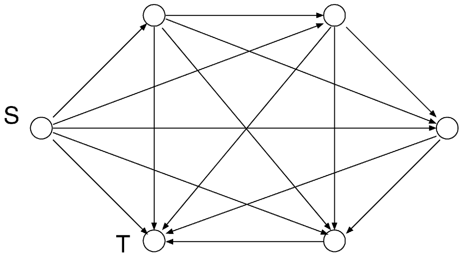

Fig. 7. An example of 6 vertex complete graph with acyclic orientation.

Example 3. Let $G = (V, E)$ be a complete graph with acyclic orientation (Fig. 7). That is, $E = \{(v_i, v_j) \mid 1 \leqslant i < j \leqslant n\}$. The edge lengths are mutually independent random variables and each of them obeys $N(\mu, \sigma)$.

In this example, π_i is always chosen as a path that passes all vertices with subscription numbers larger than i. Hence the distribution of the length of π_i is $N((n-i+1)\mu, (n-i+1)\sigma^2)$.

As for computation of C_i, A-DAG creates a racket graph R_i with its shaft π_i and its head, a parallel graph of $i-1$ edges. Let $(v_j, v_i)(j < i)$ be one of the head edges of R_i. The length distribution of (v_j, v_i) is given as $N(\nu_j + \mu, \rho_j^2 + \sigma^2)$, where ν_j and ρ_j are the mean and the variance of C_j. Then A-DAG calls A-RACKET for R_i, which creates a parallel graph $\mathrm{Par}(R_i)$ of $i-1$ edges and the j-th edge of $\mathrm{Par}(R_i)$ has its length that obeys $N(\nu_j - (n-j)\mu, \rho_j^2 - (n-j)\sigma^2)$.

Since both of the height and the maximum in-degree are given as $n-1$, the error $\tilde{F}^{-1}(a) - F^{-1}(a)$ is $O(\sqrt{n^3 \log n})$ by Corollary 2.

6. Concluding remarks

Many applications need linear time algorithms as they need process huge instances. Motivated by this, this paper has presented a linear time algorithm for approximating, in the sense below, the length of a longest path of a given DAG, where the length of each edge is given as a random variable following a normal distribution. Let $F(x)$ be the distribution function of the length of a longest path of the DAG. Our algorithm computes a normal distribution function $\tilde{F}(x)$ that satisfies $\tilde{F}(x) \leqslant F(x)$ as long as $F(x) \geqslant a$, given a constant a $(1/2 \leqslant a < 1)$. In other words, it computes an upper bound $1 - \tilde{F}(x)$ on the tail probability $1 - F(x)$, provided $x \geqslant \tilde{F}^{-1}(a)$.

Next we have demonstrated the performance of A-DAG by using as instances a standard benchmark set ITC'99 of logical circuits, and observed that A-DAG exhibits sufficiently good performance. We have also conducted a worst case analysis to bound the error $\tilde{F}^{-1}(a) - F^{-1}(a)$; assuming that the variance of an edge length is bounded by a constant, we showed that the error is $O(\sqrt{h^3 \ln d})$, where h and d are the height and the maximum in-degree of G. This worst case result is by no means a welcomed one, but there are some good news. Firstly, the experimental results using ITC'99 do show the correlation between the height and the error. Secondly, as long as the circuits in ITC'99 are concerned, the height h is around $n^{1/4}$, where n is the number of gates when n is large. Thus the error can be bounded from above by $n^{1/2}$ practically.

We used Monte Carlo simulations to draw $F(x)$. Since A-DAG does not obtain the correct value $F^{-1}(a)$, one might argue that Monte Carlo simulation would be superior to A-DAG. As argued earlier, we would like to emphasize two points. Firstly, A-DAG returns a normal distribution in terms of its mean and variance, which is more useful for further investigation than just knowing $F^{-1}(a)$. Secondly, we need to repeat Monte Carlo simulation many times to calculate $F(x)$ accurately. If we impose the same linear time constraint on Monte Carlo simulation, we can repeat it only a constant number of times, and hence we cannot guarantee the accuracy.

Before closing, let us list some open problems. An open question is to design an algorithm for the case $0 < a < 1/2$. As readers might have noticed, all algorithms shown before A-DAG correctly work for any $0 < a < 1$. Thus there is a good chance to extend A-DAG to handle the case $0 < a < 1/2$.

From standpoints of applications, obvious requests are other linear time algorithms to solve other important stochastic problems possibly under more realistic models. In the paper we mentioned that our model for a gate can be extended to treat wire delay. However, in reality assuming that the distribution of delay follows a normal distribution may be too strong, because the normal distribution gives some probability of negative values which might not be appropriate, for example, in the context of calculating delays in logical circuits. Other important problems include the flow, the minimum cut, the minimum spanning tree problems.

Recent works on approximation algorithms usually provide approximation ratio guarantees with the time and/or the space complexity as parameters. Although we can construct a linear time algorithm by appropriately tune the parameters, those algorithms may not be able to fully use ideas that are applicable particularly to linear time algorithms. It may be interesting to consider a bound on approximation ratio assuming a linear time approximation algorithm.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful suggestions. The authors are grateful to Yusuke Matsunaga for introducing us the ITC'99 benchmark set. This work was supported in part by a Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture of Japan. This work was supported in part by Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists.

Appendix A. Approximation of $\mathrm{Inv}(F_1, F_2, \ldots, F_k; y)$

For $i = 1, 2, \ldots, k$, let $F_i(x)$ be a normal distribution function with mean μ_i and variance σ_i^2. In this paper, $\mathrm{Inv}(F_1, F_2, \ldots, F_k; y)$ is assumed to return the correct value of x_0 in $O(k)$ time, where x_0 is a (unique) real number such that $F_1(x_0)F_2(x_0)\ldots F_k(x_0) = y$ holds. We justify this assumption.

When $k = 1$, we can calculate $\mathrm{Inv}(F_1; y) = F_1^{-1}(y) = x_0$ by referring to the table of the standard normal distribution Φ, since $x_0 = F_1^{-1}(y) = \mu_1 + \sigma_1 \Phi^{-1}(y)$. When $k \geqslant 2$, we may apply e.g., the Newton's method or the binary search method referring to the table of the standard normal distribution, to calculate x_0. Our assumption essentially assumes that the Newton's method or the binary search method terminates in a constant number of repetitions and produces the correct root up to the required level of accuracy. We have used this method in the experiments in Section 5.1 and confirmed that its performance is good. We hence suggest to use it in real applications. However, its time complexity has not been formally discussed. For the completeness, in the following, we describe another method that approximates x_0 in $O(k)$ time with the level of accuracy required by our algorithm A-DAG.

Let $\mu_{\max} = \max_{1 \leqslant i \leqslant k} \mu_i$ and $\sigma_{\max} = \max_{1 \leqslant i \leqslant k} \sigma_i$. We can have that $\mu_{\max} = \mu_1$ without loss of generality. Let $F_M(x)$ be the distribution function of a normal distribution $N(\mu_{\max}, \sigma_{\max}^2)$. It is easy to see that $F_M(x) \leqslant F_i(x)$ for any $1 \leqslant i \leqslant k$ and $\mu_{\max} \leqslant x$, which implies that $\mathrm{Inv}(F_1, F_2, \ldots, F_k; y) < F_M^{-1}(y^{(1/k)})$ if $1/2 \leqslant y^{(1/k)} < 1$. We use $F_M^{-1}(y^{(1/k)})$ as the approximation of $\mathrm{Inv}(F_1, F_2, \ldots, F_k; y)$, which takes $O(k)$ time to compute. Note that we do not have to consider the case $0 < y^{(1/k)} < 1/2$.

Let us bound the error. Since $F_1(x) \geqslant \prod_{1 \leqslant i \leqslant k} F_i(x)$ for all x, we have $F_1^{-1}(y) \leqslant \mathrm{Inv}(F_1, F_2, \ldots, F_k; y)$ for all $0 < y < 1$. Since $\Phi^{-1}(y^{(1/k)}) = \Theta(\sqrt{\log k})$ by Lemma 11, $F_M^{-1}(y^{(1/k)}) - F_1^{-1}(y) = O(\sigma\sqrt{\log k})$. The fact that $O(\sigma\sqrt{\log k})$-error does not affect the analysis of A-DAG is easy to see, and its proof is left to readers.

References

[1] M.O. Ball, C.J. Colbourn, J.S. Proban, Network reliability, in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), Handbooks in Operations Research and Management Science, vol. 7: Network Models, Elsevier Science B.V., 1995, pp. 673–762.

[2] M. Berkelaar, Statistical delay calculation, a linear time method, in: Proc. the International Workshop on Timing Analysis (TAU '97), 1997, pp. 15–24.

[3] C.E. Clark, The greatest of a finite set of random variables, Operations Research 9 (1961) 145–152.

[4] C.E. Clark, The PERT model for the distribution of an activity time, Operations Research 10 (1962) 405–406.

[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, 1979.

[6] G. Grimmett, D. Stirzaker, Probability and Random Processes, third ed., Oxford U.P., 2001.

[7] M. Hashimoto, H. Onodera, A performance optimization method by gate sizing using statistical static timing analysis, IEICE Trans. Fundamentals E83-A (12) (2000) 2558–2568.

[8] ITC99 benchmark home page, http://www.cerc.utexas.edu/itc99-benchmarks/bench.html.

[9] J.E. Kelley Jr., Critical-path planning and scheduling: Mathematical basis, Operations Research 10 (1962) 912–915.

[10] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, third ed., Holt, Rinehart and Winston, USA, 1976.

[11] D.G. Malcolm, J.H. Roseboom, C.E. Clark, W. Fazar, Application of a technique for research and development program evaluation, Operations Research 7 (1959) 646–669.

[12] J.J. Martin, Distribution of the time through a directed, acyclic network, Operations Research 13 (1965) 46–66.

[13] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.

[14] S. Tsukiyama, M. Tanaka, M. Fukui, Estimation method for probabilistic delay of critical path on combinational logical circuits, in: IEICE Proc. 13th Workshop on Circuits and Systems, 2000, pp. 131–135 (in Japanese).