Probabilistic Approach to Trust: Ideas, Algorithms, and Simulations

Pattama Jaksurat
Department of Computer Science
Chiang Mai University
Chiang Mai 50200, Thailand
Email: pattama@chiangmai.ac.th

Eric Freundenthal,
Martine Ceberio,
and Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
Emails: {efreudenthal,mceberio,vladik}@utep.edu

***Abstract*— In traditional security systems, for each task, we either trust an agent or we don't. If we trust an agent, we allow this agent full access to this particular task. This agent can usually allow his trusted sub-agents the same access, etc. If a trust management system only uses "trust" and "no trust" options, then a person should trust everyone in this potentially long chain. The problem is that trust is rarely a complete trust, there is a certain probability of distrust. So, when the chain becomes long, the probability of a security leak increases. It is desirable to keep track of trust probabilities, so that we should only delegate to agents whose trust is above a certain threshold. In this paper, we present efficient algorithms for handling such probabilities.**

I. Probabilistic Approach is Needed

A. Traditional Approach to Trust Management: Brief Idea

In traditional security systems (see, e.g., [23]), for each task, we either trust an agent or we don't. If we trust an agent, we allow this agent full access to this particular task. For example, I trust a bank, where I keep my money, to handle my account.

This agent can usually allow his trusted sub-agents the same access, etc. For example, the bank can outsource some money operations to another company and trust this company to handle its accounts. Since I trust the bank, and bank trusts the company, I therefore have to trust the company that handles my account.

B. Traditional Approach to Trust Management: Main Problem

The problem with the traditional approach is that trust is rarely a complete trust. For example, I trust a bank, where I keep my money, to handle my account. I know that there have been cases when banks cheated on clients, but overall, my trust is pretty high, say $p = 99.9\%$ (i.e., distrust is $d = 0.1\%$).

The bank, in its turn, trusts a company in India – to which this bank has outsourced to handle my account – with a certain high probability. That company trusts its own employees, etc.

If a trust management system only uses "trust" and "no trust" options, then I should trust everyone in this potentially long chain. However, when the chain becomes long, the probability of a security leak increases, and the resulting probability of distrust may get much higher than my original 0.1%.

C. What Needs to Be Done

It is desirable to keep track of trust probabilities, so that we should only delegate to agents whose trust is above a certain threshold $\widetilde{p}$.

II. Probabilistic Approach: Idea

To implement the above idea, we can write down the rules describing who directly trusts whom and with what probability. The objective of the resulting Qualitative Trust Management System is, given such rules and the two agents f and s, to determine the probability $p_t(f, s)$ with which the first agent f should trust the second agent s.

Such systems exist:

- a system in which, crudely speaking, all direct trusts are assumed to be of the same probability is described in [7];
- a more complex system in which direct trusts may have different probabilities is described in [12].

In this talk, we will describe new efficient algorithms for computing the corresponding probabilities $p_t(f, s)$.

III. Independent Case: Towards Precise Formulation of the Problem

Let us formulate the problem in precise terms.

A. Input Data: Formal Description

We have a finite set A; it elements are called *agents*.

For some pairs (a, b) of agents, we are given a number $p_0(a, b) > 0$ from the interval $(0, 1]$. This number is called a *probability* with which agent a directly trusts agent b.

B. Desired Output: Informal Description

Informally, our objective is to describe, for given two agents f and s, the probability $p_t(f, s)$ with which the agent f trusts the agent s.

C. Graphs: A Natural Description of Input Data

From the mathematical viewpoint, it is reasonable to describe this input as a directed graph $G = (A, E)$, in which:

- the agents are vertices, and
- an edge $(a, b) \in E$ connects those pairs of vertices a and b for which we know the probability of direct trust.

D. Independence: Reasonable Assumption

All we know is the probabilities of direct trust. Since we have no information on the dependence between different direct trust links, it makes sense to assume that the corresponding events are independent; see, e.g., [26].

Under this independence assumption, we can formulate the problem in precise terms.

E. Desired Output: Formal Description

We have a graph $G = (A, E)$, in which there is a probability $p_0(a, b)$ assigned to every edge. We can now describe a random subgraph (A, E_r) ($E_r \subseteq E$) of the original graph as follows:

- for every edge $(a, b) \in E$, the probability that this edge is present in E_r is equal to $p_0(a, b)$;
- for two different edges, the events describing their presence is E_r are statistically independent.

In other words, for each edge $(a, b) \in E$:

- this edge belongs to E_r with probability $p_0(a, b)$, and
- this edge does not belong to E_r with probability $1 - p_0(a, b)$.

Since edges are statistically independent, we can provide an explicit formula for the probability $p(E')$ that the resulting random graph E_r coincides with a given subgraph $E' \subseteq E$:

$$p(E') = \left(\prod_{(a,b)\in E'} p_0(a, b)\right) \cdot \left(\prod_{(a,b)\notin E'} (1 - p_0(a, b))\right). \quad (1)$$

These values $p(E')$ form a probability distribution on the set of all subsets $E' \subseteq E$. Based on this probability distribution, we can then determine the desired probability $p_t(f, s)$ as the probability that in the random graph E_r, there is a directed path from f to s. If we denote the existence of such a path by $f \stackrel{E_r}{\to} s$, we can then describe the desired probability $p_t(f, s)$ as follows:

$$p_t(f, s) = \sum_{E':f\stackrel{E'}{\to}s} p(E'). \quad (2)$$

IV. Algorithm for the Independent Case

A. Monte-Carlo Simulation: Main Idea

Since the edges are assumed to be statistically independent, we can use the following Monte-Carlo simulation algorithm to generate a random graph E_r:

We loop over all edges $(a, b) \in E$, and for each edge (a, b), we keep it in E_r with probability $p_0(a, b)$. Specifically, for each edge, we do the following:

- we run a standard random number generator that generates numbers uniformly distributed on the interval $[0, 1]$; as a result, we get a value $\xi \in [0, 1]$;
- then, we compare ξ with $p_0(a, b)$:
  - if $\xi \le p_0(a, b)$, we keep the edge (a, b) in E_r;
  - otherwise, we delete the edge (a, b) from the graph E_r.

Since the random variable ξ is uniformly distributed on the interval $[0, 1]$, the probability that ξ belongs to any interval $[x, y]$ is equal to the width $y - x$ of this interval. In particular, the probability that $\xi \le p_0(a, b)$, i.e., that ξ belongs to the interval $[0, p_0(a, b)]$, is equal to $p_0(a, b)$.

As a result of this process, we get different subgraphs E' with probability that is described by the formula (1). Thus, the desired probability $p_t(f, s)$ is equal to the probability that in thus generated random graph E_r, there is a directed path from f to s.

We can therefore estimate this probability $p_t(f, s)$ as the frequency of this event. In other words:

- we select a number of iteration N;
- then, N times, we generate the corresponding random graph E_r and check whether in this graph, there is a path from f and s;
  - there are known algorithms for efficiently checking the existence of such a path; see, e.g., [6];
- we then estimate $p_t(f, s)$ as the ratio $f \stackrel{\text{def}}{=} M/N$, where M is the number of generated graphs in which f and s were connected.

It is well known that after N simulations, the frequency $f = M/N$ estimates the desired probability p with the accuracy $\sim \sqrt{\dfrac{p \cdot (1 - p)}{N}}$ (see, e.g., [26]). Specifically, the standard deviation of the difference $f - p$ is equal to this value, and for large N, the distribution of this difference is approximately normal. Thus, e.g.:

- with certainty 95% (corresponding to 2σ bounds), we can conclude that
$$|p - f| \le 2 \cdot \sqrt{\frac{p \cdot (1 - p)}{N}};$$
- with certainty 99.9% (corresponding to 3σ bounds), we can conclude that
$$|p - f| \le 3 \cdot \sqrt{\frac{p \cdot (1 - p)}{N}};$$
- with certainty $1 - 10^{-8}$ (corresponding to 6σ bounds), we can conclude that
$$|p - f| \le 6 \cdot \sqrt{\frac{p \cdot (1 - p)}{N}}.$$

B. Monte-Carlo Simulation: Problem

In many applications of trust, the desired level of trust $\widetilde{p}$ is very high. For example, for a bank with a thousand employees around the country, trust probability 99.9% may sound very reasonable, but it means that at any given moment of time, at least one of the employees is stealing money – this is clearly not an acceptable situation. For problems related to national security, the desired level of trust must be even higher.

Since we want to check whether the actual probability $p_t(f, s)$ is above the threshold level of trust $\widetilde{p}$, the estimation error $|p - f|$ of our estimate f of the probability $p = p_t(f, s)$ must be much smaller than the difference $1 - \widetilde{p}$ between the ideal trust 1 and the desired threshold $\widetilde{p}$ – otherwise, even in

the ideal case when $f = 1$, we would still not be able to guarantee that the actual probability p exceeds the threshold.

Thus, for trust problems, we must estimate the probability $p_t(f, s)$ with the accuracy $\varepsilon \cdot (1 - \widetilde{p})$, where $\varepsilon < 1$ and $1 - \widetilde{p} < 10^{-3}$. In other words, we would like to find the value f for which, with a given certainty δ, we can guarantee that $|p - f| \le \varepsilon \cdot (1 - \widetilde{p})$.

If the given certainty corresponds to the $k \cdot \sigma$ bounds for the normal distribution, then, after N iterations, we can guarantee the accuracy

$$|p - f| \le k \cdot \sqrt{\frac{p \cdot (1 - p)}{N}}.$$

So, to guarantee that $|p - f| \le \varepsilon \cdot (1 - \widetilde{p})$, we must guarantee that $k \cdot \sqrt{\dfrac{p \cdot (1 - p)}{N}} \le \varepsilon \cdot (1 - \widetilde{p})$, i.e., that

$$N \ge \frac{k^2}{\varepsilon^2} \cdot \frac{p \cdot (1 - p)}{(1 - \widetilde{p})^2}.$$

For $p \approx 1$ and $1 - p \approx 1 - \widetilde{p} \approx 10^3$, we get

$$N \ge 10^3 \cdot (k^2/\varepsilon^2) \approx 10^4.$$

For more accurate estimations, we need even larger number of simulations. These simulations take time. How can we decrease the time that is needed for these simulations?

Comment. Our problem can be reformulated in terms of propositional formulas. Indeed, to every edge $(a, b) \in E$, we can assign a propositional variable $c_{a,b}$ that is true if and only if the nodes a and b are connected in the resulting random graph E_r. The condition that there is a path from f to s can be described as a propositional formula F in terms of these variables $c_{a,b}$. In these terms, the desired probability $p_t(f, s)$ can be described as a probability that the formula F is true when each variable $c_{a,b}$ is true with probability $p_0(a, b)$ and the corresponding random variables are independent.

Thus, our problem can be viewed as a particular case of the general Stochastic Satisfiability problem SSAT: given a propositional formula $F(x_1, \ldots, x_n)$ with n random variables and the probabilities $p_1, \ldots, p_n$ of these variables being true, find the probability P that F is true under the assumption that x_i are independent random variables. It is known that in general, the problem of computing P exactly is NP-hard, and the problem of computing the small value P with a good relative accuracy is difficult.

There have been many good algorithms for solving this general problem (see, e.g., [14], [15]), but the problem remains very difficult even for the case when all the probabilities p_i are equal to 0.5. Our case is different because in our case, the difficulty comes not from NP-hardness but from the fact that the original probabilities are very small. It turns out that in this particular case, a simple efficient algorithm is possible.

C. How to Solve the Problem: An Idea

The main problem with the Monte-Carlo simulation comes from the fact that in the trust problems, the probabilities $p_0(a, b)$ of direct trust are close to 1 – or, equivalently, that the probabilities $m_0(a, b) \stackrel{\text{def}}{=} 1 - p_0(a, b)$ of direct mistrust are very small.

Since the method does not work well when the values $m_0(a, b)$ are small, a reasonable idea is:

- to compute the value $p_t^{(\lambda)}(f, s)$ (or, equivalently, $m_t^{(\lambda)}(f, s) \stackrel{\text{def}}{=} 1 - p_t^{(\lambda)}(f, s)$) for larger original mistrust values $m_0^{(\lambda)}(a, b) \stackrel{\text{def}}{=} \lambda \cdot m_0(a, b)$, for several different values $\lambda > 1$ – and then
- use extrapolation to reconstruct the desired value $m_t^{(\lambda)}(f, s) = m_t^{(1)}(f, s)$.

How can we do this?

The formula (2) describes $p(f, s)$ as a sum of terms $p(E')$, and, according to the formula (1), each of these terms $p(E')$ is a polynomial in terms of the variables $p_0(a, b)$. Thus, when we substitute the expressions $p_0(a, b) = 1 - m_0(a, b)$ into the formulas (1) and (2), we conclude that $m_t(f, s)$ is a polynomial in terms of $m_0(a, b)$. Each polynomial can be represented as a sum of 0-th order terms, linear terms, quadratic terms, etc., i.e., as the following sum:

$$m_t(f, s) = m_0 + m_1 + \ldots + m_k + \ldots, \tag{3}$$

where:

- m_0 is a constant;
- m_1 is the sum of all the terms that are linear in $m_0(a, b)$;
- m_2 is the sum of all the terms that are quadratic in $m_0(a, b)$;
- etc.

The value m_0 corresponds to the case when all the mistrust levels are 0s ($m_0(a, b) = 0$), so all the trusts are absolute: $p_0(a, b) = 1 - m_0(a, b) = 1 - 0 = 1$. In this case, every edge (a, b) from the original graph E is present in E' with probability 1. Thus, if in the original graph E, we have a path from f to s, then, with probability 1, this same path leads from f to s in $E_r = E$. (And if in E, there is no path from f to s, then there is no such path in a subgraph E_r either, so $p_t(f, s) = 0$ anyway.)

So, with the exception of the degenerate (and easy-to-detect) case when f and s are not even originally connected, for $m_0(a, b) = 0$, we have $p_t(a, b) = 1$ – hence $m_t(f, s) = 0$. Since in this case, $m_0(f, s) = m_0$, we conclude that $m_0 = 0$.

Let d denote the ordinal number of the first non-zero term in the formula (3); then, (3) takes the following form:

$$m_t(f, s) = m_d + m_{d+1} + \ldots \tag{4}$$

Since the values $m_0(a, b)$ are very small, all d-th order terms are much larger than all the $(d+1)$-th order terms, i.e., $m_d \gg m_{d+1}$, and with a good accuracy, $m_t(f, s) \approx m_d$.

If we use the values $m_0^{(\lambda)}(a, b) = \lambda \cdot m_0(a, b)$ instead of the original values $m_0(a, b)$, then we similarly get

$$m_t^{(\lambda)}(f, s) = m_d^{(\lambda)} + m_{d+1}^{(\lambda)} + \ldots \tag{5}$$

When we replace $m_0(a, b)$ by $\lambda \cdot m_0(a, b)$, then each p-th order term – the sum of the products of p values $m_0(a, b)$ –

is multiplied by λ^p. Thus, $m_p^{(\lambda)} = \lambda^p \cdot m_p$, hence:

$$m_t^{(\lambda)}(f,s) = \lambda^d \cdot m_d + \lambda^{d+1} \cdot m_{d+1} + \ldots \quad (6)$$

When the value of λ is not too large, we still have $m_t^{(\lambda)}(f,s) \approx m_d^{(\lambda)}$ hence $m_t^{\lambda)}(f,s) \approx \lambda^d \cdot m_d$. Since, as we have mentioned, $m_d \approx m_t(f,s)$, we thus conclude that:

$$m_t^{(\lambda)}(f,s) \approx \lambda^d \cdot m_t(f,s). \quad (7)$$

This formula enables us to perform the desired extrapolation.

Indeed, if we know the values $m_t^{(\lambda_l)}(f,s)$ corresponding to different factors $\lambda_1, \ldots, \lambda_m$ ($m > 2$), then, from the formula (7), we conclude that

$$m_t^{(\lambda_l)}(f,s) \approx \lambda_l^d \cdot m_t(f,s).$$

Turning to logarithms of both sides, we conclude that

$$d \cdot a_l + x \approx b_l, \quad (8)$$

where $x \stackrel{\text{def}}{=} \ln(m_t(f,s))$, $a_l \stackrel{\text{def}}{=} \ln(\lambda_l)$ and $b_l \stackrel{\text{def}}{=} \ln(m_t^{(\lambda_l)}(f,s))$.

We can use the Least Square Method to estimate d and x. Once we know $x = \ln(m_t(f,s))$, we can now reconstruct the desired value $m_t(f,s)$ as $\exp(x)$ and $p_t(f,s) = 1 - m_t(f,s)$ as $1 - \exp(x)$. Thus, we arrive at the following algorithm.

D. New Algorithm: Description and Implementation

Suppose that we are given the graph (A, E), two nodes f and s, and the values $p_0(a,b)$ – or, equivalently, the values $m_0(a,b) = 1 - p_0(a,b)$. Our objective is to estimate the value $p_t(f,s)$ with a given accuracy ε.

To solve this problem, we do the following. We select a reasonable value N – the number of iterations used in the Monte-Carlo approach; e.g., we select $N = 25$ or $N = 50$ or $N = 100$. First, we try the above Monte-Carlo method with the original values $m_0(a,b)$. The original Monte-Carlo simulation technique does not work well when $p_t(f,s) \approx 1$ and the corresponding accuracy $\sim 1/\sqrt{N}$ is much larger than ε. In this case, we do the following:

- We try increasing values of λ, e.g., $\lambda = 2, 4, 8$, etc., and apply the Monte-Carlo method to the corresponding mistrust values $m_0^{(\lambda)}(a,b) = \lambda \cdot m_0(a,b)$. We continue this process until we reach a value λ for which $m_t^{(\lambda)}(f,s)$ is reliably different from 0 – i.e., for which $m_t^{(\lambda)}(a,b) \gg 1/\sqrt{N}$. We will denote this value λ by λ_1.
- Once we reach this value λ_1, we try several ($m \geq 2$) larger values $\lambda_1 < \lambda_2 < \ldots < \lambda_m$. For each of these values, we apply the Monte-Carlo method and get an estimate for $m_t^{(\lambda_l)}(f,s)$.
- Based on these estimates, we:
  - compute the values $a_l = \ln(\lambda_l)$ and $b_l = \ln(m_t^{(\lambda_l)}(f,s))$;
  - apply the Least Squares Method to solve the system of linear equations (8) with unknowns k and x, and
  - estimate $p_t(a,b)$ as $1 - \exp(x)$.

V. Case When Trusts Depend on Independent Factors

A. Formulation of the Problem

The same approach can be applied when trusts are, in general, not independent, but they depend on several common factors which are independent random variables. For example, for computer connections, our trust in a connection from a to b may come from third parties certifying that these connections are trustworthy. In this case, our trust in these connections comes from our trust in these third parties.

If the same third party certifies two different connections (a,b) and (a',b'), then, of course, the variables $c_{a,b}$ and $c_{a',b'}$ representing these connections are *not* independent because they depend on the same third party. We can, however, assume that the third parties themselves are indeed independent entities.

There are many different ways how our trust in a connection depends on the their parties:

- In some cases, there is exactly one third party that certifies the connection. In this case, our trust in this connection is exactly equal to our trust in this third party.
- It is also possible that several third parties certifies the connection. In this case, we should trust this connection if at least one of these certifiers is trustworthy: either the first one, or the second one, etc.
- It may also be that our trust is based on a joint statement by two third parties, in which case, to trust the connection, we must trust both third parties.

In general, we can have more general propositional formulas relating our trust in the connection with our trust in these different third parties.

In precise terms, in addition to the graph (A, E), we have a finite list of "third parties" $t_1, \ldots, t_n$. For each potential third party t_i, we know the probability $p_i \approx 1$ with which we trust this party; based on these probabilities p_i, we can compute the probabilities $m_i = 1 - p_i$ with which we mistrust the corresponding parties.

For each connection $(a,b) \in E$, we have a propositional formula $F_{a,b}(v_1, \ldots, v_n)$ that describes our trust in this connection as a function of the Boolean variables v_i describing our trust in third parties t_i. For example:

- if a connection was certified by exactly one third party t_3, then $F_{a,b}(v_1, \ldots, v_n) = v_3$;
- if each of the two third parties t_3 and t_5 certifies this connection, then $F_{a,b}(v_1, \ldots, v_n) = v_3 \vee v_5$;
- if the connection is certified by a joint declaration of third parties t_4 and t_6, then $F_{a,b}(v_1, \ldots, v_n) = v_4 \,\&\, v_6$.

B. Monte-Carlo Approach

Since the third parties are independent, we can use the Monte-Carlo technique to estimate the desired probability $p_t(f,s)$. Specifically, on each iteration, we do the following:

- First, for each third party t_i, we set the trust v_i in this party to be true with probability p_i and false with probability $1 - p_i$.

- Then, for each connection $(a,b) \in E$, we place this connection in the random graph E_r if the corresponding formula $F_{a,b}(v_1,\ldots,v_n)$ become true after substituting the values v_i obtained on the first stage.
- Finally, we check whether there is a path from f to s in the resulting graph E_r.

We then estimate the desired probability $p_t(f,s)$ as the ratio M/N, where N is the overall number of iteration, and M is the number of iterations in which there was a path from f to s in the corresponding graph E_r.

C. Problem

When the probability $p_t(f,s)$ is small, we will need a large number of iterations to get a good estimate of $p_t(f,s)$.

D. Solution

To decrease the number of iterations, we can use the same idea as above:

- We try increasing values of λ, e.g., $\lambda = 2, 4, 8$, etc., and apply the Monte-Carlo method to the corresponding mistrust values $m_i^{(\lambda)} = \lambda \cdot m_i$. We continue this process until we reach a value λ for which $m_t^{(\lambda)}(f,s)$ is reliably different from 0 – i.e., for which $m_t^{(\lambda)}(a,b) \gg 1/\sqrt{N}$. We will denote this value λ by λ_1.
- Once we reach this value λ_1, we try several ($m \geq 2$) larger values $\lambda_1 < \lambda_2 < \ldots < \lambda_m$. For each of these values, we apply the Monte-Carlo method and get an estimate for $m_t^{(\lambda_l)}(f,s)$.
- Based on these estimates, we:
  - compute the values $a_l = \ln(\lambda_l)$ and $b_l = \ln(m_t^{(\lambda_l)}(f,s))$;
  - apply the Least Squares Method to solve the system of linear equations (8) with unknowns k and x, and
  - estimate $p_t(a,b)$ as $1 - \exp(x)$.

VI. Possibly Dependent Case: Formulation of the Problem

A. Motivation

In the above text, we assumed that either all the trusts are statistically independent or at least that the trusts depend on statistically independent factors. In reality, however, different trusts may be dependent (correlated) – e.g., if these trusts come from the assurances of the same third party.

In this more realistic situation, depending on the degree of correlation, we may get different values of the resulting trust $p_t(f,s)$. In critical systems, it is reasonable to guarantee the trust only we are guaranteed that $p_t(f,s)$ exceeds the threshold $\widetilde{p}$, i.e., if we know that all possible values of $p_t(f,s)$ are greater than or equal to the threshold.

To check this requirement, it is, of course, necessary and sufficient to check the smallest possible value $\underline{p}_t(a,b)$ of $p_t(f,s)$ exceeds the threshold: $\underline{p}_t(f,s) \geq \widetilde{p}$.

Thus, we must be able to compute this "worst-case" trust probability $\underline{p}_t(f,s)$.

B. Precise Formulation of the Problem

Similarly to the dependent case, we have a graph (A,E); for each edge $(a,b) \in E$, we are given the probability $p_0(a,b)$.

We consider all possible probability distributions $p(E')$ on the set of all subgraphs $E' \subseteq E$ that are consistent with this information, i.e., for which, for every $(a,b) \in E$, we have

$$\sum_{E':(a,b)\in E'} p(E') = p_0(a,b).$$

For every two edges f and s and for each such probability distribution $p(E')$, we can define $p_t(f,s)$ as the overall probability that there is a path from f to s:

$$p_t(f,s) = \sum_{E':f \stackrel{E'}{\to} s} p(E').$$

We can now define the desired worst-case probability $\underline{p}_t(f,s)$ as the exact lower bound of all such values $p_t(f,s)$:

$$\underline{p}_t(f,s) \stackrel{\text{def}}{=}$$

$$\inf\{p_t(f,s) \mid p \text{ is consistent with the given information}\}.$$

Our objective is to compute this value $\underline{p}_t(f,s)$.

C. This Problem Is Difficult to Solve

In the dependence case, we knew the exact distribution $p(E')$, so we could use the Monte-Carlo simulation techniques and estimate the desired value $p_t(f,s)$. In the more general case when there is a possible dependence, we may have several different probability distributions $p(E')$ consistent with the given information. For each of these distributions, we can still use the Monte-Carlo simulation and compute the corresponding probability $p_t(f,s)$. However, there are, in general, infinitely many such distributions; so, we cannot find the smallest possible value $\underline{p}_t(f,s)$ of the probability $p_t(f,s)$ by simply simulating all such distributions.

D. This Problem is a Part of a General Problem

How can we efficiently solve the problem of computing $\underline{p}_t(f,s)$?

The difficulty of solving this problem comes from the fact that instead of a single probability distribution, our knowledge is consistent with several different probability distributions. In other words, instead of a single distribution p, we have a *class* $\mathcal{P}$ of possible probability distributions, and our objective is to find the smallest possible value of a statistical characteristic (namely, the probability $p_t(f,s)$) over all the distributions from this class.

Problem of this type – and their potential applications – have been described, in a general context, in the monographs [20], [27]; for further developments, see, e.g., [1], [2], [3], [4], [5], [8], [9], [10], [11], [13], [16], [17], [18], [19], [21], [22], [24], [25], [28] and references therein.

In this paper, we will use ideas from these monographs and papers. As a result, we arrive at the following algorithm.

VII. Possibly Dependent Case: Algorithm and Justification

A. Preliminary Definitions

In a graph (A, E), we say that a sequence $\gamma = (a_0, a_1, \ldots, a_n)$ is a *path* from a_0 to a_n if for every i, the pair (a_i, a_{i+1}) is an edge.

The *length* $\ell(\gamma)$ of the path γ is defined as the sum of the lengths $d_0(a_i, a_{i+1})$ of the corresponding edges:

$$\ell(\gamma) \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} d_0(a_i, a_{i+1}).$$

B. Algorithm: Idea

The desired value $\underline{p}_t(f, s)$ is equal to $\max(1 - \underline{d}_t(f, s), 0)$, where $\underline{d}_t(f, s)$ is the length of the shortest path from f to s, i.e.,

$$\underline{d}_t(f, s) \stackrel{\text{def}}{=} \min\{\ell(\gamma) \,|\, \gamma \text{ is a path from } f \text{ to } s\}.$$

C. Algorithm: Implementation

There exists efficient algorithms for finding the length of the shortest path; see, e.g., [6].

The efficient shortest path algorithm was originally proposed by Dijkstra.

D. Algorithm: justification

1°. Let us first show that for every distribution $p(E')$ that is consistent with the given information, we have $d_t(f, s) \le \underline{d}_t(f, s)$, where $d_t(f, s) \stackrel{\text{def}}{=} 1 - p_t(f, s)$.

Let $\gamma_0 = (a_0, a_1, \ldots, a_n)$ be the shortest path from $a_0 = f$ to $a_n = s$. Then, by definition of the function $\underline{d}_t(f, s)$, we have

$$\underline{d}_t(f, s) = d_0(a_0, a_1) + \ldots + d_0(a_{n-1}, a_n), \qquad (9)$$

where $(a_i, a_{i+1}) \in E$ for all i. If all the connections (a_i, a_{i+1}) are in the random graph E', then in E', there exists a path from $f = a_0$ to $a_n = s$ – namely, the path γ_0. So, if there is no path from s to f, then at least one of the connections (a_i, a_{i+1}) is not present in the random graph.

Let us denote:

- the event that there is no path from f to s by $N_t(f, s)$, and
- the event that there is no direct connection from a_i to a_{i+1} by $N_0(a_i, a_{i+1})$.

Then:

- the probability of $N_t(f, s)$ is equal to $d_t(f, s)$, while
- the probability of each event $N_0(a_i, a_{i+1})$ is equal to $d_0(a_i, a_{i+1})$.

In these terms, the above logical conclusion takes the following form: if the event $N_t(f, s)$ occurs, then at least one of the events $N_0(a_i, a_{i+1})$ must have occurred:

$$N_t(f, s) \supset (N_0(a_0, a_1) \vee \ldots \vee N_0(a_{n-1}, a_n)).$$

Therefore, the probability $d_t(f, s)$ of the event $N_t(f, s)$ cannot exceed the probability of the disjunction:

$$d_t(f, s) \le P(N_0(a_0, a_1) \vee \ldots \vee N_0(a_{n-1}, a_n)).$$

It is well known that the probability of a disjunction

$$S_1 \vee \ldots \vee S_n$$

of arbitrary n events $S_1, \ldots, S_n$ cannot exceed the sum of the corresponding probabilities $p(S_1) + \ldots + p(S_n)$. In our case, this means that $d_t(f, s)$ cannot exceed the sum of the probabilities $p(N_0(a_0, a_1)) + \ldots + p(N_0(a_{n-1}, a_n))$, i.e., that $d_t(f, s) \le d_0(a_0, a_1) + \ldots + d_0(a_{n-1}, a_n)$.

Due to (9), this means that $d_t(f, s) \le \underline{d}_t(f, s)$.

2°. From $d_t(f, s) = 1 - p_t(f, s) \le \underline{d}_t(f, s)$, it follow that $p_t(f, s) \ge 1 - \underline{d}_t(f, s)$. Since $p_t(f, s)$ is a probability, it is a non-negative number, so

$$p_t(f, s) \ge \max(1 - \underline{d}_t(f, s), 0).$$

In other words, for every distribution $p(E')$ that is consistent with the given information, the corresponding probability $p_t(f, s)$ is larger than or equal to $\max(1 - \underline{d}_t(f, s), 0)$. Thus, the infimum $\underline{p}_t(f, s)$ of all such values $p_t(f, s)$ is also smaller than or equal to this number, i.e.,

$$\underline{p}_t(f, s) \ge \max(1 - \underline{d}_t(f, s), 0).$$

3°. To complete the proof, we will produce an example of a probability distribution $p(E')$ for which $p_t(f, s) \le \max(1 - \underline{d}_t(f, s), 0)$. Then, for $\underline{p}_t(f, s) \le p_t(f, s)$, we will get $\underline{p}_t(f, s) \le \max(1 - \underline{d}_t(f, s), 0)$, hence

$$\underline{p}_t(f, s) = \max(1 - \underline{d}_t(f, s), 0).$$

To describe the corresponding distribution, we will use the standard projection π of the real line $\mathbb{R}$ onto the interval $[0, 1)$ that assigns to each real number x its fractional part $\pi(x) \stackrel{\text{def}}{=} x - \lfloor x \rfloor$. This projection has a simple geometric interpretation: we interpret the interval $[0, 1)$ as a circle of circumference 1, and we "wrap up" the real line around this circle.

It is easy to see that in this wrapping, the length of an interval is preserved as long as it does not exceed 1. Thus, for any interval $I = [x, y] \subseteq \mathbb{R}$ of length $y - x \le 1$, its projection $\pi(I)$ is either an interval, or a pair of intervals, and the total length of the set $\pi(I)$ is equal to the length $y - x$ of the original interval.

The corresponding distribution is located on the graphs $E(\omega)$ corresponding to different real numbers $\omega \in [0, 1)$; these graphs will be describe below. The probability of different graphs $E(\omega)$ is described by the uniform distribution on the interval $[0, 1)$.

The graphs $E(\omega)$ are described as follows. For every two nodes a and b for which $(a, b) \in E$, we consider the interval $I(a, b) \stackrel{\text{def}}{=} [\underline{d}_t(f, a), \underline{d}_t(f, a) + d_0(a, b)]$ of length $d_0(a, b)$. For

every $\omega \in [0,1)$, the edge (a,b) belongs to the graph $E(\omega)$ if and only if $\omega \notin \pi(I(a,b))$.

3.1°. Let us prove that thus defined distribution is indeed consistent with the original information, i.e., with all the given values $p_0(a,b)$.

Indeed, since the distribution on ω is uniform, the probability that $\omega \in \pi(I(a,b))$ is equal to the total length of the set $\pi(I(a,b))$. Due to the above property of the projection, this total length is equal to the length of the original interval $I(a,b)$, i.e., to $d_0(a,b)$. Thus, the probability that $\omega \in \pi(I(a,b))$ is equal to $d_0(a,b)$. Therefore, the probability of the opposite event $\omega \notin \pi(I(a,b))$ is equal to $1 - d_0(a,b)$. By definition of $d_0(a,b)$, the value $1 - d_0(a,b)$ is exactly $p_0(a,b)$.

So, the probability that the edge (a,b) belongs to the graph $E(\omega)$ is exactly $p_0(a,b)$.

3.2°. Let us prove that for every path $\gamma = (a_0, \ldots, a_n)$ that starts at $a_0 = f$, if all the edges $(a_0,a_1), \ldots, (a_{n-1},a_n)$ from this path belong to the graph $E(\omega)$, then $\omega \geq \underline{d}_t(a_0,a_n)$.

We will prove this statement by induction over the length n of the path.

3.2.1°. *Induction base.* For $n = 0$, we have $\underline{d}_t(a_0,a_0) = 0$, so the desired inequality is clearly true.

3.2.2°. *Induction step.* Let us assume that the desired property is true for n, and we have added one more edge (a_n,a_{n+1}) to the path γ. Let us prove that the desired property holds for the extended path.

If all $n+1$ edges $(a_0,a_1), \ldots, (a_{n-1},a_n), (a_n,a_{n+1})$ belong to the graph $E(\omega)$, then the first n edges also belong to the graph. By the induction assumption, this means that $\omega \geq \underline{d}_t(f,a_n)$.

If $\underline{d}_t(f,a_n) \geq 1$, then – since ω is from the interval $[0,1)$ – this inequality means that no such ω exists; so, the graph $E(\omega)$ simply cannot contain all the edges from the path γ. In this case, the statement that we are trying to prove is trivially true for an enlarged path – because false implies everything.

It is therefore sufficient to only consider the case when $\underline{d}_t(a_0,a_n) < 1$. In this case, by definition of $E(\omega)$, the fact that $(a_n,a_{n+1}) \in E$ means that

$$\omega \notin \pi([\underline{d}_t(f,a_n), \underline{d}_t(f,a_n) + d_0(a_n,a_{n+1})]).$$

Since $\underline{d}_t(f,a_n) < 1$, the projection $pi(\ldots)$ starts with the value $\underline{d}_t(f,a_n)$. Whether this projection is a single interval or two intervals depends on whether $\underline{d}_t(f,a_n) + d_0(a_n,a_{n+1}) < 1$ or not. Let us consider both possibilities:

- If $\underline{d}_t(f,a_n) + d_0(a_n,a_{n+1}) \geq 1$, then the projection contains all the values from $\underline{d}_t(f,a_n)$ to 1. Since $\omega \geq \underline{d}_t(f,a_n)$ and ω cannot belong to this projection, we conclude that no such ω is possible, so the desired property is trivially true.
- If $\underline{d}_t(f,a_n) + d_0(a_n,a_{n+1}) < 1$, then the projection coincides with the original interval

$$[\underline{d}_t(f,a_n), \underline{d}_t(f,a_n) + d_0(a_n,a_{n+1})].$$

Since $\omega \geq \underline{d}_t(f,a_n)$ and ω cannot belong to this projection, we conclude that

$$\omega \geq \underline{d}_t(f,a_n) + d_0(a_n,a_{n+1}).$$

Let us continue the analysis of the second case. By definition, $\underline{d}_t(f,a_{n+1})$ is the length of the shortest path from f to a_{n+1}. In particular, when we add the edge (a_n,a_{n+1}) to the shortest path from f to a_n, we conclude that

$$\underline{d}_t(f,a_{n+1}) \leq \underline{d}_t(f,a_n) + d_0(a_n,a_{n+1}).$$

Hence, the inequality $\omega \geq \underline{d}_t(f,a_n) + d_0(a_n,a_{n+1})$ implies that $\omega \geq \underline{d}_t(f,a_{n+1})$.

In both cases, the induction step is proven, and so is the result.

3.3°. Due to Part 3.2 of this proof, if there is a path from f to s in a graph $E(\omega)$, then $\omega \geq \underline{p}_t(f,s)$. Thus, the probability $p_t(f,s)$ that there exists such a path does not exceed the probability that a uniformly distributed number $\omega \in [0,1)$ is $\geq \underline{p}_t(f,s)$. In other words,

$$p_t(f,s) \leq \max(1 - \underline{p}_t(f,s), 0).$$

The statement is proven, so the algorithm has been justified.

VIII. Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0365, by NSF grants EAR-0112968, EAR-0225670, and EIA-0321328, by the Army Research Laboratories grant DATM-05-02-C-0046, and by the NIH grant 3T34GM008048-20S1.

The authors are very thankful to Luc Longpré (University of Texas at El Paso) and to Dan Wallach and Anwis Das (Rice University) for valuable discussions, and to the anonymous referees for the important suggestions.

References

[1] J. B. Beck, V. Kreinovich, and B. Wu, "Interval-Valued and Fuzzy-Valued Random Variables: From Computing Sample Variances to Computing Sample Covariances", In: M. Lopez, M. A. Gil, P. Grzegorzewski, O. Hrynewicz, and J. Lawry (eds.), *Soft Methodology and Random Information Systems*, Springer-Verlag, 2004, pp. 85-92.

[2] D. Berleant, "Automatically verified arithmetic with both intervals and probability density functions", *Interval Computations*, 1993, No. 2, pp. 48–70.

[3] D. Berleant, "Automatically verified arithmetic on probability distributions and intervals", In: R. B. Kearfott and V. Kreinovich, eds., *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[4] D. Berleant and C. Goodman-Strauss, "Bounding the results of arithmetic operations on random variables of unknown dependency using intervals", *Reliable Computing*, 1998, Vol. 4. No. 2, pp. 147–165.

[5] D. Berleant, L. Xie, and J. Zhang, "Statool: A Tool for Distribution Envelope Determination (DEnv), an Interval-Based Algorithm for Arithmetic on Random Variables", *Reliable Computing*, 2003, Vol. 9, No. 2, pp. 91–108.

[6] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[7] A. Das and D. S. Wallach, "Distributed authentication using web of trust", *Abstracts of the 2004 South Central Information Security Symposium SCISS'04*, Houston, Texas, April 24, 2004, p. 15.

[8] S. Ferson, *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.

[9] S. Ferson, L. Ginzburg, V. Kreinovich, and J. Lopez, "Absolute Bounds on the Mean of Sum, Product, etc.: A Probabilistic Extension of Interval Arithmetic", *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, Toronto, Canada, May 23–25, 2002, pp. 70–72.

[10] S. Ferson, L. Ginzburg, V. Kreinovich, H. T. Nguyen, and S. A. Starks, "Uncertainty in Risk Analysis: Towards a General Second-Order Approach Combining Interval, Probabilistic, and Fuzzy Techniques", *Proceedings of FUZZ-IEEE'2002*, Honolulu, Hawaii, May 12–17, 2002, Vol. 2, pp. 1342–1347.

[11] S. Ferson, D. Myers, and D. Berleant, *Distribution-free risk analysis: I. Range, mean, and variance*, Applied Biomathematics, Technical Report, 2001.

[12] E. Freudenthal and V. Karamcheti, *QTM: Trust Management with Quantified Stochastic Attributes*, NYU Computer Science Technical Report TR2003-848, available at http://csdocs.cs.nyu.edu/Dienst/UI/2.0/Describe/ncstrl.nyu_cs%fTR2003-848

[13] L. Granvilliers, V. Kreinovich, and N. Mueller, "Novel Approaches to Numerical Software with Result Verification", In: R. Alt, A. Frommer, R. B. Kearfott, and W. Luther (eds.), *Numerical Software with Result Verification*, International Dagstuhl Seminar, Dagstuhl Castle, Germany, January 19–24, 2003, Revised Papers, Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.

[14] R. M. Karp and M. G. Luby, *A new Monte-Carlo Method for Estimating the Failure Probability of an N-Component System*, University of California at Berkeley, Computer Science Division, Technical Report 83/117, 1983.

[15] R. M. Karp and M. G. Luby, "Monte-Carlo algorithms for enumeration and reliability problems", *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, 1983, pp. 54–63.

[16] V. Kreinovich, "Probabilities, Intervals, What Next? Optimization Problems Related to Extension of Interval Computations to Situations with Partial Information about Probabilities", *Journal of Global Optimization*, 2004, Vol. 29, No. 3, pp. 265–280.

[17] V. Kreinovich, S. Ferson, and L. Ginzburg, "Exact Upper Bound on the Mean of the Product of Many Random Variables With Known Expectations", *Reliable Computing*, 2003, Vol. 9, No. 6, pp. 441–463.

[18] V. Kreinovich and L. Longpré, "Computational complexity and feasibility of data processing and interval computations, with extension to cases when we have partial information about probabilities", In: V. Brattka, M. Schröder, K. Weihrauch, and N. Zhong, *Proceedings of the Conference on Computability and Complexity in Analysis CCA'2003*, Cincinnati, Ohio, USA, August 28–30, 2003, pp. 19–54.

[19] V. Kreinovich, G. N. Solopchenko, S. Ferson, L. Ginzburg, and R. Aló, "Probabilities, intervals, what next? Extension of interval computations to situations with partial information about probabilities", *Proceedings of the 10th IMEKO TC7 International Symposium on Advances of Measurement Science*, St. Petersburg, Russia, June 30–July 2, 2004, Vol. 1, pp. 137–142.

[20] V. P. Kuznetsov, *Interval Statistical Models*, Radio i Svyaz, Moscow, 1991 (in Russian).

[21] W. A. Lodwick and K. D. Jamison, "Estimating and Validating the Cumulative Distribution of a Function of Random Variables: Toward the Development of Distribution Arithmetic", *Reliable Computing*, 2003, Vol. 9, No. 2, pp. 127–141.

[22] R. E. Moore and W. A. Lodwick, "Interval Analysis and Fuzzy Set Theory", *Fuzzy Sets and Systems*, 2003, Vol. 135, No. 1, pp. 5–9.

[23] *Proceedings of the Second International Conference on Trust Management*, Oxford, UK, March 29–April 1, 2004.

[24] H. Regan, S. Ferson, and D. Berleant, "Equivalence of five methods for bounding uncertainty", *International Journal of Approximate Reasoning*, 2004, Vol. 36, No. 1, pp. 1–30.

[25] N. C. Rowe, "Absolute bounds on the mean and standard deviation of transformed data for constant-sign-derivative transformations", *SIAM Journal of Scientific Statistical Computing*, 1988, Vol. 9, pp. 1098–1113.

[26] H. M. Wadsworth Jr., *Handbook of statistical methods for engineers and scientists*, McGraw-Hill, N.Y., 1990.

[27] P. Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman & Hall, N.Y., 1991.

[28] R. Williamson and T. Downs, "Probabilistic arithmetic I: numerical methods for calculating convolutions and dependency bounds", *International Journal of Approximate Reasoning*, 1990, Vol. 4, pp. 89–158.