Information Systems Frontiers 7:4/5, 337–358, 2005

Propagation Models for Trust and Distrust in Social Networks

Cai-Nicolas Ziegler and Georg Lausen
Institut für Informatik, Universität Freiburg, Georges-Köhler-Allee, Gebäude 51, 79110 Freiburg i.Br., Germany
E-mail: cziegler@informatik.uni-freiburg.de; lausen@informatik.uni-freiburg.de

Abstract. *Semantic Web endeavors have mainly focused on issues pertaining to knowledge representation and ontology design. However, besides understanding information metadata stated by subjects, knowing about their credibility becomes equally crucial. Hence, trust and trust metrics, conceived as computational means to evaluate trust relationships between individuals, come into play. Our major contribution to Semantic Web trust management through this work is twofold. First, we introduce a classification scheme for trust metrics along various axes and discuss advantages and drawbacks of existing approaches for Semantic Web scenarios. Hereby, we devise an advocacy for local group trust metrics, guiding us to the second part which presents Appleseed, our novel proposal for local group trust computation. Compelling in its simplicity, Appleseed borrows many ideas from spreading activation models in psychology and relates their concepts to trust evaluation in an intuitive fashion. Moreover, we provide extensions for the Appleseed nucleus that make our trust metric handle distrust statements.*

Key Words. *trust, semantic web, trust propagation, spreading activation models, balance theory, web of trust*

1. Introduction

In our world of information overload and global connectivity leveraged through the Web and other types of media, social trust (McKnight and Chervany, 1996) between individuals becomes an invaluable and precious good. Hereby, trust exerts an enormous impact on decisions whether to believe or disbelieve information asserted by other peers. Belief should only be accorded to statements from people we deem trustworthy. Hence, trust assumes the role of an instrument for complexity reduction (Luhmann, 1998). However, when supposing huge networks such as the Semantic Web, trust judgements based on personal experience and acquaintanceship become unfeasible. In general, we accord trust, concisely defined by Mui as the "subjective expectation an agent has about another's future behavior based on the history of their encounters" (Mui, Mohtashemi and Halberstadt, 2002), to only small numbers of people. These people, again, trust another limited set of people, and so forth. The network structure emanating from our very person, composed of trust statements linking individuals, constitutes the basis for trusting people we do not know personally. Playing an important role for the conception of Semantic Web trust infrastructure, the latter structure has been dubbed "Web of Trust" (Golbeck, Parsia and Hendler, 2003).

Its effectiveness has been underpinned through empirical evidence from social psychology and sociology, indicating that transitivity is an important characteristic of social networks (Holland and Leinhardt, 1972; Rapoport, 1963). To the extent that communication between individuals becomes motivated through positive affect, drive towards transitivity can also be explained in terms of Heider's famous "balance theory" (Heider, 1958), i.e., individuals are more prone to interact with friends of friends than unknown peers.

Hence, we might be tempted to adopt the policy of trusting all those people who are trusted by persons we trust, exploiting transitivity in social networks. Trust would thus propagate through the network and become accorded whenever two individuals can reach each other via at least one trust path. However, common sense tells us we should not rely upon this strategy. More complex metrics are needed in order to more sensibly evaluate trust between two persons. Among other features, these metrics must take into account subtle social and psychological aspects of trust and suffice criteria of computability and scalability, likewise.

The paper is organized as follows. In order to assess diverse properties of metrics, Section 2.1 briefly

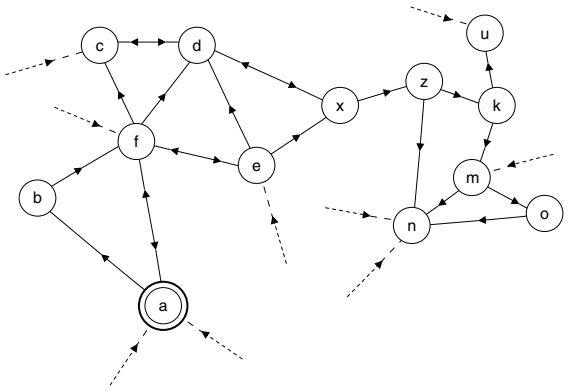

Fig. 1. *Sample web of trust for agent a.*

introduces existing trust metrics and classifies them according to our proposed classification scheme. An investigation of trust metric classes and their fitness for Semantic Web scenarios follows in Section 2.2, along with an overview of our asserted trust model. Besides, Section 2.2 exposes the urging need for *local group* trust metrics and gives examples of possible application scenarios. Section 3 forms the second part of the paper and explicitly deals with these local group trust metrics. We briefly sketch the well-known Advogato trust metric and introduce our novel Appleseed trust metric in Section 3.2. Appleseed constitutes the major contribution of this paper and represents our own approach to local group trust computation. Many of its ideas and concepts borrow from spreading activation models, which simulate human semantic memory. Section 3.3 matches Appleseed and Advogato against each other, discussing advantages and drawbacks of either approach. Furthermore, results of experiments conducted to evaluate the behavior of Appleseed under diverse conditions are illustrated in Section 3.4. Section 3.5 indicates possible modifications and gives some implementation details, while Section 3.6 briefly presents the testbed we used to base all our experiments and comparisons upon. Eventually, in Section 4.1, semantics and implications of distrust are discussed, followed by the integration of distrust into the Appleseed framework in Section 4.2.

2. *Trust in Social Networks*

Trust represents an invaluable and precious good one should award deliberately. Trust metrics compute quantitative *estimates* of how much trust an agent *a* should accord to its peer *b*, taking into account trust ratings from other persons on the network. These metrics should also act "deliberately", not overly awarding trust to persons or agents whose trustworthiness is questionable.

2.1. Classification of trust metrics

Applications for trust metrics and trust management (Blaze, Feigenbaum and Lacy, 1996) are rife and not confined to the Semantic Web. First proposals for metrics date back to the early nineties, where trust metrics were deployed in various projects to support the Public Key Infrastructure (Zimmermann, 1995). Metrics proposed in Levien and Aiken (1998), Reiter and Stubblebine (1997), Maurer (1996), and Beth, Borcherding and Klein (1994) count among the most popular ones for public key authentication and have initiated fruitful discussions. New areas and research fields apart from PKI have come to make trust metrics gain momentum. Peer-to-peer networks, ubiquitous and mobile computing, and rating systems for online communities, where maintenance of explicit certification authorities is not feasible anymore, have raised the research interest in trust. The whole plethora of available metrics can hereby be defined and characterized along various classification axes. We identify three principal dimensions with distinctive features. These axes are not orthogonal, though, for various features impose restrictions on the feature range of other dimensions. Mind that some of the below mentioned categories have already been defined in prior work. For instance, Guha (2003) differentiates between local and global trust, and distinctive features between scalar and group trust metrics are discussed in Levien (2003). However, to our knowledge, no explicit categorization of trust metrics along various axes, supplemented with an analysis of axis interaction, exists. We therefore regard the classification scheme provided below as one major contribution of this paper. Its results are also synthesized in Fig. 2.

2.1.1. Network perspective. The first dimension influences semantics assigned to the values computed. Trust metrics may basically be subdivided into ones with *global*, and ones with *local* scope. Global trust metrics take into account *all* peers and trust links connecting them. Global trust ranks are assigned to an individual based upon complete trust graph information. Many global trust metrics, such as those presented

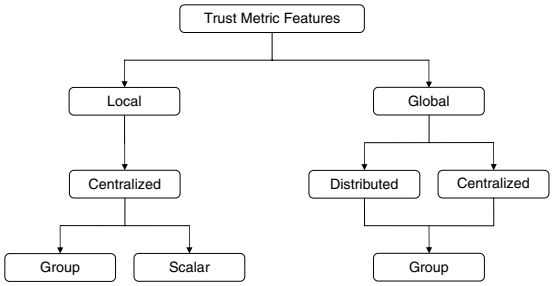

Fig. 2. *Trust metric classification.*

in Kamvar, Schlosser and Garcia-Molina (2003), Guha (2003), and Richardson, Agrawal and Domingos (2003), borrow their ideas from the renowned PageRank algorithm Page et al. (1998) to compute web page reputation. The basic intuition behind the approach is that nodes should be ranked higher the better the rank of nodes pointing to them. Obviously, the latter approach works for trust and page reputation likewise.

Trust metrics with local scope, on the other hand, take into account personal bias. Interestingly, some researchers claim that only local trust metrics are "true" trust metrics, since global ones compute overall reputation rather than personalized trust (Mui, Mohtashemi and Halberstadt, 2002). Local trust metrics take the agent for whom to compute trust as an additional input parameter and are able to operate on *partial* trust graph information. The rationale behind local trust metrics is that persons an agent a trusts may be completely different from the range of individuals that agent b deems trustworthy. Local trust metrics exploit structural information defined by personalized webs of trust. Hereby, the personal web of trust for individual a is given through the set of trust relationships emanating from a and passing through nodes it trusts either directly or indirectly, as well as the set of nodes reachable through these relationships. Merging all webs of trust engenders the global trust graph. Local trust metrics comprise Levien's Advogato trust metric (Levien and Aiken, 2000), metrics for modelling the Public Key Infrastructure (Beth, Borcherding and Klein, 1994; Maurer, 1996; Reiter and Stubblebine, 1997), Golbeck's metrics for Semantic Web trust (Golbeck, Parsia and Hendler, 2003), and Sun Microsystems's Poblano (Chen and Yeager, 2003). The latter work hereby strongly resembles Abdul-Rahman and Hailes Abdul-Rahman and Hailes (1997).

2.1.2. Computation locus. The second axis refers to the place where trust relationships between individuals are evaluated and quantified. Local or centralized approaches perform all computations in one single machine and hence need to be granted full access to relevant trust information. The trust data itself may hereby be distributed over the network. Most of the beforementioned metrics count among the class of centralized approaches.

Distributed metrics for the computation of trust and reputation, such as those described in Richardson, Agrawal and Domingos (2003), Kamvar, Schlosser and Garcia-Molina (2003), and Sankaralingam, Sethumadhavan and Browne (2003), equally deploy the load of computation on every trust node in the network. Upon receiving trust information from its predecessor nodes in the trust graph, an agent a merges the data with its own trust assertions and propagates synthesized values to its successor nodes. The entire process of trust computation is necessarily asynchronous and its convergence depends on the eagerness or laziness of nodes to propagate information. Another characteristic feature of distributed trust metrics refers to the fact that they are inherently global. Though the individual computation load is decreased with respect to centralized computation approaches, nodes need to store trust information about *any other* node in the system.

2.1.3. Link evaluation. The third dimension distinguishes scalar and group trust metrics. According to Levien Levien (2003), scalar metrics analyze trust assertions independently, while group trust metrics evaluate groups of assertions "in tandem". PageRank Page et al. (1998) and related approaches count among global group trust metrics, for the reputation of one page depends on the ranks of referring pages, thus entailing parallel evaluation of relevant nodes thanks to mutual dependencies. Advogato (Levien and Aiken, 2000) represents an example for local group trust metrics. Most other trust metrics count among the category of scalar ones, tracking trust paths from sources to targets and not performing parallel evaluations of groups of trust assertions. Hence, another basic difference between scalar and group trust metrics refers to their functional design. In general, scalar metrics compute trust between two given individuals a and b taken from set V of all agents.

On the other hand, group trust metrics generally compute trust ranks for *sets* of individuals in V. Hereby, global group trust metrics assign trust ranks for every

$a \in V$, while local ones may also return ranked subsets of V. Note that complete trust graph information is only important for *global* group trust metrics, but not for *local* ones. Informally, local group trust metrics may be defined as metrics to compute *neighborhoods* of trusted peers for an individual a. As input parameters, these trust metrics take an individual $a \in V$ for which to compute the set of peers it should trust, as well as an amount of trust the latter wants to share among the most trustworthy agents. For instance, in Levien and Aiken (2000), the amount of trust is said to correspond to the number of agents that a wants to trust. The output is hence given by a trusted subset of V.

Note that scalar trust metrics are inherently local, while group trust metrics do not impose any restrictions on features for other axes.

2.2. Semantic web trust

Most presented metrics and trust models have been proposed for scenarios other than the Semantic Web. In fact, research in trust infrastructure and metrics for the latter network of metadata still has to come of age and gain momentum. Before discussing specific requirements and fitness properties of trust metrics along those axes proposed before, we need to define one common trust model on which to rely upon for the Semantic Web. Some steps towards one such common model have already been taken and incorporated into the FOAF (Dumbill, 2002) project. FOAF is an abbreviation for "Friend of a Friend" and aims at enriching personal homepages with machine-readable content encoded in RDF statements. Besides various other information, these publicly accessible pages allow their owners to nominate all individuals part of the FOAF universe they know, thus weaving a "web of acquaintances" (Golbeck, Parsia and Hendler, 2003). Golbeck has extended the FOAF schema to also contain *trust* assertions with values ranging from 1 to 9, where 1 denotes complete distrust and 9 absolute trust towards the individual for which the assertion has been issued (Golbeck, Parsia and Hendler, 2003). Hereby, her assumption that trust and distrust represent symmetrically opposed concepts perfectly aligns with Abdul-Rahman and Hailes's work (Abdul-Rahman and Hailes, 2000).

The model that we adopt is quite similar to FOAF and its extensions, but only captures the notion of trust and lack of trust, instead of trust and distrust. Note that zero trust and distrust are *not* the same (Marsh, 1994a) and may hence not be intermingled. Explicit modelling of distrust has some serious implications for trust metrics and will hence be discussed separately in Section 4. Mind that only few research endeavors investigated the implementation of distrust into trust models, e.g., Jøsang, Gray, and Kinateder (2003) and Guha (2003), Guha, Raghavan, and Tomkins (2004).

2.2.1. Trust model. In this section, we present the constituents of our model for the Semantic Web trust infrastructure. As is the case for FOAF, we assume that all trust information is publicly accessible for any agent in the system through machine-readable personal homepages distributed over the network. This assumption may yield privacy concerns and will be discussed and justified later.

- *Agent set* $V = \{a_1, \ldots, a_n\}$. Similar to the FOAF approach, we assume agents $a \in V$ to be represented and uniquely identified by the URI of their machine-readable personal homepage.
- *Partial trust function set* $T = \{W_{a_1}, \ldots, W_{a_n}\}$. Every agent a is associated with one partial trust function $W_a : V \rightarrow [0, 1]^{\perp}$, which corresponds to the set of trust assertions that a has stated on its machine-readable homepage. In most cases, these functions will be very sparse as the number of individuals for which an agent is able to assign explicit trust ratings is much smaller than the total number n of agents on the Semantic Web:

$$W_{a_i}(a_j) = \begin{cases} p, & \text{if trust}(a_i, a_j) = p \\ \perp, & \text{if no rating for } a_j \text{ from } a_i \end{cases}$$

Note that the higher the value of $W_{a_i}(a_j)$, the more trustworthy a_i deems a_j. Conversely, $W_{a_i}(a_j) = 0$ means that a_i considers a_j to be not trustworthy at all. The assignment of trust through continuous values between 0 and 1 and their adopted semantics is in perfect accordance with (Marsh, 1994a), where possible stratifications of trust values are proposed. Our trust model defines one directed trust graph with nodes being represented by agents $a \in V$ and directed edges from nodes a_i to nodes a_j being trust statements with weight $W_{a_i}(a_j)$.

For convenience, we furthermore introduce the partial function $W : V \times V \rightarrow [0, 1]^{\perp}$ which we define as the union of all partial functions $W_a \in T$.

2.2.2. Trust metrics for the semantic web. Trust and reputation ranking metrics have primarily been

used for public key certification (Reiter and Stubblebine, 1996,1997; Levien and Aiken, 1998; Maurer, 1996; Beth, Borcherding and Klein, 1994), rating and reputation systems part of online communities (Guha, 2003; Levien and Aiken, 2000; Levien, 2003), peer-to-peer networks (Kamvar, Schlosser and Garcia-Molina, 2003; Sankaralingam, Sethumadhavan and Browne, 2003; Kinateder and Rothermel, 2003; Kinateder and Pearson, 2003; Aberer and Despotovic, 2001), and also mobile computing fields (Eschenauer, Gligor and Baras, 2002). Each of these scenarios favors different trust metrics. For instance, reputation systems for online communities tend to make use of centralized trust servers that compute global trust values for all users on the system (Guha, 2003). On the other hand, peer-to-peer networks of moderate size rely upon distributed approaches that are in most cases based upon PageRank (Kamvar, Schlosser and Garcia-Molina, 2003; Sankaralingam, Sethumadhavan and Browne, 2003).

The Semantic Web, however, is expected to be made up of millions of nodes a representing agents. The fitness of *distributed* approaches to trust metric computation, such as depicted in Richardson, Agrawal and Domingos (2003) and Kamvar, Schlosser and Garcia-Molina (2003), is hence limited by virtue of various reasons:

- *Trust data storage.* Each agent a needs to store trust information about any other agent b on the Semantic Web. Agent a uses this information in order to merge it with own trust beliefs and propagates the synthesized information to trusted agents. Even though we might expect the size of the Semantic Web to be several orders of magnitude smaller than the traditional Web, the number of agents which to keep trust information for will still exceed storage capabilities of "normal" agents.
- *Convergence.* The structure of the Semantic Web is diffuse and not subject to some higher ordering principle or hierarchy. Furthermore, the process of trust propagation is necessarily asynchronous. As the Semantic Web is huge in size with possibly numerous antagonist or idle agents, convergence of trust values might take a very long time.

The huge advantage of distributed approaches, on the other hand, is the immediate availability of computed trust information for any other agent in the system as well as the fact that agents have to disclose their trust assertions only to peers they trust (Richardson, Agrawal and Domingos, 2003). For instance, suppose that a declares its trust in b to be 0.1, which is very low. Hence, a might want b not to know about that fact. As distributed metrics only propagate synthesized trust values from nodes to successor nodes in the trust graph, a would not have to disclose its trust statements to b.

As it comes to centralized, i.e., locally computed, metrics, full trust information access is required for agents inferring trust. Hence, online communities based on trust require their users to disclose all trust information to the community server, but not necessarily to other peers (Guha, 2003). Privacy is thus maintained. On the Semantic Web and in the area of ubiquitous and mobile computing, however, it is not only some central authority which computes trust. Any agent might want to do so. Our own trust model, as well as trust models proposed in Golbeck, Parsia and Hendler (2003), Eschenauer, Gligor and Baras (2002), and Abdul-Rahman and Hailes (1997), are hence based upon the assumption of publicly available trust information. Though privacy concerns may persist, this assumption is vital due to the mentioned deficiencies of distributed computation models. Moreover, centralized *global* metrics, such as depicted in Guha (2003) and Page et al. (1998), also fail to fit the requirements imposed by the Semantic Web: due to the huge number of agents issuing trust statements, only dedicated server clusters could be able to manage the whole bulk of trust relationships. For small agents and applications roaming the Semantic Web, global trust computation is not feasible.

The traditional as well as the Semantic Web bear significant traits of small-world networks (Golbeck, Parsia and Hendler, 2003). Small worlds theory has been investigated extensively by Stanley Milgram, social psychologist at Harvard University. His hypothesis, commonly referred to as "six degrees of separation", states that members of any large social network are connected to each other through short chains of intermediate acquaintances (Gray et al., 2003). Relating his research results to trust on the Semantic Web, we come to conclude that average trust path lengths between any two individuals are small. Hence, locally computed *local* trust metrics considering trust paths from trust sources to trust targets, such as the ones proposed for PKI (Reiter and Stubblebine, 1996,1997; Levien and Aiken, 1998; Maurer, 1996; Beth, Borcherding and Klein, 1994), may be expected to suitably lend themselves to the Semantic Web. In contrast to global metrics, no clustering of massive CPU power is required to compute trust.

Besides centrally computed *scalar* trust metrics taking into account personal bias, we advocate *local group* trust metrics for the Semantic Web. These metrics bear several welcome properties with respect to computability and complexity, which may be summarized as follows:

- *Partial trust graph exploration.* Global metrics require a priori full knowledge of the entire trust network. Distributed metrics store trust values for all agents in the system, thus implying massive data storage demands. On the other hand, when computing trusted *neighborhoods*, the trust network only needs to be explored partially: originating from the trust source, one only follows those trust edges that seem promising, i.e., bearing high trust weights, and which are not too far away from the trust source. Inspection of personal, machine-readable homepages is thus performed in a just-in-time fashion. Hence, prefetching bulk trust information is not required.
- *Computational scalability.* Tightly intertwined with partial trust graph exploration is computational complexity. Local group trust metrics scale well to any social network size, as only tiny subsets of relatively constant size are visited. This is not the case for global trust metrics.

By the time of this writing, local group trust metrics have been subject to comparatively sparse research interest and none, to our best knowledge, have been proposed for the Semantic Web. However, we believe that local group trust metrics will play an important role for trust-based communities on the Semantic Web. Application scenarios for group trust are rife. In order to not go beyond the scope of this article, we will give just one detailed example dealing with trust in metadata statements:

The Semantic Web basically consists of metadata assertions that machines can understand by virtue of ontology sharing. However, since the number of agents able to publish statements is vast, credibility in those statements should be limited. The issue of trust in Semantic Web content has already been addressed in Gil and Ratnakar (2002). Herein, the authors propose a centralized system which allows issuing statements and analyzing their reliability and credibility. Complementary to this work by Gil and Ratnakar, the W3C Annotea Project intends to provide an infrastructure for assigning annotations to statements (Kahan, 2001). These statements could also include statements about the credibility of certain metadata. Supposing such an environment and supposing an agent a who wants to reason about the credibility of an assertion s found on the Semantic Web, local group trust metrics could play an important role in its quest: not being able to judge the credibility of s on its own, a could refer to its personal web of trust and compute its n most trusted peers. The latter trust neighborhood is now taking part in an opinion poll where a wants to know about the credibility its trusted peers assign to s. Technically, this could be achieved by searching Annotea servers for statements by a's peers about s. The eventual decision whether to believe s or not could then be made by averaging the credibility ratings of its trusted peers. Similar models with distributed reputation systems based on trust have been proposed in Kinateder and Rothermel (2003).

3. Local Group Trust Metrics

Local group trust metrics, in their function as means to compute trust neighborhoods, have not been subject to mainstream research until now. Actually, significant research has been limited to the work done by Levien (2003), Levien and Aiken (1998), having conceived the Advogato group trust metric. This section provides an overview of Advogato and introduces our own Appleseed trust metric, eventually comparing both approaches.

3.1. Outline of advogato maxflow

The Advogato maximum flow trust metric has been proposed by Levien and Aiken (2000) in order to discover which users are trusted by members of an online community and which are not. Hereby, trust is computed by a centralized community server and considered relative to a seed of users enjoying supreme trust. However, the metric is not only applicable to community servers, but also to *arbitrary* agents which may compute *personalized* lists of trusted peers and not one single global ranking for the whole community they belong to. In this case, the agent itself constitutes the singleton trust seed. The following paragraphs briefly introduce basic concepts. For more detailed information, refer to Levien and Aiken (2000,1998), and Levien (2003).

3.1.1. Trust computation steps. Local group trust metrics compute sets of agents trusted by those being

part of the trust seed. In case of Advogato, its input is given by an integer number n, which is supposed to be equal to the number of members to trust (Levien and Aiken, 2000), as well as the trust seed s, being a subset of the entire set of users V. The output is a characteristic function that maps each member to a boolean value indicating trustworthiness:

$$\text{Trust}_M : 2^V \times \mathbb{N}_0^+ \to (V \to \{\text{true}, \text{false}\})$$

The trust model underlying Advogato does *not* provide support for weighted trust relationships in its original version. Hence, trust edges extending from individual x to y express blind, i.e., full, trust of x in y. Metrics for PKI maintenance suppose similar models. Maximum integer network flow computation (Ford and Fulkerson, 1962) was investigated by Reiter and Stubblebine (1997), Reiter and Stubblebine (1996) in order to make trust metrics more reliable. Levien adopted and extended this approach for group trust in his Advogato metric.

Capacities $C_V : V \to \mathbb{N}$ are assigned to every community member $x \in V$ based upon the shortest-path distance from the seed to x. Hereby, the capacity of the seed itself is given by the input parameter n mentioned before, whereas the capacity of each successive distance level is equal to the capacity of the previous level l divided by the average outdegree of trust edges $e \in E$ extending from l. The trust graph obtained hence contains one single source, which is the set of seed nodes considered one single "virtual" node, and multiple sinks, i.e., all nodes other than those defining the seed. Capacities $C_V(x)$ constrain nodes. In order to apply Ford-Fulkerson maximum integer network flow (Ford and Fulkerson, 1962), the underlying problem has to be formulated as single-source/single-sink, having capacities $C_E : E \to \mathbb{N}$ constrain edges instead of nodes. Hence, Algorithm 1 is applied to the old directed graph $G = (V, E, C_V)$, resulting in a new graph structure $G' = (V', E', C_{E'})$.

Figure 3 and 4 depicts the outcome of converting node-constrained single-source/multiple-sink graphs into single-source/single-sink ones with capacities constraining edges.

Conversion is followed by simple integer maximum network flow computation from the trust seed to the super-sink. Eventually, trusted agents x are exactly those peers for which there is flow from "negative" nodes x^- to the super-sink. An additional constraint needs to be introduced, requiring flow from x^- to the super-sink whenever there is flow from x^- to x^+. The latter constraint assures that node x does not only serve as an intermediate for the flow to pass through, but is actually added to the list of trusted agents when reached by network flow. However, the standard implementation of Ford-Fulkerson traces shortest paths to the sink first (Ford and Fulkerson, 1962). Therefore, the above constraint is satisfied implicitly already.

```
function transform (G = (V, E, C_V)) {
  set E' ← ∅, V' ← ∅;
  for all x ∈ V do
    add node x⁺ to V';
    add node x⁻ to V';
    if C_V(x) ≥ 1 then
      add edge (x⁻, x⁺) to E';
      set C_E'(x⁻, x⁺) ← C_V(x) − 1;
      for all (x, y) ∈ E do
        add edge (x⁺, y⁻) to E';
        set C_E'(x⁺, y⁻) ← ∞;
      end do
      add edge (x⁻, supersink) to E';
      set C_E'(x⁻, supersink) ← 1;
    end if
  end do
  return G' = (V', E', C_E');
}
```

Algorithm 1. *Trust graph conversion.*

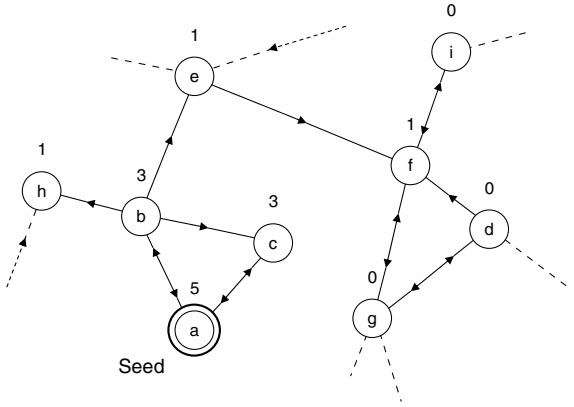

Fig. 3. *Trust graph before conversion for Advogato.*

Example 1 (Advogato trust computation). Suppose the trust graph depicted in Fig. 3 and 4. The only seed node is a with initial capacity $C_V(a) = 5$. Hence, taking into account the outdegree of a, nodes at unit distance from the seed, i.e., nodes b and c, are assigned capacities $C_V(b) = 3$ and $C_V(c) = 3$, respectively. The average outdegree of both nodes is 2.5 so that second level nodes e and h obtain unit capacity. When

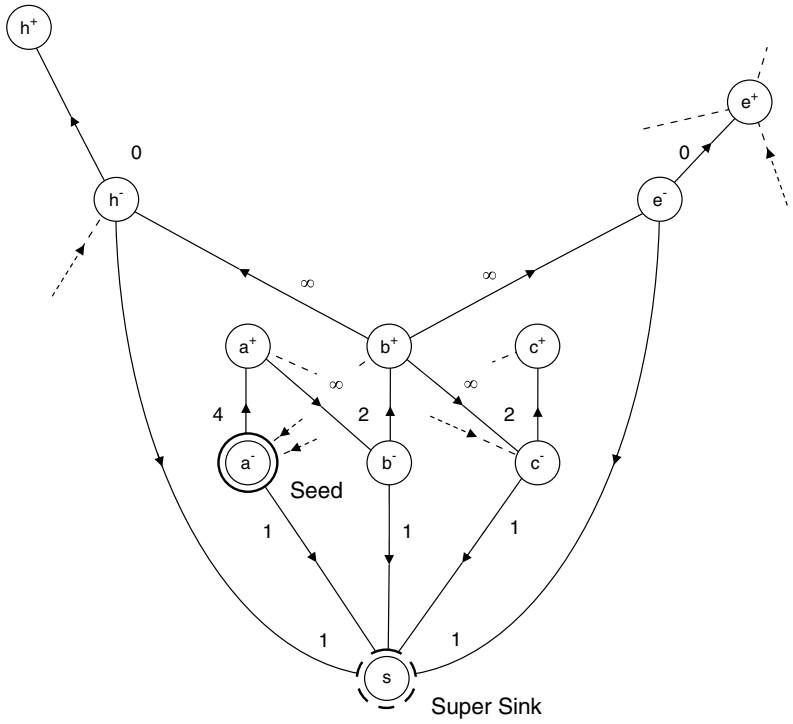

Fig. 4. *Trust graph after conversion for Advogato.*

computing maximum integer network flow, agent a will accept itself, b, c, e, and h as trustworthy peers.

3.1.2. Attack-resistance properties. Advogato has been designed with resistance against massive attacks from malicious agents outside of the community in mind. Therefore, an upper bound for the number of "bad" peers chosen by the metric is provided in Levien and Aiken (2000), along with an informal security proof to underpin its fitness. Resistance against malevolent users trying to break into the community may already be observed in the example depicted by Fig. 1, supposing node n to be "bad": though agent n is trusted by numerous persons, it is deemed less trustworthy than, for instance, x. While there are fewer agents trusting x, these agents enjoy higher trust reputation than the numerous persons trusting n. Hence, it is not just the *number* of agents trusting an individual i, but also the trust *reputation* of these agents that exerts an impact on the trust assigned to i. PageRank (Page et al., 1998) works in a similar fashion and has been claimed to possess similar properties of attack-resistance like the Advogato trust metric (Levien, 2003). In order to make the concept of attack-resistance more tangible, Levien proposes the "bottleneck property" as common feature of attack-resistant trust metrics. Informally, this property states that the "trust quantity accorded to an edge $s \rightarrow t$ is not significantly affected by changes to the successors of t" (Levien, 2003). Moreover, attack-resistance features of various trust metrics are discussed in detail in Levien and Aiken (1998) and Twigg and Dimmock (2003).

3.2. Appleseed trust metric

The Appleseed trust metric constitutes the main contribution of this work and is our novel proposal for local group trust metrics. In contrast to Advogato, being inspired by maximum network flow computation, the basic intuition of Appleseed is motivated by spreading activation models. Spreading activation models have first been proposed by Quillian (1968) in order to simulate human comprehension through semantic memory. They are commonly described as "models of retrieval from long-term memory in which activation subdivides among paths emanating from an activated mental representation" (Smith et al., 2003). By the time of this writing, the seminal work of Quillian has been ported to a whole plethora of other disciplines, such as latent

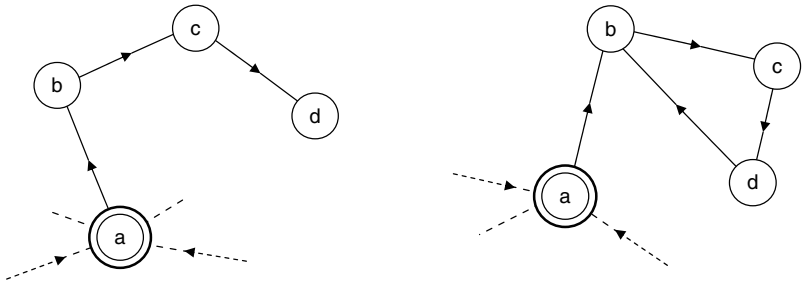

Fig. 5. *Node chains and rank sinks.*

semantic indexing (Ceglowski, Coburn and Cuadrado, 2003) and text illustration (Hartmann and Strothotte, 2002). As an example, we will briefly introduce the spreading activation approach adopted in Ceglowski, Coburn and Cuadrado (2003) for semantic search in contextual network graphs in order to then relate Appleseed to the former work.

3.2.1. Searches in contextual network graphs. The graph model underlying search strategies in contextual network graphs is almost identical in structure to the one presented in Section 2.2.1, i.e., edges $(x, y) \in E \subseteq V \times V$ connecting nodes $x, y \in V$. Edges are assigned continuous weights through $W : E \rightarrow [0, 1]$. Source node s to start the search from is activated through an injection of energy e, which is then propagated to other nodes along edges according to some set of simple rules: all energy is fully divided among successor nodes with respect to their normalized local edge weight, i.e., the higher the weight of an edge $(x, y) \in E$, the higher the portion of energy that flows along that edge. Furthermore, supposing average outdegrees greater than one, the closer node x to the injection source s, and the more paths leading from s to x, the higher the amount of energy flowing into x. To eliminate endless, marginal and negligible flow, energy streaming into node x must exceed threshold T in order not to run dry. The described approach is captured formally by Algorithm 2, which propagates energy recursively.

```
procedure energize (e ∈ ℝ₀⁺, s ∈ V) {
  energy(s) ← energy(s) + e;
  e' ← e / Σ_{(s,n)∈E} W(s,n);
  if e > T then
    ∀(s,n) ∈ E : energize (e' · W(s,n), n);
  end if
}
```

Algorithm 2. *Recursive energy propagation.*

3.2.2. Trust propagation. Algorithm 2 shows the basic intuition behind spreading activation models. In order to tailor these models to trust computation, later to become the Appleseed trust metric, serious adaptations are necessary. For instance, procedure energize(e, s) registers *all* energy e that passed through node x, accumulated in energy(x). Hence, energy(x) represents the *rank* of x. Higher values indicate higher node rank. However, at the same time, all energy contributing to the rank of x is passed *without loss* to its successor nodes. Interpreting energy ranks as trust ranks thus implies numerous issues of semantic consistency as well as computability. Consider the graph depicted on the left-hand side of Fig. 5. Applying spreading activation according to Ceglowski, Coburn and Cuadrado (2003), trust ranks of nodes b and d will be identical. However, common sense tells us that d should be accorded *less* trust than b, since its shortest-path distance to the trust seed is higher. Trust decay is commonly agreed upon (Guha, 2003; Jøsang, Gray, and Kinateder, 2003), for people tend to trust individuals trusted by immediate friends more than individuals trusted only by friends of friends. The right-hand side of Fig. 5 entails even more serious implications. All energy, or trust, respectively, distributed along edge (a, b) becomes trapped in a cycle and will never be accorded to any other nodes but those being part of that cycle, i.e., b, c, and d. These nodes will eventually acquire infinite trust rank. Obviously, the bottleneck property (Levien, 2003) does not hold. Similar issues occur with simplified versions of PageRank (Page et al., 1998), where cycles accumulating infinite rank are dubbed "rank sinks".

3.2.3. Spreading factor. We handle both issues, i.e., trust decay in node chains and the elimination of rank sinks, by tailoring the algorithm to rely upon our global spreading factor d. Hereby, let in(x) denote the energy influx into node x. Parameter d then denotes the portion

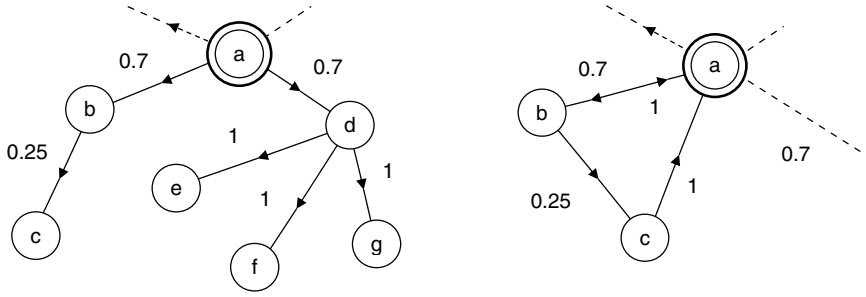

Fig. 6. *Issues with trust normalization.*

of energy $d \cdot \text{in}(x)$ that the latter node distributes among successors, while retaining $(1 - d) \cdot \text{in}(x)$ for itself. For instance, suppose $d = 0.85$ and energy quantity $\text{in}(x) = 5.0$ flowing into node x. Then, the total energy distributed to successor nodes amounts to 4.25, while energy rank energy(x) of x increases by 0.75. Special treatment is necessary for nodes with zero outdegree. For simplicity, we assume all nodes to have an outdegree of at least one, which makes perfect sense, as will be shown later.

The spreading factor concept is very intuitive and, in fact, very close to real models of energy spreading through networks. Observe that the overall amount of energy in the network, after initial activation in^0, does not change over time. More formally, suppose that $\text{energy}(n) = 0$ for all $n \in V$ before injection in^0 into source s. Then the following equation holds in every computation step of our modified spreading algorithm, incorporating the concept of spreading factor d:

$$\sum_{x \in V} \text{energy}(x) = \text{in}^0 \tag{1}$$

Spreading factor d may also be seen as the ratio between *direct* trust in x and trust in the ability of x to *recommend* others as trustworthy peers. For instance, Beth, Borcherding and Klein (1994) and Maurer (1996) explicitly differentiate between *direct* trust edges and *recommendation* edges.

We generally assume $d = 0.85$, though other values may also seem reasonable. For instance, having $d \leq 0.5$ allows agents to keep most of the trust they are granted for themselves and only pass small portions of trust to their peers. Observe that low values for d favor trust proximity to the source of trust injection, while high values allow trust to also reach nodes which are more distant. Furthermore, the introduction of spreading factor d is crucial for making Appleseed retain Levien's bottleneck property, as will be shown in later sections.

3.2.4. Rank normalization. Algorithm 2 makes use of edge weight normalization, i.e., the quantity $e_{x \to y}$ of energy distributed along (x, y) from x to successor node y depends on its *relative* weight, i.e., $W(x, y)$ compared to the sum of weights of all outgoing edges of x:

$$e_{x \to y} = d \cdot \text{in}(x) \cdot \frac{W(x, y)}{\sum_{(x,s) \in E} W(x, s)}$$

Normalization is common practice to many trust metrics, among those PageRank (Page et al., 1998), EigenTrust (Kamvar, Schlosser and Garcia-Molina, 2003), and AORank (Guha, 2003). However, while normalized reputation or trust seem reasonable for models with plain, non-weighted edges, serious interferences occur when edges are weighted, as is the case for our trust model adopted in Section 2.2.1.

For instance, refer to the left-hand side of Fig. 6 for unwanted effects: the amounts of energy that node a accords to successors b and d, i.e., $e_{a \to b}$ and $e_{a \to d}$, respectively, are identical in value. Note that b has issued only *one* trust statement $W(b, c) = 0.25$, telling that its trust in c is rather weak. On the other hand, d assigns *full* trust to individuals e, f, and g. Nevertheless, the overall trust rank for d will be much higher than for any successor of d, for c is accorded $e_{a \to b} \cdot d$, while e, f, and g only obtain $e_{a \to d} \cdot d \cdot 1/3$ each. Hence, c will be trusted *three times* as much as e, f, and g, which is not reasonable at all.

3.2.5. Backward trust propagation. The above issue has already been discussed in Kamvar, Schlosser and Garcia-Molina (2003), but no solution was proposed

therein, arguing that "substantially good results" were achieved despite the drawbacks. We propose to alleviate the problem by making use of backward propagation of trust to the source: when computing the metric, additional "virtual" edges (x, s) from every node $x \in V \setminus \{s\}$ to the trust source s are created. These edges are assigned full trust $W(x, s) = 1$. Existing backward links (x, s), along with their weights, are "overwritten". Intuitively, every node is supposed to blindly trust the trust source s, see Figure 6. The impacts of adding backward propagation links are threefold:

- *Mitigating relative trust.* Again, we refer to the left-hand graph in Figure 6. Trust distribution in the underlying case becomes much fairer through backward propagation links, for c now only obtains $e_{a\to b} \cdot d \cdot (0.25/(1 + 0.25))$ from source s, while e, f, and g are accorded $e_{a\to d} \cdot d \cdot (1/4)$ each. Hence, trust ranks of both e, f, and g amount to 1.25 times the trust assigned to c.
- *Avoidance of dead ends.* Dead ends, i.e., nodes x with zero outdegree, require special treatment in our computation scheme. Two distinct approaches may be adopted. First, the portion of incoming trust $d \cdot \text{in}(x)$ supposed to be passed to successor nodes is completely discarded, which contradicts our constraint of no energy leaving the system. Second, instead of retaining $(1 - d) \cdot \text{in}(x)$ of incoming trust, x keeps *all* trust for itself. The latter approach is also not sensible as it encourages users to not issue trust statements for their peers. Luckily, with backward propagation of trust, all nodes are implicitly linked to the trust source s, so that there are no more dead ends to consider.
- *Favoring trust proximity.* Backward links to the trust source s are favorable for nodes close to the source, as their eventual trust rank will increase. On the other hand, nodes further away from s are penalized.

Overly rewarding nodes close to the source is not beyond dispute and may pose some issues. In fact, it represents the tradeoff we have to pay for both welcome aspects of backward propagation.

3.2.6. Nonlinear trust normalization. In addition to backward propagation, an integral part of Appleseed, we propose supplementary measures to decrease the negative impact of trust distribution based on relative weights. Situations where nodes y with poor ratings from x are awarded high overall trust ranks, thanks to the low outdegree of x, have to be avoided. Taking the squares of local trust weights provides an appropriate solution:

$$e_{x\to y} = d \cdot \text{in}(x) \cdot \frac{W(x, y)^2}{\sum_{(x,s)\in E} W(x, s)^2}$$

As an example, refer to node b in Figure 6. With squared normalization, the total amount of energy flowing backward to source a increases, while the amount of energy flowing to the poorly trusted node c decreases significantly. Accorded trust quantities $e_{b\to a}$ and $e_{b\to c}$ amount to $d \cdot \text{in}(b) \cdot (1/1.0625)$ and $d \cdot \text{in}(b) \cdot (0.0625/1.0625)$, respectively. More serious penalization of poor trust ratings can be achieved by selecting powers above two.

3.2.7. Algorithm outline. Having identified modifications to apply to spreading activation models in order to tailor them for local group trust metrics, we are now able to formulate the core algorithm of Appleseed. Input and output are characterized as follows:

$$\text{Trust}_A : V \times \mathbb{R}_0^+ \times [0, 1] \times \mathbb{R}^+ \to (\text{trust} : V \to \mathbb{R}_0^+)$$

The first input parameter specifies trust seed s, the second trust injection e, parameter three identifies spreading factor $d \in [0, 1]$, and the fourth argument binds accuracy threshold T_c, which serves as one of two convergence criteria. Similar to Advogato, the output is an assignment function of trust with domain V. However, Appleseed allows *rankings* of agents with respect to the trust accorded. Advogato, on the other hand, only assigns boolean values indicating presence or absence of trust.

Appleseed works with *partial* trust graph information. Nodes are accessed only when needed, i.e., when reached by energy flow. Trust ranks $\text{trust}(x)$, which correspond to $\text{energy}(x)$ in Algorithm 2, are initialized to 0. Any unknown node u hence obtains $\text{trust}(u) = 0$. Likewise, virtual trust edges for backward propagation from node x to the source are added at the moment that x is discovered. In every iteration, for those nodes x reached by flow, the amount of incoming trust is computed as follows:

$$\text{in}(x) = d \cdot \sum_{(p,x)\in E} \left(\text{in}(p) \cdot \frac{W(p, x)}{\sum_{(p,s)\in E} W(p, s)} \right)$$

Incoming flow for x is hence determined by all flow that predecessors p distribute along edges (p, x). Note that the above equation makes use of linear normalization of relative trust weights. Replacement of linear by

nonlinear normalization according to Section 3.2.6 is straight-forward, though. The trust rank of x is updated as follows:

$$\text{trust}(x) \leftarrow \text{trust}(x) + (1 - d) \cdot \text{in}(x)$$

However, trust networks generally contain cycles and thus allow no topological sorting of nodes. Hence, the computation of $\text{in}(x)$ for reachable $x \in V$ is inherently recursive. Several iterations for all nodes are required in order to make computed information converge towards the least fixpoint. We give a criterion that has to be satisfied for convergence, relying upon accuracy threshold T_c briefly introduced before.

Definition 1 (Termination). Suppose that $V_i \subseteq V$ represents the set of nodes that were discovered until step i, and $\text{trust}_i(x)$ the current trust ranks for all $x \in V$. Then the algorithm terminates when the following condition is satisfied after step i:

$$\forall x \in V_i : \text{trust}_i(x) - \text{trust}_{i-1}(x) \leq T_c \qquad (2)$$

Informally, Appleseed terminates when changes of trust ranks with respect to the prior iteration $i - 1$ are not greater than accuracy threshold T_c.

Moreover, when supposing spreading factor $d > 0$, accuracy threshold $T_c > 0$, and trust source s part of some connected component $G' \subseteq G$ containing at least two nodes, convergence, and thus termination, is guaranteed. The following paragraph gives an informal proof:

Proof (Convergence of Appleseed): Assume that f_i denotes step i's quantity of energy flowing through the network, i.e., all the trust that has not been captured by some node x through function $\text{trust}_i(x)$. It follows from Equation (1) that in^0 constitutes the *upper boundary* of trust energy floating through the network, and f_i can be computed as below:

$$f_i = \text{in}^0 - \sum_{x \in V} \text{trust}_i(x)$$

Since $d > 0$ and $\exists (s, x) \in E, x \neq s$, the sum of current trust ranks $\text{trust}_i(x)$ of all $x \in V$ is *strictly increasing* for increasing i. Consequently, $\lim_{i \to \infty} f_i = 0$ holds. Moreover, since termination is defined by some fixed accuracy threshold $T_c > 0$, there exists some step k such that $\lim_{i \to k} f_i \leq T_c$. □

3.3. Comparison of advogato and appleseed

Both Advogato and Appleseed are implementations of local group trust metrics. Advogato has already proven its efficiency in practical usage scenarios such as the Advogato online community, though lacking quantitative fitness information. Its success is mainly measured by indirect feedback, such as the amount of spam messages posted on Advogato, which has been claimed to be rather low. In order to evaluate the fitness of Appleseed as an appropriate approach to group trust computation, we intend to relate our novel approach to Advogato for comparison:

- *Attack-resistance.* This property defines the behavior of trust metrics in case of malicious nodes trying to invade into the system. For the evaluation of attack-resistance capabilities, we have briefly introduced the "bottleneck property" in Section 3.1.2, which holds for Advogato. In order to recapitulate, suppose that s and t are nodes and connected through trust edge (s, t). Node s is assumed good, while t is an attacking agent trying to make good nodes trust malevolent ones. In case the bottleneck property holds, manipulation "on the part of bad nodes does not affect the trust value" (Levien, 2003). Clearly, Appleseed satisfies the bottleneck property, for nodes cannot raise their impact by modifying the structure of trust statements they issue. Bear in mind that the amount of trust accorded to agent t *only* depends on its predecessors and does not increase when t adds more nodes. Both spreading factor d and normalization of trust statements ensure that Appleseed becomes equally as attack-resistant as Advogato.
- *Trust weight normalization.* We have indicated before that issuing multiple trust statements dilutes trust accorded to successors. According to Guha (2003), this does not comply with real world observations, where statements of trust "do not decrease in value when the user trusts one more person [...]". The malady that Appleseed suffers from is common to many trust metrics, notably those based upon finding principal eigenvectors (Page et al., 1998; Kamvar, Schlosser and Garcia-Molina, 2003; Richardson, Agrawal and Domingos, 2003). On the other hand, the approach pursued by Advogato does not penalize trust relationships asserted by eager trust dispensers, for node capacities do not depend on local information. Remember that capacities of nodes pertaining to level l are assigned based on the capacity of level

$l-1$ as well as the *overall* outdegree of nodes part of this level. Hence, Advogato encourages agents issuing numerous trust statements, while Appleseed penalizes overly abundant trust certificates.

- *Deterministic trust computation.* Appleseed is deterministic with respect to the assignment of trust rank to agents. Hence, for any arbitrary trust graph $G = (V, E, W)$ and for every node $x \in V$, linear equations allow to characterize the amount of trust assigned to x, as well as the quantity that x accords to its successor nodes. Advogato, however, is nondeterministic. Though the *number* of trusted agents, and therefore the computed maximum flow size, is determined for given input parameters, the set of agents itself is not. Changing the order in which trust assertions are issued may yield different results. For example, suppose $C_V(s) = 1$ holds for trust seed s. Furthermore, assume s has issued trust certificates for two agents, b and c. The actual choice between b or c as trustworthy peer with maximum flow only depends on the order in which nodes are accessed.
- *Model and output type.* Basically, Advogato supports non-weighted trust statements only. Appleseed is more versatile by virtue of its trust model based on weighted trust certificates. In addition, Advogato returns one set of trusted peers, whereas Appleseed assigns *ranks* to agents. These ranks allow to select most trustworthy agents first and relate them to each other with respect to their accorded rank. Hereby, the definition of thresholds for trustworthiness is left to the user who can thus tailor relevant parameters to fit different application scenarios. For instance, raising the application-dependent threshold for the selection of trustworthy peers, which may be either an absolute or relative value, allows for enlarging the neighborhood of trusted peers. Appleseed is hence more adaptive and flexible than Advogato.

3.4. Parameterization and experiments

Appleseed allows numerous parameterizations of its input variables. Discussions of parameter instantiations and caveats thus constitute indispensable complements to our contribution. Moreover, we provide experimental results exposing observed effects of parameter tuning. Note that all experiments have been conducted on data obtained from "real" social networks: several web crawling tools were written to mine the Advogato community web site and extract trust assertions stated by its more than 8,000 members. Hereafter, we converted all trust data to our trust model proposed in Section 2.2.1. Notice that the Advogato community server supports four different levels of peer certification, namely "Observer", "Apprentice", "Journeyer", and "Master". We mapped these qualitative certification levels to quantitative ones, assigning $W(x, y) = 0.25$ for x certifying y as "Observer", $W(x, y) = 0.5$ for an "Apprentice", and so forth. The Advogato community grows rapidly and our crawler extracted $3,224,101$ trust assertions. Heavy preprocessing and data cleansing was inevitable, eliminating reflexive trust statements $W(x, x)$ and shrinking trust certificates to reasonable sizes. Note that some eager Advogato members issued more than two thousand trust statements, yielding an overall average outdegree of 397.69 assertions per node. Common sense tell us that this figure is beyond dispute. Applying our set of extraction tools, we tailored the test data obtained from Advogato to our needs and extracted trust networks with specific average outdegree for the experimental analysis.

3.4.1. Trust injection. Trust values $\text{trust}(x)$ computed by the Appleseed metric for source s and node x may differ greatly from explicitly assigned trust weights $W(s, x)$. We have already mentioned before that computed trust ranks may *not* be interpreted as absolute values, but rather in comparison with ranks assigned to all other peers. In order to make assigned rank values more tangible, though, one might expect that tuning the trust injection in^0 to satisfy the following proposition will align computed ranks and explicit trust statements:

$$\forall (s, x) \in E : \ \text{trust}(x) \in [W(s, x) - \epsilon, W(s, x) + \epsilon]$$

However, when assuming reasonably small ϵ, the approach does not succeed. Recall that *computed* trust values of successor nodes x to s do not only depend on assertions made by s, but also on trust ratings asserted by other peers. Hence, perfect alignment of explicit trust ratings with computed ones cannot be accomplished. However, we propose alignment heuristics, incorporated into Algorithm 4, which have proven to work remarkably well in diverse test scenarios. The basic idea is to add another node i and edge (s, i) with $W(s, i) = 1$ to the trust graph $G = (V, E, W)$, treating (s, i) as an indicator to tell whether trust injection in^0 is "good" or not. Consequently, parameter in^0 has to be adapted in order to make $\text{trust}(i)$ converge towards $W(s, i)$. The trust metric computation is hence repeated with different values for in^0 until convergence

```
function Trust_A (s ∈ V, in^0 ∈ R_0^+, d ∈ [0,1], T_c ∈ R^+) {
  set in_0(s) ← in^0, trust_0(s) ← 0, i ← 0;
  set V_0 ← {s};
  repeat
    set i ← i + 1;
    set V_i ← V_{i−1};
    ∀x ∈ V_{i−1} : set in_i(x) ← 0;
    for all x ∈ V_{i−1} do
      set trust_i(x) ← trust_{i−1}(x) + (1 − d) · in_{i−1}(x);
      for all (x,u) ∈ E do
        if u ∉ V_i then
          set V_i ← V_i ∪ {u};
          set trust_i(u) ← 0, in_i(u) ← 0;
          add edge (u,s), set W(u,s) ← 1;
        end if
        set w ← W(x,u) / Σ_{(x,u')∈E} W(x,u');
        set in_i(u) ← in_i(u) + d · in_{i−1}(x) · w;
      end do
    end do
    set m = max_{y∈V_i} {trust_i(y) − trust_{i−1}(y)};
  until (m ≤ T_c)
  return (trust : {(x, trust_i(x)) | x ∈ V_i});
}
```

Algorithm 3. *Appleseed trust metric.*

```
function Trust_heu (s ∈ V, d ∈ [0,1], T_c ∈ R^+) {
  add node i, edge (s,i), set W(s,i) ← 1;
  set in^0 ← 20, ε ← 0.1;
  repeat
    set trust ← Trust_A (s, in^0, d, T_c);
    in^0 ← adapt (W(s,i), trust(i), in^0);
  until trust(i) ∈ [W(s,i) − ε, W(s,i) + ε]
  remove node i, remove edge (s, i);
  return Trust_A (s, in^0, d, T_c);
}
```

Algorithm 4. *Adding weight alignment heuristics.*

of explicit and computed trust value for i is achieved. Eventually, edge (s, i) and node i are removed and the metric computation is performed one more time. Experiments have shown that our imperfect alignment heuristics yield computed ranks trust(x) for direct successors x of trust source s which come close to previously specified trust statements $W(s, x)$.

3.4.2. Spreading factor. Small values for d tend to overly reward nodes close to the trust source and penalize remote ones. Recall that low d allows nodes to retain most of the incoming trust quantity for themselves, while large d stresses the recommendation of trusted individuals and makes nodes distribute most of the assigned trust to their successor nodes:

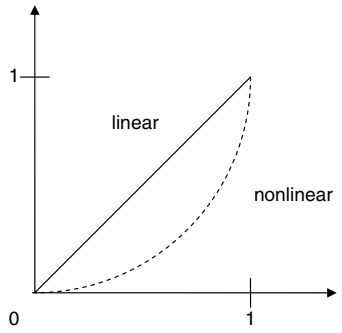

Fig. 7. *Linear and nonlinear normalization.*

Experiment 1 (Impact of parameter d). We compare distributions of computed rank values for three diverse instantiations of d, namely $d_1 = 0.1$, $d_2 = 0.5$, and $d_3 = 0.85$. Our setup is based upon a social network with an average outdegree of six trust assignments and 384 nodes reached by trust energy spreading from our designated trust source. We furthermore suppose $\text{in}^0 = 200$, $T_c = 0.01$, and linear weight normalization. Computed ranks are classified into 11 histogram cells with nonlinear cell width. Obtained output results are displayed in Fig. 8. Mind that we have chosen *logarithmic* scales for the vertical axis in order to render the diagram more legible. For d_1, we observe that this value engenders the highest amount of nodes x with ranks trust$(x) \geq 25$. On the other hand, virtually no ranks ranging from 0.2 to 1 are assigned, while the number of nodes with ranks smaller than 0.05 is again much higher for d_1 than for both d_2 and d_3. Instanti-

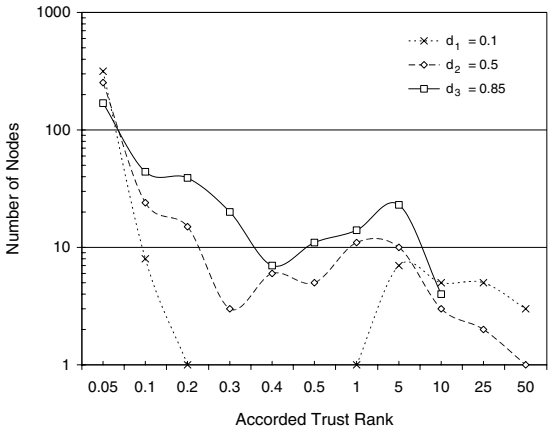

Fig. 8. *Spreading factor impact.*

ation $d_3 = 0.85$ constitutes the counterpart of d_1. No ranks with trust$(x) \geq 25$ are accorded, while interim ranks between 0.1 and 10 are much more likely for d_3 than for both other instantiations of spreading factor d. Consequently, the number of ranks below 0.05 is lowest for d_3.

The experiment demonstrates that high values for parameter d tend to distribute trust more evenly, neither overly rewarding nodes close to the source, nor penalizing remote ones too rigidly. On the other hand, low d assigns high trust ranks to very few nodes, namely those which are closest to the source, while the majority of nodes obtains very low trust rank. We propose to set $d = 0.85$ for general use.

3.4.3. Accuracy and convergence rate. We already mentioned before that the Appleseed algorithm is inherently recursive. Parameter T_c constitues the ultimate criterion for termination. We will show through an experiment that convergence is reached very fast, no matter how huge the number of nodes trust is flowing through, and no matter how large the initial trust injection.

Experiment 2 (Convergence rate). The trust network we consider has an average outdegree of five trust assignments per node. The number of nodes for which trust ranks are assigned amounts to 572. We suppose $d = 0.85$, $T_c = 0.01$, and linear weight normalization. Two runs are computed, one with trust activation $\text{in}_1 = 200$, the other with initial energy $\text{in}_2 = 800$. Fig. 9 demonstrates the rapid convergence of both runs. Though the trust injection for the second run is four times as high as for the first, convergence is reached in only few more iterations: run one takes 38 iterations, run two terminates after 45 steps.

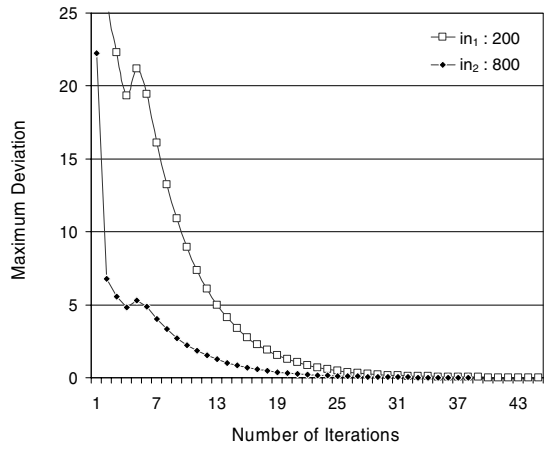

Fig. 9. *Convergence of Appleseed.*

For both runs, we assumed accuracy threshold $T_c = 0.01$, which is extremely small and accurate beyond necessity already. However, experience taught us that convergence takes place rapidly even for very large networks and high amounts of trust injected, so that assuming the latter value for T_c imposes no scalability issues. In fact, the amount of nodes taken into account for trust rank assignment in the above example well exceeds practical usage scenarios: mind that the case at hand demands 572 documents to be fetched from the Web, complaisantly supposing that these pages containing personal trust information for each node are cached after their first access. Hence, we may well claim that the actual bottleneck of group trust computation is not the Appleseed metric itself, but downloads of trust resources from the network. This bottleneck might also be the reason for selecting thresholds T_c greater than 0.01, in order to make the algorithm terminate after fewer node accesses.

3.5. Implementation and extensions

Appleseed was implemented in JAVA, based upon Algorithm 3. We applied moderate fine-tuning and supplemented our metric with an architectural cushion in order to access "real" machine-readable RDF homepages. Other notable modifications to the core algorithm are discussed briefly:

- *Maximum number of nodes.* We supplemented the set of input parameters by yet another argument M, which specifies the maximum number of nodes to unfold. This extension hinders trust energy from overly covering vast parts of the entire network. Note that accessing the personal, machine-readable homepages, which contain trust information required for the metric computation, represents the actual computation bottleneck. Hence, expanding as few nodes as possible is highly desirable. When choosing reasonably large M, for instance, twice the number of agents assumed trustworthy, we may expect to not miss any relevant nodes: mind that Appleseed proceeds breadth-first and thus considers close nodes first, which are more eligible for trust than distant ones.
- *Upper-bounded trust path lengths.* Another approach to sensibly restrict the number of nodes unfolded relies upon upper-bounded path lengths. The idea

of constraining path lengths for trust computation has been adopted before by Reiter and Stubblebine (1996) and within the X.509 protocol (Housely et al., 1999). Depending on the overall trust network connectivity, we opt for maximum path lengths between three and six, well aware of Milgram's "six degress of separation" paradigm (Milgram, 1992). In fact, trust decay is inherent to Appleseed, thanks to spreading factor d and backward propagation. Stripping nodes at large distances from the seed therefore only marginally affects the trust metric computation results while providing major speed-ups at the same time.

- *Zero trust retention for the source.* Third, we modified Appleseed to hinder trust source s from accumulating trust energy, essentially introducing one novel spreading factor $d_s = 1.0$ for the seed only. Consequently, all trust is divided among peers of s and none retained, which is reasonable. Remember that s wants to discover trustworthy agents and not assign trust rank to itself. Convergence may accelerate, since $\text{trust}_{i+1}(x) - \text{trust}_i(x)$ used to be maximal for seed node s, thanks to backward propagation of trust. Furthermore, supposing the same trust quantity in^0 injected, assigned trust ranks become greater in value, also enlarging gaps between neighbors in trust rank.

3.6. Testbed for local group trust metrics

Trust metrics and models for trust propagation have to be intuitive, underpinning the need for the application of Occam's Razor. Humans must be able to comprehend *why* agent a was accorded higher trust rank than b and come to similar results when asked for a personal judgement. Consequently, we implemented our own testbed, which visually displays social networks, allows zooming of specific nodes and layouts these appropriately, with minimum overlap. We made use of the yFiles (Wiese, Eiglsperger and Kaufmann, 2001) library to perform all sophisticated graph drawing. Moreover, our testbed permits to parameterize Appleseed through dialogs. Detailed output is provided, both graphical and textual. Graphical results comprise the highlighting of nodes with trust ranks above certain thresholds, while textual results return quantitative trust ranks of all accessed nodes, numbers of iterations, and so forth. We also implemented the Advogato trust metric and incorporated the latter into our testbed. Hereby, our implementation of Advogato does not require a priori complete trust graph information, but accesses nodes "just in time", similar to Appleseed. All experiments were conducted on top of the testbed application.

4. Distrust

Distrust is one of the most controversial topics and issues to cope with, especially when considering trust metrics and trust propagation. Most approaches completely ignore distrust and only consider full trust or degrees of trust (Levien and Aiken, 1998; Mui, Mohtashemi and Halberstadt, 2002; Beth, Borcherding and Klein, 1994; Maurer, 1996; Reiter and Stubblebine, 1996; Richardson, Agrawal and Domingos, 2003). Others, among those (Abdul-Rahman and Hailes, 1997; Chen and Yeager, 2003; Aberer and Despotovic, 2001; Golbeck, Parsia and Hendler, 2003), allow for distrust ratings, though, but do not consider the subtle semantic differences pertaining to the distinct notions of trust and distrust. Consequently, according to Gans et al. (2001), "distrust is regarded as just the other side of the coin, that is, there is generally a symmetric scale with complete trust on one end and absolute distrust on the other." Furthermore, some researchers equate the notion of distrust with lack of trust information. Contrarily, in his seminal work on the essence of trust, Marsh (1994a) has already pointed out that those two concepts, i.e., lack of trust and distrust, may not be intermingled. For instance, in absence of trustworthy agents, we might be more prone to accept recommendations from persons we do not trust, probably because of lack of prior experiences Marsh (1994a), than from persons we explicitly distrust, resulting from past bad experiences or deceit. However, even Marsh pays little attention to the specifics of distrust.

Gans et al. (2001) were among the first to recognize the importance of distrust, stressing the fact that "distrust is an irreducible phenomenon that cannot be offset against any other social mechanisms", hence including trust. In their work (Gans et al., 2001), an explicit distinction between confidence, trust, and distrust is made. Moreover, the authors indicate that distrust might be highly relevant to social networks. Its impact is not inherently negative, but may also influence the network in an extremely positive fashion. However, the primary focus of the latter work is on methodology issues and planning, not considering trust assertion evaluations and propagation through appropriate metrics.

Guha (2003), Guha, Raghavan, and Tomkins (2004) eventually acknowledges the immense role of distrust with respect to trust propagation applications, arguing that "distrust statements are very useful for users to debug their Web of Trust". For example, suppose that agent a blindly trusts b, which again blindly trusts c, which blindly trusts d. However, a completely distrusts d. The latter distrust statement hence ensures that a will *not* accept beliefs and ratings from d, regardless of agent a trusting b trusting c trusting d.

4.1. Semantics of distrust

The non-symmetrical nature of distrust and trust, being two perfect dichotomies, has already been recognized by recent sociological research (Lewicki, McAllisterand and Bies, 1998). In this section, we investigate the differences between distrust and trust pertaining to possible inferences and the propagation of statements.

4.1.1. Distrust as negated trust. Interpreting distrust as negation of trust was adopted by many trust metrics, among those trust metrics proposed in Abdul-Rahman and Hailes (1997), Jøsang, Gray, and Kinateder (2003), Abdul-Rahman and Hailes (2000), and Chen and Yeager (2003). Basically, these metrics compute trust values by analyzing *chains* of trust statements from source s to target t, eventually merging them to obtain an aggregate value. Each chain hereby becomes synthesized into one single number through *weighted multiplication* of trust values along trust paths. Serious implications resulting from assuming that trust concatenation relates to multiplication (Richardson, Agrawal and Domingos, 2003), and distrust to negated trust, manifest when agent a distrusts b, which distrusts c:

$$\neg\,\text{trust}(a, b) \wedge \neg\,\text{trust}(b, c) \models \text{trust}(a, c)$$

Jøsang, Gray, and Kinateder (2003) are well aware of this rather unwanted effect but do not deny its correctness, for the enemy of your enemy could well be your friend. Guha, on the other hand, indicates that two distrust statements cancelling out each other most often does not reflect desired behavior (Guha, 2003). We adopt the opinion of Guha and claim that distrust may not be interpreted as negated trust.

4.1.2. Propagation of distrust. The "conditional transitivity" (Abdul-Rahman and Hailes, 1997) of trust is commonly agreed upon and constitutes the foundation and pivotal premiss that all trust metrics rely upon. However, no consensus in literature has been achieved as it comes to the *degree* of transitivity and the decay rate of trust. Many approaches therefore explicitly distinguish between recommendation trust and direct trust (Jøsang, Gray, and Kinateder, 2003; Abdul-Rahman and Hailes, 1997; Maurer, 1996; Beth, Borcherding and Klein, 1994; Chen and Yeager, 2003) in order to keep apart the transitive fraction of trust from the non-transitive one. Hence, in these works, only the *ultimate* edge within the trust chain, i.e., the one linking to the trust target, needs to be direct, while all others are supposed to be recommendations. For the Appleseed trust metric, this distinction is made through the introduction of the global spreading factor d. However, the conditional transitivity property of trust does not equally extend to distrust. The case of double negation through distrust propagation has already been considered. Now suppose, for instance, that a distrusts b, which trusts c. Supposing distrust to propagate through the network, we may come to make the following inference:

$$\text{distrust}(a, b) \wedge \text{trust}(b, c) \models \text{distrust}(a, c)$$

This inference is more than questionable, for a penalizes c simply for being trusted by an agent that a distrusts. Obviously, this assumption is not sound and does not reflect expected real-world behavior. We assume that distrust does not allow to make direct inferences of any kind. This conservative assumption makes us stay on the "safe" side and is in perfect accordance with (Guha, 2003).

4.2. Incorporating distrust into appleseed

We compare our implementation of distrust with Guha's approach, who supposes an identical model of distrust. Guha computes trust by means of one global group trust metric, similar to PageRank (Page et al., 1998). For distrust, he proposes two candidate approaches. The first one directly integrates distrust into the iterative eigenvector computation and comes up with one single measure combining both trust and distrust. However, in networks dominated by distrust, the iteration might not converge. The second proposal first computes trust ranks by trying to find the dominant eigenvector, and then computes separate distrust ranks in one single step, based upon the iterative computation of trust ranks. Suppose that D_a is the set of agents who distrust a:

$$\text{DistrustRank}(a) = \frac{\sum_{b \in D_a} \text{TrustRank}(b)}{|D_a|}$$

The problem we perceive with this approach refers to *superimposing* the computation of distrust ranks *after* trust rank computation, which may yield some strange behavior: suppose an agent a which is highly controversial by engendering ambiguous sentiments, i.e., on the one hand, there are numerous agents that trust a, and on the other hand, there are numerous agents which distrust a. With the approach proposed by Guha, a's impact through asserting distrust into other agents is huge, resulting from its immense positive trust rank. However, common sense tells us this should not be the case, for a is subject to tremendous distrust itself, thus levelling out its high trust rank.

Hence, for our own approach, we intend to directly incorporate distrust into the iterative process of the Appleseed trust metric computation, and not superimpose distrust afterwards. Several pitfalls have to be avoided, such as the risk of non-convergence in case of networks dominated by distrust (Guha, 2003). Furthermore, in absence of distrust statements, we want the distrust-enhanced Appleseed algorithm, which we denote by Trust_{A^-}, to yield results identical to those engendered by the original version Trust_A.

4.2.1. Normalization and distrust. First, the trust normalization procedure has to be adapted. We hereby suppose the more general case which does not necessarily assume linear normalization but normalization of weights to the power of q, as has been discussed in Section 3.2.6. Let $\text{in}(x)$, the trust influx for agent x, be positive. As usual, we denote the global spreading factor by d, and quantified trust statements from x towards y by $W(x, y)$. Function $\text{sign}(x)$ returns the sign of value x. Note that from now on, we assume $W : E \to [-1, +1]$, for degrees of *distrust* need to be expressed as well. Then the trust quantity $e_{x\to y}$ distributed from x to successor node y is computed as follows:

$$e_{x\to y} = d \cdot \text{in}(x) \cdot \text{sign}(W(x, y)) \cdot w, \qquad (3)$$

where

$$w = \frac{|W(x, y)|^q}{\sum_{(x,s)\in E} |W(x, s)|^q}$$

The accorded quantity $e_{x\to y}$ becomes negative if $W(x, y)$ is negative, i.e., if x distrusts y to a certain extent. For the relative weighting, the *absolute* values $|W(x, s)|$ of all weights are considered. Otherwise, the denominator could become negative, or positive trust statements could become boosted unduly. The latter would be the case if the sum of positive trust ratings slightly outweighed the sum of negative ones, making the denominator converge towards zero. An example demonstrates the computation process:

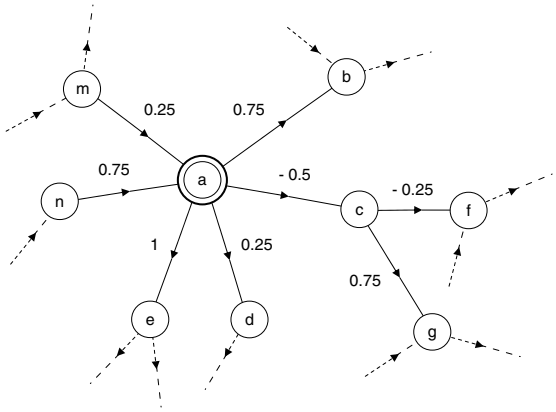

Fig. 10. *Network augmented by distrust.*

Example 2 (Distribution of Trust and Distrust). We assume the trust network as depicted in Fig. 10. Let the trust energy influx into node a be $\text{in}(a) = 2$, and global spreading factor $d = 0.85$. For simplicity reasons, backward propagation of trust to the source is not considered. Moreover, we suppose *linear* weight normalization, thus $q = 1$. Consequently, the denominator of the normalization equation is $|0.75| + |-0.5| + |0.25| + |1| = 2.5$. The trust energy that a distributes to b hence amounts to $e_{a\to b} = 0.51$, whereas the energy accorded to the distrusted node c is $e_{a\to c} = -0.34$. Furthermore, we have $e_{a\to d} = 0.17$ and $e_{a\to e} = 0.68$.

Observe that trust energy becomes lost during distribution, for the sum of energy accorded along outgoing edges of a amounts to 1.02, while 1.7 was provided for distribution. The effect results from the negative trust weight $W(a, c) = -0.5$.

4.2.2. Distrust allocation and propagation. We now analyze the case where the influx $\text{in}(x)$ for agent x is *negative*. In this case, the trust allocated for x will also be negative, i.e., $\text{in}(x) \cdot (1 - d) < 0$. Moreover, the energy $\text{in}(x) \cdot d$ that x may distribute among its successor nodes will naturally be negative as well. The implications are those which have been mentioned in Section 4.1, i.e., distrust as negation of trust and propagation of distrust. For the first case, refer to node f in Figure 10 and assume $\text{in}(c) = -0.34$, which is derived from Example 2. The trusted agent a distrusts c

which distrusts f. Eventually, f would be accorded $d \cdot (-0.34) \cdot (-0.25)$, which is *positive*. For the second case, node g would be assigned a *negative* trust quantity $d \cdot (-0.34) \cdot (0.75)$, simply for being trusted by f, which is commonly distrusted. Both unwanted effects can be avoided by not allowing distrusted nodes to distribute any energy at all. Hence, more formally, we introduce a novel function $\text{out}(x)$:

$$\text{out}(x) = \begin{cases} d \cdot \text{in}(x), & \text{if in}(x) \geq 0 \\ 0, & \text{else} \end{cases} \tag{4}$$

The function then has to replace $d \cdot \text{in}(x)$ when computing the energy distributed along edges from x to successor nodes y:

$$e_{x \to y} = \text{out}(x) \cdot \text{sign}(W(x, y)) \cdot w, \tag{5}$$

where

$$w = \frac{|W(x, y)|^q}{\sum_{(x,s) \in E} |W(x, s)|^q}$$

This design decision perfectly aligns with our assumptions made in Section 4.1 and prevents the inference of unwanted side-effects mentioned before. Furthermore, one can see easily that the modifications introduced do not change the behavior with respect to Algorithm 3 when not considering relationships of distrust.

4.2.3. Convergence. Even in networks largely or entirely dominated by distrust, our enhanced version of Appleseed is guaranteed to converge. We therefore briefly outline an informal proof, knowing about the convergence of the core Appleseed algorithm, which has been shown before by Proof 1:

Proof 2 (Convergence in presence of distrust): Recall that only positive trust influx $\text{in}(x)$ becomes propagated, which has been indicated in Section 4.2.2. Hence, all we need to show is that the overall quantity of *positive* trust distributed in computation step i cannot be augmented through the presence of distrust statements. In other words, suppose that $G = (V, E, W)$ defines an arbitrary trust graph, containing quantified trust statements, but no distrust, i.e., $W : E \to [0, 1]$. Now consider another trust graph $G' = (V, E \cup D, W')$ which contains additional edges D, and weight function $W' = W \cup (D \to [-1, 0[)$. Hence, G' augments G by additional distrust edges between nodes taken from V. We now perform two parallel computations with enhanced Appleseed, one operating on G and the other on G'. In every step, and for every trust edge $(x, y) \in E$ for G, the distributed energy $e_{x \to y}$ is greater or equal than for its equivalent counterpart on G', for the denominator of the fraction given in Eq. (5) can only become *greater* through additional distrust outedges. Second, for the computation performed on G', negative energy distributed along edge (x, y) can only *reduce* the trust influx for y and may hence even accelerate convergence. □

However, as one might already have observed from the proof, there exists one serious implication arising from having distrust statements in the network. The overall accorded trust quantity does not equal the initially injected energy anymore. Moreover, in networks dominated by distrust, the overall trust energy sum may even be *negative*.

Experiment 3 (Network impact of distrust). We intend to analyze the number of iterations until convergence and the overall accorded trust rank of five networks. The structures of all these graphs are identical, being composed of 623 nodes with an average indegree and outdegree of 9. The only difference applies to the assigned weights, where the first graph contains no distrust statements at all, while 25% of all weights are negative for the second, 50% for the third, and 75% for the fourth. The fifth graph only contains distrust statements. Appleseed parameters are identical for all five runs, having backward propagation enabled, an initial trust injection $\text{in}^0 = 200$, spreading factor $d = 0.85$, convergence threshold $T_c = 0.01$, linear weight normalization, and no upper bound on the number of nodes to unfold. The left-hand side of Figure 11 clearly demonstrates that the number of iterations until convergence, given on the vertical axis, decreases with the proportion of distrust increasing, observable along the horizontal axis. Likewise, the overall accorded trust rank, indicated on the vertical axis of the right-hand side of Figure 11, decreases rapidly with increasing distrust, eventually dropping below zero. The same experiment was repeated for another network with 329 nodes, an average indegree and outdegree of 6, yielding similar results.

The effects observable in Experiment 3 only marginally affect the ranking itself, for trust ranks are interpreted *relative* to each other. Moreover, compensation for lost trust energy may be achieved by boosting the initial trust injection in^0.

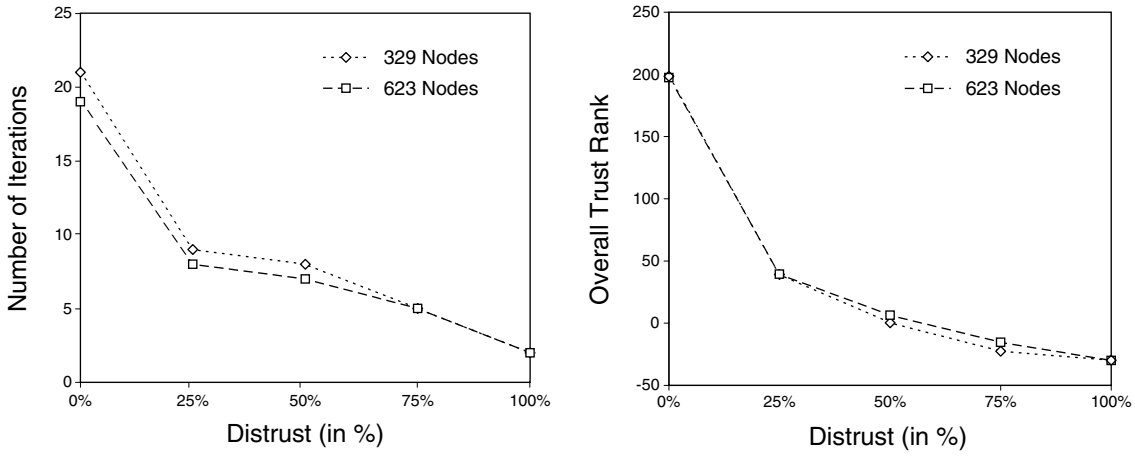

Fig. 11. *Network impact of distrust.*

5. *Discussion*

In this work, we have introduced various axes to classify trust metrics with respect to diverse criteria and features. Furthermore, we have advocated the need for local group trust metrics, eventually presenting Appleseed, our main contribution made. Through our proposed trust model, we have situated Appleseed within the Semantic Web universe. However, we believe that Appleseed suits other application scenarios likewise, such as group trust in online communities, open rating systems, ad-hoc and peer-to-peer networks.

For instance, Appleseed could support peer-to-peer-based file-sharing systems in reducing the spread of self-replicating inauthentic files by virtue of trust propagation (Kamvar, Schlosser and Garcia-Molina, 2003). In that case, explicit trust statements, resulting from direct interaction, reflect belief in someone's endeavor to provide authentic files.

Moreover, we have provided ample discussions of semantics and propagation models of distrust, owing to the fact that the latter concept has remained rather unattended by research. Details of its incorporation into the core Appleseed framework have also been provided.

However, several open issues for future research remain. Though having described ranking mechanisms and ways to align direct and indirect, i.e., computed, trust relationships by means of heuristics, an actual policy for eventual *boolean* decision-taking with respect to which agents to grant trust and which to deny has not been considered. Note that possible criteria are application-dependent. For some, one might want to select the n most trustworthy agents. For others, all agents with ranks above given thresholds may be eligible.

We strongly believe that local group trust metrics, such as Advogato and Appleseed, will become subject to substantial research for diverse computing domains within the near future. For instance, the Appleseed core currently undergoes integration into our decentralized, Semantic Web-based recommender system (Ziegler, 2004; Ziegler and Lausen, 2004), playing an essential role in its overall conception.

At any rate, success or failure of Appleseed, Advogato, and other group trust metrics largely depend on the leverage that candidate application scenarios are able to unfold.

Acknowledgments

We are very grateful to Lule Ahmedi, Nathan Dimmock, Kai Simon, and Paolo Massa for insightful comments and fruitful discussions, which helped us improve the quality of the paper.

Notes

1. Recall the definition of trust given before, telling that trust is a "subjective expectation".
2. Mind that in this context, "local" refers to the *place of computation* and not network perspective.
3. Supposing identical parameterizations for the metrics in use, as well as similar network structures.
4. Though various levels of peer certification exist, their proposed interpretation does not correspond to weighted trust relationships.
5. With respect to seed node a.

6. The terms "energy" and "trust" are used interchangeably in this context.
7. By relying on predicate calculus expressions, we greatly simplify through supposing that trust, and hence distrust, is fully transitive.

References

Abdul-Rahman A, Hailes S. A distributed trust model. In: *New Security Paradigms Workshop*, Cumbria, UK, September 1997:48–60.

Abdul-Rahman A, Hailes S. Supporting trust in virtual communities. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Maui, HW, USA, January 2000.

Aberer K, Despotovic Z. Managing trust in a peer-2-peer information system. In Paques H, Liu L, and Grossman D (eds.) *Proceedings of the Tenth International Conference on Information and Knowledge Management*, ACM Press, 2001:310–317.

Beth T, Borcherding M, Klein B. Valuation of trust in open networks. In: *Proceedings of the 1994 European Symposium on Research in Computer Security*, 1994:3–18.

Blaze M, Feigenbaum J, Lacy J. Decentralized trust management. In: *Proceedings of the 17th Symposium on Security and Privacy*, Oakland, CA, USA. IEEE Computer Society Press 1996:164–173.

Ceglowski M, Coburn A, Cuadrado J. Semantic search of unstructured data using contextual network graphs, June 2003.

Chen R, Yeager W. Poblano: A distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems, Santa Clara, CA, USA, February 2003.

Dumbill E. Finding friends with XML and RDF, June 2002. IBM's XML Watch.

Eschenauer L, Gligor V, Baras J. On trust establishment in mobile ad-hoc networks. Technical Report MS 2002-10, Institute for Systems Research, University of Maryland, MD, USA, October 2002.

Ford L, Fulkerson R. *Flows in Networks*. Princeton, NJ, USA:Princeton University Press, 1962.

Gans G, Jarke M, Kethers S, Lakemeyer G. Modeling the impact of trust and distrust in agent networks. In: *Proceedings of the Third International Bi-Conference Workshop on Agent-oriented Information Systems*, Montreal, Canada, May 2001.

Gil Y, Ratnakar V. Trusting information sources one citizen at a time. In: *Proceedings of the First International Semantic Web Conference*, Sardinia, Italy, June 2002.

Golbeck J, Parsia B, Hendler J. Trust networks on the semantic web. In: *Proceedings of Cooperative Intelligent Agents*, Helsinki, Finland, August 2003.

Gray E, Seigneur J-M, Chen Y, Jensen C. Trust propagation in small worlds. In Nixon P, and Terzis S (eds.), *Proceedings of the First International Conference on Trust Management*, volume 2692 of *LNCS*, Springer-Verlag, 2003:239–254.

Guha R. Open rating systems. Technical report, Stanford Knowledge Systems Laboratory, Stanford, CA, USA, 2003.

Guha R, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. In: *Proceedings of the Thirteenth International World Wide Web Conference*, New York, NY, USA, May 2004. ACM Press, 2004.

Hartmann K, Strothotte T. A spreading activation approach to text illustration. In: *Proceedings of the 2nd International Symposium on Smart Graphics*, Hawthorne, NY, USA: ACM Press, 2002:39–46.

Heider F. *The Psychology of Interpersonal Relations*. New York, NY, USA, Wiley, 1958.

Holland P, Leinhardt S. Some evidence on the transitivity of positive interpersonal sentiment. *American Journal of Sociology*, 1972;77:1205–1209.

Housely R, Ford W, Polk W, Solo D. Internet X.509 public key infrastructure, January 1999. Internet Engineering Task Force RFC 2459.

Jøsang A, Gray E, Kinateder M. Analysing topologies of transitive trust. In: *Proceedings of the Workshop of Formal Aspects of Security and Trust*, Pisa, Italy, September 2003.

Kahan J, Koivunen M-R, Prud' Hommeaux E, Swick R. Annotea - an open RDF infrastructure for shared web annotations. In: *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, China, May 2001:623–632.

Kamvar S, Schlosser M, Garcia-Molina H. The EigenTrust algorithm for reputation management in P2P networks. In: *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungary, May 2003.

Kinateder M, Pearson S. A privacy-enhanced peer-to-peer reputation system. In: *Proceedings of the 4th International Conference on Electronic Commerce and Web Technologies*, volume 2378 of *LNCS*, Prague, Czech Republic, September 2003. Springer-Verlag.

Kinateder M, Rothermel K. Architecture and algorithms for a distributed reputation system. In Nixon P, Terzis S (eds.), *Proceedings of the First International Conference on Trust Management*, volume 2692 of *LNCS*, Springer-Verlag, 2003:1–16.

Levien R. Attack Resistant Trust Metrics. PhD thesis, UC Berkeley, Berkeley, CA, USA, 2003.

Levien R, Aiken A. Attack-resistant trust metrics for public key certification. In: *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, USA, January 1998.

Levien R, Aiken A. An attack-resistant, scalable name service, 2000. Draft submission to the Fourth International Conference on Financial Cryptography.

Lewicki R, McAllister D, Bies R. Trust and distrust: New relationships and realities. *Academy of Management Review*, 1998;23(12):438–458.

Luhmann N. *Trust and Power*. Chichester, UK: Wiley, 1979.

Marsh S. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK, 1994a.

Marsh S. Optimism and pessimism in trust. In Ramirez J (eds.), *Proceedings of the Ibero-American Conference on Artificial Intelligence*, Caracas, Venezuela, McGraw-Hill, 1994.

Maurer U. Modelling a public key infrastructure. In Bertino E, editor, *Proceedings of the 1996 European Symposium on Research in Computer Security*, volume 1146 of *LNCS*, Springer-Verlag, 1996:325–350.

McKnight H, Chervany N. The meaning of trust. Technical Report MISRC 96-04, Management Informations Systems Research Center, University of Minnesota, MN, USA, 1996.

Milgram S. The small world problem. In Sabini J, and Silver M (eds.), *The Individual in a Social World—Essays and Experiments* 2nd edition. New York, NY, USA, McGraw-Hill, 1992.

Mui L, Mohtashemi M, Halberstadt A. A computational model of trust and reputation. In: *Proceedings of the 35th Hawaii International Conference on System Sciences*, Big Island, HI, USA, January 2002:188–196.

Page L, Brin S, Motwani R, Winograd T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

Quillian R. Semantic memory. In Minsky M, (ed.), *Semantic Information Processing*, Boston, MA, USA, MIT Press, 1968:227–270.

Rapoport A. Mathematical models of social interaction. In Luce D, Bush R, and Galanter E (eds.), *Handbook of Mathematical Psychology*, volume 2. New York, NY, USA Wiley, 1963.

Reiter M, Stubblebine S. Path independence for authentication in large-scale systems. In: *ACM Conference on Computer and Communications Security*, 1996:57–66.

Reiter M, Stubblebine S. Toward acceptable metrics of authentication. In: *Proceedings of the IEEE Symposium on Security and Privacy*, 1997:10–20.

Richardson M, Agrawal R, Domingos P. Trust management for the semantic web. In: *Proceedings of the Second International Semantic Web Conference*, Sanibel Island, FL, USA, September 2003.

Sankaralingam K, Sethumadhavan S, Browne J. Distributed pagerank for P2P systems. In: *Proceedings of the Twelfth International Symposium on High Performance Distributed Computing*, Seattle, WA, USA, June 2003.

Smith E, Nolen-Hoeksema S, Fredrickson B, Loftus G. *Atkinson and Hilgards's Introduction to Psychology*. Thomson Learning, Boston, MA, USA, 2003.

Twigg A, Dimmock N. Attack-resistance of computational trust models. In: *Proceedings of the Twelfth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises: Enterprise Security (Special Session on Trust Management)*, Linz, Austria, June 2003:275–280.

Wiese R, Eiglsperger M, Kaufmann M. yfiles—visualization and automatic layout of graphs. In: *Proceedings of the 9th International Symposium on Graph Drawing*, volume 2265 of *LNCS*, Heidelberg, Germany, Springer-Verlag, January 2001:453–454.

Ziegler C-N. Semantic web recommender systems. In Lindner W, and Perego A (eds.), *Proceedings of the Joint ICDE/EDBT Ph.D. Workshop 2004*, Heraklion, Greece. Crete University Press, March 2004.

Ziegler C-N, Lausen G. Analyzing correlation between trust and user similarity in online communities. In: Jensen C, Poslad S, and Dimitrakos T (eds.), *Proceedings of the 2nd International Conference on Trust Management*, volume 2995 of *LNCS*, Oxford, UK, March 2004. Springer-Verlag, 251–265.

Zimmermann P. *The Official PGP User's Guide*. Boston, MA, USA, MIT Press, 1995.

Cai-Nicolas Ziegler is a post-doctoral researcher at DBIS, the Databases and Information Systems group of the University of Freiburg, Germany. He studied Computer Science at the University of Passau, Germany, and Université Laval, Québec, receiving his Diploma (equivalent to MSc) in 2003. Cai-Nicolas obtained his PhD in Computer Science from the University of Freiburg in 2005.

His primary research interests cover collaborative filtering applications and recommender systems, as well as computational trust models on the verge of human-computer interaction.

Georg Lausen is head of the research group on Databases and Information Systems (DBIS) at the University of Freiburg, Germany. He received his Diploma in Industrial Engineering in 1978, PhD in 1982, and his postdoctoral lecture qualification (Habilitation) in 1985 from the University of Karlsruhe (TH). He was associate professor for Information Technology and Integration Problems at the Technical University of Darmstadt from 1986 to 1987. From 1987 to 1994 he was full professor for databases and informations systems at the University of Mannheim, and since 1994 at the University of Freiburg.

His current research interests comprise information integration, internet technologies, and Web services.