Computing and Informatics, Vol. 34, 2015, 167–184

THE COST OF TRUST IN THE DYNAMICS OF BEST ATTACHMENT

Vincenza Carchiolo, Alessandro Longheu
Michele Malgeri, Giuseppe Mangioni

Dipartimento di Ingegneria Elettrica
Elettronica e Informatica (DIEEI)
Università degli Studi di Catania – Italy
e-mail: {Vincenza.Carchiolo, Alessandro.Longheu,
Michele.Malgeri, Giuseppe.Mangioni}@dieei.unict.it

Abstract. The need of trust is growing in several contexts as e-commerce, virtual communities, distributed on-line services and many others as an essential precautionary component for users during interactions with strangers, either other people or virtual agents. Generally trust metrics endorse the principle "the higher the trust, the more legitimate that user will be"; a consequence is that getting trusted must require some effort, otherwise all participants would easily achieve high trustworthiness. In this work we study how a user can achieve and preserve a good trust and what costs it requires over time; we also investigate some heuristics that allow reducing the complexity in exploring the rank-effort space especially for large networks.

Keywords: Trust, network dynamics, rank, complex networks, best attachment

Mathematics Subject Classification 2010: 05-C82, 91-D30

1 INTRODUCTION

The need of underlying trust networks is growing in several contexts such as e-commerce, virtual communities, distributed on-line services and many others [1, 2, 3]. Considering for instance the case of e-commerce, the exponential growth of Internet in the last decade determined new types of customer-seller behaviours in addition to those based on the traditional *face-to-face* pattern; such new interactions often require trust for positive achievements [4].

Beyond the nature of the service per se, the way they are accessed is also relevant, especially in the post-PC era where the role of embedded, distributed and mobile devices is becoming central; on one hand this led to new and fascinating concepts as *pervasive* and *ubiquitous* computing, *everyware*, *Internet of Things* and others [5, 6, 7], on the other, this also shifted the question of trust to mobile devices that are still not ready to effectively support this need [8].

Despite their differences, in all of these scenarios trust has become an essential precautionary component for users to help them during interactions with (possibly total) strangers, either real people or virtual agents[1] [10]; trust values are usually used to rank actors (nodes of the overlay trust network) so that only the most *reliable* are considered.

Over these years, a plethora of trust models and frameworks have been developed [28, 29] generally they endorse the principle *'the higher the trust, the more legitimate that user/agent will be'*. A first consequence is that getting trusted in a network must require some cost[2], otherwise all participants would easily achieve a high (and therefore useless) trust. In addition, we focus here on the network dynamics, i.e. changing in rank classification due to nodes joining or leaving the network; note that others adopt a different meaning for *dynamics*, e.g. [17] defines it as the evolution of trust over time due to the change in nodes behaviour.

Given these premises, our goal is to study how a node can achieve and possibly preserve a good trust (therefore a high rank) and what costs this requires over time.

In particular, we consider a new node that joins the network and which aim is to get a high rank with a low cost. It establishes the rank to achieve, and it also assigns a budget it can spend to increase its rank. During its lifetime, it initially establishes a certain number of trust links with other existing nodes to get trusted (and to improve its rank); after a while, the node wants to increase its rank keeping the cost as low as possible, thus a couple of question arise:

1. which new nodes should it connect to?
2. should it preserve existing trust links or not?
3. what are costs associated with these actions?

In a few words, we want to investigate the *trajectory* the node has to follow to guarantee the best trade-off between its rank and related costs.

This scenario is frequent in several context, for instance, let us consider a company that wants to increase its visibility on the marketplace; to do this, it plans both a target position to achieve and a corresponding capital investment for its promotion it has to support over years. Our goal then can be reformulated as the search for the best advertising plan for that company.

[1] in this paper, we will use the term agent, peer, person or node indifferently

[2] in the following, we will use the term *effort*

This task can be hard to solve, especially when large trust networks are considered, as frequently happens in real scenarios; to get an answer for the best rank-cost trade-off without performing exhaustive experiments with all possible links combinations in this paper we propose a heuristic aiming at exploiting just *relevant* links combinations, preserving both the effectiveness (rank enhancement) and the efficiency (less costs as possible). Moreover, we compare this approach with a suboptimal proposal that aim at finding the best rank for a given effort. Some considerations arise from this comparison, as illustrated later.

This work was originally presented in [27], where first experiments on the proposed heuristic were presented. Here, we aim at validate such results through the comparison with another approach. The paper is organized as follows. In Section 2 we provide definitions and the model of trust network we are focusing on whereas in Section 3 we propose a heuristic to avoid the complexity of considering exhaustive approach. In Section 4 we show simulations used to validate the proposed heuristic, together with the comparison with a different heuristic, finally providing concluding remarks and future works in Section 6.

2 THE MODEL OF THE TRUST NETWORK

The scenario we consider is that of a trust network, modeled as a graph where the nodes (N) are agents (persons, devices, resources etc.) and the arcs (E) represent trusting relationships, each labeled with a measure ($\mathcal{L}$) of the trust value according to a given metric. This model is widely accepted in literature [18, 19, 20] and largely inspired by the PageRank algorithm [21] that allows us to assess a steady global trust value associated to each node.

Although the trustworthiness per se is important to study the evolution of the network, a key role here is played by the *rank*, i.e. the placement each node achieves with respect to others. We point out that a rank of a node is proportional to its trustworthiness, i.e. the more trusted the node, the better its rank. Here we simply defined the rank as the position a node gets when they are ordered in a descending way according to their trust values; other analytical trust-vs-rank relationships could be considered, however this is out of our scope.

Moreover, in the following we do not impose any specific metric for trust. Among many proposals that exist in literature [14, 15, 16, 17, 22], we used EigenTrust [14] in our experiments since it is one of the simplest yet efficient metric in evaluating global trust; note, however, that any other metric can be adopted, since our proposal does not rely on specific metric features.

A further hypothesis is that the agent joining the trust network is supposed to be *honest*, so it simply aims at gaining the best rank and has no other goals, as for instance using its (good) trust to subvert others' trust values; if so, a limit to the trust achievable by a node must be established, though we are not addressing such cases here.

To complete the scenario overview, we also have to formalize the attachment *effort* cited in the introduction; indeed, in real networks an existing node does not trust new ones unless it assumes some responsibility and/or offers some service, for instance in P2P networks a peer must guarantee a given bandwidth or a minimum number of replica files, or in a social network a person must somehow sacrifice his/her privacy to be known (hence, trusted) by others. We then define the *effort* as the cost the agent X bears to persuade another agent Y to trust it; of course, if X aims at being trusted by more nodes it has to spend more, therefore the effort of X is:

$$\mathit{effort}_X = \sum_j c_{jX} \tag{1}$$

where c_{jX} are the normalized local trust values X receives from his neighbours j according to the EigenTrust metric we chose.

Simply put, the effort measures how many trust links the new agent must collect from others until it obtains a given trust; note that using values provided by EigenTrust does not affect this definition, i.e. any other metric can be used as well.

Finally, for the sake of simplicity, before the new agent X joins the network, all existing arcs are labeled with 1.0 as trust value. We choose this setting in order to avoid the distribution of trust values affects our simulation results. Note that the 1.0 trust value just refers to each *direct* trust, the global trust of each agent is instead evaluated with the EigenTrust metric and falls in the range [0,1];

As soon as X joins the network, some questions arise:

1. which (existing) nodes should X connect to in order to get a good rank since the beginning?
2. should X retain previously established links when adding new ones or a better rank could be achieved by also replacing some of the existing trust links?

In a few words, which is the best set of trust links that allows a node to achieve the best rank?

The idea to investigate a non-random network attachment is also present in other works, e.g. [23], where focusing on social networks leads to consider the assortativity [24] and the clustering effect [25] as properties that determine a dynamic behavior of non-random networks. The study we perform here is conducted on both random and scale-free networks, and we do not assume any hypothesis on the type of network (e.g. social, biological, etc.).

The simplest approach to study how trust values affect ranks evolution is a *brute force* algorithm, where we attempt to consider all configurations and compare the results [26]. However, this approach is too complex for (even not so) large size networks: given a node X attempting to join a network with N nodes, assuming that local trust-values are $\{0, 1\}$, the available configurations to analyze are $2^N - 1$. Being this a value that easily explodes even for little networks, we need a criterion for discarding those configurations that will not lead to a high rank. In the next section we present our heuristic approach based on some statistical considerations.

3 REDUCING THE EFFORT-RANK SPACE

In order to study a strategy that allows a joining node to gain the best rank, we are interested in exploring the effort-rank space. For a network with N nodes this means to compute the rank for all links combinations a joining node can have with the N nodes of the network; since such number is $2^N - 1$, this makes practically unfeasible to study many real networks. For instance, the network used in our experiments has $N = 20$ nodes meaning that a brute force algorithm has to take into account 1 048 575 configurations. This approach permits to study networks whose dimension is up to 25–30 nodes (a network with 30 nodes requires 1 073 741 823 configurations to be analyzed), but real networks can be even bigger. To solve this problem, we want to reduce the configurations by discarding those not *relevant* via the heuristic approach explained in the following.

The driving idea is based on the fact that both rank and effort can range over a limited amount of values, therefore it is possible to collapse all configurations achieving a given rank into just one.

Each node can assume a trust-based rank bounded to an upper limit that linearly depends on the number N of nodes of a network: it ranges indeed from 1 to N (where 1 is the best and N the worst rank). Therefore, a node attempting to join a network whose size is N can gain a rank in the range $[1, N+1]$, so we can study no more than $N + 1$ configurations. Moreover, also the effort_X linearly depends on N and ranges from 1 to N in the hypothesis that local trust values are in $\{0, 1\}$.

Thanks to the linear dependence of both rank and effort, we can simply find the upper bound of the number of points in the effort-rank space that is $N \cdot (N+1)$, much lesser than $2^N - 1$. This can be explained by the fact that several configurations (corresponding to several trust-values) actually provide the same rank.

Since we are interested in the rank a node can assume, it would be interesting to find a way to directly explore this space. This leads to a considerable decrease in the number of configurations to compute. For instance, for a network with $N = 20$ nodes, a brute force algorithm has to take into account 1 048 575 configurations, whilst the number of rank points are 421 $(20 \cdot (20 + 1))$ at most, thanks to the fact that many configurations map to one rank, for example $\mathit{effort}_X = 10$ leads to the same rank 184 756 times.

Based on these considerations, the best approach (i.e. with less computational time cost) to explore the effort-rank space is to compute at most $N \cdot (N + 1)$ points. The problem is to select exactly $N \cdot (N + 1)$ attachment configurations among the possible $2^N - 1$ that guarantee the maximum coverage. In other words, we want to exclude those attachment configurations that produce the same points in the effort-rank plane. To do this we need a heuristic that permits to select ideally just one attachment configuration per point.

The heuristic we propose in this work is based on the analysis of the statistical distributions of the attachment configurations in the effort-rank plane. To introduce it we need some notations.

We call in^X the in-set of a node X (i.e. the set of nodes for which an outgoing link to the node X exists):

$$in^X = \{j : (j, X) \in E\} \tag{2}$$

We also refer to this set as the *attachment configuration* of the node X. Note that the cardinality of in^X is equal to the effort_X just in the case the direct local trust values are $\{0, 1\}$.

A new node joining the network can be linked to several different nodes, thus originating several different attachment configurations. Let us suppose to enumerate such configurations by using a subscript index. So, let us consider for a given effort two attachment configurations of the node X, say in_p^X and in_q^X; we define their distance as:

$$d\left(in_p^X, in_q^X\right) = \left|in_p^X - in_q^X\right| \tag{3}$$

i.e. the cardinality of the difference between the two in-sets of the node X. Note that the minimum distance d is 1, while the maximum value is given by:

$$\mathit{maximum}\left\{d\left(in_p^X, in_q^X\right)\right\} = \min\left\{\left|in_p^X\right|, N - \left|in_p^X\right|\right\} \tag{4}$$

To find a good heuristic in exploring the effort-rank space, we examined the statistical distribution of the distance, considering all the configurations having the same effort. In Table 1 we show some of the $2^N - 1$ probabilities that two configurations with the maximum distance either belong (P_{in}^{maxd}) or not (P_{out}^{maxd}) to the same point in the effort-rank space. In other terms, given an attachment configuration A, the table reports the chance that another configuration B whose distance from A is the maximum (as defined by Equation (4)) leads to a different rank value (different point in the space). The Table 1 reports only the first points, i.e. with effort 1, 2 and 3.

Except for the case with effort 1 and rank 20 where only one point is present, Table 1 highlights that the configurations whose distance is maximum have a high probability to get a different point of the plane. This distribution suggests a strategy to generate attachment configuration: for a given effort, the configuration must be chosen in such a way that the distance between any two of them is maximum.

In summary, the heuristic we propose is to start with a given configuration and, for each value of the effort, to generate a sequence of other attachment configurations whose distance from the previous one is the maximum (as given by Equation (4)). Then we apply again the previous step by starting from a new attachment configuration just in case we have not found all the $N + 1$ points of the plane. This approach allows to decrease drastically the number of attachment configurations to analyze as shown in details in the next section.

Effort	**Rank**	P_{in}^{maxd}	P_{out}^{maxd}
1	20	1	0
2	20	0.3312	0.6688
2	19	0.5337	0.4663
2	18	0.0514	0.9486
3	19	0.0082	0.9918
3	18	0.0159	0.9841
3	17	0.1258	0.8742
3	16	0.2041	0.7959
3	15	0.1978	0.8022
3	14	0.1448	0.8552
3	13	0.1151	0.8849
3	12	0.0710	0.9290
3	11	0.0507	0.9493
3	10	0.0146	0.9854
3	9	0.0088	0.9912
3	8	0.0015	0.9985
...	...	...	...

Table 1. Probability to stay/to escape in/from a given point of the effort/trust-rank plane

4 TESTING THE HEURISTIC

The heuristic discussed in the previous section facilitates the study of dynamics and behavior of large networks, but we still have to validate our approach by comparing the results provided by the brute force algorithm with those coming from the proposed heuristic. Given the cited computational limit concerning the size of network, we compare such results for networks with size of 20 nodes.

To avoid the introduction of biasing we synthesized several networks with different distributions (random, scale-free) and created several ad-hoc topologies (we call them *regular*) that preserve the generality of outcoming networks. Each group of networks has been studied using both the proposed heuristic and brute force algorithm.

The Figure 1 reports the effort-rank graph for a network of 20 nodes analyzed with the brute force algorithm, therefore all the possible attachment configurations are present. Each point in the graph may represent several configurations (some points result from more than 156 000 input configurations) raising the complexity and time of simulations. To qualitatively distinguish such points, different colors are used to indicate the *density*, in particular the more input configurations map to the same point, the darker that point is represented in figure.

The Figure 2 reports the same graph of Figure 1 applied to the same networks but using the proposed heuristic. The figures highlight that results are a good approximation since almost all points (120 over 160) have been found, showing the same pattern (dynamics) and limits; the points that have been discarded are not

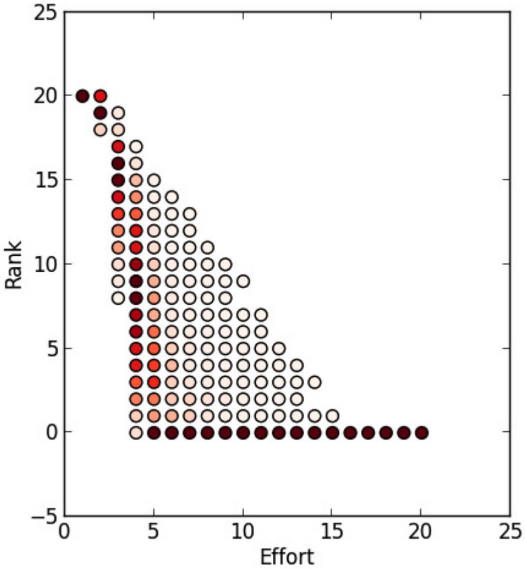


Figure 1. Regular net – brute force analysis

particularly meaningful for the attachment strategy and the rank distribution (being the pattern preserved from Figure 1 to Figure 2). Also note that in columns where the heuristic approach does not capture all points, e.g. that for *effort* = 3, the most significant points to define the correct range of rank are still detected. The outcome of capturing the correct range is very important to define the correct attachment strategy, i.e. the points to connect to obtain the best rank starting from current configuration. We believe this is the most common case, since, in a real world, a node is not likely to drop (possibly recent) connections established with other nodes.

Of course the density is not preserved by the heuristic approach due to heuristic itself that tries to find as less configurations as possible for each point in the effort-rank space.

We also compared the brute force algorithm with the heuristic on synthesized random and scale-free networks, both with a dimension of 20 nodes. The Figures 3 and 4 show the results that are similar for all networks we analyzed, thus confirming that the heuristic approach captures both the general dynamics and most of the significant points to define a step-by-step attachment strategy aiming at obtaining that best rank with less effort.

The time complexity we obtained is about 90 % of the computational time thanks to the reduction of the configurations to consider: about 10 000 instead of the 1 048 575.

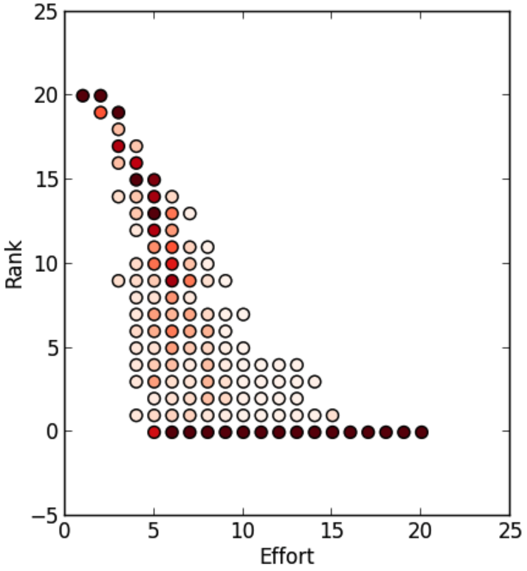


Figure 2. Regular net – heuristic analysis

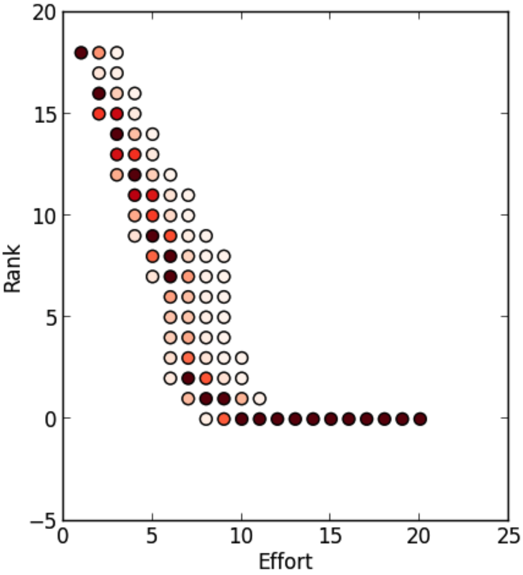
5 TRYING TO GET THE BEST RANK

In the previous section we introduced a heuristic that allows exploring the effort-rank space more efficently than the brute force approach, while mainly preserving its effectiveness. Sometimes, in real trust networks the interest is focused mainly on those configurations that provide the best rank for a given value of the effort, for instance a company may establish a certain budget to be spent over a couple of years and its managers are interested on what position they can strive for the company on the marketplace with such an investment. In this case, we do not need to map the whole effort-rank space, but we just want to know the best rank for each effort value.

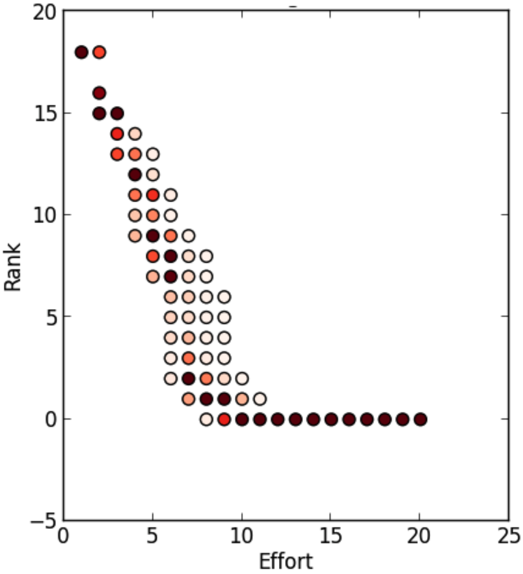
This goal can be viewed as a special case of the problem we addressed before with the proposed heuristic and to this purpose we can adopt the approach proposed in [30]. In particular, in the heuristic proposed to achieve the best rank (actually, Pagerank in [30]) for a new node X, the node first receives a backlink from the most trusted node in the network. Then, X is backlinked by the most trusted node of this new network, and this approach is followed until the given number of (back)links is reached.

A brief comparison with our heuristic shows that

1. the number of backlinks represents the effort,
2. the rank is anyway trust-based (Eigentrust vs Pagerank) and

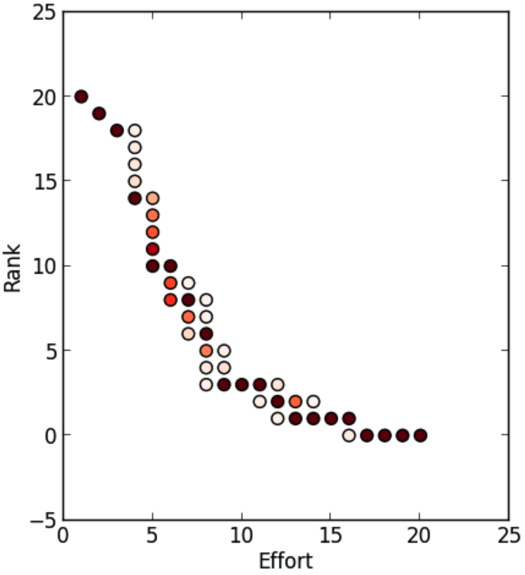


a)

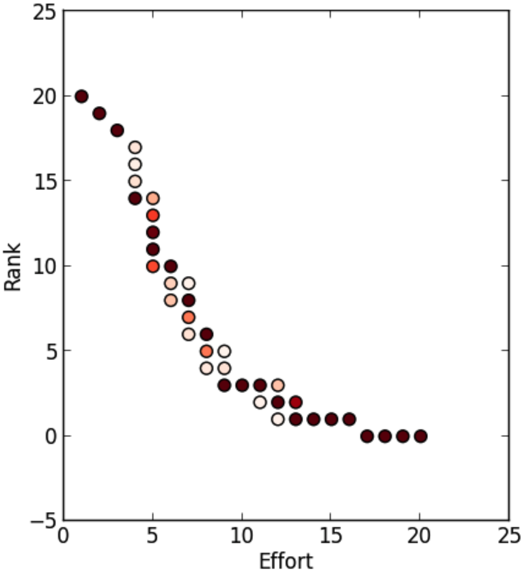


b)

Figure 3. Random networks simulations: a) brute force, b) heuristic



a)



b)

Figure 4. Scale-free networks simulations: a) brute force, b) heuristic

3. the main difference is that in [30] previously established links are preserved, whereas we do not impose such a constraint.

For what concerns the complexity, in [30] it is polynomial, since at each step just one backlink among existing nodes is created, whereas our heuristic is slower and more time consuming, even if the two heuristics actually aim at providing different information: just optimal effort-rank combinations in [30] vs the whole effort-rank space – the one we propose.

To compare both heuristics, and also to get a validation feedback about our proposal, we applied the algorithm described in [30] to the same networks used in the previous section. Figure 5 represents the regular network case, whereas Figure 6 presents results for random networks and finally Figure 7 shows the case of scale-free networks.

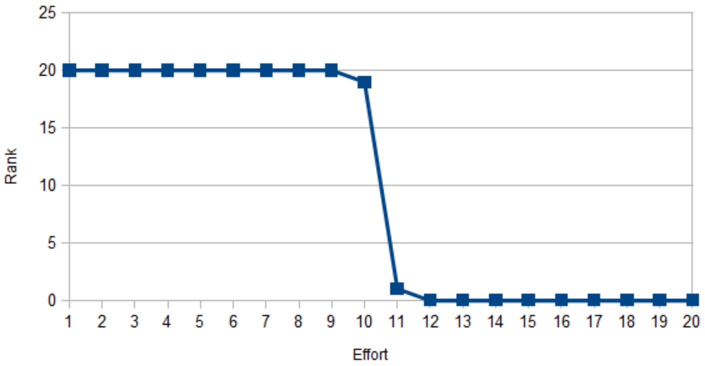


Figure 5. Regular networks

Similarly as for our heuristic, we performed several tests using different networks for each type (random, scale-free as well as regular), always with 20 nodes; in this sense, note that Figures 5, 6 and 7 actually show only one result for each type of network, though results for all networks always show the same trend of the sample figures.

In particular, we note that for all networks the rank trend shows a slope whose position depends on the number of links and also on network being considered (indeed, we observed the discontinuity in different positions for different networks even of the same type); probably this could also depend on the small size of the networks considered, therefore this deserves further studies on larger networks.

The most important remark we point out is the fact that Olsen's heuristic actually find a rank that could also not be the best achievable for a given effort. For instance, in Figure 5 for an effort 3 we can reach a rank 20, whereas the brute force approach (Figure 1) allows reaching a rank 8, significantly better. Our heuristic

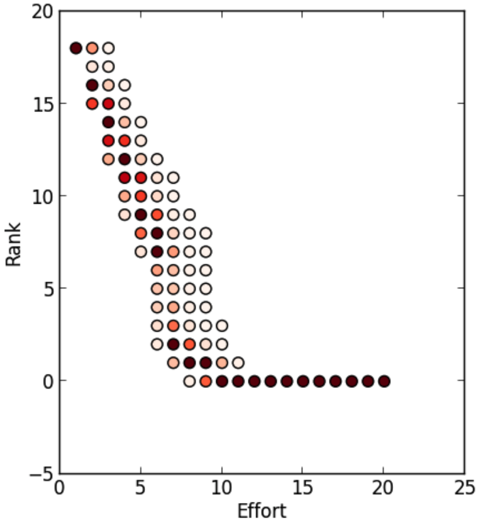


Figure 6. Random network

probably does not allow to reach the best rank as in the brute force case, as we illustrated previously, though in Figure 2 we note that the effort 3 allows reaching a rank 9 (worse than 8 but better than 20). The same fact occurs for other effort values and also for other network types; for instance in random networks (Figure 6), for effort 3 we have rank 18, whereas the same effort determines rank 13 with our heuristic (Figure 3 b)) and 12 with brute force approach (Figure 3 a)). Similar be-

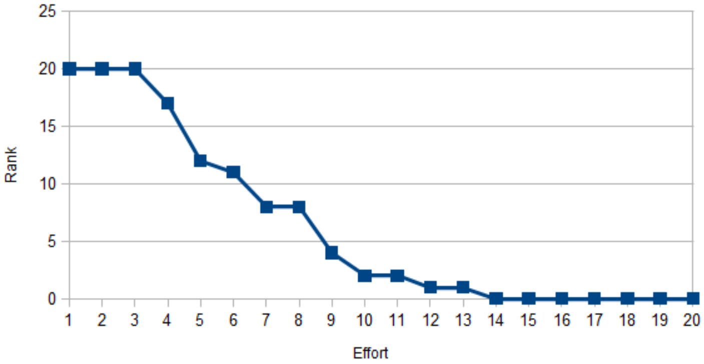
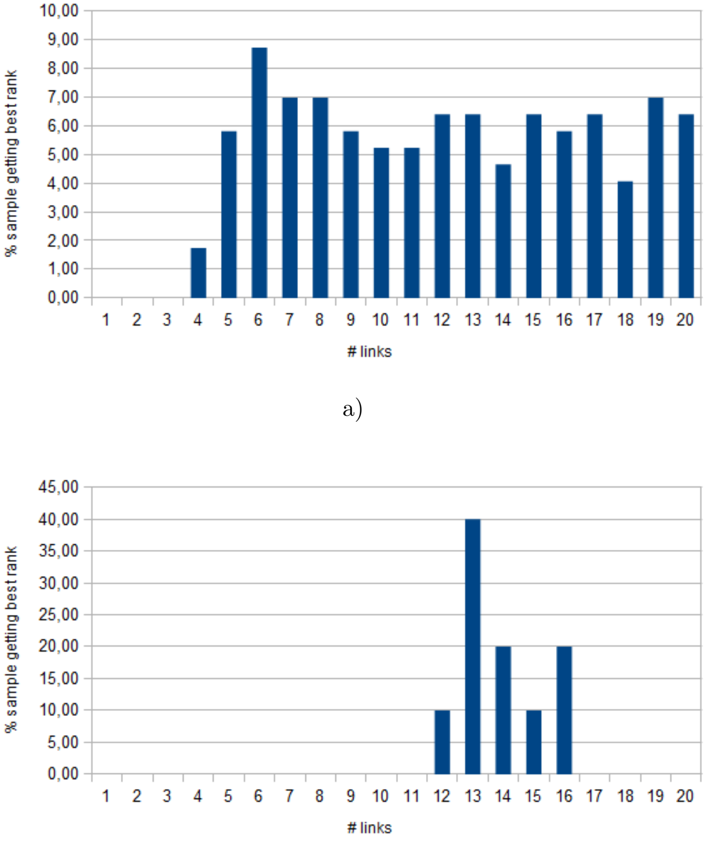


Figure 7. Scale free networks



a)

b)

Figure 8. Getting the best rank: a) ad-hoc network, a) scale-free network

haviour is present in scale-free networks, e.g. with effort 4 we get rank 17 in Figure 7 and 14 for both brute force and our heuristic (see Figure 4). Scale-free networks also show that getting the best rank is quite difficult (see Figure 7), whereas the results for networks with a more regular topology spread over the complete range of rank values.

We can conclude that Olsen's algorithm does not provide the best rank, though is very efficent in finding a suboptimal solution. Our heuristic is more time consuming but results represent a better approximation of the exhaustive (brute force) case and, moreover, it also provides (almost) a complete map of the effort-rank space. Results also show that preserving existing backlinks while trying to increase the rank (as in Olsen's approach) does not lead to good ranks, rather each time we increase the number of backlinks by 1 we may discard some or all of previous backlinks to achieve a better position (as we do in our proposal).

Finally, one more question to answer is *how many links are needed to get the best rank*; a correct answer allows an agent to correctly evaluate duration and cost in advance. The Olsen's approach highlights that results strong depend on the network topology, see Figure 8 for the case of scale-free and regular (ad-hoc) networks.

6 FINAL REMARKS

The interest in algorithms and mechanism to evaluate trust and reputation in large networks is becoming central due to the spreading of social-based networks, mainly if non central authority or server could simplify the scattering of information.

Unfortunately the study of huge networks is often not feasible due to complexity matter, therefore some heuristic is necessary to increase the dimension of the network being studied.

The paper presents and compares two approaches aiming at reducing the complexity to use algorithm based on random walker that strongly depends on the number of nodes and arcs of the networks.

A problem the both approaches try to solve is *to find k arcs connecting a new node X that maximizes the trust of X*, but the strategy is quite different: the first aims at exploiting the locality of nodes and explore all configuration that get same results, the second is a typical greedy algorithm that finds the next best solution without analyzing backtracking.

We present the result of simulation on several simple networks whose topology and characteristics is as simple as an exhaustive approach can be used in order to compare the heuristic with real behavior.

REFERENCES

[1] Thatcher, J.—McKnight, D.—Baker, E.—Arsal, R.—Roberts, N.: The Role of Trust in Postadoption IT Exploration: An Empirical Examination of Knowledge Management Systems. Engineering Management, IEEE Transactions, Vol. 58, 2011, No. 1, pp. 56–70.

[2] Huang, J.—Fox, M. S.: An Ontology of Trust: Formal Semantics and Transitivity. Proceedings of the 8th International Conference on Electronic Commerce (ICEC '06), ACM, New York, NY, USA, pp. 259–270.

[3] Wang, Y. D.—Emurian, H. H.: An Overview of Online Trust: Concepts, Elements, and Implications. Computers in Human Behavior, Vol. 21, 2005, No. 1, pp. 105–125.

[4] Fung, R.—Lee, M.: Ec-Trust (Trust in Electronic Commerce): Exploring the Antecedent Factors. Proceedings of the 5th Americas Conference on Information Systems. 1999, pp. 517–519.

[5] Armbrust, M.—Fox, A.—Griffith, R.—Joseph, A. D.—Katz, R.—Konwinski, A.—Lee, G.—Patterson, D.—Rabkin, A.—Stoica, I.—Zaha-

ria, M.: A View of Cloud Computing. Commun. ACM, Vol. 53, 2010, No. 4, pp. 50–58.

[6] Hansmann, U.—Nicklous, M. S.—Stober, T.: Pervasive Computing Handbook. Springer-Verlag New York, Inc., New York, NY, USA 2001.

[7] Nieuwdorp, E.: The Pervasive Discourse: An Analysis. Comput. Entertain., Vol. 5, 2007, No. 2, p. 13.

[8] Vasudevan, A.—Owusu, E.—Zhou, Z.—Newsome, J.—McCune, J. M.: Trustworthy Execution on Mobile Devices: What Security Properties Can My Mobile Platform Give Me? Proceedings of the 5th international conference on Trust and Trustworthy Computing (TRUST '12), Springer-Verlag, Berlin, Heidelberg, 2012, pp. 159–178.

[9] Hasan, Z.—Krischkowsky, A.—Tscheligi, M.: Modelling User-Centered-Trust (UCT) in Software Systems: Interplay of Trust, Affect and Acceptance Model. In: Katzenbeisser, S., Weippl, E., Camp, L., Volkamer, M., Reiter, M., Zhang, X. (Eds.): Trust and Trustworthy Computing, Springer, Berlin, Heidelberg, LNCS, Vol. 7344, 2012, pp. 92–109.

[10] Vance, A.—Elie-Dit-Cosaque, C.—Straub, D.: Examining Trust in Information Technology Artifacts: The Effects of System Quality and Culture. J. Manage. Inf. Syst., Vol. 24, 2008, No. 4, pp. 73–100.

[11] Santos, N.—Gummadi, K. P.—Rodrigues, R.: Towards Trusted Cloud Computing. Proceedings of the 2009 Conference on Hot Topics in Cloud Computing (HotCloud' 09), USENIX Association, Berkeley, CA, USA, 2009.

[12] Dietrich, K.—Winter, J.: Implementation Aspects of Mobile and Embedded Trusted Computing. Proceedings of the 2nd International Conference on Trusted Computing (Trust '09), Springer-Verlag, Berlin, Heidelberg, 2009, pp. 29–44.

[13] Wang, Y.—Norcie, G.—Cranor, L. F.: Who Is Concerned about What? A Study of American, Chinese and Indian Users' Privacy Concerns on Social Network Sites. Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST '11), Springer-Verlag, Berlin, Heidelberg, 2011, pp. 146–153.

[14] Kamvar, S. D.—Schlosser, M. T.: The Eigentrust Algorithm for Reputation Management in P2P Networks. Proceedings of the Twelfth International World Wide Web Conference, 2003.

[15] Zhou, R.—Hwang, K.—Cai, M.: Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks. IEEE Trans. on Knowl. and Data Eng., Vol. 20, 2008, No. 9, pp. 1282–1295.

[16] Zhou, R.—Hwang, K.: Powertrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. IEEE Trans. Parallel Distrib. Syst., Vol. 18, 2007, No. 4, pp. 460–473.

[17] Walter, F. E.—Battiston, S.—Schweitzer, F.: Personalised and Dynamic Trust in Social Networks. In: Bergman, L. D., Tuzhilin, A., Burke, R. D., Felfernig, A., Schmidt-Thieme, L. (Eds.): RecSys, ACM, 2009, pp. 197–204.

[18] Marsh, S.: Formalising Trust as a Computational Concept. Technical report, University of Stirling, Ph.D. thesis, 1994.

[19] Golbeck, J. A.: Computing and Applying Trust in Web-Based Social Networks. Ph.D. thesis, College Park, MD, USA, 2005, Chair-Hendler, James.

[20] Walter, F. E.—Battiston, S.—Schweitzer, F.: A Model of a Trust-Based Recommendation System on a Social Network. Journal of Autonomous Agents and Multi-Agent Systems, Vol. 16, 2008.

[21] Berkhin, P.: A Survey on Pagerank Computing. Internet Mathematics, Vol. 2, 2005, No. 1, pp. 73–120.

[22] Xiong, L.—Liu, L.: Peertrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Trans. Knowl. Data Eng., Vol. 16, 2004, No. 7, pp. 843–857.

[23] Allodi, L.—Chiodi, L.—Cremonini, M.: Modifying Trust Dynamics Through Cooperation and Defection in Evolving Social Networks. Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST '11), Springer-Verlag, Berlin, Heidelberg, 2011, pp. 131–145.

[24] Newman, M. E.: Assortative Mixing in Networks. Physical Review Letters, Vol. 89, 2002, No. 20, 208701.

[25] Opsahl, T.—Panzarasa, P.: Clustering in Weighted Networks. Social Networks, Vol. 31, 2009, No. 2, pp. 155–163.

[26] Carchiolo, V.—Longheu, A.—Malgeri, M.—Mangioni, G.: Gain the Best Reputation in Trust Networks. In: Brazier, F., Nieuwenhuis, K., Pavlin, G., Warnier, M., Badica, C. (Eds.): Intelligent Distributed Computing V, Studies in Computational Intelligence, Springer, Berlin, Heidelberg, Vol. 382, 2012, pp. 213–218.

[27] Carchiolo, V.—Longheu, A.—Malgeri, M.—Mangioni, G.: A Heuristic to Explore Trust Networks Dynamics. In: Zavoral, F., Jung, J. J. and Badica, C. (Eds.): Intelligent Distributed Computing VII, Proceedings of the 7th International Symposium on Intelligent Distributed Computing (IDC 2013), Prague, Czech Republic, September 2013, Studies in Computational Intelligence, Springer, Berlin, Heidelberg, Vol. 511, 2014, pp. 67–76.

[28] Riegelsberger, J.—Sasse, M.—McCarthy, J.: The Mechanics of Trust: A Framework for Research and Design. International Journal of Human-Computer Studies, Vol. 62, 2005, No. 3, pp. 381–422, ISSN 1071-5819, `http://dx.doi.org/10.1016/j.ijhcs.2005.01.001`.

[29] Corritore, C. L.—Kracher, B.—Wiedenbeck, S.: On-Line Trust: Concepts, Evolving Themes, a Model. International Journal of Human-Computer Studies – Special Issue: Trust and Technology, Vol. 58, 2003, No. 6, pp. 737–758.

[30] Olsen, M.—Viglas, A.: On the Approximability of the Link Building Problem. Theoretical Computer Science, Elsevier BV, 2013.

Vincenza Carchiolo is Full Professor of computer science at the DIEEI, University of Catania. She received her degree in electrical engineering from the University of Catania in 1983. Her research interests include information retrieval, query languages, distributed systems, and formal languages. She co-authored more than 70 scientific papers in refereed journals and conferences.

Alessandro Longheu is a computer science adjunct researcher at the DIEEI, University of Catania. He received his M.Sc. degree in computer engineering in 1997 and his Ph.D. degree in 2001. He has been a research fellow and Adjunct Professor of computer science at the University of Catania and University KORE of Enna. His research interests include complex networks and trust, e-learning, production workflows and information retrieval. He co-authored more than 60 scientific papers in refereed journals and conferences.

Michele Malgeri is Associate Professor of computer science at the DIEEI, University of Catania. He received his degree in electrical engineering from the University of Catania in 1983. His research interests include information retrieval, security, distributed systems, trust and reputation. He is a PC member of several conferences and workshops. He co-authored more than 70 scientific papers in refereed journals and conferences.

Giuseppe Mangioni is Assistant Professor of computer science at the DIEEI, University of Catania. He received his degree in computer engineering in 1995 and his Ph.D. degree in 2000 from the University of Catania. His research interests include peer-to-peer systems, trust and reputation systems, self-organizing and self-adaptive systems and complex networks. He co-authored more than 70 scientific papers in refereed journals and conferences.