Bounding Variance and Expectation of Longest Path Lengths in DAGs

Jeff Edmonds* Supratik Chakraborty†

Abstract

We consider the problem of computing bounds on the variance and expectation of the longest path length in a DAG from knowledge of variance and expectation of edge lengths. We focus primarily on the case where all edge lengths are non-negative and the DAG has a single source and sink node. We present analytic bounds for various simple DAG structures, and present a new algorithm to compute bounds for more general DAG structures. Our algorithm is motivated by an analogy with balance of forces in a network of "strange" springs.

1 Introduction

Consider a directed acyclic graph (DAG) G with a single source node s and a single sink node t, in which each edge $i = \langle a, b\rangle$ has a non-negative weight x_i. Such DAGs are commonly used to represent timed precedence constraints between jobs or events (e.g. timed marked graphs [10, 1], PERT charts [5], task graphs [7] and precedence constraint diagrams [8]). The edge weights in such a DAG correspond to delays between jobs or events. Hence, we will refer to edge weights and edge delays interchangeably. The starting time of the job associated with the source node s is assumed to be 0. The starting time of every other job b ($\neq s$) is defined to be $\max_{a \in Parent(b)}$(starting time of $a + x_{\langle a,b\rangle}$). If all edge delays are constant, the starting time of the job associated with b can be determined by computing the longest path length from s to b in G [3, 9]. If, however, the edge delays are random, the starting time of a job is determined by a random variable. Let X_G be the random variable denoting the starting time of the job associated with the sink node t of DAG G. If the joint probability distribution of the x_i's is known, techniques for computing the distribution of the sum and maximum of random variables [6, 4] can be used to obtain the distribution of X_G. Monte Carlo simulations [11] can also be used to study the distribution of X_G in such cases. However, specifying the joint probability distribution of all x_i's amounts to specifying all joint moments of x_i's. In a practical setting, this often involves making idealized assumptions. An interesting question to ask, therefore, is how well can we characterize X_G given only the first few moments of each x_i. Such a characterization must hold across all joint distributions of x_i's that preserve the first few moments of every x_i. This has potential applications in statistical timing analysis and performance analysis, and motivates our current work.

We are interested in studying bounds on the moments of X_G as a function of G and moments of each individual x_i. Specifically, suppose we know the mean m_i and variance v_i, but not the complete distribution, of the delay x_i of each edge $i = \langle a, b\rangle$ in G. We wish to establish bounds on the mean, m_G, and variance, v_G, of X_G, where the random variables x_i can be dependent in arbitrary ways (including being independent). This problem was studied earlier in [2], where a dynamic programming algorithm for computing conservative bounds on m_G and v_G was proposed, and experimentally validated against a few distributions. Unfortunately, the approach in [2] neither computes tight bounds of m_G or v_G, nor helps in identifying probability distributions of x_i's that lead to maximum or minimum values of m_G and v_G. In this paper, we try to address these deficiencies partly. Specifically, we identify tight upper bounds of m_G and v_G and also probability distributions that achieve these bounds. The corresponding problems for lower bounds still remain open.

Let P be the set of paths from s to t in G. Each path $p \in P$ can be thought of as the set of edges $i = \langle a, b\rangle$ along the path. The starting time X_G of the job associated with the single sink node t is delayed by the fact that jobs along every st path must complete sequentially. In other words, $X_G = \text{Max}_{p\in P}\left[\sum_{i\in p} x_i\right]$. Two extreme examples of DAGs are *series* and *parallel* graphs. A DAG G is a *series* graph if it consists of only one st path. In this case, $X_G = \sum_i x_i$. A DAG G is a *parallel* graph if it consists only of multiple st-edges. In this case, $X_G = \text{Max}_i x_i$. These extreme cases have been studied earlier in different contexts, e.g. in the study of linear combinations of random variables, and in the study of order statistics [4]. The situation for *series-parallel* graphs is, however, more complicated than one would expect. The problem for a general DAG with a single source and single sink node is even more complicated, and is the primary focus of this paper.

*York University, Canada. jeff@cs.yorku.ca. Supported in part by NSERC Canada.

†IIT Bombay, India. supratik@cse.iitb.ac.in

Our contributions can be summarized as follows:

1. We introduce a special kind of probability distribution called *cake distribution* for edge delays. This allows us to independently control the mean and variance of path delays while ensuring that the mean and variance of edge delays stay unchanged.

2. We present tight upper bounds of the mean and variance of X_G when edge delays are dependent in arbitrary ways, and present techniques for computing these bounds. We also identify cake distributions of edge delays that cause these bounds to be achieved.

3. We present lower bounds of the mean and variance of X_G that are not always achievable, but can be achieved under certain conditions.

4. We show a continuum of values for the mean and variance of X_G. We also show that extreme values in this continuum can be achieved simultaneously, within small factors.

5. We show that the the maximum variance of X_G in a series-parallel graph can be obtained by recursively applying the expressions for maximum variance in series and parallel graphs. However, a similar recursive application does not give tight bounds for the maximum mean of X_G in a series-parallel graph.

The remainder of this paper is organized as follows. Section 2 introduces *cake distributions* and discusses some properties of these distributions. In Section 3, we present a technique for computing a tight upper bound of the variance of X_G. We also identify edge delay distributions that cause this bound to be achieved. Section 4 presents tight upper bounds of the mean of X_G and identifies corresponding edge delay distributions. In Section 5, we present lower bound results, which are, however, not necessarily tight. Section 6 discusses the above problems for the important special case of series-parallel graphs. Finally, we conclude in Section 7.

2 Random variables and cake distributions

A convenient way to represent a random variable x_i is as a function $f_i : [0,1] \to \Re^{\geq 0}$. For clarity of exposition, we will abuse notation and use x_i to denote both the random variable and the corresponding function. In order to choose a value for x_i, we choose r uniformly at random from $[0,1]$, and then the value of the random variable x_i is given by the function $x_i(r)$. By choosing different functions $[0,1] \to \Re^{\geq 0}$, random variables with different probability distributions can be specified.

Consider a set of random variables $\{x_1, x_2, \dots x_n\}$. In general, there may be k groups in the set such that variables within the same group are dependent, while all variables in one group are independent of those in another group. In order to choose values for all the variables, we choose a real value r uniformly randomly in $[0,1]$, and then derive k uniformly randomly distributed real values $r_1, r_2, \dots r_k$ from r, such that each $r_j \in [0,1]$. One way of doing this is to have the w^{th} digit in the decimal representation of r_j to be the $(w.k+j)^{th}$ digit in the decimal representation of r for all $w \in \aleph_0$ and $j \in \{1, \dots k\}$. Since r is chosen uniformly randomly in $[0,1]$, the variables $r_1, r_2, \dots r_k$ are independent and uniformly random in $[0,1]$ as well. The values of all variables in the j^{th} group of the set $\{x_1, x_2, \dots x_n\}$ can now be obtained by evaluating the corresponding functions with r_j as the argument. Alternatively, the functions can be specified to take r as an argument, derive r_j from it and then give the values of the corresponding random variables. Thus, arbitrary dependencies (including independence) of a set of random variables can be represented by choosing the functions $[0,1] \to \Re^{\geq 0}$ appropriately. We will assume all random variables are represented as functions in this way. Choosing values for a set of random variables therefore amounts to choosing a single real value r uniformly randomly in $[0,1]$ and evaluating the corresponding functions.

For a random variable x_i represented in this way, the expected value of x_i is the area under the curve $x_i(r)$ between $r=0$ and $r=1$. Thus, $m_i = \mathrm{Exp}[x_i] = \int_{r\in[0,1]} x_i(r)\ \delta r$. Similarly, the second moment is given by $u_i = \mathrm{U}[x_i] = \mathrm{Exp}[x_i^2] = \int_{r\in[0,1]} x_i(r)^2\ \delta r$. Finally, the variance is given by $v_i = \mathrm{Var}[x_i] = u_i - m_i^2$.

Suppose we wish to know how $\mathrm{Exp}[x_i]$ and $\mathrm{Var}[x_i]$ change when the function $x_i : [0,1] \to \Re^{\geq 0}$ is changed infinitesimally. In order to produce such an infinitesimal change, we must change $x_i(r)$ by an infinitesimal amount in an infinitesimally small interval in $[0,1]$. In view of this, we choose two infinitesimals, δr and δx. To make things less confusing, we will assume that the domain $[0,1]$ of x_i is divided into slices of width δr such that the function $x_i(r)$ has a constant value within each slice. In the following, when we say that $x_i(\widehat{r})$ *is increased by* δx, we mean that the value of $x_i(r)$ is increased by δx for all r in the slice $[\widehat{r}, \widehat{r}+\delta r]$. We will now consider what effect such a change has on $\mathrm{Exp}[x_i]$, $\mathrm{Exp}[x_i]^2$, $\mathrm{U}[x_i]$, and $\mathrm{Var}[x_i]$.

LEMMA 2.1. *Increasing $x_i(\widehat{r})$ by δx increases* $\mathrm{Exp}[x_i]$ *by $\delta r \delta x$, and increases* $\mathrm{Var}[x_i]$ *by $2\Delta_i(\widehat{r})\delta r\delta x$, where $\Delta_i(\widehat{r}) = x_i(\widehat{r}) - \mathrm{Exp}[x_i]$.*

Proof. $\mathbf{Exp[x_i]}$**:** As given above, the expected value of x_i is $\int_{r\in[0,1]} x_i(r)\ \delta r$. Increasing $x_i(\widehat{r})$ by δx

increases this area by a small rectangle of area $\delta r \delta x$. Hence, $\text{Exp}[x_i]$ increases by $\delta r \delta x$.

$\text{Exp}[\mathbf{x_i}]^2$: The chain rule gives that $\frac{\delta m^2}{\delta x} = 2m \cdot \frac{\delta m}{\delta x}$. Hence, the square of the expectation $\text{Exp}[x_i]^2$ increases by $2.\text{Exp}[x_i].\delta r \delta x$.

$\text{U}[\mathbf{x_i}]$: The value $x_i(\widehat{r})$ increases by δx. Hence, by the chain rule the value $x_i(\widehat{r})^2$ increases by $2.x_i(\widehat{r}).\delta x$. The area under the curve $x_i(r)^2$, i.e., $\int_{r\in[0,1]} x_i(r)^2 \ \delta r$, has this change occur in a block of width δr. Hence, the second moment $\text{U}[x_i] = \text{Exp}[{x_i}^2]$ increases by $2.x_i(\widehat{r}).\delta r \delta x$.

$\text{Var}[\mathbf{x_i}]$: We know $\text{Var}[x_i] = \text{U}[x_i] - \text{Exp}[x_i]^2$. Hence, the variance $\text{Var}[x_i]$ increases by $[2.x_i(\widehat{r}).\delta r \delta x] - [2.\text{Exp}[x_i].\delta r \delta x] = 2.\Delta_i(\widehat{r}).\delta r \delta x$. ■

We will have occasion to use Lemma 2.1 later in Section 3.

One of the challenges in choosing distributions for edge delays x_i such that the mean or variance of X_G is maximised (or minimised), is the interplay between its expected value and its variance. *Cake distributions*, as defined below, attain "independence" between these two measures. These distributions are called *cakes* because the function $x_i : [0,1] \to \Re^{\geq 0}$ defining the distribution looks like a cake with infinitesimally thin candles on it. The cake itself is flat and accounts for the expectation of the distribution, but does not contribute to its variance. In contrast, each candle is infinitesimally thin and either has zero height or is infinitely high. Each infinitely high candle, being infinitesimally thin, contributes only to the variance of the distribution, but not to its expectation. As we will see, cake distributions are particularly useful for proving several bounds we are interested in.

For purposes of our discussion, all cake distributions are assumed to have candles in the same predefined locations $\{r^q \mid q \in Q\}$ where Q is a finite index set. Furthermore, all candles have the same fixed width ϵ^2, where ϵ is an infinitesimal (approaching zero). What changes from one cake distribution to the next is the height m of the cake and the height of each of its candles.

Suppose the random edge delay x_i has a cake distribution, where the height of the cake is m_i. Let the height of the candle at location r^q for edge delay x_i be $h_{\langle i,q\rangle}$. To help us better make the connection between candle heights and variance, we will associate with each candle the parameter $v_{\langle i,q\rangle}$, where $h_{\langle i,q\rangle} = \frac{\sqrt{v_{\langle i,q\rangle}}}{\epsilon}$. More formally, the distribution of x_i is specified by the tuple $(m_i, \{v_{\langle i,q\rangle} \mid q \in Q\})$, and is defined as follows.

$$x_i(r) = \begin{cases} h_{\langle i,q\rangle} & \text{if } r \in [r^q, r^q + \epsilon^2],\ q \in Q,\ h_{\langle i,q\rangle} > 0 \\ m_i & \text{otherwise} \end{cases}$$

Lemma 2.2. *Let x_i have a cake distribution with parameters $(m_i, \{v_{\langle i,q\rangle} \mid q \in Q\})$. Then* $\text{Exp}[x_i] = m_i$ *and* $\text{Var}[x_i] = \sum_{q\in Q} v_{\langle i,q\rangle}$.

Proof. $\text{Exp}[x_i] = \int_{r\in[0,1]} x_i(r) \ \delta r = (1 - \epsilon^2.|Q|).m_i + \sum_{q\in Q} \epsilon^2.h_{\langle i,q\rangle}$. Since ϵ is an infinitesimal, $|Q|$ is finite and $h_{\langle i,q\rangle} = \frac{\sqrt{v_{\langle i,q\rangle}}}{\epsilon}$, it follows that $\text{Exp}[x_i] = m_i$. Similarly, $\text{Exp}[x^2] = \int_{r\in[0,1]} (x_i(r))^2 \ \delta r = (1 - \epsilon^2.|Q|).m^2 + \sum_{q\in Q} \epsilon^2.{h_{\langle i,q\rangle}}^2$. For the same reasons as above, it now follows that $\text{Exp}[x^2] = m^2 + \sum_{q\in Q} v_{\langle i,q\rangle}$. Therefore, $\text{Var}[x_i] = \text{Exp}[(x_i)^2] - m^2 = \sum_{q\in Q} v_{\langle i,q\rangle}$. ■

Suppose our goal is to distribute the variance v_i of each edge delay x_i among the different candle locations in a way that maximizes $\text{Var}[X_G]$. If G is a series graph, i.e. $X_G = \sum_i x_i$, then the desire is for each x_i to put its entire candle height in the same location. On the other hand, if G is a parallel graph, i.e., $X_G = \text{Max}_i x_i$, then the desire is for the x_i's to put their candle heights in different locations, so that none of the non-zero candle heights are subsumed by others. If G is an arbitrary graph, there is a complex balance between these two desires in order to maximize X_G. What is clear, however, is that a number of different candle locations may be needed. In the extreme, the number of non-zero candle locations will be at most the number of edges in the DAG G. For now, however, we will have one candle location r^p for each st-path $p \in P$. In other words, the index set Q referred to above is identified with the set P of st-paths in G. The height $h_{\langle i,p\rangle}$ of the candle at location r^p for edge delay x_i will be non-zero only if edge i is in the path p (henceforth denoted $i \in p$). This ensures that each path p dominates $X_G(r) = \text{Max}_{p\in P}\left[\sum_{i\in p} x_i(r)\right]$ when r is within its own candle, i.e. between r^p and $r^p + \epsilon$. It also ensures that $\text{Var}[x_i] = \sum_{p\ni i} v_{\langle i,p\rangle}$, where $p \ni i$ denotes $p \in \{\pi \mid \pi \in P \wedge i \in \pi\}$.

Lemma 2.3. *Suppose each edge i in DAG G has a cake distribution with parameters $(m_i, \{v_{\langle i,p\rangle} \mid p \in P\})$. It follows that the resulting distribution of X_G is also a cake distribution with parameters $(m_G, \{v_{\langle G,p} \mid p \in P\})$, where $m_G = \text{Max}_{p\in P}\left[\sum_{i\in p} m_i\right]$ and $v_{\langle G,p\rangle} = \left[\sum_{i\in p} \sqrt{v_{\langle i,p\rangle}}\right]^2$ for each $p \in P$.*

Proof. By definition, $X_G = \text{Max}_{p\in P} \sum_{i\in p} x_i$. Tracing out $X_G(r)$ for each $r \in [0,1]$, we see that X_G itself has a cake distribution. Specifically, when r is not in a candle, i.e. $r \notin [r^p, r^p + \epsilon]$ for $p \in P$, we have $x_i(r) = m_i$ for all edges i, by definition of a cake distribution. Therefore, $X_G(r) = \text{Max}_{p\in P}\left[\sum_{i\in p} m_i\right]$

for $r \notin [r^p, r^p + \epsilon]$ and $p \in P$. Hence the height of the overall cake is $m_G = \text{Max}_{p\in P}\left[\sum_{i\in p} m_i\right]$.

For $r \in [r^p, r^p + \epsilon]$, we have $x_i(r) = h_{\langle i,p\rangle} = \frac{\sqrt{v_{\langle i,p\rangle}}}{\epsilon}$ for every edge i in p, and $x_i(r) = 0$ for every edge $i \notin p$. This gives the height of the candle at r^p for X_G as $h_{\langle G,p\rangle} = X_G(r^p) = \text{Max}_{q\in P}\left[\sum_{i\in q} h_{\langle i,p\rangle}\right]$. By our construction, the cake distribution for x_i (corresponding to edge i) has a candle of zero height at location r^q if edge i is not in the path q. Also, all $h_{\langle i,p\rangle}$'s are non-negative for $i \in p$. It follows that $h_{\langle G,p\rangle} = \sum_{i\in p)} h_{\langle i,p\rangle}$.

Translating between heights $h_{\langle G,p\rangle}$ and parameters $v_{\langle G,p\rangle}$ gives $\frac{\sqrt{v_{\langle G,p\rangle}}}{\epsilon} = \left[\sum_{i\in p}\frac{\sqrt{v_{\langle i,p\rangle}}}{\epsilon}\right]$ or equivalently, $v_{\langle G,p\rangle} = \left[\sum_{i\in p}\sqrt{v_{\langle i,p\rangle}}\right]^2$. ■

It follows from Lemma 2.2 and Lemma 2.3 that $\text{Exp}[X_G] = m_G = \text{Max}_{p\in P}\left[\sum_{i\in p} m_i\right]$ and $\text{Var}[X_G] = \sum_{p\in P} v_{\langle G,p\rangle} = \sum_{p\in P}\left[\sum_{i\in p}\sqrt{v_{\langle i,p\rangle}}\right]^2$.

3 A tight upper bound of $\text{Var}[X_G]$

We now consider an arbitrary DAG G with a single source and single sink node, and present an algorithm, motivated by balance of forces in a system of strange "springs", to compute a tight upper bound of $\text{Var}[X_G]$. We also show that cake distributions of edge delays allow us to achieve this bound in an arbitrary DAG.

Let μ_i be a real number in $(0, 1]$ associated with edge i in G such that for every st-path $p \in P$, $\sum_{i\in p}\mu_i = 1$. That such an assignment of μ_i's exists can be shown by arranging the nodes in G along a straight line of length 1. Let λ_a denote the location of node a along this line. We fix the source node s at location $\lambda_s = 0$ and the sink node t at location $\lambda_t = 1$. All other nodes are placed between these two end points in a linear/topological ordering of the DAG. In other words, for every edge $i = \langle a, b\rangle$, we ensure that $0 \le \lambda_a < \lambda_b \le 1$. If we now choose $\mu_i = \lambda_b - \lambda_a$ for every edge $i = \langle a, b\rangle$, we obtain the desired assignment of μ_i's. The above argument also shows that there are multiple (in fact, infinite) ways of assigning μ_i's such that $0 < \mu_i \le 1$ and $\sum_{i\in p}\mu_i = 1$ for every st-path $p \in P$.

In the following, we will use $\vec{\mu_i}$ to denote a vector of assignments of μ_i to edges i in G such that the above constraints are satisfied. Similarly, we will use $\vec{x_i}$ to denote a vector of probability distributions of random variables x_i corresponding to edges i in G, such that $\text{Var}[x_i] = v_i$ and $\text{Exp}[x_i] = m_i$.

The following result, though simple, will prove particularly useful in several subsequent proofs. Let $\vec{y_i}$ and $\vec{z_i}$ be vectors of non-negative real values such that $0 \le z_i \le 1$ and $\sum_i z_i = 1$. Let $S = \sum_i \frac{y_i}{z_i}$.

LEMMA 3.1. $\text{Min}_{\vec{z_i}} S = \left[\sum_i \sqrt{y_i}\right]^2$.

Proof. Let $\vec{z_i^\star}$ be a vector of assignments that minimizes S subject to the constraints $0 \le z_i \le 1$ and $\sum_i z_i = 1$. Then, the derivative $\frac{\delta S}{\delta z_i}$ must be zero at $z_i = z_i^\star$, where δz_i is a change that respects the constraints on z_i's. Since $\sum_i z_i^\star = 1$, there must be at least one i such that $z_i^\star > 0$. Let $\epsilon^\star = \text{Min}\left[\{z_i^\star \mid z_i^\star > 0\}\right]$, and let j be such that $z_j^\star = \epsilon^\star$. We now choose ϵ such that $0 < \epsilon < \epsilon^\star$ and a k distinct from j, and increase $z_k^\star$ by ϵ and decrease $z_j^\star$ by ϵ. This ensures that $\sum_i z_i = 1$ and $0 \le z_i \le 1$ for all i. We can now compute $\frac{\delta S}{\delta z_i}$ as $-\frac{y_k}{(z_k^\star)^2} + \frac{y_j}{(z_j^\star)^2}$. Setting this to zero gives $\frac{y_k}{(z_k^\star)^2} = \frac{y_j}{(z_j^\star)^2}$. In other words, $z_k^\star = \sqrt{y_k}\cdot\frac{z_j^\star}{\sqrt{y_j}}$. Since $1 = \sum_k z_k^\star = \sum_k \sqrt{y_k}\cdot\frac{z_j^\star}{\sqrt{y_j}}$, we get $z_j^\star = \frac{\sqrt{y_j}}{\left[\sum_k \sqrt{y_k}\right]}$. Hence $z_i^\star = \frac{\sqrt{y_i}}{\left[\sum_k \sqrt{y_k}\right]}$, for all i. Plugging this in gives $S = \sum_i \frac{y_i}{z_i^\star} = \sum_i y_i \cdot \frac{\left[\sum_k \sqrt{y_k}\right]}{\sqrt{y_i}} = \left[\sum_i \sqrt{y_i}\right]\cdot\left[\sum_k \sqrt{y_k}\right]$, as required. ■

The primary result of this section can now be stated as follows.

THEOREM 3.1. $\text{Max}_{\vec{x_i}}\text{Var}[X_G] = \text{Min}_{\vec{\mu_i}}\sum_i \frac{v_i}{\mu_i}$, *where the values* $\vec{\mu_i}$ *are constrained so that every* μ_i *lies in* $(0, 1]$ *and for every st-path* $p \in P$, $\sum_{i\in p}\mu_i = 1$. *Furthermore, there is an algorithm that computes* $\vec{\mu_i^*}$ *such that* $\sum_i \frac{v_i}{\mu_i^*} = \text{Min}_{\vec{\mu_i}}\sum_i \frac{v_i}{\mu_i}$.

To prove Theorem 3.1, we will first present an algorithm based on balance of forces in a system of "strange" springs that allows us to compute $\vec{\mu_i^*}$. We will then show that $\text{Max}_{\vec{x_i}}\text{Var}[X_G]$ is bounded above and below by $\sum_i \frac{v_i}{\mu_i^*}$.

Computing $\vec{\mu_i^*}$ by a spring algorithm: For purposes of this discussion, we view nodes in the DAG as balls of unit mass, and edges in the DAG as "strange" springs connecting the balls. The balls corresponding to the source node s and sink node t are *fixed* at a distance 1 apart, and are not allowed to move. Balls corresponding to all other nodes are free to move. These are initially arranged in a straight line between s and t in a linear ordering of the DAG, as discussed above. Using the notation introduced earlier, let λ_a be the location of the ball corresponding to node a, where $\lambda_s = 0$ and $\lambda_t = 1$. The spring corresponding to edge $i = \langle a, b\rangle$ exerts an outward repelling force on the balls corresponding to nodes a and b. The "strange" part about these springs is that the force F_i pushing a and b apart is given by $F_i = \frac{v_i}{(\mu_i)^2}$, where v_i, the variance of x_i, is the analogue of the spring constant, and $\mu_i = \lambda_b - \lambda_a$ is the separation between the two ends of the spring. The inverse square law dependence of forces on separation is reminiscent of electrical force laws between charged particles or gravitational force laws between bodies with gravitational mass. However, we choose to use the analogy with springs since not every ball directly exerts force on every other ball in our setting, unlike charged particles or bodies with gravitational mass. Once we let go of all the balls except those corresponding to s and t in our setting, the spring forces set the balls in motion along the straight line joining s and t. If we dampen the movements, the potential plus kinetic energy of the system must decay until all balls come to rest in a state in which the potential energy of the system is minimized. The force on every ball, except those corresponding to s and t, must be balanced in this state. For each edge $i = \langle a, b\rangle$, the value of μ_i^* can then be read off as the distance $\lambda_b^* - \lambda_a^*$ when the system comes to rest. We will call the above technique for obtaining $\vec{\mu_i^*}$ the *spring "algorithm"*. In practice, non-linear constraint solving techniques must be used to solve the set of constraints corresponding to zero net force on each ball other than s and t, while ensuring $\lambda_a < \lambda_b$ for each edge $\langle a, b\rangle$ in G.

LEMMA 3.2. *Let $\vec{\mu_i^*}$ be the values obtained from the spring algorithm when the spring system comes to rest. Then $\sum_i \frac{v_i}{\mu_i^*} = \mathrm{Min}_{\vec{\mu_i}} \sum_i \frac{v_i}{\mu_i}$.*

Proof. The first step is to prove that there is a one-to-one mapping between the domain of values $\vec{\mu_i}$ allowed by Theorem 3.1 and those allowed by the spring algorithm. In one direction, note that the values $\vec{\mu_i}$ produced by the algorithm have the property that for every st-path $p \in P$, $\sum_{i\in p} \mu_i = 1$. This is because the μ_i's are lengths of edges along a path spanning from $\lambda_s = 0$ to $\lambda_t = 1$. Also no μ_i produced by the algorithm can be negative. This is because initially $\lambda_b > \lambda_a$ for every edge $i = \langle a, b\rangle$. For λ_b to subsequently become less than λ_a, the spring system must go through a state where λ_a is arbitrarily close to λ_b, and hence μ_i is arbitrarily close to 0. However, given the force laws of our springs, the force of the spring corresponding to edge $\langle a, b\rangle$ must then increase without bounds, pushing a and b apart. Conversely, if the values $\vec{\mu_i}$ have the property that for every $p \in P$, $\sum_{i\in p} \mu_i = 1$, then the "positions" λ_a of nodes defined by $\lambda_a = \sum_{i\in \text{ any path from } s \text{ to } a} \mu_i$ is well-defined. To see this, consider any two paths p and p' from s to a. We claim that $\sum_{i\in p} \mu_i = \sum_{i\in p'} \mu_i$. To see why this is so, let p'' be some path from a to t. Note that both $\langle p, p''\rangle$ and $\langle p', p''\rangle$ are paths from s to t and hence are in the set of paths P. Hence, $\left[\sum_{i\in p} \mu_i\right] + \left[\sum_{i\in p''} \mu_i\right] = 1 = \left[\sum_{i\in p'} \mu_i\right] + \left[\sum_{i\in p''} \mu_i\right]$. This implies $\sum_{i\in p} \mu_i = \sum_{i\in p'} \mu_i$.

The remaining step is to prove that the $\vec{\mu_i^*}$ returned by the spring algorithm ensures that the value $V = \sum_i \frac{v_i}{\mu_i}$ is minimized. The derivative of V at $\vec{\mu_i^*}$ is obtained by considering an infinitesimal legal change in $\vec{\mu_i^*}$. A legal change in $\vec{\mu_i^*}$ is achieved by moving the ball corresponding to some node a from its current position λ_a to $\lambda_a + \epsilon$ in the direction of the ball corresponding to the sink node t. This increases μ_i^* for each edge $i \in In(a)$, where $In(a)$ denotes the set of edges $\langle b, a\rangle$ entering a. Similarly, the legal change decreases $\mu_{i'}^*$ for each edge $i' \in Out(a)$, where $Out(a)$ is the set of edges $\langle a, b'\rangle$ leaving a. From the force equation for our springs, increasing μ_i^* by ϵ decreases $\frac{v_i}{\mu_i^*}$ by $\frac{v_i}{(\mu_i^*)^2}.\epsilon$. Note that $\frac{v_i}{(\mu_i^*)^2}$ is also equal to the force F_i exerted by the spring corresponding to edge i when the separation is μ_i^*. Hence, the overall derivative is $\frac{\delta V}{\epsilon} = -\sum_{i\in In(a)} F_i + \sum_{i'\in Out(a)} F_{i'}$. Since the net force on every ball other than those corresponding to nodes s and t is 0 when the system of springs comes to rest, we must have $\sum_{i\in In(a)} F_i$ (total force pushing a towards t) $= \sum_{i'\in Out(a)} F_{i'}$ (total force pushing a towards s). Therefore, $\frac{\delta V}{\epsilon} = 0$ at $\vec{\mu_i^*}$. Since the legal change in $\vec{\mu_i^*}$ moves the ball corresponding to a towards the ball corresponding to t, by the inverse square law of forces for our springs, $\sum_{i'\in Out(a)} F_{i'}$ increases and $\sum_{i\in In(a)} F_i$ reduces due to the legal change in $\vec{\mu_i^*}$. Therefore, $\frac{\delta V}{\epsilon} = -\sum_{i\in In(a)} F_i + \sum_{i'\in Out(a)} F_{i'}$ increases with ϵ, giving rise to a positive second derivative of V at $\vec{\mu_i^*}$. Since we have already shown above that $\frac{\delta V}{\epsilon} = 0$ at $\vec{\mu_i^*}$, it follows that $V = \sum_i \frac{v_i}{\mu_i}$ is minimized at $\vec{\mu_i^*}$. ■

For a more physical interpretation of the same proof, notice that $\sum_i \frac{v_i}{\mu_i}$ is the potential energy of the system. Since energy is force times distance, the potential energy of a spring is obtained by integrating the force F_i needed to push one end of the spring from infinity to its current location, namely $\int_{\mu_i=\infty}^{\mu_i} F_i \delta\mu_i = \int_{\mu_i=\infty}^{\mu_i} \frac{v_i}{(\mu_i)^2} \delta\mu_i = \frac{v_i}{\mu_i}$. Therefore, the total potential energy of the system is $\sum_i \frac{v_i}{\mu_i}$. As stated, the algorithm finds a state of minimum potential energy.

LEMMA 3.3. *There exists an algorithm whose input is a DAG G, and variance v_i and mean m_i of each edge i in G, and whose output is a cake distribution $(m_i, \{v_{\langle i,p\rangle} \mid p \in P\})$ for each x_i such that $\mathrm{Var}[X_G] = \sum_{q\in P} v_{\langle G,q\rangle} = \sum_i \frac{v_i}{\mu_i^*}$, and for every edge i in G, $\mathrm{Var}[x_i] = \sum_{p\in P} v_{\langle i,p\rangle} = v_i$ and $\mathrm{Exp}[X_i] = m_i$.*

Proof. Recall the spring algorithm and consider the spring system in its final state of rest. Let $\mu_i^* = \lambda_b^* - \lambda_a^*$ be the length of the spring for edge $i = \langle a, b\rangle$ and $F_i = \frac{v_i}{(\mu_i^*)^2}$ be the force in this spring. Note that $\forall p \in P$, $\sum_{i\in p} \mu_i^* = 1$ and that for each node $a \in$

$G \setminus \{s,t\}$, $\sum_{i \in In(a)} F_i = \sum_{i' \in Out(a)} F_{i'}$. Lemma 3.4 then produces the contribution $v_{\langle G,p \rangle}$ of each path $p \in P$ to the variance of X_G such that $F_i = \sum_{p \ni i} v_{\langle G,p \rangle}$. Finally, the contribution $v_{\langle i,p \rangle}$ of each candle to the variance of x_i is given by $v_{\langle i,p \rangle} = \frac{v_{\langle G,p \rangle} \cdot v_i}{F_i}$ if $i \in p$, and 0 otherwise.

Lemma 2.2 gives $\text{Exp}[x_i] = m_i$, and $\text{Var}[x_i] = \sum_{p \in P} v_{\langle i,p \rangle} = \sum_{p \ni i} v_{\langle i,p \rangle}$. This translation from paths $p \in P$ to paths $p \ni i$ going through edge i is possible because $v_{\langle i,q \rangle}$ is zero unless the path p includes edge i. Plugging in the value for $v_{\langle i,p \rangle}$ gives $\text{Var}[x_i] = \sum_{p \ni i} \frac{v_{\langle G,p \rangle} \cdot v_i}{F_i} = \frac{v_i}{F_i} \cdot \sum_{p \ni i} v_{\langle G,p \rangle}$. The requirement given by Lemma 3.4 simplifies this to $\text{Var}[x_i] = \frac{v_i}{F_i} \cdot F_i = v_i$, as required.

Lemma 2.2 also gives the variance of X_G to be $\text{Var}[X_G] = \sum_{p \in P} v_{\langle G,p \rangle}$. For each path p, we have that $\sum_{i \in p} \mu_i^* = 1$. Hence, $\text{Var}[X_G] = \sum_{p \in P} \left[\sum_{i \in p} \mu_i^* \right] v_{\langle G,p \rangle} = \sum_i \mu_i^* \left[\sum_{p \ni i} v_{\langle G,p \rangle} \right]$. The requirement given by Lemma 3.4 simplifies this to $\text{Var}[X_G] = \sum_i \mu_i^* \left[F_i \right] = \sum_i \mu_i^* \left[\frac{v_i}{(\mu_i^*)^2} \right] = \sum_i \frac{v_i}{\mu_i^*}$, as required. ∎

Lemma 3.4. *There exists an algorithm whose input is F_i such that for each node $a \in G \setminus \{s,t\}$, $\sum_{i \in In(a)} F_i = \sum_{i' \in Out(a)} F_{i'}$, and whose output is the contribution $v_{\langle G,p \rangle}$ of each path $p \in P$ to the variance of X_G such that $F_i = \sum_{p \ni i} v_{\langle G,p \rangle}$.*

Proof. For each node $a \in G \setminus \{s,t\}$, let $h_a = \langle a,b \rangle$ be the "first" edge out of a in some ordering of its edges. Let $H = \{h_a \mid a \in G \setminus \{s,t\}\}$ be the set of these first edges and let $\overline{H} = G \setminus H$ be all the remaining edges in G. We now do a depth (or breadth) first search of the DAG from the source s, building the search tree. The only requirement we impose during this search is that the first edge traversed from every node a must be h_a.

For each edge $i = \langle a,b \rangle \in \overline{H}$, let $p_i \in P$ be the path that follows the search tree edges from the source s to node a, then follows the edge $i = \langle a,b \rangle$, and keeps following the first edges $h_{a'}$ from node b onward until the sink t is reached. Note this path must eventually find t because G is a DAG with t being the only sink.

Define M to be the $|\overline{H}| \times |\overline{H}|$ matrix such that for all $i,j \in \overline{H}$, $M_{\langle i,j \rangle} = 1$ iff edge i is in path p_j. We claim that M is invertible. This is because if we assume that the rows and columns are sorted based on the order in which the search finds the edges in $\overline{H}$, then the diagonal is all ones (edge i is in path p_i) and the lower triangle is all zeros. To see why the lower triangle is all zeros, note that path p_j has the property that it does not contain those edges from $\overline{H}$ that were found in the search later than j, i.e. for which $i > j$

For each $i,j \in \overline{H}$, we compute the required value $v_{\langle G,p_j \rangle}$ from the known values F_i using the matrix operation $\langle v_{\langle G,p_1 \rangle}, v_{\langle G,p_2 \rangle}, \ldots, v_{<G,p_{|\overline{H}|}>} \rangle^T = M^{-1} \langle F_1, F_2, \ldots, F_{|\overline{H}|} \rangle^T$. For all the remaining paths, we set $v_{\langle G,p \rangle} = 0$. Note that this algorithm is polynomial-time because the number of paths/candles that have non-zero height is $|Edges(G) - Nodes(G) + 2|$.

What remains is to prove that for every edge i, the required statement $F_i = \sum_{p \ni i} v_{\langle G,p \rangle}$ is true. Let us start with edges $i \in \overline{H}$. The matrix operation $\langle F_i \rangle^T = M \langle v_{\langle G,p_j \rangle} \rangle^T$ ensured that $F_i = \sum_{\langle p \ni i,\ p \in p_{\overline{H}} \rangle} v_{\langle G,p \rangle}$, where we only consider paths $p \in p_{\overline{H}}$ that are associated with an edge $j \in \overline{H}$. However, $\sum_{\langle p \ni i\ p \notin p_{\overline{H}} \rangle} v_{\langle G,p \rangle} = 0$, because $v_{\langle G,p \rangle} = 0$ for all the other paths. Hence, the desired result follows.

This leaves proving that for every edge $i \in H$, the required statement $F_i = \sum_{p \ni i} v_{\langle G,p \rangle}$ is true. We prove this by induction on the distance of the edge from the source s. Recall $H = \{h_a \mid a \in G \setminus \{s,t\}\}$ is the set of "first" edges h_a out of node a. As a base case of our induction, all edges out of s are in $\overline{H}$, and therefore $F_i = \sum_{p \ni i} v_{\langle G,p \rangle}$ for all such edges i. Now, consider edge $h_a \in H$ for some node $a \in G \setminus \{s,t\}$. By way of the inductive hypothesis, the statement has been proved for all edges coming into node a. All edges leaving a, except for the edge h_a, are in $\overline{H}$ and hence the statement is true for them as well. We will now prove $F_{h_a} = \sum_{p \ni h_a} v_{\langle G,p \rangle}$. To do this, we use the fact that the spring system settled into a state of rest. Hence for each node $a \in G \setminus \{s,t\}$, $\sum_{i \in In(a)} F_i = \sum_{i' \in Out(a)} F_{i'}$. This gives

$$
\begin{aligned}
F_{h_a} &= \left[\sum_{i \in In(a)} F_i \right] - \left[\sum_{i' \in Out(a) - h_a} F_{i'} \right] \\
&= \left[\sum_{i \in In(a)} \sum_{p \ni i} v_{\langle G,p \rangle} \right] - \left[\sum_{i' \in Out(a) - h_a} \sum_{p \ni i'} v_{\langle G,p \rangle} \right] \\
&= \left[\sum_{p \ni a} v_{\langle G,p \rangle} \right] - \left[\sum_{\langle p \ni a,\ p \not\ni h_a \rangle} v_{\langle G,p \rangle} \right] \\
&= \sum_{p \ni h_i} v_{\langle G,p \rangle}
\end{aligned}
$$

This completes the proof. ∎

Lemma 3.5. $\text{Max}_{\vec{x_i}} \text{Var}[X_G] \geq \text{Min}_{\vec{\mu_i}} \sum_i \frac{v_i}{\mu_i} = \sum_i \frac{v_i}{\mu_i^*}$

The proof of Lemma 3.5 follows from Lemma 3.3. To complete the proof of Theorem 3.1, we need to show the following.

Lemma 3.6. $\text{Max}_{\vec{x_i}} \text{Var}[X_G] \leq \text{Min}_{\vec{\mu_i}} \sum_i \frac{v_i}{\mu_i} = \sum_i \frac{v_i}{\mu_i^*}$.

Proof. Let $\vec{\mu_i}$ be an arbitrary real-valued vector such that for every edge i in G, $0 < \mu_i \leq 1$ and $\sum_{i \in p} \mu_i = 1$ for all st-paths $p \in P$. Our proof proceeds by showing that $\mathrm{Var}[X_G] \leq \sum_i \frac{v_i}{\mu_i}$ for every $\vec{x_i}$ such that $\mathrm{Var}[x_i] = v_i$. This is obtained as a special case of a more general result which states that if $\mathrm{Z}[\vec{x_i}] = \mathrm{Var}[X_G] - \sum_i \frac{\mathrm{Var}[x_i]}{\mu_i}$, then $\mathrm{Z}[\vec{x_i}] \leq 0$ for all $\vec{x_i}$.

Towards this goal, let $\vec{x_i^\bullet}$ be a vector of edge delay distributions that maximizes $\mathrm{Z}[\vec{x_i}]$, and gives the smallest $\mathrm{Var}[X_G]$ among all vectors $\vec{x_i}$ that maximize $\mathrm{Z}[\vec{x_i}]$. We consider two cases below.

- Suppose $\mathrm{Var}[X_G] = 0$ for $\vec{x_i^\bullet}$. Since $\mathrm{Var}[x_i^\bullet]$ is the expectation of a square, $\sum_i \frac{\mathrm{Var}[x_i^\bullet]}{\mu_i} \geq 0$. Hence, $\mathrm{Z}[\vec{x_i^\bullet}] = \mathrm{Var}[X_G] - \sum_i \frac{\mathrm{Var}[x_i^\bullet]}{\mu_i} \leq 0$. Since $\vec{x_i^\bullet}$ maximizes $\mathrm{Z}[\vec{x_i}]$, it follows that $\mathrm{Z}[\vec{x_i}] \leq 0$ for all $\vec{x_i}$.
- Suppose $\mathrm{Var}[X_G] > 0$ for $\vec{x_i^\bullet}$. In this case, Lemma 3.7 tells us that there exists a way of changing $\vec{x_i^\bullet}$ such that either $\mathrm{Var}[X_G]$ decreases or $\mathrm{Z}[\vec{x_i}]$ increases (or both). In case $\mathrm{Z}[\vec{x_i}]$ increases, we end up contradicting the fact that $\vec{x_i^\bullet}$ maximizes $\mathrm{Z}[\vec{x_i}]$. In case $\mathrm{Z}[\vec{x_i}]$ stays the same and $\mathrm{Var}[X_G]$ decreases, we get more than one vector $\vec{x_i}$ that maximize $\mathrm{Z}[\vec{x_i}]$. However, since $\mathrm{Var}[X_G]$ decreases, this contradicts the fact that $\vec{x_i^\bullet}$ gives the smallest $\mathrm{Var}[X_G]$ among all vectors $\vec{x_i}$ that maximize $\mathrm{Z}[\vec{x_i}]$. Therefore, $\mathrm{Var}[X_G]$ cannot be strictly positive for $\vec{x_i^\bullet}$.

It follows that $\mathrm{Var}[X_G] = 0$ for $\vec{x_i^\bullet}$, and hence $\mathrm{Z}[\vec{x_i}] \leq \mathrm{Z}[\vec{x_i^\bullet}] \leq 0$. The theorem follows from this result. ∎

LEMMA 3.7. *If* $\mathrm{Var}[X_G] > 0$ *for a given* $\vec{x_i}$*, there is a way of changing* $\vec{x_i}$ *so that either* $\mathrm{Var}[X_G]$ *decreases or* $\mathrm{Z}[\vec{x_i}]$ *increases or both.*

In order to prove Lemma 3.7, we will first prove Lemma 3.8, which is almost the reverse of what we want. Nevertheless, it turns out that Lemma 3.8 is easier to prove since it has fewer negatives in it, and helps in proving Lemma 3.7.

LEMMA 3.8. *If* $\mathrm{Var}[X_G] > 0$ *for a given* $\vec{x_i}$*, there is a way of changing* $\vec{x_i}$ *so that* $\mathrm{Var}[X_G]$ *increases and* $\mathrm{Z}[\vec{x_i}]$ *either stays the same or decreases.*

Proof. Our goal is to increase $\mathrm{Var}[X_G]$. There are two ways of increasing the variance of the random variable X_G. We could either increase $X_G(r)$ at those points $r \in [0,1]$ where $X_G(r) > \mathrm{Exp}[X_G]$, or we could decrease $X_G(r)$ at points $r \in [0,1]$ where $X_G(r) < \mathrm{Exp}[X_G]$. We will do the former. For this purpose, let $D_G(r) = X_G(r) - \mathrm{Exp}[X_G]$ and let $\widehat{r} \in [0,1]$ be a point at which $\Delta_G(\widehat{r}) > 0$. Since $\mathrm{Var}[X_G] > 0$, such a point must exist.

Lemma 2.1 considered the effect of increasing the edge delay x_i for one edge. We will now consider the effect of changing a whole path of edge delays. Let $\widehat{p}$ be a winning path in $X_G(\widehat{r}) = \mathrm{Max}_{p \in P}\left[\sum_{i \in p} x_i(\widehat{r})\right]$. For each edge $i \in \widehat{p}$, we increase $x_i(\widehat{r})$ by $\mu_i \delta x$, where $\vec{\mu_i}$ is the fixed vector of values chosen in the proof of Lemma 3.6. We will now consider what effect this has on $\sum_i \frac{\mathrm{Var}[x_i]}{\mu_i}$, $\mathrm{Var}[X_G]$, and $\mathrm{Z}[\vec{x_i}]$.

$\mathbf{\sum_i \frac{Var[x_i]}{\mu_i}}$**:** By Lemma 2.1, increasing $x_i(\widehat{r})$ by $\mu_i.\delta x$ increases the variance $\mathrm{Var}[x_i]$ by $2.\mu_i.\Delta_i(\widehat{r}).\delta r \delta x$. Hence, doing this for each edge in the path $\widehat{p}$ increases the sum $\sum_i \frac{\mathrm{Var}[x_i]}{\mu_i}$ by $\sum_{i \in \widehat{p}} \frac{2.\mu_i.\Delta_i(\widehat{r}).\delta r \delta x}{\mu_i} = \sum_{i \in \widehat{p}} 2.\Delta_i(\widehat{r}).\delta r \delta x$.

$\mathrm{Var}[\mathbf{X_G}]$**:** By definition, $\widehat{p}$ is a winning path in $X_G(\widehat{r}) = \mathrm{Max}_{p \in P}\left[\sum_{i \in p} x_i(\widehat{r})\right]$. Increasing $x_i(\widehat{r})$ by $\mu_i.\delta x$ for every $i \in \widehat{p}$ increases $\sum_{i \in \widehat{p}} x_i(\widehat{r})$, and hence $X_G(\widehat{r})$, by $\sum_{i \in \widehat{p}} \mu_i.\delta x = \delta x$. The reason for the last simplification is that $\vec{\mu_i}$ was chosen such that $\forall p \in P$, $\sum_{i \in p} \mu_i = 1$. By Lemma 2.1, increasing $X_G(\widehat{r})$ by δx increases the variance $\mathrm{Var}[X_G]$ by $2.\Delta_G(\widehat{r}).\delta r \delta x$. By our choice of $\widehat{r}$, we have $\Delta_G(\widehat{r}) > 0$. Hence, $\mathrm{Var}[X_G]$ increases, as required.

$\mathrm{Z}[\vec{\mathbf{x_i}}]$**:** By definition, $\mathrm{Z}[\vec{x_i}] = \mathrm{Var}[X_G] - \sum_i \frac{\mathrm{Var}[x_i]}{\mu_i}$. Hence, $\mathrm{Z}[\vec{x_i}]$ changes by $[2.\Delta_G(\widehat{r}).\delta r \delta x] - \left[\sum_{i \in \widehat{p}} 2.\Delta_i(\widehat{r}).\delta r \delta x\right] = 2.(M - N).\delta r \delta x$, where $M = X_G(\widehat{r}) - \sum_{i \in \widehat{p}} x_i(\widehat{r})$, and $N = \mathrm{Exp}[X_G] - \sum_{i \in \widehat{p}} \mathrm{Exp}[x_i]$.

We now examine the two sub-expressions M and N obtained above.

$M = \mathbf{X_G(\widehat{r})} - \sum_{\mathbf{i \in \widehat{p}}} \mathbf{x_i(\widehat{r})}$**:** Because $\widehat{p}$ is a winning path in $X_G(\widehat{r}) = \mathrm{Max}_{p \in P}\left[\sum_{i \in p} x_i(\widehat{r})\right]$, therefore $X_G(\widehat{r}) = \sum_{i \in \widehat{p}} x_i(\widehat{r})$. Hence, $M = X_G(\widehat{r}) - \sum_{i \in \widehat{p}} x_i(\widehat{r}) = 0$.

$N = \mathrm{Exp}[\mathbf{X_G}] - \sum_{\mathbf{i \in \widehat{p}}} \mathrm{Exp}[\mathbf{x_i}]$**:** As we will see in Lemma 5.1, $\mathrm{Exp}[X_G] \geq \sum_{i \in p} \mathrm{Exp}[x_i]$ for all st-paths p in P. It follows that $N = \mathrm{Exp}[X_G] - \sum_{i \in \widehat{p}} \mathrm{Exp}[x_i] \geq 0$.

Therefore, the change in $\mathrm{Z}[\vec{x_i}]$, i.e., $2.(M - N)$, is non-positive. Hence, $\mathrm{Z}[\vec{x_i}]$ either stays the same or decreases, as required. ∎

We can now turn Lemma 3.8 around to get the result we really want, i.e. Lemma 3.7.

Proof of Lemma 3.7: Similar to the proof of Lemma 3.8, we choose a point $\widehat{r} \in [0,1]$ such that $\Delta_G(\widehat{r}) = X_G(\widehat{r}) - \mathrm{Exp}[X_G] > 0$. Let $\widehat{p}$ be a winning path in $X_G(\widehat{r}) = \mathrm{Max}_{p\in P}\left[\sum_{i\in p} x_i(\widehat{r})\right]$. Instead of increasing $x_i(\widehat{r})$ by $\mu_i\delta x$ for each edge $i \in \widehat{p}$ (as in the proof of Lemma 3.8), we now decrease each $x_i(\widehat{r})$ by these amounts. There are two cases to consider.

In the first case, suppose $X_G(\widehat{r})$ does in fact decrease. Since $X_G(\widehat{r})$ was greater than $\mathrm{Exp}[X_G]$ (since $\Delta_G(\widehat{r}) > 0$), this decreases $\mathrm{Var}[X_G]$, and we are done.

In the second case, suppose $X_G(\widehat{r})$ does not decrease even though we decreased $x_i(\widehat{r})$ by $\mu_i\delta x$ for each edge $i \in \widehat{p}$. The proof of Lemma 3.8 then tells us that $\mathrm{Z}[\vec{x_i}]$ increases by $\sum_{i\in\widehat{p}} 2.\Delta_i(\widehat{r}).\delta r\delta x$. If we can now show that $\sum_{i\in\widehat{p}}\Delta_i(\widehat{r}) > 0$, we will be done. We show this by proving below that there exists a point $\widehat{r} \in [0,1]$ for which $\Delta_G(\widehat{r}) > 0$ and $\sum_{i\in\widehat{p}}\Delta_i(\widehat{r}) > 0$.

Let $R = \{r \in [0,1] \mid X_G(r) > \mathrm{Exp}[X_G]\}$. Since $\mathrm{Var}[X_G] > 0$ by assumption, R must be non-empty. Hence the cumulative width of all r's in R, i.e. $|R|$, must be > 0. Let $A = \frac{1}{|R|}\cdot\int_{r\in R} X_G(r)\delta r$ denote the average of $X_G(r)$ within R. Clearly by the definition of R, we have $A > \mathrm{Exp}[X_G]$. With abuse of notation, let $\widehat{p}(r)$ denote a winning path in $X_G(r) = \mathrm{Max}_{p\in P}\left[\sum_{i\in p} x_i(r)\right]$, for every $r \in [0,1]$. If there is some $r \in R$ for which $\sum_{i\in\widehat{p}(r)}\Delta_i(r) > 0$, then we can simply choose $\widehat{r}$ to be this r, and we are done. Otherwise, we must have $\forall r \in R$, $\sum_{i\in\widehat{p}(r)} x_i(r) \le \sum_{i\in\widehat{p}(r)}\mathrm{Exp}[x_i]$. However, we show below that this leads to a contradiction.

Assume that $\forall r \in R$, $\sum_{i\in\widehat{p}(r)} x_i(r) \le \sum_{i\in\widehat{p}(r)}\mathrm{Exp}[x_i]$, if possible. Let us reconsider the average $A = \frac{1}{|R|}\cdot\int_{r\in R} X_G(r)\delta r$. By the definition of X_G and $\widehat{p}(r)$, this is equal to $\frac{1}{|R|}\cdot\int_{r\in R}\left[\sum_{i\in\widehat{p}(r)} x_i(r)\right]\delta r$. By our assumption, this is at most $\frac{1}{|R|}\cdot\int_{r\in R}\left[\sum_{i\in\widehat{p}(r)}\mathrm{Exp}[x_i]\right]\delta r$. The last expression, in turn, is at most $\frac{1}{|R|}\cdot\int_{r\in R}\mathrm{Max}_{p\in P}\left[\sum_{i\in p}\mathrm{Exp}[x_i]\right]\delta r$. Since the integrand no longer depends on r, it can be factored out to give $\frac{1}{|R|}\cdot\mathrm{Max}_{p\in P}\left[\sum_{i\in p}\mathrm{Exp}[x_i]\right]\cdot\int_{r\in R} 1.\delta r$. This simplifies to $\mathrm{Max}_{p\in P}\left[\sum_{i\in p}\mathrm{Exp}[x_i]\right]$. Finally, Lemma 5.1 gives that this is at most $\mathrm{Exp}[X_G]$. Putting all the parts together, we find that $A = \frac{1}{|R|}\cdot\int_{r\in R} X_G(r).\delta r \le \mathrm{Exp}[X_G]$. However, we have already shown that $A > \mathrm{Exp}[X_G]$. Hence we have a contradiction! ∎

Well known measures of a DAG G are its *height* h and *width* w. The *height* h of G is defined to be the number of edges in the longest path from the source s to the sink t. The *width* w of G is the minimum number of st-paths needed to cover each edge of the graph at least once. The following lemma uses the height and width of a DAG to bound the maximum variance of X_G.

LEMMA 3.9. *If G has height h and width w, then* $\frac{1}{w}\left[\sum_i\sqrt{v_i}\right]^2 \le \mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G] \le h\sum_i v_i$.

Proof. By Theorem 3.1, $\mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G] = \mathrm{Min}_{\vec{\mu_i}}\sum_i\frac{v_i}{\mu_i} = \sum_i\frac{v_i}{\mu_i^*}$. Recall from the spring algorithm that for every edge $i = \langle a,b\rangle$ in G, $\mu_i^* = \lambda_b^* - \lambda_a^*$, where λ_a^* and λ_b^* are the locations of the balls corresponding to nodes a and b respectively, when the system of springs is at rest. If, instead, we put the balls at other locations and use the corresponding μ_i's, then the sum $\sum_i\frac{v_i}{\mu_i}$ can only increase.

Let us now define the location λ'_a of node a to be the maximum number of edges in a path from s to a divided by h. The number of edges in a path from s to s is clearly zero giving $\lambda'_s = 0$, as required. By definition of the height h of G, the maximum number of edges in a path from s to t is h. Hence, $\lambda'_t = 1$, as required. For each edge $i = \langle a,b\rangle$ in G, the maximum number of edges in a path from s to b is at least one more than that in a path from s to a. Hence, $\mu'_i = \lambda'_b - \lambda'_a \ge \frac{1}{h}$. It follows that $\mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G] = \mathrm{Min}_{\vec{\mu_i}}\sum_i\frac{v_i}{\mu_i} \le \sum_i\frac{v_i}{\mu'_i} \le h.\sum_i v_i$. This proves one part of the lemma.

For the other part, let $\widehat{P}$ be a set of st-paths that cover each edge of G at least once, and let $|\widehat{P}| = w$. Because each edge is covered at least once, we have $\sum_i\mu_i \le \sum_{p\in\widehat{P}}\sum_{i\in p}\mu_i$. By definition, for every path, $\sum_{i\in p}\mu_i = 1$. Hence, $\sum_i\mu_i \le w$.

By Lemma 3.1, we know that subject to the constraint $\sum_i\mu_i = 1$, $\mathrm{Min}_{\vec{\mu_i}}\sum_i\frac{v_i}{\mu_i} = \left[\sum_i\sqrt{v_i}\right]^2$. Scaling all μ_i's by a factor of w gives that subject to $\sum_i\mu_i = w$, $\mathrm{Min}_{\vec{\mu_i}}\sum_i\frac{v_i}{\mu_i} = \frac{1}{w}\left[\sum_i\sqrt{v_i}\right]^2$. Reducing the μ_i's can only increase $\frac{v_i}{\mu_i}$. Hence, subject to $\sum_i\mu_i \le w$, we must have $\mathrm{Min}_{\vec{\mu_i}}\sum_i\frac{v_i}{\mu_i} \ge \frac{1}{w}\left[\sum_i\sqrt{v_i}\right]^2$. ∎

4 A tight upper bound of $\mathrm{Exp}[X_G]$

We now consider the problem of obtaining a tight upper bound of $\mathrm{Exp}[X_G]$ for an arbitrary DAG G with a single source node s and an single sink node t. For every st-path $p \in P$, let τ_p be a positive real number in $[0,1]$ such that $\sum_{p\in P}\tau_p = 1$. For each edge i in G, we define $\tau_i = \sum_{p\ni i}\tau_p$. Clearly, $0 \le \tau_i \le 1$ for all edges i. Similar to notation used earlier, we will use $\vec{\tau_p}$ and $\vec{\tau_i}$ to denote a vector of assignments of τ_p to paths $p \in P$ and a vector of assignments of τ_i to edges i in G, respectively, such that the above constraints are satisfied.

THEOREM 4.1. *For every DAG G,* $\text{Max}_{\vec{x_i}}\text{Exp}[X_G] = \text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$, *where the values* $\vec{\tau_i}$ *are constrained so that there exists a value* τ_p *for each path* $p \in P$ *such that* $\forall i,\ \sum_{p \ni i} \tau_p = \tau_i$, $0 \le \tau_p \le 1$, *and* $\sum_{p \in P} \tau_p = 1$.

Proof. Our proof has two parts. In the first part, given an expectation m_i and a variance v_i for each edge $i \in G$, we construct a vector of distributions $\vec{x_i}$ such that $\text{Exp}[X_G] \ge \text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$. In the second part, we show that $\text{Max}_{\vec{x_i}}\text{Exp}[X_G] \le \text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$.

By definition, $X_G(r) = \text{Max}_{p\in P}\left[\sum_{i\in p} x_i(r)\right]$. For each path $p \in P$, let R_p be the set of r values for which p is the longest path defining X_G, and let $\tau_p = |R_p|$ be the width of this set. For each edge $i \in G$, let $R_i = \cup_{p\ni i} R_p$ be the set of r's for which x_i contributes to the longest path, and let $\tau_i = |R_i| = \sum_{p\ni i}\tau_p$ be the width of this set. Clearly τ_i is the probability that x_i is in the winning path and is able to contribute to $\text{Exp}[X_G]$. We will use $M_i = \int_{r\in R_i} x_i(r)\ \delta r$ to denote the amount that x_i contributes to $\text{Exp}[X_G]$.

Part I: Given an m_i and v_i for each edge $i \in G$, let $\vec{\tau_i^*}$ be the values that realize the maximum $\text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$. Let $\vec{\tau_p^*}$ be the corresponding values for all st-paths in P, such that $\forall i,\ \tau_i^* = \sum_{p\ni i}\tau_p^*$ and $\sum_p \tau_p^* = 1$. We partition the probability space $r \in [0,1]$ into $|P|$ disjoint regions R_p such that for each path $p \in P$, $|R_p| = \tau_p^*$. This partition is possible because $\sum_p \tau_p^* = 1$. For each edge i, we also define $R_i = \cup_{p\ni i} R_p$, such that $|R_i| = \sum_{p\ni i}\tau_p^* = \tau_i^*$. For every distribution $\vec{x_i}$, we then have:

$$\begin{aligned}
\text{Exp}[X_G] &= \int_{r\in[0,1]} X_G(r)\ \delta r \\
&= \int_{r\in[0,1]} \text{Max}_{p'\in P}\left[\sum_{i\in p'} x_i(r)\right]\ \delta r \\
&= \sum_{p\in P}\int_{r\in R_p} \text{Max}_{p'\in P}\left[\sum_{i\in p'} x_i(r)\right]\ \delta r \\
&\ge \sum_{p\in P}\int_{r\in R_p}\left[\sum_{i\in p} x_i(r)\right]\ \delta r \\
&= \sum_i \sum_{p\ni i}\int_{r\in R_p} x_i(r)\ \delta r \\
&= \sum_i \int_{r\in R_i} x_i(r)\ \delta r \ = \ \sum_i M_i
\end{aligned}$$

Note that the inequality arises because the path p' that wins when $r \in R_p$ may be different from p in general.

We now construct a distribution $\vec{x_i^*}$ such that $\sum_i M_i$ for this distribution equals $\sum_i \text{Min}\left[\tau_i^* m_i + \sqrt{\tau_i^*(1-\tau_i^*)v_i},\ m_i\right]$ $= \text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$

We consider two constructions. In the first construction, let $M_i = \tau_i^* m_i + \sqrt{\tau_i^*(1-\tau_i^*)v_i}$. The distribution of x_i is then given by $x_i(r) = \frac{M_i}{|R_i|}$ if $r \in R_i$, and $\frac{m_i - M_i}{|\overline{R_i}|}$ otherwise. It can be verified that $\text{Exp}[x_i] = m_i$ and $\text{Var}[x_i] = m_i$. In the second construction, we define $M_i = m_i$, $u_i = v_i + m_i^2$, and $\tau_i' = \frac{m_i^2}{u_i}$. The key requirement for using the second construction is that $\tau_i^* \ge \tau_i'$. If this requirement is satisfied, we define R_i' to be some subset of R_i of size $|R_i'| = \tau_i'$. The distribution of x_i is then given by $x_i(r) = \frac{M_i}{|R_i'|}$ if $r \in R_i'$, and zero otherwise. Again, it can be checked that $\text{Exp}[x_i] = m_i$ and $\text{Var}[x_i] = m_i$. Thus, if $\tau_i^* < \tau_i'$, we use the first construction, and use the second construction otherwise. Furthermore, it can be shown that if $0 \le \tau_i < \tau_i'$, then $0 \le \tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i} < m_i$. For $\tau_i' \le \tau_i \le 1$, we have $\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i} \ge m_i$. Therefore, the above distribution ensures that $M_i = \text{Min}\left[\tau_i^* m_i + \sqrt{\tau_i^*(1-\tau_i^*)v_i},\ m_i\right]$. Hence, $Max_{\vec{x_i}}\text{Exp}[X_G] \ge \text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$.

Part II: Let $\vec{x_i^{\#}}$ be the vector of distributions that maximizes $\text{Exp}[X_G]$. Let $\tau_p^{\#}$, $\tau_i^{\#}$ and $M_i^{\#}$ be the τ_p, τ_i and M_i that arise from $\vec{x_i^{\#}}$. Using the same computation of $\text{Exp}[X_G]$ as done in part I above, we find that $\text{Exp}[X_G]^{\#} = \text{Max}_{\vec{x_i}}\text{Exp}[X_G] = \sum_i M_i^{\#}$. Note that we have an equality instead of the inequality in part I because the path winning path p' is the same as p by definition of R_p.

We will now show that for every distribution x_i, if $\text{Exp}[x_i] = m_i$ and $\text{Var}[x_i] = v_i$, then $M_i = \int_{r\in R_i} x_i(r)\ \delta r \le \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$. Since $x_i(r) \ge 0$, M_i is clearly at most $\text{Exp}[x_i] = m_i = \int_r x_i(r)\ \delta r$. Similarly,

$$\begin{aligned}
\text{U}[x_i] &= \int_{r\in[0,1]} x_i(r)^2\ \delta r \\
&= \left[\int_{r\in R_i} x_i(r)^2\ \delta r\right] + \left[\int_{r\notin R_i} x_i(r)^2\ \delta r\right] \\
&\ge |R_i|\cdot\left[\frac{M_i}{|R_i|}\right]^2 + |\overline{R_i}|\cdot\left[\frac{m_i - M_i}{|\overline{R_i}|}\right]^2 \\
&= \left[\frac{M_i^2}{\tau_i}\right] + \left[\frac{(m_i - M_i)^2}{1-\tau_i}\right]
\end{aligned}$$

The inequality above comes from the fact that if we know that M_i of x_i's mass lies within R_i and the remaining $m_i - M_i$ lies outside of this region, then the second moment of x_i is minimized by having $x_i(r)$ constant within each of these two regions. From this, we get that

$$\begin{aligned}\mathrm{Var}[x_i] &= \mathrm{U}[x_i] - m_i^2 \\ &\geq \left[\frac{M_i^2}{\tau_i}\right] + \left[\frac{(m_i - M_i)^2}{1-\tau_i}\right] - m_i^2\end{aligned}$$

Since we must have $\mathrm{Var}[x_i] = v_i$, it can be shown that $M_i \leq \tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}$. Specifically, $M_i^{\#} \leq \mathrm{Min}\left[\tau_i^{\#} m_i + \sqrt{\tau_i^{\#}(1-\tau_i^{\#})v_i}, m_i\right]$.

It follows that $\mathrm{Max}_{\vec{x_i}} \mathrm{Exp}[X_G] = \mathrm{Exp}[X_G]^{\#} = \sum_i M_i^{\#} \leq \mathrm{Min}\left[\tau_i^{\#} m_i + \sqrt{\tau_i^{\#}(1-\tau_i^{\#})v_i},\ m_i\right]$. Therefore, $\mathrm{Max}_{\vec{x_i}} \mathrm{Exp}[X_G] \leq \mathrm{Max}_{\vec{\tau_i}} \sum_i \mathrm{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$. Parts I and II together prove Theorem 4.1. ■

We now wish to obtain simpler bounds of $\mathrm{Max}_{\vec{x_i}} \mathrm{Exp}[X_G]$, and also provide an algorithm to approximate the expression for $\mathrm{Max}_{\vec{x_i}} \mathrm{Exp}[X_G]$ given by Theorem 4.1.

Theorem 4.2. *For every DAG G,*

$$\begin{aligned}\mathrm{Max}_{\vec{x_i}} \mathrm{Exp}[X_G] &\leq \mathrm{Max}_{\vec{\tau_i}} \left[\textstyle\sum_i \tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right] \\ &\leq \mathrm{Max}_{p \in P} \left[\textstyle\sum_{i \in p} m_i\right] + \sqrt{\mathrm{Min}_{\vec{\mu_i}} \textstyle\sum_i \frac{v_i}{\mu_i}}.\end{aligned}$$

Furthermore, the optimal τ_i's can be derived from the above formulation.

We will first prove a few simple results that will eventually lead to a proof of Theorem 4.2. We begin with the following observation.

Lemma 4.1. *Let $U(\tau_i) = \mathrm{Min}\left[\tau_i m_i + \sqrt{\tau_i v_i},\ m_i\right]$ and $V(\tau_i) = \mathrm{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$. Then $\frac{1}{2}U(\tau_i) \leq V(\tau_i) \leq U(\tau_i)$.*

Proof. The second inequality is obvious because $(1-\tau_i) \leq 1$. The first inequality follows from the fact that $\tau_i.(1-\tau_i) \geq \frac{1}{4} \geq \frac{1}{4}.\tau_i$. ■

The next natural step would be to understand which of $\tau_i m_i + \sqrt{\tau_i v_i}$ and m_i is lesser for each τ_i. However, we will leave this until later, and presently focus on bounding $\mathrm{Max}_{\vec{\tau_i}} \sum_i \left[\tau_i m_i + \sqrt{\tau_i v_i}\right]$. Let $A(\vec{\tau_i}) = \sum_i \tau_i m_i$ and $B(\vec{\tau_i}) = \sum_i \sqrt{\tau_i v_i}$. Instead of obtaining $\mathrm{Max}_{\vec{\tau_i}} [A(\vec{\tau_i}) + B(\vec{\tau_i})]$ directly, we will maximize A and B independently.

Lemma 4.2. $\frac{1}{2}[\mathrm{Max}_{\vec{\tau_i}} A(\vec{\tau_i})] + [\mathrm{Max}_{\vec{\tau_i}} B(\vec{\tau_i})] \leq \mathrm{Max}_{\vec{\tau_i}} [A(\vec{\tau_i}) + B(\vec{\tau_i})] \leq [\mathrm{Max}_{\vec{\tau_i}} A(\vec{\tau_i})] + [\mathrm{Max}_{\vec{\tau_i}} B(\vec{\tau_i})]$.

Proof. It is clear that $\mathrm{Max}_{\vec{\tau_i}} [A(\vec{\tau_i}) + B(\vec{\tau_i})] \leq [\mathrm{Max}_{\vec{\tau_i}} A(\vec{\tau_i})] + [\mathrm{Max}_{\vec{\tau_i}} B(\vec{\tau_i})]$. Let $\vec{\tau_i}'$ be the values that optimize $A(\vec{\tau_i}')$ and $\vec{\tau_i}''$ be those that optimize $B(\vec{\tau_i}'')$. Let $\vec{\tau_i}$ be such that $\tau_i = \frac{1}{2}[\tau_i' + \tau_i'']$. Note that all requirements on $\vec{\tau_i}$ are met because it is a linear combination of $\vec{\tau_i}'$ and $\vec{\tau_i}''$.

$$\begin{aligned}\frac{1}{2}A(\vec{\tau_i}') + \frac{1}{2}B(\vec{\tau_i}'') &= \frac{1}{2}\left[\sum_i \tau_i' m_i\right] + \frac{1}{2}\left[\sum_i \sqrt{\tau_i'' v_i}\right] \\ &\leq \left[\sum_i \tau_i m_i\right] + \left[\sum_i \sqrt{\tau_i v_i}\right] \\ &\quad \text{for every } \tau_i \\ &= [A(\vec{\tau_i}) + B(\vec{\tau_i})] \\ &\leq \mathrm{Max}_{\vec{\tau_i}} [A(\vec{\tau_i}) + B(\vec{\tau_i})]\end{aligned}$$

This completes the proof. ■

Lemma 4.3. $\mathrm{Max}_{\vec{\tau}} [\sum_i \tau_i m_i] = \mathrm{Max}_p \sum_{i \in p} m_i$.

Proof.

$$\begin{aligned}\mathrm{Max}_{\vec{\tau_i}} \left[\sum_i \tau_i m_i\right] &= \mathrm{Max}_{\vec{\tau_i}} \sum_i \left(\sum_{p \ni i} \tau_p\right) m_i \\ &= \mathrm{Max}_{\langle \sum_p \tau_p = 1 \rangle} \sum_p \tau_p \sum_{i \in p} m_i\end{aligned}$$

Since $0 \leq \tau_p \leq 1$ for all $p \in P$ and $\sum_{p \in P} \tau_p = 1$, the maximum value of $\sum_p \tau_p \sum_{i \in p} m_i$ occurs when $\tau_p = 1$ for the path p with the maximum value of $\sum_{i \in p} m_i$ and $\tau_{p'} = 0$ for all other paths p'. Therefore, $\mathrm{Max}_{\vec{\tau}} [\sum_i \tau_i m_i] = \mathrm{Max}_p \sum_{i \in p} m_i$. ■

Lemma 4.4. $\mathrm{Max}_{\vec{\tau}} \left[\sum_i \sqrt{\tau_i v_i}\right] \leq \sqrt{\mathrm{Min}_{\vec{\mu_i}} \sum_i \frac{v_i}{\mu_i}}$.

Proof. In a "primal-dual" sort of way, it is sufficient to consider any fixed setting for $\vec{\tau}$ and any fixed setting for $\vec{\mu_i}$, and prove that

$$\sum_i \sqrt{\tau_i v_i} \leq \sqrt{\sum_i \frac{v_i}{\mu_i}}$$

Given these fixed $\vec{\tau}$ and $\vec{\mu_i}$, let us define $a_i = \tau_i \cdot \mu_i$. The following is a useful property of the a_i's.

$$\begin{aligned}\sum_i a_i &= \sum_i \tau_i \cdot \mu_i = \sum_i \left[\sum_{p \ni i} \tau_p\right] \cdot \mu_i \\ &= \sum_p \tau_p \left[\sum_{i \in p} \mu_i\right] = \sum_p \tau_p \cdot 1 = 1\end{aligned}$$

The second last equality is because of the restriction on $\vec{\mu_i}$ that $\sum_{i\in p}\mu_i = 1$ and the last equality is because of the restriction on $\vec{\tau}$ that $\sum_p \tau_i = 1$. We can now get the bound we want. Specifically, $\sum_i \frac{v_i}{\mu_i} = \sum_i \frac{\tau_i v_i}{a_i}$. However, by Lemma 3.1, $\sum_i \frac{\tau_i v_i}{a_i} \geq \left[\sum_i \sqrt{\tau_i v_i}\right]^2$ since $\tau_i v_i > 0$ and $\sum_i a_i = 1$. ■

We are now ready to complete the proof of Theorem 4.2. *Proof of Theorem 4.2:* Theorem 4.1 and Lemmas 4.1, 4.2, 4.3, and 4.4 give $\text{Max}_{\vec{x_i}}\text{Exp}[X_G] \leq \text{Max}_{\vec{\tau_i}}\left[\sum_i \tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right] \leq \text{Max}_{p\in P}\left[\sum_{i\in p} m_i\right]$ $+ \sqrt{\text{Min}_{\vec{\mu_i}} \sum_i \frac{v_i}{\mu_i}}$. It is interesting that the second term in the above upper bound is $\sqrt{Max_{\vec{x_i}}\text{Var}[X_G]}$.

Furthermore, Theorem 3.1 gives a spring algorithm to determine the optimal $\vec{\mu_i^*}$ in the proof of Lemma 4.4. We will now use these to find the optimal $\vec{\tau_i^*}$ in Lemma 4.4, and hence in Theorem 4.2. Recall we used Lemma 3.1 in the proof of Lemma 4.4 to show that the maximum value of $sum_i \frac{\tau_i v_i}{a_i}$ is $\left[\sum_i \sqrt{\tau_i v_i}\right]^2$. In fact, the proof of Lemma 3.1 also gives the optimal values of a_i as $\frac{\sqrt{\tau_i v_i}}{\sqrt{V}}$, where $\sqrt{V} = \sum_i \sqrt{\tau_i v_i}$. However, as shown in the proof of Lemma 4.4, the optimal value of $\sum_i \sqrt{\tau_i v_i}$ $= \sqrt{\sum_i \frac{v_i}{\mu_i}}$. Recalling that we defined $a_i = \tau_i \cdot \mu_i$, we get $\tau_i^* = \frac{v_i}{(\mu_i^*)^2.V} = \frac{v_i}{(\mu_i^*)^2.\left(\sum_j \frac{v_j}{\mu_j^*}\right)}$. ■

Thus, the same spring algorithm described in Section 3 gives us the optimal $\vec{\mu_i}$ and also the optimal $\vec{\tau}$. Hence the same algorithm can be used to compute $\text{Max}_{\vec{x_i}}\text{Var}[X_G]$ exactly and also an upper bound of $\text{Max}_{\vec{x_i}}\text{Exp}[X_G]$.

The values $\vec{\tau_i}$ are constrained so that there exists a value τ_p for each path $p \in P$ such that $\forall i$, $\sum_{p\ni i}\tau_p = \tau_i$ and $\sum_{p\in P}\tau_p = 1$.

THEOREM 4.3. *Given $\vec{\tau_i}$ (computed from Theorem 4.2), there is a linear program to find the corresponding $\vec{\tau_p}$. The dual of this linear program involves finding the optimal $\vec{\mu_i}$ such that $\sum_{i\in p}\mu_i = 1$ for all $p \in P$.*

Proof. Let $\vec{\tau_i}$ be the vector of τ_i values given to us. Let $\vec{\tau_p}$ be the vector of τ_p values that we are looking for. Let M be a matrix such that for each path p and edge i, we $M[i,p] = 1$ if and only if $i \in p$. Finally let $\vec{1}$ be the vector of consisting of $|P|$ ones. The required linear program is as follows.

$$\text{Min } \vec{\tau_p} \cdot \vec{1}^T$$

$$\text{Subject to } M \cdot \vec{\tau_p} = \vec{\tau_i}$$

Note that the objective function is $\sum_p \tau_p$, which we hope to be at most one. The constraints effectively encode that for each variable edge i, $\sum_{p\ni i}\tau_p = \tau_i$. The primal-dual theorem states that the optimal value for this primal is the same as the optimal value for the dual problem. Hence, it is sufficient to prove that the optimal value of the dual is at most one. The dual is stated as follows.

$$\text{Max } \vec{\mu_i} \cdot \vec{\tau_i}^T$$

$$\text{Subject to } M^T \cdot \vec{\mu_i} = \vec{1}$$

Note that the requirements on the unknown variables μ_i are that for each path $p \in P$, $\sum_{i\in p}\mu_i = 1$. Indeed, these are the same requirements on μ_i that we have used throughout the paper. ■

THEOREM 4.4. *There exists a greedy algorithm to determine for each variable i, whether $M_i = \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$ is equal to $\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}$ or m_i. This, in turn, gives a quick algorithm to approximate $\text{Max}_{\vec{x_i}}\text{Exp}[X_G] = \text{Max}_{\vec{\tau_i}} \sum_i \text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$ within a factor of four.*

Proof. To convey the intuition, a reasonable starting point would be to find $\vec{\tau_i}$ that maximizes $\text{Max}_{\vec{\tau_i}} \sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$. However, the problem with this solution is that for some edges i, we will have $\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right] \geq m_i$, and hence $\text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}, m_i\right]$ will decrease these terms. Recalling the proof of Theorem 4.1, this occurs when the corresponding edge delay variables x_i are allocated too much of the limited resource τ_i: specifically, $\tau_i \geq \frac{m_i^2}{u_i}$. Decreasing these τ_i's to $\frac{m_i^2}{u_i}$ may allow other edge delay variables x_j to have their τ_j's increased. This in turn may allow the overall expression expression $sum_i\text{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}, m_i\right]$ to increase once again.

Before discussing the final algorithm, we present an idealized version of it. The intuition behind the working of this idealized algorithm is that we start by computing an optimal distribution of τ_i's where $\sum_{p\in P}\tau_p$ is bounded not by 1, but by some infinitesimal ϵ. This forces all τ_i's to be infinitesimal as well, and thereby ensures that each τ_i is less than the threshold of $\frac{m_i^2}{u_i}$. Next, we slowly increase the bound on $\sum_{p\in P}\tau_p$. Each variable's allocation τ_i therefore increases in turn. Each time τ_i (corresponding to edge delay variable x_i) reaches the limit $\frac{m_i^2}{u_i}$ that it is allowed use, this variable is set aside in a set S and the corresponding τ_i is fixed to its maximum value of $\frac{m_i^2}{u_i}$. The bound on $\sum_{p\in P}\tau_p$ continues to increase in this fashion until it reaches the overall limit, i.e., $\sum_p \tau_p = 1$. More formally, given values $\vec{\tau_i}$ for every edge i, let $\vec{\tau_p}$ be the values for which $\sum_p \tau_p$ is minimized subject to the constraint $\forall i,\ \sum_{p\ni i}\tau_p = \tau_i$. This can easily be found by a linear program. Let $Q(\vec{\tau_i}) = \sum_{pinP}\tau_p$, and $U(\vec{\tau_i}) = \sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$. The code for the idealized algorithm can then be presented as follows.

algorithm $\mathsf{FindExp}(G, \langle m_i, v_i\rangle, i \in \{1\ldots n\})$
Inputs: DAG G, (m_i, v_i) for every edge i
Returns: $\vec{\tau_i}$ that maximizes $\mathrm{Exp}[X_G]$

- Let $q = \epsilon$
- Let $\vec{\tau_i}$ maximize $U(\vec{\tau_i})$ subject to $Q(\vec{\tau_i}) = q$.
- Let $S = \emptyset$
- **loop**
  - **loop** increasing q continuously

    Loop Invariants:

    (i) $\vec{\tau_i}$ maximizes $U(\vec{\tau_i})$ subject to $Q(\vec{\tau_i}) = q$.

    (ii) τ_i's for all x_i's in S are fixed.

    * As q increases, adjust $\vec{\tau_i}$ to maintain loop invariant.

      /* $\forall i \notin S$, τ_i increases continually */
    * **if** $(\exists i \notin S$ s.t. $\tau_i = \frac{m_i^2}{u_i})$ exit loop
    * **if** $(q = 1)$ **return**$(\vec{\tau_i})$

    end loop
  - Add i to S

  end loop

end algorithm

The correctness of the idealized algorithm is clear. Unfortunately, it is not implementable as is. The first change needed to make it more implementable is to remove the $(1-\tau_i)$ in the square root. Lemma 4.1 proves that this changes the result by at most a factor of two. The next change is to instead compute $\mathrm{Max}_{\vec{\tau_i'}}\left[\sum_i \tau_i m_i\right] + \mathrm{Max}_{\vec{\tau_i''}}\left[\sum_i \sqrt{\tau_i'' v_i}\right]$ and then to let $\tau_i = \frac{1}{2}[\tau_i' + \tau_i'']$. Lemma 4.2 proves that this changes the result by at most another factor of two and Lemmas 4.3 and 4.4 describe how to compute these values. A useful thing to know is that if each τ_i' is changed by a multiplicative factor of c then both $A = \mathrm{Max}_{\vec{\tau_i'}}\left[\sum_i \tau_i m_i\right]$ and $Q(\vec{\tau_i}')$ change by the same multiplicative factor c. On the other hand, changing each τ_i'' in this way changes $B = \mathrm{Max}_{\vec{\tau_i''}}\left[\sum_i \sqrt{\tau_i'' v_i}\right]$ by a multiplicative factor of $\sqrt{c}$. In both cases, this proves that if $Q(\vec{\tau_i}) = q$ changes by the same multiplicative factor c, then the $\vec{\tau_i}'$ and the $\vec{\tau_i}''$ that optimize A and B remain the same except that each τ_i changes by this multiplicative factor of c. The first advantage of this fact is that the effect of the step in the idealized code that finds the $\vec{\tau_i}$ subject to $Q(\vec{\tau_i}) = q$ can just as well be achieved by finding the $\vec{\tau_i}$ subject to $Q(\vec{\tau_i}) = 1$, and then multiplying the resulting $\vec{\tau_i}$ by q. The second, even more significant, advantage is that the inner loop that continually increases q can be changed in a way that makes the recalculation of the $\vec{\tau_i}$ much easier. Let $\vec{\tau_i}^{old}$ be the current values when the inner loop starts in a given iteration of the outer loop. Note that by the loop invariant, $\vec{\tau_i}^{old}$ maximizes $\sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$ subject to two conditions: (i) $Q(\vec{\tau_i}) = q_{old}$, and (ii) $\forall i \in S$, $\tau_i \geq \frac{m_i^2}{u_i}$. Instead of the inner loop increasing q from q^{old}, it can increase the multiplicative factor c continuously from 1. The new values, $\vec{\tau_i}^{new}$ are obtained as $Mult(c, \vec{\tau_i}^{old})$, which is defined as $\vec{\tau_i}^{old}$ except that for each $i \notin S$, τ_i is increased by a factor of c. Note that $q^{new} = Q(\vec{\tau_i}^{new}) = \sum_{p\in P}\tau_p$ is not simply $c\cdot\vec{\tau}_{iold}$ in general, since the τ_i's do not change for $i \in S$. The optimal τ_p's therefore need to be recomputed. The key, however, is that the loop invariant will still be maintained because the resulting $\vec{\tau_i}^{new}$ will give the optimal values for maximizing $\sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$ subject to $Q(\vec{\tau_i}) = q^{new}$. The final change needed to make the algorithm implementable is that the inner loop cannot increase c continuously, but must compute a value $c \geq 1$ that either makes $\tau_i = \frac{m_i^2}{u_i}$ for some $i \notin S$, or makes $q = 1$. The former is easy because if $\tau_i^{new} = c\cdot\tau_i^{old} = \frac{m_i^2}{u_i}$, then $c = \frac{m_i^2}{\tau_i^{old}\cdot u_i}$. Hence, all that is required is to let $c = \mathrm{Min}_{i\notin S}\frac{m_i^2}{\tau_i^{old}\cdot u_i}$.

It is interesting that the spring algorithm for finding the optimal $\vec{\tau_i}$ actually needs to be run only once at the beginning. ∎

5 Lower bounds and continuum results

In this section, we present lower bounds and continuum results for the mean and variance of X_G for an arbitrary DAG G with a single source node s and a single sink node t. Unlike the upper bounds presented in the previous two sections, our lower bounds are not necessarily tight.

Lemma 5.1. *For every DAG G and for every vector of distributions $\vec{x_i}$ for the underlying edge delay variables,* $\mathrm{Exp}[X_G] \geq \mathrm{Max}_{p\in P}\sum_{i\in p} m_i$.

Proof. Let $\widehat{p}$ be the path that maximizes $\sum_{i\in p} m_i$.

Then

$$
\begin{aligned}
\text{Exp}[X_G] &= \int_{r\in[0,1]} X_G(r)\ \delta r \\
&= \int_{r\in[0,1]} \text{Max}_{p\in P}\left[\sum_{i\in p} x_i(r)\right]\ \delta r \\
&\geq \int_{r\in[0,1]} \left[\sum_{i\in\widehat{p}} x_i(r)\right] \delta r \\
&= \sum_{i\in\widehat{p}} \left[\int_{r\in[0,1]} x_i(r)\ \delta r\right] \\
&= \sum_{i\in\widehat{p}} \text{Exp}[x_i]
\end{aligned}
$$

Therefore, $\text{Exp}[X_G] \geq \text{Max}_{p\in P}\sum_{i\in p} m_i$. ∎

Recall the construction of cake distributions in Lemma 3.3 for maximizing $\text{Var}[X_G]$. That construction ensures that $\text{Exp}[X_G] = \text{Max}_{p\in P}\sum_{i\in p} m_i$. Hence, the maximum variance and minimum expectation can be simultaneously achieved for DAGs with arbitrary dependence of the distributions of x_i's.

There exist several cases where $Min_{\vec{x_i}}\text{Var}[X_G] = 0$, although each of the x_i's has non-zero variance. Note that 0 is the minimum possible value of $\text{Var}[X_G]$ since it is the expectation of a square. For example, consider a series graph with two edges. Suppose $x_1(r) = K$ if $r \in [0, 0.5)$ and 0, elsewhere. Suppose further that $x_2(r) = K$ if $r \in [0.5, 1]$ and 0, elsewhere. Clearly, $x_1 + x_2 = K$ for all $r \in [0, 1]$. Hence $\text{Var}[X_G] = 0$, although the x_i's themselves have non-zero variance.

Lemma 5.2. *Given two random variables x and y, suppose we form a third random variable z by flipping a coin with probability $c \in [0, 1]$. If the coin flip comes up as heads, we let z take the value of x, otherwise we let z take the value of y. Then* $\text{Exp}[z] = c.\text{Exp}[x] + (1-c).\text{Exp}[y]$, $\text{Var}[z] \geq c.\text{Var}[x] + (1-c).\text{Var}[y]$, *and* $\text{U}[z] = c.\text{U}[x] + (1-c).\text{U}[y]$.

Proof. Follows from simple algebra and probability. ∎

Theorem 5.1. *For every DAG G, for all $V \in$* $[\text{Min}_{\vec{x_i}}\text{Var}[X_G], \text{Max}_{\vec{x_i}}\text{Var}[X_G]]$, *there exists $\vec{x_i}$ such that* $\text{Var}[X_G] = V$. *The same continuum result holds for* $\text{Exp}[X_G]$ *and $U[X_G]$ (second moment) as well.*

Proof. Let $\vec{x_i}$ be the distribution of variables that minimizes $\text{Var}[X_G]$ to V_{min}, and let $\vec{y_i}$ be the distribution that maximizes $\text{Var}[Y_G]$ to V_{max}. By extending our earlier notation, we will use Y_G to denote the random variable representing the longest path length from s to t in DAG G when each edge i has the distribution y_i. For each edge i, we now form a third random variable z_i by flipping a coin with probability $c \in [0, 1]$. If the flip gives heads, then the z_i's take the values of x_i's; otherwise the z_i's take the values of y_i's. Therefore, the random variable Z_G behaves like X_G with probability c and behaves like Y_G with probability $1-c$. Hence, by Lemma 5.2, $\text{Exp}[Z_G] = c.\text{Exp}[X_G] + (1-c).\text{Exp}[Y_G]$ and $\text{U}[Z_G] = c.\text{U}[X_G] + (1-c).\text{U}[Y_G]$. It follows that

$$
\begin{aligned}
\text{Var}[Z_G] &= \text{U}[Z_G] - (\text{Exp}[Z_G])^2 \\
&= c.\text{U}[X_G] + (1-c).\text{U}[Y_G] \\
&\quad - (c.\text{Exp}[X_G] + (1-c).\text{Exp}[Y_G])^2
\end{aligned}
$$

Note that if $c = 0$ then $\text{Var}[Z_G] = \text{Var}[X_G] = V_{min}$, and if $c = 1$ then $\text{Var}[Z_G] = \text{Var}[Y_G] = V_{max}$. Hence by the mean value theorem, for every $V \in [V_{min}, V_{max}]$, there exists a $c \in [0, 1]$ such that $\text{Var}[Z_G] = V$. ∎

Lemma 5.3. *There exists a vector of distributions $\vec{x_i}$ such that within a factor of three,* $\text{Var}[X_G] = \text{Max}_{\vec{x_i}}\text{Var}[X_G]$, $\text{Exp}[X_G] = \text{Max}_{\vec{x_i}}\text{Exp}[X_G]$ *and* $\text{U}[X_G] = \text{Max}_{\vec{x_i}}\text{U}[X_G]$.

Proof. Let $\vec{w_i}$ be the distribution that maximizes the variance to $V_{max} = \text{Max}_{\vec{w_i}}\text{Var}[W_G]$. Similarly, let $\vec{x_i}$ be the distribution that maximizes the expectation to $M_{max} = \text{Max}_{\vec{x_i}}\text{Exp}[X_G]$, and let $\vec{y_i}$ be the variables that maximizes the second moment to $U_{max} = \text{Max}_{\vec{y_i}}\text{U}[Y_G]$. As in the proof of Theorem 5.1, let us define a random variable z_i for each edge i such that z_i is equal to one of w_i, x_i or y_i based on a flip of a fair three-way coin. Hence, by Lemma 5.2, $\text{Var}[Z_G] \geq \frac{1}{3}[\text{Var}[W_G] + \text{Var}[X_G] + \text{Var}[Y_G]] \geq \frac{1}{3}\text{Var}[W_G] = \frac{1}{3}V_{max}$. Thus, $\frac{1}{3}V_{max} \leq \text{Var}[Z_G] \leq V_{max}$. By a similar argument, the result for the expectation and second moment of Z_G can also be obtained. ∎

Lemma 5.4. *Within a factor of three,* $\text{Max}_{\vec{x_i}}\text{U}[X_G] = \text{Max}_{\vec{x_i}}\text{Var}[X_G] + (\text{Max}_{\vec{x_i}}\text{Exp}[X_G])^2$.

Proof. A standard equation for variance is $\text{Var}[X_G] = \text{U}[X_G] - \text{Exp}[X_G]^2$. Hence, $\text{U}[X_G] = \text{Var}[X_G] + \text{Exp}[X_G]^2$. Clearly, this is at most $\text{Max}_{\vec{x_i}}\text{Var}[X_G] + [\text{Max}_{\vec{x_i}}\text{Exp}[X_G]]^2$. By using a version of Lemma 5.3, one can simultaneously obtain V_{max} within a factor of a and E_{max} with a factor of $1-a$. This gives ${E_{max}}^2$ within a factor of $(1-a)^2$. Solving $a = (1-a)^2$ gives $\frac{1}{a} = 2.62$. The result follows by choosing a to be this value ($\frac{1}{2.62}$). ∎

6 Series-parallel graphs

In the previous few sections, we looked at arbitrary DAGs with a single source node and a single sink node. In this section, we consider an special class of DAGs called series-parallel graphs. We show that the bounds and algorithms presented in earlier sections can be simplified for this class of DAGs.

A DAG G is said to be a *series* graph if it consists of a single path of edges from the source s to the sink t. Therefore, $X_G = \mathrm{Max}_{p\in P} \sum_{i\in p} x_i = \sum_i x_i$.

A DAG G is said to be a *parallel* graph if it consists only of multiple st-edges. Thus, $X_G = \mathrm{Max}_{p\in P} \sum_{i\in p} x_i = \mathrm{Max}_i x_i$.

The class of *series-parallel* graphs is defined inductively as follows. As a base case, the graph consisting of a single st-edge is a series-parallel graph. Given a set of series-parallel graphs $G_1, G_2, \ldots, G_q$, a graph G obtained by linking $G_1, G_2, \ldots, G_q$ sequentially, so that the sink t_j of one is the source s_{j+1} of the next, is a series-parallel graph. Similarly, the graph obtained by placing $G_1, G_2, \ldots, G_q$ in parallel, so that all of them have a common source node s and a common sink node t, is a series-parallel graph.

THEOREM 6.1. *If G is a series graph, then $X_G = \sum_i x_i$. For such a graph, $\mathrm{Max}_{\vec{x_i}}\mathrm{Exp}[X_G] = \mathrm{Min}_{\vec{x_i}}\mathrm{Exp}[X_G] = \sum_i m_i$. In addition, $\mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G] = \left(\sum_i \sqrt{v_i}\right)^2$.*

Proof. The result about the mean follows from the linearity of expectation. By Theorem 3.1 $\mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G] = \mathrm{Min}_{\vec{\mu_i}} \frac{v_i}{\mu_i}$. However, since $\sum_i \mu_i = 1$ in a series graph, we know from Lemma 3.1 that $\mathrm{Min}_{\vec{\mu_i}} \frac{v_i}{\mu_i} = \left(\sum_i \sqrt{v_i}\right)^2$. ■

To understand $\left[\sum_i \sqrt{v_i}\right]^2$ better, let all the v_i be equal. Then $\left[\sum_i \sqrt{v_i}\right]^2 = [n.\sqrt{v}]^2 = n^2.v = n.\sum_i v_i$.

THEOREM 6.2. *If G is a parallel graph, then $X_G = \mathrm{Max}_i x_i$. For such a graph, $\mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G] = \sum_i v_i$. In addition, $\mathrm{Min}\left[\sum_i m_i,\ \mathrm{Max}_i m_i + \sqrt{\sum_i v_i}\right]$ approximates $\mathrm{Max}_{\vec{x_i}}\mathrm{Exp}[X_G]$ within a factor of four.*

Proof. G consists of only the source s, the sink t, and many parallel edges from s to t. The constraints on $\vec{\mu_i}$ give that $\mu_i = 1$ and hence $\mathrm{Max}_{\vec{x_i}} Var[X_G] = \mathrm{Min}_{\vec{\mu_i}} \sum_i \frac{v_i}{\mu_i} = \sum_i v_i$.

To reason about $\mathrm{Max}_{\vec{x_i}}\mathrm{Exp}[X_G]$, note that G consists of only the source s, the sink t, and many parallel edges from s to t. Thus each edge is contained in its own path, giving $\tau_i = \sum_{p\ni i} \tau_p = \tau_p$. Hence, the requirement that $\sum_p \tau_p = 1$ simplifies to $\sum_i \tau_i = 1$. This makes it even more clear how τ_i is a scarce resource that must be partitioned between the variables.

Given the expectation m_i, variance v_i, and second moment $u_i = v_i + m_i$ for each variable x_i, let us set the availability of resource (τ_i) such that each variable is exactly at the balancing point between the two possible outcomes of $\mathrm{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$. Recall from the proof of Theorem 4.1 that this balancing point occurs when the variable is allocated $\tau_i' = \frac{m_i^2}{u_i}$. Hence, let $q = \sum_i \tau_i' = \sum_i \frac{m_i^2}{u_i}$. As in the proof of Theorem 4.4, let $\vec{\tau_i}$ be the assignment of τ_i's that maximizes $\sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$ subject to $Q(\vec{\tau_i}) = q$. We claim that this maximization happens at $\tau_i = \tau_i'$. This is because with $\tau_i = \tau_i'$, each variable achieves the balance between the two possible outcomes of $\mathrm{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i},\ m_i\right]$, namely $M_i = \tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}$ and m_i. If we allocate a greater value of τ_i to some variable x_i, it would not increase M_i; however if we allocate a lesser value of τ_i, it would decrease M_i. Hence, this allocation maximizes $\sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$.

Now that we understand this balancing point q, let us try either side of it. If $q < 1$, then in reality, more resource is available. This will increase each τ_i, which in turn will increase the value of $\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}$. This means that $\mathrm{Min}\left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}, m_i\right]$ will evaluate to m_i. Thus, we will have $\mathrm{Exp}[X_G] = \sum_i m_i$.

On the other hand, if $q > 1$, then in reality, less resource is available. This will require us to decrease each τ_i, which in turn will decrease the value of $\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}$. However, this will will make $\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}$ smaller than m_i. This gives that $\mathrm{Exp}[X_G] = \sum_i \left[\tau_i m_i + \sqrt{\tau_i(1-\tau_i)v_i}\right]$.

From Theorem 4.1, Lemmas 4.1, 4.2, 4.3, and 4.4 and the above analysis for parallel graphs, it follows that $\mathrm{Min}\left[\sum_i m_i,\ \mathrm{Max}_i m_i + \sqrt{\sum_i v_i}\right]$ approximates $\mathrm{Max}_{\vec{x_i}}\mathrm{Exp}[X_G]$ within a factor of four. ■

THEOREM 6.3. *If G is a* series-parallel *graph, then one can apply the rules for maximum variance in Theorems 6.1 and 6.2 to recursively to obtain $\mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_G]$.*

Proof. We will prove this by induction on the depth of recursion of the series-parallel graph G. The sequential and parallel bounds, $\left[\sum_i \sqrt{v_i}\right]^2$ and $\sum_i v_i$ respectively, clearly equal v_i for a single edge. Now let us assume that the theorem holds for $d-1$ levels of recursion, and consider a graph G with d levels of recursion. Suppose that G consists of a number of sub-DAGs $G_1, G_2, \ldots, G_q$. For each sub-DAG G_j, let $v_j = \mathrm{Max}_{\vec{x_i}}\mathrm{Var}[X_{G_j}]$. By the induction hypothesis, these are equal to the values obtained by applying the sequential and parallel rules recursively. We now consider two cases depending on whether G is formed by combining these sub-DAGs sequentially or in parallel.

Sequential: Suppose G is formed by linking the sub-DAGs $G_1, G_2, \ldots, G_q$ together sequentially so that the sink t_j of one is the source s_{j+1} of the next.

By Theorem 3.1, $\text{Max}_{\vec{x_i}}\text{Var}[X_G] = \text{Min}_{\vec{\mu_i}} \sum_{i\in G} \frac{v_i}{\mu_i}$, where $\mu_i = \lambda_b - \lambda_a$ is the length of the spring for edge i in the steady state of the spring system. For each sub-DAG G_j, all its nodes must lie between its source node s_j and its sink node t_j. Let $\mu_j = \lambda_{t_j} - \lambda_{s_j}$ be the lengths that G_j takes up in the spring system. Because the G_i's are linked sequentially between $\lambda_s = 0$ and $\lambda_t = 1$, we have that $\sum_j \mu_j = 1$.

Theorem 3.1 also gives us that for each sub-DAG G_j, $v_j = \text{Max}_{\vec{x_i}}\text{Var}[X_{G_j}] = \text{Min}_{\vec{\widehat{\mu_i}}} \sum_{i\in G_j} \frac{v_i}{\widehat{\mu_i}}$ where $\widehat{\mu}_i = \widehat{\lambda}_b - \widehat{\lambda}_a$ are the lengths of the springs for edge i in the steady state of the spring system. These nodes are spread between $\widehat{\lambda}_{s_j} = 0$ and $\widehat{\lambda}_{t_j} = 1$. If, however, this system was compressed so that $\widehat{\lambda}_{t_j} - \widehat{\lambda}_{s_j}$ was no longer 1 but equal to the μ_j defined for the whole graph G, then the length of each edge of G_j would be compressed from $\widehat{\mu}_i$ to $\mu_j \cdot \widehat{\mu}_i = \mu_i$. Hence, $\sum_{i\in G_j} \frac{v_i}{\widehat{\mu}_i}$ changes from $v_j = \sum_{i\in G_j} \frac{v_i}{\widehat{\mu}_i}$ to $\frac{v_j}{\mu_j} = \sum_{i\in G_j} \frac{v_i}{\mu_j \cdot \widehat{\mu}_i} = \sum_{i\in G_j} \frac{v_i}{\mu_i}$. We can conclude that $\text{Max}_{\vec{x_i}}\text{Var}[X_G] = \text{Min}_{\vec{\mu_i}} \sum_{i\in G} \frac{v_i}{\mu_i} = \text{Min}_{\vec{\mu_i}} \sum_j \sum_{i\in G_j} \frac{v_i}{\mu_i} = \text{Min}_{\vec{\mu_j}} \sum_j \frac{v_j}{\mu_j}$. As we have seen, we have the single constraint on $\vec{\mu_j}$ that $\sum_j \mu_j = 1$. The proof that $\text{Min}_{\vec{\mu_j}} \sum_j \frac{v_j}{\mu_j} = \left(\sum_j \sqrt{v_j}\right)^2$ is identical to that in Theorem 6.1.

Parallel: For the second case, suppose that G is formed by putting the sub-DAGs in parallel with a common source node s and a common sink node t. Because each sub-DAG G_j has $\lambda_{s_j} = 0$ and $\lambda_{t_j} = 1$ in the spring system, it follows that $\mu_j = 1$ and does not need to be compressed. It follows that $\text{Max}_{\vec{x_i}}\text{Var}[X_G] = \text{Min}_{\vec{\mu_i}} \sum_{i\in G} \frac{v_i}{\mu_i} = \text{Min}_{\vec{\mu_i}} \sum_j \sum_{i\in G_j} \frac{v_i}{\mu_i} = \text{Min}_{\vec{\mu_j}} \sum_j v_j = \sum_j v_j$. ■

THEOREM 6.4. *There is a* series-parallel *graph G and values m_i and v_i of expectations and variances of the underlying edge delay variables, such that $\text{Max}_{\vec{x_i}}\text{Exp}[X_G]$ is much less than that obtained by applying the series and parallel rules in Theorems 6.1 and 6.2 recursively.*

Proof. Consider the following counter example. Let G have two disjoint and similar *st* paths, each containing $n+1$ edges. For $i \in [0,1]$, the delay variable of the first edge in the i^{th} path will be denoted $x_{\langle i,0\rangle}$. We let its mean be $m_{\langle i,0\rangle} = m$ and its variance be $v_{\langle i,0\rangle} = 0$. For $j \in [1..n]$, the j^{th} edge in the i^{th} path will be denoted $x_{\langle i,j\rangle}$. We let its mean be $m_{\langle i,j\rangle} = \epsilon$ and its variance be $v_{\langle i,0\rangle} = v$.

Note that when computing $\text{Min}\left[\tau_{\langle i,j\rangle} m_{\langle i,j\rangle} + \sqrt{\tau_{\langle i,j\rangle}(1-\tau_{\langle i,j\rangle})v_{\langle i,j\rangle}},\ m_{\langle i,j\rangle}\right]$, $\tau_{\langle i,j\rangle} m_{\langle i,j\rangle}$ will be the minimum for the two first edges because $v_{\langle i,0\rangle} = 0$, and $m_{\langle i,j\rangle}$ will be the minimum for the rest of the edges because $m_{\langle i,j\rangle} = \epsilon$. Hence we can easily use Theorems 3.1 and 4.1 to bound the expectation and variance of X_G.

$$\begin{aligned}
&\text{Max}_{\vec{x_{\langle i,j\rangle}}}\text{Exp}[X_G] \\
&= \text{Max}_{\vec{\tau_{\langle i,j\rangle}}} \textstyle\sum_{i\in[0,1]} \sum_{j\in[0,n]} \text{Min}\left[H_{\langle i,j\rangle},\ m_{\langle i,j\rangle}\right], \\
&\quad \text{where } H_{\langle i,j\rangle} = \tau_{\langle i,j\rangle} m_{\langle i,j\rangle} + \sqrt{\tau_{\langle i,j\rangle}(1-\tau_{\langle i,j\rangle})v_{\langle i,j\rangle}} \\
&= \tau_{\langle 0,0\rangle} m_{\langle 0,0\rangle} + (1-\tau_{\langle 0,0\rangle}) m_{\langle 1,0\rangle} \\
&\quad + \textstyle\sum_{i\in[0,1]} \sum_{j\in[1,n]} m_{\langle i,j\rangle} \\
&= m + 2n\epsilon
\end{aligned}$$

$$\begin{aligned}
\text{Max}_{\vec{x_i}}\text{Var}[X_G] &= \text{Min}_{\vec{\mu_i}} \sum_{i\in[0,1]} \sum_{j\in[0,n]} \frac{v_{\langle i,j\rangle}}{\mu_{\langle i,j\rangle}} \\
&= \sum_{i\in[0,1]} \sum_{j\in[1,n]} \frac{v}{1/n} \\
&= 2n \cdot nv \ =\ 2n^2 v
\end{aligned}$$

Note that $\mu_{\langle i,0\rangle}$ might as well be (almost) zero because $v_{\langle i,0\rangle} = 0$. Since for each path $i \in [0,1]$ we need $\sum_{j\in[0,n]} \mu_{\langle i,j\rangle} = 1$, by symmetry the other edges have $\mu_{\langle i,j\rangle} = \frac{1}{n}$.

Now let us bound $\text{Exp}[X_G]$ by recursively applying the bounds within the series-parallel structure of G. For $i \in [0,1]$, let G_i be the i^{th} path of G and let $X_i = \sum_{j\in[0,n]} x_{\langle i,j\rangle}$ be the variable associated with this path. We can easily use Theorem 6.1 to bound the expectation and variance of X_i.

$$\text{Max}_{\vec{x_{\langle i,j\rangle}}}\text{Exp}[X_i] = \sum_{j\in[0,n]} m_{\langle i,j\rangle} = m + n\epsilon$$

$$\begin{aligned}
\text{Max}_{\vec{x_{\langle i,j\rangle}}}\text{Var}[X_i] &= \left[\sum_{j\in[0,n]} \sqrt{v_{\langle i,j\rangle}}\right]^2 = \left[0 + n\sqrt{v}\right]^2 \\
&= n^2 v
\end{aligned}$$

Having the variance and expectation of each X_i, we can now use these within Theorem 6.2 to compute the expectation and variance of $G = \text{Max}_{i\in[0,1]} X_i$.

$$\begin{aligned}
&\text{Max}_{\vec{x_{\langle i,j\rangle}}}\text{Exp}[X_G] \\
&= \text{Min}\left[\textstyle\sum_{i\in[0,1]} \text{Exp}[X_i], J\right], \text{ where} \\
&\quad J = \text{Max}_{i\in[0,1]}\text{Exp}[X_i] + \sqrt{\textstyle\sum_{i\in[0,1]} \text{Var}[X_i]} \\
&= \text{Min}\left[2\cdot(m+n\epsilon),\ (m+n\epsilon) + \sqrt{2\cdot(n^2 v)}\right] \\
&= m + n\epsilon + n.\sqrt{2v} \text{ or } 2\cdot(m+n\epsilon) \\
&\gg m + 2n\epsilon \text{ in general.}
\end{aligned}$$

$$\text{Max}_{\vec{x_i}}\text{Var}[X_G] = \sum_{i\in[0,1]} \text{Var}[X_i] = 2n^2 v$$

Note that if v and m are sufficiently large, the maximum expectation $m + 2n\epsilon$ of X_G is much less than both $m+n\epsilon+n\sqrt{2v}$ and $2\cdot(m+n\epsilon)$ computed recursively. This proves that the recursive bound, though sound, may be very conservative.

In contrast, the maximum variance $2n^2v$ of X_G is exactly equal to the amount computed recursively. ■

7 Conclusion

In this paper, we presented tight upper bounds of the mean and variance of the longest path length in a single source, single sink DAG with non-negative edge weights. We discussed a new algorithm inspired by balance of forces in a system of strange springs to compute the maximum variance and the maximum mean of the longest path. We also presented cake distributions, and showed their importance in achieving these upper bounds. We also presented closed-form bounds for an important class of graphs called series-parallel graphs. Unfortunately, our lower bound analysis is conservative, and does not provide much insight into the nature of distributions that can achieve such lower bounds. As part of future work, we intend to work on these lower bounds, and also on the more general problem of bounding the mean and variance of the time separation of two arbitrary events (not necessarily source and sink) in a precedence constraint graph.

References

[1] P. Buchholz and P. Kemper. Numerical analysis of stochastic marked graph nets. In *PNPM '95: Proceedings of the Sixth International Workshop on Petri Nets and Performance Models*, page 32, Washington, DC, USA, 1995. IEEE Computer Society.

[2] S. Chakraborty and R. Angrish. Probabilistic timing analysis of asynchronous systems with moments of delays. In *International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 99–108. IEEE Computer Society, 2002.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (2nd edition)*. The MIT Press, 2001.

[4] H. A. David and H. N. Nagaraja. *Order Statistics (3rd edition)*. Wiley, New Jersey, 2003.

[5] A. Dennis, B. H. Wixom, and M. Roth. *System Analysis and Design (3rd edition)*. Wiley-India, 2006.

[6] W. Feller. *An Introduction to Probability: Theory and Its Applications, Vol. 1 (3rd edition)*. Wiley India, 2008.

[7] Y.-K. Kwok and I. Ahmad. Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, 7(5):506–521, 1996.

[8] P.I. Richards. Precedence constraints and arrow diagrams. *SIAM Review*, 9(3):548–553, 1967.

[9] R. Sedgewick. *Algorithms in C (3rd edition)*. Dorling Kindersley, 2008.

[10] M. Sereno. Approximate mean value analysis for stochastic marked graphs. *IEEE Trans. Softw. Eng.*, 22(9):654–664, 1996.

[11] A. Xie and P. A. Beerel. Performance analysis of asynchronous circuits and systems using stochastic timed Petri nets. In A. Yakovlev, L. Gomes, and L. Lavagno, editors, *Hardware Design and Petri Nets*, pages 239–268. Kluwer Academic Pub., March 2000.