Audun Jøsang and Touhid Bhuiyan. Optimal Trust Network Analysis with Subjective Logic.

This paper appears in:

Optimal Trust Network Analysis with Subjective Logic[*]

Audun Jøsang
UNIK Graduate Center, University of Oslo
Norway
josang@unik.no

Touhid Bhuiyan
Faculty of Information Technology, QUT, Brisbane
Australia
t.bhuiyan@qut.edu.au

Abstract

Trust network analysis with subjective logic (TNA-SL) simplifies complex trust graphs into series-parallel graphs by removing the most uncertain paths to obtain a canonical graph. This simplification could in theory cause loss of information and thereby lead to sub-optimal results. This paper describes a new method for trust network analysis which is considered optimal because it does not require trust graph simplification, but instead uses edge splitting to obtain a canonical graph. The new method is compared with TNA-SL, and our simulation shows that both methods produce equal results. This indicates that TNA-SL in fact also represents an optimal method for trust network analysis and that the trust graph simplification does not affect the result.

1 Introduction

Trust networks consist of transitive trust relationships between people, organisations and software agents connected through a medium for communication and interaction. By formalising trust relationships, e.g. as reputation scores or as subjective trust measures, trust between parties within a domain can be derived by analysing the trust graph consisting of all the paths linking the parties together. TNA-SL (Trust Network Analysis with Subjective Logic) [7, 6] takes directed trust edges between pairs as input, and can be used to derive a level of trust between arbitrary parties that are interconnected through the network. Even in case no explicit trust paths between two parties exists, subjective logic allows a level of trust to be derived through the default vacuous opinions. TNA-SL therefore has a general applicability and is suitable for many types of trust networks. This can also be combined with Bayesian reputation systems [5]. In case of a complex network with dependent paths, TNA-SL

requires simplification of the trust graph into a DSPG (Directed Series-Parallel Graph) before computing the derived trust. The simplification consists of gradually removing the most uncertain trust paths until the whole graph can be represented in a series-parallel form. As this process removes information it could intuitively be considered sub-optimal.

In this paper we describe a new method for trust network analysis which avoids this problem by splitting dependent trust edges into independent parts, so that each part can be taken into account during the computation. This approach is considered optimal because it does not remove information. Simulations show that thew new method produces the same results as TNA-SL. This implies that the information removed through trust graph simplification is irrelevant for the trust network analysis, and that TNA-SL therefore produces optimal results.

2 Transitive Trust Paths

Trust transitivity means, for example, that if Alice trusts Bob who trusts David, then Alice will also trust David. This assumes that Alice is actually aware that Bob trusts David. This could e.g. be achieved through a *recommendation* from Bob to Alice as illustrated in Fig.1, where the indexes on each arrow indicate the sequence in which the trust relationships/recommendation are formed.
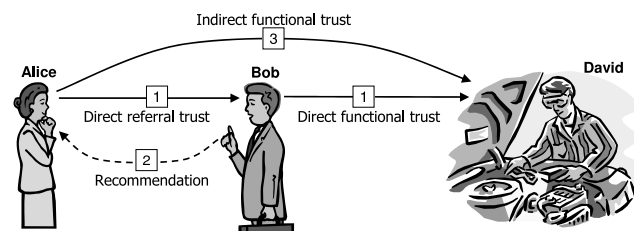

Figure 1. Transitive trust principle

The *trust scope* is the specific type(s) of trust assumed in a given trust relationship. In other words, the trusted party is relied upon to have certain qualities, and the scope is what the trusting party assumes those qualities to be.

Let us assume that Alice needs to have her car serviced, so she asks Bob for his advice about where to find a good car mechanic in town. Bob is thus trusted by Alice to know about a good car mechanic and to tell his honest opinion about that. Bob in turn trusts David to be a good car mechanic. Bob's trust in David is functional whereas Alice's trust in Bob is referral because Bob only refers to a mechanic. Both trust relationships are direct and have the same scope, namely that of being a good mechanic. Alice's trust in David is indirect because it is derived, and functional because David will actually do the job. This situation is illustrated in Fig.1, where the indexes indicate the order in which the trust relationships and recommendations are formed.

The examples above assume some sort of absolute trust between the agents along the transitive trust path. In reality trust is never absolute, and researchers have proposed to express trust as discrete verbal statements, as probabilities or other continuous measures. TNA-SL assumes that trust relationships are expressed as subjective opinions.

3   Parallel Trust Combination

It is common to collect advice from several sources in order to be better informed when making decisions. This can be modelled as *parallel trust combination* illustrated in Fig.2, where again the indexes indicate the order in which the trust relationships and recommendations are formed.
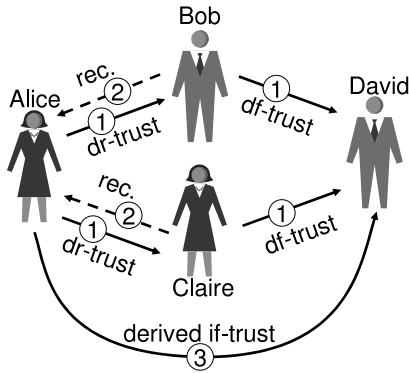

Figure 2. Parallel combination of trust paths

Let us assume again that Alice needs to get her car serviced, and that she asks Bob to recommend a good car mechanic. When Bob recommends David, Alice would like to get a second opinion, so she asks Claire whether she has heard about David. Intuitively, if both Bob and Claire recommend David as a good car mechanic, Alice's trust in David will be stronger than if she had only asked Bob. Parallel combination of positive trust thus has the effect of strengthening the derived trust.

In the case where Alice receives conflicting recommended trust, e.g. trust and distrust at the same time, she

needs some method for combining these conflicting recommendations in order to derive her trust in David. Our method, which is described in Sec.6, is based on subjective logic which is suitable for analysing such situations.

4   Structured Notation

Transitive trust networks can involve many principals, and in the examples below, capital letters A, B, C and D will be used to denote principals instead of names such as Alice and Bob.

We will use basic constructs of directed graphs to represent transitive trust networks, and add some notation elements which allow us to express trust networks in a structured way.

A single trust relationship can be expressed as a directed edge between two nodes that represent the trust source and the trust target of that edge. For example the edge $[A, B]$ means that A trusts B.

The symbol ":" will be used to denote the transitive connection of two consecutive trust edges to form a transitive trust path. The trust relationships of Fig.1 is expressed as:

$$([A, D]) = ([A, B] : [B, D]) . \tag{1}$$

Let us now turn to the combination of parallel trust paths, as illustrated in Fig.2. We will use the symbol "⋄" to denote the graph connector for this purpose. The "⋄" symbol visually resembles a simple graph of two parallel paths between a pair of agents, so that it is natural to use it for this purpose. In short notation, Alice's combination of the two parallel trust paths from her to David in Fig.2 is then expressed as:

$$([A, D]) = (([A, B] : [B, D]) \diamond ([A, C] : [C, D])) \tag{2}$$

5   Trust Graph Simplification

Trust networks can have dependent paths. Subjective logic requires trust graphs to be expressed in a canonical form that has no dependent paths. An example of graph simplification is illustrated in Fig.3.
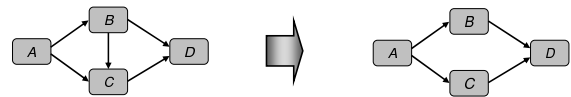

Figure 3. Network simplification by removing weakest path

The expression for the dependent graph on the left-hand side of Fig.3 would be:

$$([A,D]) = \quad (([A,B]:[B,D])$$
$$\diamond([A,C]:[C,D]) \qquad (3)$$
$$\diamond([A,B]:[B,C]:[C,D]))$$

The problem with Eq.(3) is that the edges $[A,B]$ and $[C,D]$ appear twice. TNA-SL requires graphs to be expressed in a form where an edge only appears once. This will be called a *canonical expression*, which is defined as follows:

Definition 1 (Canonical Expression) *An expression of a trust graph in structured notation where every edge only appears once is called canonical.*

A method for canonicalisation based on network simplification was described in [6, 7]. Simplification consists of removing the weakest, i.e. the least certain paths, until the network becomes a directed series-parallel graph which can be expressed on a canonical form.

Assuming that the path $([A,B]:[B,C]:[C,D])$ is the weakest path in the graph on the left-hand side of Fig.2, network simplification of the dependent graph would be to remove the edge $[B,C]$ from the graph, as illustrated on the right-hand side of Fig.3. Since the simplified graph is equal to that of Fig.2, the formal expression is the same as Eq.(2).

An alternative canonicalisation method called *edge splitting* will be described in Sec.7. The next section describes subjective logic operators for analysing trust networks.

6 Trust Computation with Subjective Logic

Subjective logic is suitable for analysing trust networks because trust relationships can be expressed as subjective opinions with degrees of uncertainty. TNA-SL requires trust relationships to be expressed as beliefs, and trust networks to be expressed as a DSPG (Directed Series-Parallel Graph) in the form of canonical expressions.

6.1 Subjective Logic Fundamentals

Subjective logic [3] is probabilistic logic that use opinions as input and output variables. Opinions explicitly express uncertainty about probability values, and can express degrees of ignorance about a subject matter such as trust. An opinion is denoted by ω_x^A which expresses A's belief in the truth of proposition x. Alternatively, an opinion can focus on an entity X expressed as ω_X^A which can be interpreted as *"Party A believes that party X is honest and reliable regarding a specific scope"*, which can be interpreted as A's trust in X within the given scope.

Binomial opinions are expressed as $\omega = (b,d,u,a)$ where b, d, and u represent belief, disbelief and uncertainty respectively, under the constraint that $b,d,u \in [0,1]$ and $b+d+u = 1$. The parameter $a \in [0,1]$ is called the base rate, and is used for computing an opinion's probability expectation value that can be determined as $E(\omega_x^A) = b + au$. More precisely, a determines how uncertainty shall contribute to the probability expectation value $E(\omega_x^A)$. In the absence of any specific evidence about a given party, the base rate determines the default trust.

The opinion space can be mapped into the interior of an equal-sided triangle as illustrated in Fig.4 where the three parameters b, d and u determine the position of the opinion point in the triangle, with $\omega_x = (0.7, 0.1, 0.2, 0.5)$ as an example.
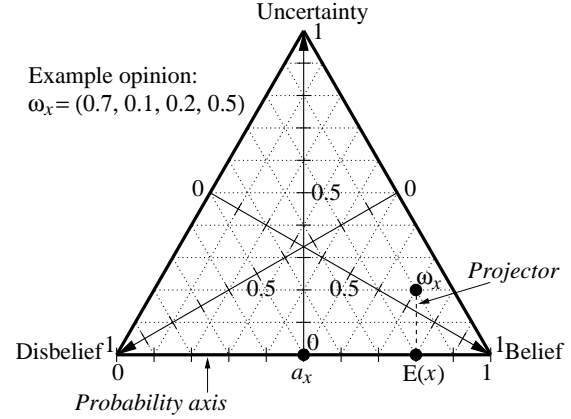

Figure 4. Opinion triangle with example

The base rate a_x is indicated by a point on the probability axis, and the projector starting from the opinion point is parallel to the line that joins the uncertainty vertex and the base rate point on the probability axis. The point at which the projector meets the probability axis determines the expectation value of the opinion, i.e. it coincides with the point corresponding to expectation value $E(\omega_x^A)$.

The probability density over binary event spaces can be expressed as Beta PDFs (probability density functions) denoted by Beta (α, β) [2]. Let r and s express the number of past observations of x and $\bar{x}$ respectively, and let a express the *a priori* or base rate, then α and β can be determined as:

$$\alpha = r + 2a, \qquad \beta = s + 2(1-a). \qquad (4)$$

The following bijective mapping between the opinion parameters and the Beta PDF parameters can be determined analytically [3].

$$\begin{cases} b = r/(r+s+2) \\ d = s/(r+s+2) \\ u = 2/(r+s+2) \\ a = \text{base rate of } x \end{cases} \iff \begin{cases} r = 2b/u \\ s = 2d/u \\ 1 = b+d+u \\ a = \text{base rate of } x \end{cases} \qquad (5)$$

This means for example that a totally ignorant opinion with $u = 1$ and $a = 0.5$ is equivalent to the uniform PDF Beta$(1,1)$. It also means that a dogmatic opinion with $u = 0$ is equivalent to a spike PDF with infinitesimal width and infinite height expressed by Beta$(b\eta, d\eta)$, where $\eta \to \infty$. Dogmatic opinions can thus be interpreted as being based on an infinite amount of evidence.

After r observations of x and s observations of $\overline{x}$ with base rate $a = 0.5$, the *a posteriori* distribution is the Beta PDF with $\alpha = r + 1$ and $\beta = s + 1$. For example the Beta PDF after observing x 7 times and observing $\overline{x}$ once is illustrated in Fig.5, which also is equivalent to the opinion illustrated in Fig.4
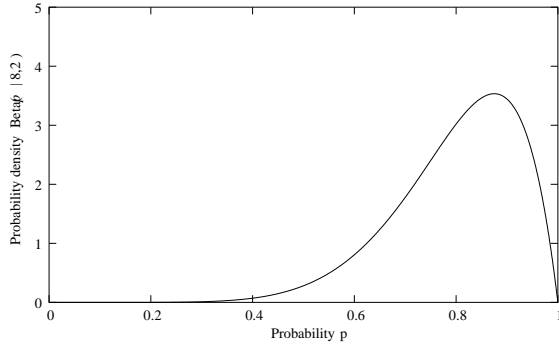

Figure 5. *A posteriori* **Beta(8,2) after 7 observations of** x **and 1 observation of** $\overline{x}$

A PDF of this type expresses the uncertain probability that a process will produce positive outcome during future observations. The probability expectation value of Fig.5 is $\mathrm{E}(p) = 0.8$.

6.2 Operators for Trust Network Analysis

Subjective logic defines a number of operators[3, 8]. Some operators represent generalisations of binary logic and probability calculus operators, whereas others are unique to belief theory because they depend on belief ownership. This presentation focuses on the *transitivity* (also called discounting) and the *fusion* (also called consensus) operators that are used for trust graph analysis. The transitivity operator can be used to derive trust from a trust path consisting of a chain of trust edges, and the fusion operator can be used to combine trust from parallel trust paths. These operators are described below.

- **Transitivity** is used to compute trust along a chain of trust edges. Assume two agents A and B where A trusts B, denoted by ω_B^A, for the purpose of judging the trustworthiness of C. In addition B has trust in C, denoted by ω_C^B. Agent A can then derive her trust in C by discounting B's trust in C with A's trust in B, denoted by

$\omega_C^{A:B}$. The symbol '$\otimes$' is used to designate this operator. The uncertainty favouring version of the transitivity operator [9] is expressed as:

$$\omega_C^{A:B} = \omega_B^A \otimes \omega_C^B$$

$$\begin{cases} b_C^{A:B} = b_B^A b_C^B \\ d_C^{A:B} = b_B^A d_C^B \\ u_C^{A:B} = d_B^A + u_B^A + b_B^A u_C^B \\ a_C^{A:B} = a_C^B . \end{cases} \quad (6)$$

The effect of transitivity discounting in a transitive chain is that uncertainty increases, not disbelief.

- **Cumulative Fusion** is equivalent to Bayesian updating in statistics. The cumulative fusion of two possibly conflicting opinions is an opinion that reflects both opinions in a fair and equal way. Let ω_C^A and ω_C^B be A's and B's trust in C respectively. The opinion $\omega_C^{A\diamond B}$ is then called the fused trust between ω_C^A and ω_C^B, denoting an imaginary agent (A,B)'s trust in C, as if she represented both A and B. The symbol '$\oplus$' is used to designate this operator. The cumulative fusion operator $\omega_C^{A\diamond B} = \omega_C^A \oplus \omega_C^B$ is expressed as:

$$\omega_C^{A\diamond B} = \omega_C^A \oplus \omega_C^B$$

$$\begin{cases} b_C^{A\diamond B} = (b_C^A u_C^B + b_C^B u_C^A)/(u_C^A + u_C^B - u_C^A u_C^B) \\ d_C^{A\diamond B} = (d_C^A u_C^B + d_C^B u_C^A)/(u_C^A + u_C^B - u_C^A u_C^B) \\ u_C^{A\diamond B} = (u_C^A u_C^B)/(u_C^A + u_C^B - u_C^A u_C^B) \\ a_C^{A\diamond B} = a_C^A \end{cases}$$

$$(7)$$

where it is assumed that $a_C^A = a_C^B$. Limits can be computed [4] for $u_C^A = u_C^B = 0$. The effect of the cumulative fusion operator is to amplify belief and disbelief and reduce uncertainty.

7 Trust Network Canonicalisation by Edge Splitting

The existence of a dependent edge in a graph is recognised by multiple instances of the same edge in the trust network expression. Edge splitting is a new approach to achieving independent trust edges. This is achieved by splitting a given dependent edge into as many different edges as there are different instances of the same edge in the trust network expression. Edge splitting is achieved by splitting one of the nodes in the dependent edge into different nodes so that each independent edge is connected to a different node.

A general directed trust graph is based on directed trust edges between pairs of nodes. It is desirable not to put

any restrictions on the possible trust edges except that they should not be cyclic. This means that the set of possible trust paths from a given source X to a given target Y can contain dependent paths. The left-hand side of Fig.6 shows an example of a trust network with dependent paths.
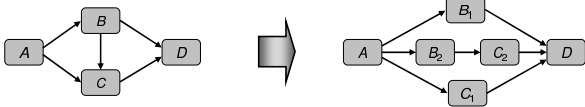

Figure 6. Edge splitting of trust network to produce independent paths

The non-canonical expression for the left-hand side trust network of Fig.6 is:

$$[A,D] = \begin{array}{l} ([A,B]:[B,D]) \\ \diamond([A,C]:[C,D]) \\ \diamond([A,B]:[B,C]:[C,D]) \end{array} \qquad (8)$$

In this expression the edges $[A,B]$ and $[C,D]$ appear twice. Edge splitting in this example consists of splitting the node B into B_1 and B_2, and the node C into C_1 and C_2. This produces the right-hand side trust network in Fig.6 with canonical expression:

$$[A,D] = \begin{array}{l} ([A,B_1]:[B_1,D]) \\ \diamond([A,C_1]:[C_1,D]) \\ \diamond([A,B_2]:[B_2,C_2]:[C_2,D]) \end{array} \qquad (9)$$

Edge splitting must be translated into opinion splitting in order to apply subjective logic. The principle for opinions splitting will be to separate the opinion on the dependent edge into two independent opinions that when cumulatively fused produce the original opinion. This can be called fission of opinions, and will depend on a fission factor ϕ that determines the proportion of evidence assigned to each independent opinion part. The mapping of an opinion $\omega = (b,d,u,a)$ to Beta evidence parameters Beta(r,s,a) according to Eq.(5), and linear splitting into two parts Beta(r_1,s_1,a_1) and Beta(r_2,s_2,a_2) as a function of the fission factor ϕ is:

$$\text{Beta}(r_1,s_1,a_1) : \begin{cases} r_1 = \frac{\phi 2b}{u} \\ \\ s_1 = \frac{\phi 2d}{u} \\ \\ a_1 = a \end{cases} \qquad (10)$$

$$\text{Beta}(r_2,s_2,a_2) : \begin{cases} r_2 = \frac{(1-\phi)2b}{u} \\ \\ s_2 = \frac{(1-\phi)2d}{u} \\ \\ a_2 = a \end{cases} \qquad (11)$$

The reverse mapping of these evidence parameters into two separate opinions according to Eq.(5) produces:

$$\omega_1 : \begin{cases} b_1 = \frac{\phi b}{\phi(b+d)+u} \\ \\ d_1 = \frac{\phi d}{\phi(b+d)+u} \\ \\ u_1 = \frac{u}{\phi(b+d)+u} \\ \\ a_1 = a \end{cases} \qquad (12)$$

$$\omega_2 : \begin{cases} b_2 = \frac{(1-\phi)b}{(1-\phi)(b+d)+u} \\ \\ d_2 = \frac{(1-\phi)d}{(1-\phi)(b+d)+u} \\ \\ u_2 = \frac{u}{(1-\phi)(b+d)+u} \\ \\ a_2 = a \end{cases} \qquad (13)$$

It can be verified that $\omega_1 \oplus \omega_2 = \omega$, as expected.

When deriving trust values from the canonicalised trust network of Eq.(8) we are interested in knowing its certainty level as compared with a simplified network, as described in [6].

We are interested in the expression for the uncertainty of ω_D^A corresponding to trust expression of Eq.(9). Since the edge splitting introduces parameters for splitting opinions, the uncertainty will be a function of these parameters. By using Eq.(6) the expressions for the uncertainty in the trust paths of Eq.(9) can be derived as:

$$\begin{array}{ll} u_D^{A:B_1} & = d_{B_1}^A + u_{B_1}^A + b_{B_1}^A u_D^{B_1} \\ u_D^{A:C_1} & = d_{C_1}^A + u_{C_1}^A + b_{C_1}^A u_D^{C_1} \\ u_D^{A:B_2:C_2} & = b_{B_2}^A d_{C_2}^{B_2} + d_{B_2}^A + u_{B_2}^A \\ & + b_{B_2}^A u_D^{B_2} + b_{B_2}^A b_{C_2}^{B_2} u_D^{C_2} \end{array} \qquad (14)$$

By using Eq.(7) and Eq.(14), the expression for the uncertainty in the trust network of Eq.(9) can be derived as:

$$u_D^A =$$
$$\frac{u_D^{A:B_1} u_D^{A:C_1} u_D^{A:B_2:C_2}}{u_D^{A:B_1} u_D^{A:C_1} + u_D^{A:B_1} u_D^{A:B_2:C_2} + u_D^{A:C_1} u_D^{A:B_2:C_2} - 2u_D^{A:B_1} u_D^{A:C_1} u_D^{A:B_2:C_2}} \qquad (15)$$

By using Eq.(12), Eq.(14) and Eq.(15), the uncertainty value of the derived trust ω_D^A according to the edge splitting principle can be computed. This value depends on the edge opinions and on the two splitting parameters ϕ_B^A and ϕ_D^C.

As an example the opinion values will be set to:

$$\omega_B^A = \omega_D^B = \omega_C^A = \omega_D^C = \omega_C^B = (0.9,0.0,0.1,0.5) \qquad (16)$$

The computed trust values for the two possible simplified graphs are:

$$(\omega_B^A \otimes \omega_D^B) \oplus (\omega_C^A \otimes \omega_D^C) \quad = (0.895,0.0,0.105,0.5) \quad (17)$$
$$\omega_B^A \otimes \omega_C^B \otimes \omega_D^C \quad = (0.729,0.0,0.271,0.5) \quad (18)$$

The uncertainty level u_D^A when combining these two graphs through edge splitting as a function of ϕ_B^A and ϕ_D^C is shown in Fig.7
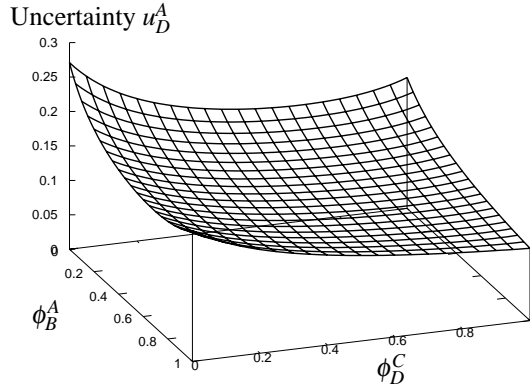

Figure 7. Uncertainty u_D^A as a function of ϕ_B^A and ϕ_D^C

The conclusion which can be drawn from this is that the optimal value for the splitting parameters are $\phi_B^A = \phi_D^C = 1$ because that is when the uncertainty is at its lowest. In fact the uncertainty can be evaluated to $u_D^A = 0.105$. This is equivalent to the trust network simplification of Eq.(17) where the edge $[B,C]$ is completely removed from the left-hand side graph of Fig.6.

The least optimal values for the splitting parameters is when $\phi_B^A = \phi_D^C = 0$, resulting in $u_D^A = 0.271$. This is equivalent to the absurd trust network simplification of Eq.(18) where the edges $[A,C]$ and $[B,D]$, and thereby the most certain trust paths are completely removed from the left-hand side graph of Fig.6.

Given the edge opinion values used in this example, $([A,B]:[B,C]:[C,D])$ is the least certain path of the left-hand side graph of Fig.6. It turns out that the optimal splitting parameters for analysing the right-hand side graph of Fig.6 produces the same result as network simplification where this particular least certain path is removed.

8   Discussion and Conclusion

We have described edge splitting as a new principle for trust network analysis with subjective logic. This method consists of splitting dependent trust edge opinions in order to avoid dependent paths, which can be considered optimal because it does not cause any loss of information. Our analysis and simulation have shown that edge splitting produces the same result as the previously described TNA-SL method of network simplification. This shows that network simplification with TNA-SL in fact does produce optimal results.

Our analysis was based on a fixed set of edge opinion values. Because of the large number of parameters involved, it is a relatively complex task to verify if our conclusion is valid for all possible trust edge opinion values, so a complete study must be the subject of future work. The present study has given a strong indication that trust network simplification produces optimal results even though edges are removed from the trust graph.

References

[1] B. Christianson and W. S. Harbison. Why Isn't Trust Transitive? In *Proceedings of the Security Protocols International Workshop*. University of Cambridge, 1996.

[2] M.H. DeGroot and M.J. Schervish. *Probability and Statistics (3rd Edition)*. Addison-Wesley, 2001.

[3] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[4] A. Jøsang. Probabilistic Logic Under Uncertainty. In *The Proceedings of Computing: The Australian Theory Symposium (CATS2007), CRPIT Volume 65*, Ballarat, Australia, January 2007.

[5] A. Jøsang, T. Bhuiyan, Y. Xu, and C. Cox. Combining Trust and Reputation Management for Web-Based Services. In *The Proceedings of the 5th International Conference on Trust, Privacy & Security in Digital Business (TrustBus2008)*, Turin, September 2008.

[6] A. Jøsang, E. Gray, and M. Kinateder. Simplification and Analysis of Transitive Trust Networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[7] A. Jøsang, R. Hayward, and S. Pope. Trust Network Analysis with Subjective Logic. In *Proceedings of the 29th Australasian Computer Science Conference (ACSC2006), CRPIT Volume 48*, Hobart, Australia, January 2006.

[8] A. Jøsang, S. Pope, and M. Daniel. Conditional deduction under uncertainty. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, 2005.

[9] A. Jøsang, S. Pope, and S. Marsh. Exploring Different Types of Trust Propagation. In *Proceedings of the 4th International Conference on Trust Management (iTrust)*, Pisa, May 2006.