# Computing bounds for delay in a stochastic network

J.-M. Fourneau[1], E. Longueville[1], Y. Ben Maissa[2], L. Echabbi[2], and H. Lotfi[2]

[1] DAVID, UVSQ, Univ. Paris-Saclay, Versailles France
Jean-Michel.Fourneau@uvsq.fr
[2] INPT, Rabat, Morocco

**Abstract.** We consider a stochastic network where the arcs are associated to discrete random variables which represent the delay. We need to compute the shortest delay (or equivalently the distance) from the source to the sink in the network. Due to the randomness, this problem is known to be hard while it has many polynomial algorithms when the arcs have deterministic lengths (or durations). We provide three approaches and algorithms to obtain stochastic bounds of the distribution of the distance. We present several examples to compare the precision and the time. The approach based on association of random variables gives very accurate results on the examples and has the smallest complexity.

## 1 Introduction

In the transportation systems in smart cities, due to the large number of sensors available, we collect a huge volume of data. The data could not be seen as deterministic anymore and we have to deal with the apparent randomness of our measures due to noise, contention, incidents. Here we propose a method to deal with this randomness for a classical problem: the computation of the distance between two nodes.

We consider a directed graph $G = (V, E)$ (digraph) which does not contain any directed cycle, such that each arc $(i, j)$ is associated with a random delay (or distance) $W(i, j)$ to join $j$ from $i$. These r.v. will be denote as $W_m$ where $m$ is the arc label. We assume that these random variables are discrete and their supports $S_m$ ($m$ is the index of the r.v.) are finite subsets of $\mathbb{R}^+$. We also assume that these random variables are independent. As the digraph does not contain any directed cycle, it is associated to a topological ordering of the nodes. The graph contains $N$ nodes and $M$ directed edges (or arcs). Among these nodes, we distinguish the first node (labelled $s$ and numbered 1). We want to compute the distance (or delay) between $s$ and any node $t$ in the graph. Let $X = (X_1, .., X_t, ., X_N)$ be the random variables associated with the distance or the delay from $s$ to all the nodes $t$. By construction, $X_s$ has a distribution with a single atom in 0 associated with a probability equal to 1.

Let $\mathcal{P}_G(s, t)$ be the set of paths $P(t)$ from $s$ to $t$ in the graph $G$. We assume without loss of generality that $\mathcal{P}_G(s, t)$ is not empty. As $G$ is a DAG, there exist

a finite number of paths from $s$ to $t$ and these paths have a finite number of edges. Let $L(P(t))$ be the delay to reach $t$ departing from $s$.

$$L(P(t)) = \sum_{(a,b)\in P(t)} W(a,b) \quad and \quad d(s,t) = Min_{P\in\mathcal{P}_G(s,t)} L(P)$$

Computing $d(s,t)$ is a difficult problem due to the randomness of $W(i,j)$ while many polynomial algorithms exist in the deterministic case. The two reasons are the size of the resulting distributions and the dependence of the path lengths when they share an arc. Indeed, even if the arcs lengths are supposed to be independent, the paths lengths are not independent. Therefore a simple computation of addition and minimum requires conditioning. Furthermore, a convolution of two distributions with size $S$ (associated with the addition of the independent r.v.) may lead to a distribution with size $S^2$. Thus each new arc added in a path may geometrically increase the number of atoms in the resulting distribution. Such problem was ignored in a recent approach [5] where distributions were modeled by polynomials: during the computation the number of monomials increase geometrically and the approach quickly becomes intractable.

A simple algorithm (with non polynomial complexity) can however be designed, using conditioning on the random variables to solve the problem for small instances when the discrete variables take values in very small sets. It is sufficient to use the Total Probability Theorem after conditioning on the states of all the random variables. Clearly

$$Pr(d(s,t)=k) = \sum_{k_1,k_2,...,k_M} 1_{D(s,t,k_1,k_2,...,k_M)=k} \prod_{m=1}^{M} Pr(W_m = k_m)$$

where $D(s,t,k_1,k_2,...,k_M)$ is the distance from $s$ to $t$ when the length of arc $m$ is $k_m$. $D(s,t,k_1,k_2,...,k_M)$ can be obtained by any deterministic algorithm (with complexity $o(M)$) to compute the distances in a directed graph. Clearly the complexity of this approach is $M \prod_{m=1}^{M} |S_m|$ (see [2] for a survey on the complexity for various delays and flow problems for networks or graphs with random discrete costs or durations).

Thus we develop several algorithms to derive stochastic bounds on the distribution of the distance. The technical part of the paper is as follows. In Section 2, we introduce strong stochastic bounds and increasing concave bounds. Based on monotonicity of $d(s,t)$, we propose two algorithms based on a reduction of the size of the supports of $W(i,j)$. This first approach based on the fact that the distance is an increasing and concave function was already used to bound max flow problem for a stochastic network [3]. Section 3 is devoted to associated random variables to present a new and more efficient approach. We prove that the distances between nodes are associated as some paths share arcs. Then we propose an algorithm to obtain lower bounds on the distance based on this property. We also develop an upper bound algorithm which relies on arc disjoint paths. To the best of our knowledge associated random variables have been considered for PERT networks but their application to the shortest distance problem is

original. These algorithms are then numerically compared on some examples in Section 4.

## 2 Stochastic bounds based on stochastic monotonicity

The complexity of the exact calculation of the distribution comes from the number of atoms. Therefore it is appealing to derive bounds when we decrease the number of atoms. In [1] we have proposed some methods which keep some quantitative and qualitative information on the results after a reduction of the number of atoms. This is obtained through the use of stochastic orderings. We begin with the definition of the orders we will use in this paper (see [6] for more information).

**Definition 1 (strong stochastic ordering).** *Let $X$ and $Y$ be two random variables, $X <_{st} Y$ if for all increasing function $\Phi$, $E[\phi(X)] \leq E[\phi(Y)]$ if the expectations exist.*

The stochastic comparison of random variables also implies that a strict inequality between their expectations as seen below.

**Property 1** *Let $X$ and $Y$ be two random variables, such that $X <_{st} Y$. If $E[X] = E[Y]$ then $X =_{st} Y$.*

We also use some orders associated with variability of the random variables to obtain tighter bounds. Let us first consider convex order which are defined as follows.

**Definition 2 (stochastic convex ordering).** *Let $X$ and $Y$ be two random variables, $X \preceq_{cx} Y$ if $E[X] = E[Y]$ and for all convex function $\phi$, $E[\phi(X)] \leq E[\phi(Y)]$ if the expectations exist.*

Here we will use the concave ordering which is easily derived from the convex ordering.

**Definition 3 (stochastic concave ordering).** *Let $X$ and $Y$ be two random variables, $X \preceq_{cv} Y$ if $Y \preceq_{cx} X$*

And finally,

**Definition 4 (increasing concave ordering).** *Let $X$ and $Y$ be two random variables, $X \preceq_{icv} Y$ if for all increasing concave function $\phi$, $E[\phi(X)] \leq E[\phi(Y)]$ if the expectations exist.*

The distance from $s$ to $t$ in the network is the define as the minimum of the path lengths from $s$ to $t$. And the length of a path is the sum of the length of the arcs inside the path. Therefore the distance is defined using operators "Min" and "+". And both operators are increasing and concave. More formally, we can define:

$$d(s,t) = f(W_1, W_2, ..., W_M),$$

and we know that $f$ is increasing and concave. We now define monotonicity for some ordering and we mention the key property for approach.

**Definition 5 ($\Psi-$Monotony).** *A function $f$ is $\Psi-$monotone if for all $X$ and $Y$ random variables such that $X \preceq_\psi Y$, then $f(X) \preceq_\psi f(Y)$.*

Due to the definitions of the orderings we considered by set of functions, the following property holds:

**Property 2** *If function $f$ is increasing, then it is $st-$monotone. Similarly, if function $f$ is increasing and concave then it is monotone for the increasing concave ordering.*

*Algorithms for st-bounds:* Computing st-bounds of $d(s,t)$ is very simple (see [3] for more details). It is sufficient to replace the distributions for the length of an arc by a "st" bound of this distribution as stated in Algo 2. If we consider a smaller support, the bound will be easier to compute. The first step consists in building "st" bounds (upper and lower for the input distributions of $W_m$. This can be done with a very simple algorithm we know describe:

**Algorithm 1** Simple "st" Bounds for input discrete distributions

1: Chose the size of the bounding distribution (say $K$) for $W_m$. Of course $K < S_m$ .
2: Divide the set $S_m$ into $K$ proper subsets $S_m^{(i)}$ ($i$ between 1 and $K$) such that all the atoms in $S_m^{(i)}$ are smaller than atoms in $S_m^{(i+1)}$. Let $l_m^{(i)}$ (resp. $u_m^{(i)}$) be the smallest (resp. largest) atom in $S_m^{(i)}$.
3: The distribution with $K$ atoms $l_m^{(i)}$ with probability $\sum_{a \in S_m^{(i)}} Pr(W_m = a)$ is a lower st-bound of $W_m$.
4: Similarly the distribution with $K$ atoms $u_m^{(i)}$ and the same probability vector is an upper st-bound of $W_m$.

In Algo. 2, we give a short presentation of the algorithm for st-lower bounds. The upper bounds are obtained by a similar arguments. We also present in Table 1 and Table 2 the results for the first example we present in section 4. Remark that upper and lower bounds share the same probability vector but they do not have the same support.

**Algorithm 2** St Bounds for the distance distributions.

1: Derive $L_m$ stochastic lower bound for $W_m$, for all $m \in [1..M]$ with the former algorithm.
2: Compute $f(L1, ..., L_M)$ by conditioning and the total probability theorem. The stochastic monotonicity implies that $f(L1, ..., L_M) <_{st} d(s,t)$.

*Algorithms for increasing concave bounds:* The approach is similar. It is sufficient to build "icv"-bounds of the input distributions. However the derivation of these

**Algorithm 3** "icv" Bounds for the distance distributions.

1: **while** The number of atoms in the bound of distribution $W_m$ is larger than the objective **do**
2: To obtain an upper bound, consider two atoms in the actual distribution, replace these atoms by a new atom which is the barycenter of the atoms.
3: To obtain a lower bound, consider a subset of at least three atoms in the actual distribution and replace them with the two extreme atoms keeping the same expectation.
4: **end while**

bounds differs significantly from the st-bound algorithm (see [3] for the details).

Again, for the sake of conciseness we just state the algorithm for "icv"-lower bounds.

**Algorithm 4** "icv" lower bounds for the distance distributions.

1: Derive $V_m$ increasing concave lower bound for $W_m$, for all $m \in [1..M]$.
2: Compute $f(V1, ..., V_M)$ by conditioning and the total probability theorem. The increasing concave monotonicity implies that $f(V1, ..., V_M) \preceq_{icv} d(s, t)$.

**Corollary 1** *Replacing all random variables $W(i, j)$ by their expectation provides an increasing concave upper bound for the distance (like in Jensen inequality). Thus, this strategy to eliminate the randomness have a systematic bias. In the following, this bound will be denoted as Fulkerson bound.*

**Property 3 (Complexity)** *Let $M$ be the number of arcs in the graph. Assuming that all input distributions have size $|S|$, that we compress the distributions to obtain input bounds with size $K$, then the algorithm needs $O(M\ |S|)$ operations to derive the input bounds and $O(M\ 2^K)$ to get the bounds on the distance.*

| Arcs | Atoms | Probability vector |
|---|---|---|
| e0,e2,e4 | { 2 5 8 10 12 15 18 20} | { 0.1 0.1 0.1 0.2 0.1 0.1 0.1 0.2 } |
| e1,e11 | {5 10 15 20 25 30 35 40} | { 0.1 0.2 0.1 0.2 0.1 0.1 0.1 0.1 } |
| e3 | { 1 2 3 4 7 8 9 12 } | {0.2 0.1 0.2 0.1 0.1 0.1 0.1 0.1 } |
| e5,e10 | { 2 3 7 8 10 11 15 16 } | {0.1 0.1 0.2 0.1 0.1 0.2 0.1 0.1 } |
| e6 | { 1 2 3 4 8 9 10 13 } | {0.1 0.2 0.1 0.1 0.2 0.1 0.1 0.1 } |
| e7 | {1 2 3 4 6 9 11 16 } | {0.1 0.2 0.1 0.1 0.1 0.1 0.1 0.2 } |
| e8,e9 | {3 10 15 20 22 25 30 35 } | {0.1 0.3 0.1 0.1 0.1 0.1 0.1 0.1 } |

**Table 1.** Input Distributions, model 1.

| Arc | Atoms | | Probability | | Atoms | | Atoms | | Probability | | Atoms | | Probability | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arc | "st" | lower | | | "st" | upper | "icv" | lower | | | "icv" | upper | | |
| e0,e2,e4 | 2 | 12 | 0.5 | 0.5 | 10 | 20 | 2 | 20 | 0.444 | 0.556 | 10 | 15 | 0.6 | 0.4 |
| e1,e11 | 5 | 25 | 0.6 | 0.4 | 20 | 40 | 5 | 40 | 0.543 | 0.457 | 15 | 30 | 0.6 | 0.4 |
| e3 | 1 | 7 | 0.6 | 0.4 | 4 | 12 | 1 | 12 | 0.636 | 0.364 | 2 | 7 | 0.4 | 0.6 |
| e5,e10 | 2 | 10 | 0.5 | 0.5 | 8 | 16 | 2 | 16 | 0.5 | 0.5 | 8 | 13 | 0.8 | 0.2 |
| e6 | 1 | 8 | 0.5 | 0.5 | 4 | 13 | 1 | 13 | 0.583 | 0.417 | 6 | 11 | 0.8 | 0.2 |
| e7 | 1 | 6 | 0.5 | 0.5 | 4 | 16 | 1 | 16 | 0.6 | 0.4 | 3 | 13 | 0.6 | 0.4 |
| e8,e9 | 3 | 22 | 0.6 | 0.4 | 20 | 35 | 3 | 35 | 0.531 | 0.469 | 15 | 30 | 0.8 | 0.2 |

**Table 2.** Stochastic Bounds with two atoms.

Note that both approaches simplify the computation to obtain bounds and that we obtain distinct bounds when we change the partition (finding the most accurate partition (and bound) is still an open problem). However the complexity remains exponential if we consider distributions which have more than one atom. The theory of associated random vectors will be used to derive less complex bounds. In many cases they are also much more accurate (at least for the lower bound).

## 3 Bounds based on associated random vectors

For a more detailed presentation of associated random variables, see [4].

**Definition 1** *The random variables $X_1, .., X_n$ are associated if, given two coordinatewise nondecreasing functions $f$ and $g$: $\mathbb{R}^n \to \mathbb{R}$,*

$$Cov(f(X_1, .., X_n), g(X_1, .., X_n)) \geq 0.$$

**Remark 1** *One can also consider non increasing functions, as $Cov(f(X), g(X)) = Cov(-f(X), -g(X))$ and $f$ is non increasing implies that $-f$ is non decreasing.*

**Remark 2** *As $Cov(f(X), g(X)) = E[f(X)g(X)] - E[f(X)]E[g(Y)]$, one can also define associated random vector as $E[f(X)g(X)] \geq E[f(X)][g(X)]$.*

As a large of the theory came from reliability, one have more results when the r.v. are Boolean.

**Property 4 (Barlow et Proschan)** *Let $X_1, ..., X_n$ be $n$ Boolean r.v., then*

$$Pr[(\prod_i X_i) = 1] \geq \prod_i Pr[X_i = 1]$$

*and*

$$Pr[(\max_i X_i) = 1] \geq \max_i Pr[X_i = 1]$$

Taking into account that the r.v. are boolean, we have: $(\max_i X_i) = 1 - \prod_i (1 - X_i)$. This elementary property is used to derive useful inequalities.

**Property 5** *Let $X$ a random vectors with $n$ associated random variables. We have:*

$$Pr(X_1 > x_1, X_2 > x_2, .., X_n > x_n)) \geq \prod_k Pr(X_k > x_k),$$

*and*

$$Pr(X_1 \leq x_1, X_2 \leq x_2, .., X_n \leq x_n)) \geq \prod_k Pr(X_k \leq x_k).$$

Proof: we define $T_i(t) = 1_{X_i > t}$ for all $i$. $T_i$ is increasing in $X_i$ and is a boolean. The random vector $(X_i)$ is associated. Thus random vector $(T_i)$ is also associated. We apply both inequalities of Prop 4 on $T_i$ to prove inequalities on $X$.

**Corollary 2** *Thus, $Pr(min_i(X_i) > x) \geq \prod_k Pr(X_i > x)$.*

Proof: it is sufficient to remark that $Pr(min_i(X_i) > x) = Pr(X_1 > x, X_2 > x, .., X_n > x))$.

To the best of our knowledge, proving an algorithm to check that discrete random variables are associated is still an open problem. Therefore association is proved using the following properties (see the next section on distance in a stochastic network). Starting with a given set of associated random variables (independence is useful here), it is rather simple to obtain new families of associated random variables with increasing transformation.

**Property 6** *Every random variable $X$ is associated with itself.*

**Property 7** *If $X = (X_1, X_2, ..., X_N)$ is a random vector such that all the $X_i$ are independent random variables, then $X$ is associated.*

**Property 8** *Every subset of an associated random vector is associated.*

**Property 9** *Let $X_1$,$X_2$, ... $X_n$ be independent of the variables $Y_1$, $Y_2$, ..., $Y_m$. Assume that $X_1$,$X_2$, ... $X_n$ are associated random variables. Assume also that $Y_1$, $Y_2$, ..., $Y_m$ are associated random variables. Then $X_1$,$X_2$, ... $X_n$, $Y_1$, $Y_2$, ..., $Y_m$ are associated random variables.*

**Property 10** *Let $X_1$,$X_2$, ... $X_n$ be $n$ associated random variables, and consider coordinatewise nondecreasing functions $f_1, .., f_K$: $\mathbb{R}^n \to \mathbb{R}$. Then random variables $Y_1 = f_1(X_1, X_2, ...X_n)$,.., $f_K(X_1, X_2, ...X_n)$, are associated.*

### 3.1 Links with $<_{st}$ ordering and independence

**Notation 1** *In the following, let us denote by $=_d$ the equality of the distributions. Let $X = (X_1, .., X_n)$ an associated random vector. For all $i$ between 1 and $n$, let us denote by $\overline{X_i}$ a random variable such that $\overline{X_i} =_d X_i$ while $\overline{X_i}$ and $\overline{X_j}$ are independent for all $j \neq i$. $\overline{X_i}$ will be denoted as the independent version of $X_i$.*

We begin with a well-known property which was used in [7, 8] to obtain bounds for PERT networks. As we do not use "Max" operator in this paper we do not give the proof of this property.

**Property 11** *Let $X = (X_1, .., X_n)$ an associated random vector. Then,*

$$Pr(max_k(X_k)) \geq Pr(max_k(\overline{X_k})).$$

*Equivalently $max_k(X_k) <_{st} max_k(\overline{X_k})$. The independent versions of the random variables provide a strong stochastic upper bound of the max (a guarantee).*

Here, the operator we use for the path length is the "Min" operator. Thus we derive a similar result for the distance.

**Property 12** *Let $X = (X_1, .., X_n)$ an associated random vector. Then, $min_k(\bar{X_k}) <_{st} min_k(X_k)$. The independent versions of the random variables provide a strong stochastic lower bound of the min.*

Proof : we consider the first relation of Prop 5, taking for all $i$ $x_i = x$.

$$Pr(X_1 > x, X_2 > x, .., X_n > x)) \geq \prod_k Pr(X_k > x).$$

The left part of the inequality is $Pr(min_k(X_k) > x))$ while the right part is:

$$\prod_k Pr(X_k > x) = \prod_k Pr(\overline{X_k} > x) = Pr(\overline{X_1} > x, \overline{X_2} > x, .., \overline{X_n} > x) = Pr(min_k(\overline{X_k}) > x).$$

Thus,

$$Pr(min_k(X_k) > x)) \geq Pr(min_k(\overline{X_k}) > x).$$

or equivalently

$$min_k(\overline{X_k}) <_{st} min_k(X_k).$$

### 3.2 Distance and Association

We now prove that in a stochastic network with independent random variables for the length, the distances between nodes are associated random variables. We begin with a technical lemma which also provides an intuition about associated random variables.

**Lemma 1.** *Let $Y$, $Z1$ and $Z2$ three independent r.v., then $Y + Z1$ and $Y + Z2$ are associated.*

Proof : $Y$, $Z1$ and $Z2$ are independents. Thus $(Y, Z1, Z2)$ is an associated random vector. We consider functions $f1$, $f2$ and $f3$: $\mathbb{R}^3 \to \mathbb{R}$ defined by:

$$f1(Y, Z1, Z2) = Y, \quad f2(Y, Z1, Z2) = Y + Z1, \quad f3(Y, Z1, Z2) = Y + Z2.$$

Clearly these three functions are increasing. Thus according to Prop. 10, random vector $(f1(Y, Z1, Z2), f2(Y, Z1, Z2), f3(Y, Z1, Z2)) = (Y, Y + Z1, Y + Z2)$ is associated. Due to Prop. 8, $(Y + Z1, Y + Z2)$ is also associated.

**Property 13** *Let $Y_P = L(P(t))$ a r.v. equal to the length of path $P(t)$ from $s$ to $t$ in digraph $G$, $Y = (Y_P)_{P \in \mathcal{P}_G(s,t)}$ is an associated random vector.*

Proof : Let $P1$ et $P2$ two paths from $s$ to $t$. We have two cases to consider:

1. $P1$ and $P2$ are arc-disjoint.
2. The intersection of $P1$ and $P2$ contains some arcs.

In the first case, $L(P1)$ and $L(P2)$ are independent as they are summations of distinct independent random variables (the length of the arcs which belong to the path). As they are independent, they are also associated according to Prop. 7.

Now assume that the intersection of $P1$ and $P2$ contains some arcs and let $Q = P1 \cap P2$. As the "+" operator in the definition of the path length is commutative, one can separate each path into two subsets of arcs such that

$$L(P1) = L(Q) + L(P1 \setminus Q) \quad and \quad L(P2) = L(Q) + L(P2 \setminus Q)$$

As $Q = P1 \cap P2$, we have $(P1 \setminus Q) \cap (P2 \setminus Q) = \emptyset$, thus $L(P1 \setminus Q)$ and $L(P2 \setminus Q)$ are independent random variables. Similarly $L(Q)$ is independent of these two r.v. and we can apply Lemma 1 to prove that $L(P1)$ and $L(P2)$ are associated.

### 3.3 Algorithm for a lower bound

Let $d^i(j)$ represent the distance from node 1 to node $j$ when all the nodes between 1 and $i$ have been taken into account to find paths. In general we cannot compute the distribution of $d^i(j)$ because of Prop. 13 without conditioning and using an exponential complexity algorithm. Instead we compute a stochastic bound $l^i(j)$. The algorithm proceeds by iteration adding one node at each iteration. Each node allows to add new paths and to decrease the shortest distance computed so far. Remember that the directed graph is a DAG and we use this property to consider the nodes in the topological order associated to the DAG. Note that $d(s,t) = d^N(t)$ (i.e. we have considered all the nodes to build paths from 1 to $t$).

**Algorithm 5** St lower bounds based on association of paths.

1: Init $l^1(u) = W(1,u)$ for all nodes $u$ which are neighbor of node $s = 1$.
2: **for all** $k$ (node number) from 2 to $N$ **do**
3:  **for all** all node $u$ neighbor of node $k-1$ **do**
4:  let $l^k(u) = min\left(\overline{l^{k-1}(u)}, \overline{l^{k-1}(k-1) + W(k-1,u)}\right)$
5:  **end for**
6: **end for**

By definition $\overline{l^{k-1}(u)}$ as the same distribution and is independent. Furthermore we know how to numerically compute the minimum of two independent distributions.

**Theorem 1** *$l^k(u)$ is a "st" lower bound for the distance: for all node index $k$ and $u$, $l^k(u) <_{st} d^k(u)$. And $l^N(t) <_{st} d(s,t)$.*

Proof : by induction on $k$.

- $k = 1$. We consider the neighbors (say $u$) of node 1. The distance to reach node $u$ is exactly the length of the arcs $W(1,u)$. Therefore $d^1(u) = W(1,u)$ and thus $l^1(u) <_{st} d^1(u)$ for all these nodes $u$.
- $k \Rightarrow k+1$. Adding new arc from $k$ to $u$ allows to decrease the distance (see Fig. 1). By construction we have:

$$d^k(u) = min\left(d^{k-1}(u), d^{k-1}(k-1) + W(k-1,u)\right).$$

By Prop. 13, we know that the path lengths are associated. Thus for all $u$,

$$min\left(\overline{d^{k-1}(u)}, \overline{d^{k-1}(k-1) + W(k-1,u)}\right) <_{st} d^k(u)$$

As "min" and "+" are increasing functions, if $x <_{st} y$ and $z$ independent of $y$ and $x$ then $min(z,x) <_{st} min(z,y)$ and $x + z <_{st} y + z$. By induction, we have : $l^{k-1}(k-1) <_{st} d^{k-1}(k-1)$. Therefore:

$$min\left(\overline{l^{k-1}(u)}, \overline{l^{k-1}(k-1) + W(k-1,u)}\right) <_{st} min\left(\overline{d^{k-1}(u)}, \overline{d^{k-1}(k-1) + W(k-1,u)}\right).$$

By transitivity: $min\left(\overline{l^{k-1}(u)}, \overline{l^{k-1}(k-1) + W(k-1,u)}\right) <_{st} d^k(u)$, and finally, $l^k(u) <_{st} d^k(u)$.

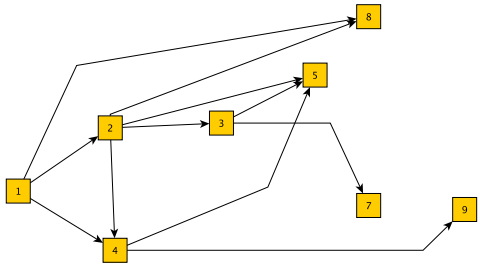


**Fig. 1.** Considering node 4 allows to modify the distance distribution to node 5 and to initialize the distance distribution to node 9.

The two basic operations in Algo. 5 is the addition and the minimum of two independent random variables. These operations obviously depend on the size of the discrete distributions. Note that each convolution operation leads to an increase of the size of the distributions and this size has to be controlled to avoid that the last steps of the algorithm deals with very large distributions.

**Corollary 3** *In Algo. 5, after each computation of $l^k(u)$, we replace $l^k(u)$ by a "st" lower bound of $l^k(u)$ with at most $K$ atoms. The algorithm still compute a "st" lower bound of $d(s,t)$ by transitivity.*

Assume that the distributions are represented by sorted lists with size $d1$ and $d2$ and let $d = max(d1, d2)$. The minimum can be computed in $O(d)$ operations and the output distribution has a size smaller than $d$. The addition of independent random variables is associated with the convolution of their distributions. Efficient algorithms exist with a complexity in $O(d\ log(d))$ and the output have a size which is at most $d^2$.

**Property 14 (Complexity of Algo 5)** *Assume that each step of the algorithm, the distributions have size $K$. Each step requires $O(K\ logK)$ operations and after calculation, $l^k(u)$ have size at most $K^2$ (due to the convolution). The extra step of lower bounds introduced in Cor. 3 has a linear complexity (in $K^2$). And the number of iterations in the nested loops of the algorithm is $M$. Therefore the complexity is $O(M\ K^2)$.*

### 3.4 Algorithms for an upper bound

We combine two arguments:

- When two paths are arc disjoint, the lengths of the paths are independent (see the first part of the proof of Prop. 13) and we know how to compute them.
- If we only consider a subset of paths, we compute a st-upper bound. More formally, if $\mathcal{A}(s,t) \subset \mathcal{P}(s,t)$, then

$$d(s,t) = Min_{P \in \mathcal{P}(s,t)} L(P) <_{st} m(s,t) = Min_{P \in \mathcal{A}(s,t)} L(P).$$

Thus, the algorithm consists in finding arc-disjoint paths (the largest set) and compute the distance associated with this set of paths.

**Algorithm 6** St Upper Bounds based on subgraphs and independence of paths.

1: Find the largest set of arc-disjoint path. This is easily done with a max flow algorithm on the original graph where all the arcs receive capacity 1. The augmenting paths provided by a max flow algorithm, on such a graph with arc capacity equal to 1, are by construction arc disjoint. Thus the max-flow algorithm return the maximal number of arc disjoint paths.
2: Compute the distribution of length for each path. The length of a path is the sum of arc length which are by assumption independent. Thus one can use convolution algorithm to obtain the length of any path.
3: Compute the distribution of the "Min" of these random variables. As the paths are arc-disjoint, the r.v. are independent and the computation of the "Min" is an easy task.

**Property 15 (Complexity)** *First step requires $O(F\ M)$ operations to obtain a $F$ arc disjoint paths and $F < M$. During step 2, the distribution of the distance*

*is obtained by the convolution of independent random variables of size smaller than $K$. Each convolution is followed by a compression of the resulting distribution (whose size is at most $K^2$). Let $D$ of the graph, the complexity of this step is $O(F\ D\ K^2)$ using the same arguments as in Prop. 14. The last step consists in computing the minimum of $F$ distributions with size $K$ and this can be done with $O(F\ K\ ln(K))$ operations. Therefore, the total complexity is $O(F\ D\ K^2)$.*

## 4 Numerical results

We begin with a toy example where it is feasible to obtain the exact solution to check the accuracy of the methods. Even if it is possible, the exact algorithm is really time consuming. The graph has 8 nodes and 12 edges (see Fig. 2). All the distributions have 8 atoms. The input distributions are given in table 1. The "st" bounds and "icv" bounds with two atoms for these distributions are given in Table 2. These distributions with 4 atoms will be given in an appendix of the full paper.

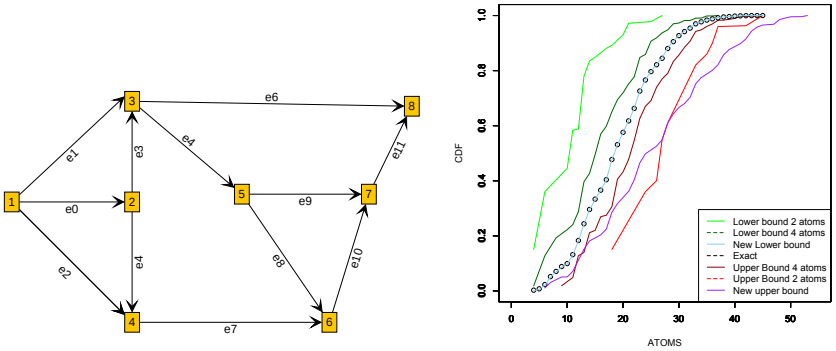


**Fig. 2.** Graph for Model 1 (left), Stochastic bounds (right).

We present in the right part of Fig. 2 the cdf of the distributions. We only present the exact result, the "st" bounds based on association and the "st" upper and lower bounds with 2 and 4 atoms in the input bounds. We do not draw the "icv" bounds to avoid confusion. Note that as the "st" bounds are not all based on the same strategy, the cdf may cross (for instance the purple and the red curve). The most important point to remark here is the tightness of the lower bound based on association.

To compare all these strategies, we compute the expectation of the distance (in Table. 3). The algorithms are very fast for this small graph: computing the distribution with the algorithms based on association need 0.01s on an ordinary laptop, while the bounds based on bounding input distributions atoms need 0.02s (resp. 14s) for bounds with 2 (resp. 4) atoms. Note that computing "st"

| Method | | Association | St Monotonicity | | "icv" Monotonicity | | Fulkerson | Exact |
|---|---|---|---|---|---|---|---|---|
| | | | 2 atoms | 4 atoms | 2 atoms | 4 atoms | | |
| Exp. | Lower bound | 19.35 | 10.88 | 16.19 | 15.02 | 18.80 | . | 19.385 |
| | Upper bound | 26 | 28.29 | 21.99 | 21.23 | 20.83 | 23 | . |

**Table 3.** Expectation of the distributions (Model1).

bounds or "icv" bounds requires the same time as we deal with distributions with the same number of atoms. Finally, the exact results are obtained after $17h$ and 26min on the same laptop.

We now study two examples: a larger graph with 26 arcs and 14 nodes (Fig. 3). As the distributions all have 8 atoms, the number of deterministic cases we have to solve with the approach based on conditioning is $8^{26}$ and this precludes to give the exact solution.

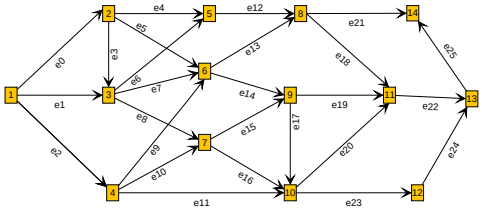


**Fig. 3.** Graph for Model 2.

For the first method which bounds the input distributions, we consider two strategies. In the first one, we only keep 2 atoms per input distributions. Thus the number of deterministic cases generated by the conditioning is $2^{26}$. To keep the number of cases smaller than $2^{30}$, the second strategy consists in bounding all the input distributions by distributions with 2 atoms, except distributions for arcs $e22$, $e23$, $e24$ and 25 which have 4 atoms. The first strategy needs 3min while the second needs 48min. Stochastic bounds based on association are very easily solved. They require less than 1s.

Remark in Table 4 that the best lower bound is provided by the association algorithm. But the best upper bound is obtained by the first approach with $2^{30}$ deterministic cases. Clearly we have to improve the strategy based on arc-disjoint paths. This point will be developed in the full version of the paper.

| Method | | Association | Input Bounds: "st" | | Input Bounds: "icv" | |
|---|---|---|---|---|---|---|
| | | | $2^{26}$ | $2^{30}$ | $2^{26}$ | $2^{30}$ |
| Expectation | lower bound | 111.660 | 76.868 | 78.596 | 92.652 | 94.198 |
| | upper bound | 183.596 | 160.149 | 153.877 | 132.016 | 131.109 |

**Table 4.** Bounds of the expectation for Model2.

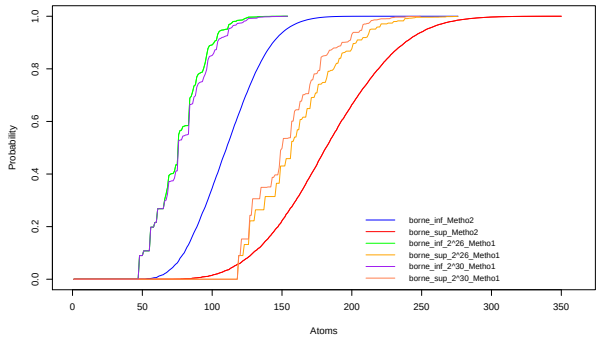


**Fig. 4.** Probability distribution for the bounds of the distance (Model 2).

Now we consider a Series-Parallel (SP for short) digraph (see Fig. 5) with 48 nodes and 54 arcs. Such a graph has a recursive construction and this provides a recursive algorithm to compute the distribution of the distance. Note however that the distributions which appear during the execution of the recursive algorithm still suffers of the size explosion problem due to the convolution operation and we have to bound them if we want to keep their size smaller than $K$. The arcs all have the same distribution of delay with 8 atoms $(1, 2, 5, 6, 12, 19, 28, 34)$ and probability vector $(0.1, 0.1, 0.05, 0.3, 0.05, 0.1, 0.2, 0.1)$. As the number of arcs is too large, we only report the results based on association. The exact algorithm based on the SP structure only requires 66s, while the bounding algorithms are faster (1.7s for the upper bound and 3.8s for the lower bound). The exact expectation is 191.66 while the lower (resp. upper) bounding distribution has an expectation equal to 174.15 (resp. 234.38). This last example (and many others we cannot report here due to the lack of space) shows that the lower bound is still quite accurate. The quality of the upper bound may depend of other graph properties and may be very bad.

## 5 Concluding Remarks

These two approaches provides some tradeoff between complexity and accuracy of the numerical results. They also suggest new approaches combining association of random variables and graph properties that we will present in the extended version of this paper. We have now derived new algorithms based on recursive construction of the graph or a subgraph to combine the "st" bounds approach with the associated r.v. results. We have also generalized some of our results to networks where the delays to cross the arcs are not independent. This allows to study new models of traffic.

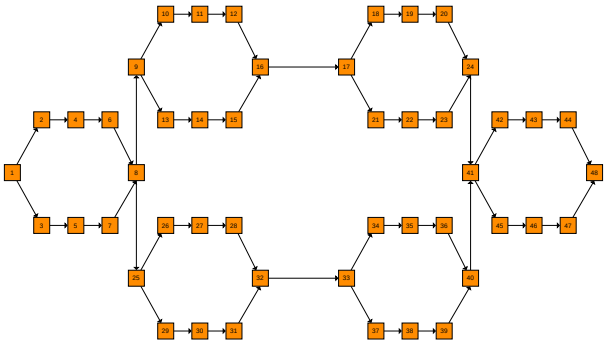

**Fig. 5.** SP-Graph for Model 3.

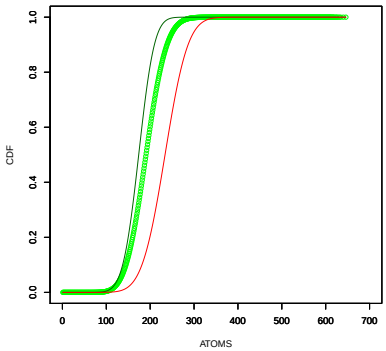


**Fig. 6.** Probability distribution for the bounds of the distance (Model 3).

## References


1. At Salaht, F., Cohen, J., Castel Taleb, H., Fourneau, J.M., Pekergin, N.: Accuracy vs. complexity: the stochastic bound approach. In: 11th International Workshop on Disrete Event Systems (WODES2012). No. 8 (2012)
2. Ball, M., Colbourn, C., Provan, J.: Network reliability. Handbooks in Operation Research and Management Science **7**, 673–762 (1995)
3. Echabbi, L., Fourneau, J., Gacem, O., Lotfi, H., Pekergin, N.: Stochastic bounds for the max flow in a network with discrete random capacities. Electronic Notes in Theoretical Computer Science **353**, 77–105 (2020), international Workshop on the Practical Application of Stochastic Modelling (PASM)
4. Esary, J.D., Proschan, F., Walkup, D.W.: Association of random variables, with applications. Ann. Math. Statist. **38**(5), 1466–1474 (10 1967)
5. Hastings, K.: Algebraic Approaches to Stochastic Optimization. https://tigerprints.clemson.edu/all_dissertations/935, PHD Clemson Univ. (2012)

6. Shaked, M., Shantikumar, J.G.: Stochastic Orders and their Applications. Academic Press, San Diego, CA (1994)
7. Shogan, A.W.: Bounding distributions for a stochastic pert network. Networks **7**, 359–381 (1977)
8. Yazici-Pekergin, N., Vincent, J.M.: Stochastic bounds on execution times of parallel programs. IEEE Trans. Software Eng. **17**(10), 1005–1012 (1991)

## Acknowledgments

H. Lotfi was financially partially supported by CAMPUS FRANCE (PHC TOUBKAL 2019 (French-Morocco bilateral program) Grant Number: 41562UA. This work was partially supported by the Paris Ile-de-France Region through a DIM RFSI grant (EPINE Project).