



A MULTI-AGENT APPROACH FOR TRUST-BASED SERVICE DISCOVERY AND SELECTION IN SOCIAL NETWORKS

AMINE LOUATI, JOYCE EL HADDAD, AND SUZANNE PINSON*

Abstract. Service discovery and selection approaches are often done using a centralized registry-based technique, which only captures common Quality of Service criteria. With more and more services offered via *social networks*, these approaches are not able to evaluate *trust* in service providers and often fail to comply with new requester's expectations. This is because these approaches are not able (i) to take into consideration the social dimension and (ii) to capitalize on information resulting from previous experiences. To address these challenges, we propose the use of *multi-agent systems* as they have demonstrated the capability to use previous interactions, knowledge representation and distributed reasoning, as well as social metaphors like trust. More precisely, in this paper, we enhance service discovery and selection processes by integrating the societal view in trust modeling. Based on relationships between agents, their previous experiences and extracted information from social network, we define a trust model built upon *social*, *expert* and *recommender*-based components. The social-based component judges whether the provider is worthwhile pursuing before using his services (viz. trust in sociability). The expert-based component estimates whether the service behaves well and as expected (viz. trust in expertise). The recommender-based component assesses for an agent whether one's can rely on its recommendations (viz. trust in recommendation). However, when searching for a service in a social network, agents (service requester and service providers) may have no direct interactions or previous experiences. This requires a method to infer trust between them. Based on a probabilistic model, we estimate trust between non adjacent agents while taking into account roles (recommender or provider) of intermediate agents. Moreover, we propose a distributed algorithm for trustworthy service discovery and selection using referral systems in social networks. Experiments demonstrate that our approach is effective and outperforms existing ones, and can deliver more trustworthy results.

Key words: Service Discovery and Selection, Multi Agent System, Trust Model, Referral Systems, Social Networks.

AMS subject classifications. 68T42, 91D30, 90B22, 62F07

1. Introduction. More and more services are proposed via *social networks* (SN). In such context, a requester needs trustful and efficient methods to discover requested services given by trustees and to choose among them. Existing service discovery and selection approaches are generally based on non-functional service descriptions usually done through quality-driven techniques using QoS attributes and centralized registries (e.g. ebXML). In most of these approaches, trust is not considered ([3, 8, 11]). Recent research works ([28, 32, 38, 50]) have shown that trustworthiness is a key factor for good service selection in order to comply with requester's expectations.

This paper proposes a new approach to discover and select services based on trust. Liu and Wang in [32] defined trust as "the belief of one participant in another, based on their interactions, in the extent to which the future action to be performed by the latter will lead to an expected outcome". Several authors have proposed to integrate the societal view into the classical service discovery process [4, 9, 28, 31]. The societal view takes into account various sources of trust information located in the social network of the trustor and considers past interaction between agents. Sabater and Vercauter distinguished in [44] three principal sources of information that agents can use to build trust: (i) individual experience between the trustor and the trustee which comes from a direct interaction between them, (ii) communicated experience which represents knowledge coming from other agents describing their interactions with the trustee and, (iii) social information which may encompasses a variety of semantic and structural knowledge useful to evaluate trust (see Sect. 4 for description).

In this work, we propose to use a multi-agent system (MAS) to model the service discovery and selection process where artificial agents represent requesters, providers and intermediate agents in the social network of the requester and interactions between agents are done through message exchange. Castelfranchi and Falcone [14] claimed that trust is a basis of interactions among artificial agents in multi-agent systems: one agent must trust a counter-party to be willing to carry out interactions and delegate a task. Consequently, we design a trust model for social networks that 1) assists agents in identifying trustworthy service providers and 2) gives agents the ability to reason about trust in their distributed decision-making process. Our trust model is based on real world characteristics of trust between people that are depicted in social science studies. Other fields are

* PSL, Université Paris-Dauphine, LAMSADE CNRS UMR 7243 Paris, France, {amine.louati,elhaddad,pinson}@lamsade.dauphine.fr

concerned with trust such as philosophy, socio-psychology and economics [2, 7, 36, 41]. All these fields show that trust could have multiple components [1, 27] and that each component can play a specific role in assessing the service trustworthiness. Some research works [29, 47, 51] focus on evaluating trust from a practical point of view by means of a reputation measure. However, this expertise-based component of trust presents several limits. As pointed out in [22, 42], trust evaluation comes from additional information sources different from those used in reputation computation such as social relationships between agents and their roles in the society. For example, [6, 32, 38, 43, 45] incorporate social network analysis as a part of the calculation of trust. Following previous works, our trust model is built upon three components namely trust in sociability, trust in expertise and trust in recommendation (see Sect. 2 for a review). First, a high sociability value indicates relevant providers and good recommenders. The interest of such a component lies in evaluating trust in a provider without relying on observations of other agents avoiding underlying subjectivity. Second, a high expertise value reflects the capability of a provider to offer good services that behaves as expected. Finally, a high recommendation value gives rise to good recommenders that one's can rely on their opinion.

During the service discovery and selection processes, agents use the aforementioned trust components to decide with whom to interact. However, in absence of direct interactions, a social and decentralized approach namely referral systems allow agents to cooperate by giving, pursuing and evaluating referrals [53]. In the literature, a number of techniques for trust inference in social networks can be found [19, 21, 33, 48, 49] but they usually focus on an evaluation based on a single trust value between adjacent agents regardless of their roles in the chain. In this paper, we propose a distributed algorithm for service discovery and selection in social networks that propagates trust values among agents using referral systems. Based on [48], we infer trust between non adjacent agents using a probabilistic model. This model takes into account roles (recommender or provider) of intermediate agents along the chain. Experiments demonstrate that our approach is better than two well-known trust methods, namely Bansal *et al.* [6], and Maaradji *et al.* [38], and can deliver more trustworthy results.

The paper is organized as follow. In the Sect. 2, we present a literature review of different trust-based approaches for service discovery and selection in social networks. Sect. 3 defines the main concepts used in this paper. Sect. 4 describes our trust model and its three components. In Sect. 5, we provide a detailed description of our trust-based discovery and selection approach. Sect. 6 discusses the experimental setup and the performance evaluation of our approach. Finally, Sect.7 concludes and presents perspectives for future work.

2. Related Work. In the literature, most of the existing Web service discovery and selection approaches use QoS attributes (e.g., latency, throughput, response time) to distinguish between functionally equivalent services. However, without trust consideration, selected services often fail to comply with requester's expectations. In the following, we briefly review the relevant trust-based service discovery and selection related work from three viewpoints.

2.1. Trust-based expertise. Many works have evaluated providers' trustworthiness from an expertise perspective [9, 25, 29, 47, 51] aiming to improve the quality of the underlying composition. Vu *et al.* [47] presented a QoS-enabled distributed service selection framework including trust and reputation management. They use some dedicated QoS registries to collect QoS feedbacks from consumers. In [51], Xu *et al.* proposed a reputation-enhanced QoS-based Web service discovery algorithm for service matching, ranking and selection based on existing Web services. Billhard *et al.* [9] examined the problem of service provider selection using an experience-based approach in which clients use confidence and reputation mechanisms to infer expectations of future providers' behavior from past experiences in similar situations. More precisely, when no previous experience is available for a particular provider/service pair, authors proposed a way to estimate confidence values based on past experiences with similar services. Lalanne *et al.* [25] proposed an approach for using quality of past experiences as a factor for service selection, including an analysis of the different factors that may affect the quality perceived by the end-user. Likewise, Li *et al.* [29] used users' feedbacks including ratings, opinions and relevant comments after use to estimate service reputation. All of the above approaches have evaluated trust from a subjective point of view based on the advertised QoS values.

2.2. Trust-based sociability. The expertise-based component evaluating trust between agents and services lacks social features, as for example relationships between users to make a significant assessment. To

address that, Da Silva and Zisman [17] proposed a trust model that takes into account different levels of trust among users, different relationships between users, and different degrees of importance that a user may have for certain QoS attributes. Some other works [6, 32, 38, 45] have investigated social network analysis techniques to evaluate the trust in sociability an agent may have in a provider. Bansal et al. [6] for example, evaluate providers' trustworthiness based on the degree centrality that gives an indication of their prestige in the network. However, this is a poor definition of trust, since it just uses a single measure, the degree centrality. Maraadji et al. [38], consider social proximity as an indicator to measure trust in sociability. Sierra used in [45] prestige and centrality measures to compute the strength of information flow when there is no complete independence in the opinions expressed by agents. Recently, Liu and Wang presented in [32] a service provider selection model considering both adjacent and end-to-end constraints, based on the quality of trust and a complex social network structure. This complex social network takes into consideration three concepts which are trust, social relationships and recommendation role reflecting better the real-world situations.

These works present several limits, most of them do not take into account the semantic information and deal with social networks with only one type of relationships. To cope with these limits, we integrate in the present work semantic information by including profiles and relationship types between agents in trust modeling to enhance its effectiveness and expressiveness.

2.3. Trust-based recommendation. The goal of a trust-based recommendation system is to generate personalized recommendations by aggregating the opinions of users in the trust network. According to Golbeck [19], recommendation techniques that analyze trust networks provide more accurate and highly personalized results than other classical collaborative filtering techniques. Golbeck proposed a FilmTrust site for generating predictive movie recommendations from trust in social network. This trust-based movie recommendation is grounded on knowledge extracted from annotations and user ratings added in the system. Hang and Singh [20] use a graph-based approach to recommend a node in a social network using similarity in trust networks. Al-Sharawneh and Williams developed in [5] an approach to model systems with autonomous interacting agents. Through a set of simulations, they investigated the impact of Goldbaum's innovative "Follow the Leader" strategy in social networks in the context of Web service selection. They use a recommender system that guides a user to select the best service that matches his requirements and preferences. Massa and Aversani [39] proposed a trust-based recommendation system where it is possible to search for trustworthy users by exploiting trust propagation over the trust network [40]. Trust-based recommendation techniques provide a way of disseminating trust information within a social network which is the basis for inferring trust between non adjacent users. However, similarly to trust-based expertise approaches, they only depend on subjective information provided by users.

3. Preliminaries. We describe in this section the overall concepts used in this work, i.e., social networks, services, user needs, and software agents.

Social Networks. We consider a Multi-Relation Social Network (MRSN) [46] which takes into account different types of relationships linking two nodes. For example, in Fig. 3.1 there are three different relationships between nodes: family, friend, and colleague relationships. Unlike multiplex networks [30] that deal with multi-layered graphs defined over the same set of nodes where each layer represents a different relationship, we assume that there is one single layer which contains different types of relationships. A multi-relation social network is modeled by a graph, where nodes represent agents and, an edge between two agents indicates a symmetric social relationship between them. More formally, a multi-relation social network and neighborhood are defined as follows:

DEFINITION 1. *Given a set $V = \{a_1, a_2, \dots, a_n\}$ of agents and a set R of types of symmetric relationships with $R = \{R_1, R_2, \dots, R_r\}$, a multi-relation social network (MRSN) is a connected graph $G = \langle V, E_1, E_2, \dots, E_r \rangle$ where $E_i \subseteq V \times V \forall i \in \{1, \dots, r\}$ is the set of edges w.r.t the i -th relationship and $\forall i \neq i', E_i \cap E_{i'} = \emptyset$. Let $\rho : E \mapsto R$ be a function that links edges to the relationship they represent, i.e. an edge $(a_k, a_j) \in E_i$ represents a social relationship of type R_i between a_k and a_j .*

DEFINITION 2. *Given a MRSN graph $G = (V, E)$, the neighborhood of an agent $a_k \in V$ w.r.t. a type of relationship $R_i \in \{R_1, R_2, \dots, R_r\}$, denoted $N_{R_i}(a_k)$, is defined as $N_{R_i}(a_k) = \{a_j \in V \mid (a_k, a_j) \in E_i\}$.*

In the MRSN, each agent a_k interacts with a subset of agents, called the social acquaintances SA_k . This

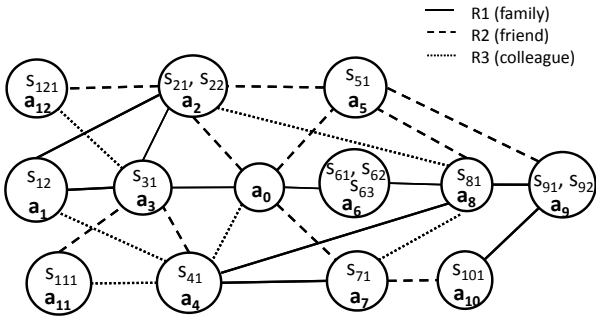


Fig. 3.1: Example of a MRSN

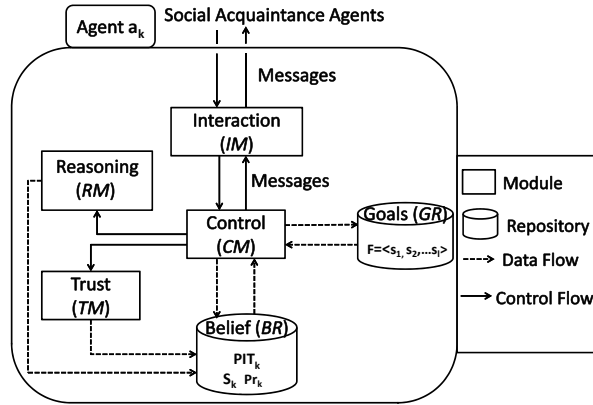


Fig. 3.2: Trust-based Agent Architecture

set represents a_k 's local view in the MRSN such as $SA_k = \bigcup_{R_i \in R} N_{R_i}(a_k)$.

Services. A service is described in terms of functionality, inputs, outputs, and non-functional attribute values. We define a service as follows:

DEFINITION 3. A service s is a n -uplet $(in, out, f, q^1, \dots, q^d)$ where in is a set of inputs required to use the service, out is a set of outputs provided at the completion of the service, f is a functionality describing the provided capacity, and q^1, \dots, q^d are the advertised values of the d non-functional criteria.

User needs. A user communicates his needs by expressing a set of required services and his preferences over relationship types and trust threshold values.

DEFINITION 4. A query Q is a 5-uplet $(F, U, \alpha, \beta, \mu)$ where $F = \{s_1, s_2, \dots, s_l\}$ is a set of the required services, $U : R \mapsto [0, 1]$ is an utility function expressing user's preferences over relationship types in the social network, $\alpha \in [0, 1]$ is a trust in sociability threshold, $\beta \in [0, 1]$ is a trust in recommendation threshold and $\mu \in [0, 1]$ is a trust in expertise threshold.

The utility of a relationship R_i , where $i \in \mathbb{N}^*$ is the rank of R_i in the preference order, can be defined as follows $U(R_i) = \frac{1}{2^{i-1}}$. For example, consider the MRSN in Fig. 3.1 with three relationship types $R_1 = family$, $R_2 = friend$ and $R_3 = colleague$, and a user's preferences over them such as $family \succ friend \succ colleague$. Therefore, $U(R_1) = 1, U(R_2) = \frac{1}{2}$ and $U(R_3) = \frac{1}{4}$.

Software Agents. We have chosen a deliberative architecture as in [12, 18] in order to enable agents to evaluate their trust in other agents before interacting with them. It is composed of four modules and two data repositories as shown in Fig. 3.2. The four modules are: the reasoning module \mathcal{RM} , the trust module \mathcal{TM} , the control module \mathcal{CM} and the interaction module \mathcal{IM} . The two data repositories are: the belief repository \mathcal{BR} and the goal repository \mathcal{GR} .

DEFINITION 5. An agent a_k is defined as a 6-components structure $\langle \mathcal{BR}, \mathcal{GR}, \mathcal{RM}, \mathcal{TM}, \mathcal{CM}, \mathcal{IM} \rangle$ with:

- $\mathcal{BR} = \langle Pr_k, S_k, PIT_k \rangle$, the belief repository with Pr_k a profile¹, $S_k = \{s_{k1}, \dots, s_{km_k}\}$ a set of offered services and PIT_k a Personal Interaction Table. Each record in PIT_k (see Tab. 3.1) contains the following elements: an acquaintance agent $a_j \in SA_k$, the profile Pr_j of a_j , the social acquaintances set SA_j of a_j and the set of services S_j provided by a_j . This information is acquired through interactions among agents.
- \mathcal{GR} , the goal repository which encompasses the required services needed to solve user's query.

¹A profile consists of a set of items structured into a set of fields, each field containing one or several values (e.g., gender=[female], music-likes=[folk, jazz, pop]).

Table 3.1: Personal Interaction Table Example (PIT_0 of agent a_0 in Fig. 3.1)

Acquaintance	Profile	Social Acquaintances	Offered services
a_2	Pr_2	$SA_2 = \{a_0, a_1, a_3, a_5, a_8, a_{12}\}$	$S_2 = \{s_{21}, s_{22}\}$
a_3	Pr_3	$SA_3 = \{a_0, a_1, a_2, a_4, a_{11}, a_{12}\}$	$S_3 = \{s_{31}\}$
a_4	Pr_4	$SA_4 = \{a_0, a_1, a_3, a_7, a_8, a_{11}\}$	$S_4 = \{s_{41}\}$
a_5	Pr_5	$SA_5 = \{a_0, a_2, a_8, a_9\}$	$S_5 = \{s_{51}\}$
a_6	Pr_6	$SA_6 = \{a_0, a_8\}$	$S_6 = \{s_{61}, s_{62}, s_{63}\}$
a_7	Pr_7	$SA_7 = \{a_0, a_4, a_8, a_{10}\}$	$S_7 = \{s_{71}\}$

- \mathcal{RM} , the reasoning module representing the matching function. A matching function between a service $s_{kl} \in S_k$ and a service $s \in F$ is defined as follows:
 $matching(s, s_{kl}) = True \Leftrightarrow (s_{kl}.in \subseteq s.in) \wedge (s.out \subseteq s_{kl}.out) \wedge (s_{kl}.f \equiv s.f)$.
- \mathcal{TM} , the trust module which computes all trust measures that an agent a_k has with its social acquaintances before interacting with them.
- \mathcal{CM} , the control module which manages agent's behavior and guides its decision-making in the discovery and selection process.
- \mathcal{IM} , the interaction module which structures the messages built by the agent a_k and handles the received ones.

We now make some assumptions about agents and their behavior:

- *Assumption 1.* Agents are cooperative and have the good will to share their experiences with others.
- *Assumption 2.* Agents have limited view in the social network, they only know agents that belong to their social acquaintance set.
- *Assumption 3.* Agents perform decentralized decision-making in contacting other agents.

Agents are endowed with a bounded set of services and act on behalf of their associated users to discover and select trustworthy providers with good services. During the service discovery process, some agents may be good providers and some others may not be good providers but may be well connected and thus, may recommend good providers. We consider four different roles that an agent may have during the discovery process:

- *requester:* a requester is the agent that receives the query from its associated user containing the required services. It is the initiator of the discovery and selection processes. It determines from its acquaintances set the most trustworthy acquaintances and sends them the query.
- *provider:* a provider is an agent which offers one or many requested services. When an agent receives a query, it determines how to fulfill it with its offered services. If there is a matching between desired services and offered ones, it claims to be a provider and participates to the search of other potential providers by propagating the query to its trustworthy acquaintances.
- *recommender:* a recommender is an agent which has no required services but may participate to the query solving by discovering good providers. Unlike provider agent, a recommender agent queries only trustworthy acquaintances providing required services.
- *stopper:* a stopper is an agent that cannot participate to the query solving as it can neither provide a service nor recommend a provider. In the discovery process, this role is seen as a lock that blocks the query propagation in the social network. This allows us to limit the search space and reduce the computational cost.

4. Trust Model. This section presents our trust model for service discovery and selection processes in social networks. First, we present the different sources of information that we used for trust evaluation. Then, we describe subsequently the three trust components of our model.

4.1. Sources of trust information. Trust is a complex social phenomenon which is the basis of all social interactions [1]. Thus, any computational model of trust must be designed based on how trust works between people in society. As stated in [44], trust models that use only one source of information usually fail to provide a significant trust value of the trustee agent. Moreover, in a MAS context, trust evaluation is considered as an

internal reasoning from different sources of information that leads the agent to build a belief about the behavior of another agent [22]. For these reasons, we consider four different sources of information to build our trust model:

- *Individual experience*: It models the direct interaction between the trustor (i.e. agent which evaluates) and the trustee (i.e. agent which is evaluated). Based on its personal previous experiences, the trustor estimates how the trustee would behave in the current interaction. This type of information is used to compute trust in recommendation and is a part of the calculation of trust in expertise.
- *Communicated experience*: In the individual experience we considered only the direct interaction between the trustor and trustee. In the absence of direct interactions, indirect interactions or opinions of other agents can be consulted. This mechanism is called the referral system, often used for locating services through trusted referrals.
- *Social information*: In addition to agent interactional history (which are used in the two previous types of information), knowledge extracted from social networks can be a rich source for gathering information. Such information can be classified into two categories; semantic and structural. Semantic information includes trustor and trustee profiles and the relationship types between them. Structural information are derived using social network analysis techniques. It encompasses the position of the trustee in the social network graph, the proximity between the trustor and trustee and their neighborhood structure. This information is used to evaluate the trust in sociability between two agents.
- *Certified information provided by the trustee*: In the previous cases, the trustor needs to collect the required information itself. However, the trustee can also seek the trust of the trustor by presenting arguments about its trustworthiness. In this paper, such arguments are non-functional attributes (e.g., reliability, specialization, and experience rating) of the different services that it offers. In contrast to communicated experience which needs to be collected by the trustor, the trustee stores and provides such certified information on request to gain the trust of the trustor. As we mentioned before, a part of the calculation of trust in expertise is based on individual experience. The other part is based on this type of information.

Our trust model integrates these four sources of information and it is able to provide a trust evaluation that helps agents in their decision-making process when they want to interact. The problem of various sorts of disinformation and inaccuracy will be considered in future work. Integrating these various sources will also enhance the precision and the expressiveness of the trust model. This will be verified subsequently in our empirical evaluation (cf. Sect. 6).

4.2. Trust in Sociability (ST). Trust in sociability evaluates the level of social trust an agent a_k may have in another agent a_j . It is computed by extracting information from MRSN such as the structure of the graph, agents' profiles that contain personal data and information about their interests, and relationship types between agents. Based on the analysis of the MRSN graph and the extracted information, three measures are computed which are: *the social position* (SPo), *the social proximity* (SPr) and *the social similarity* (SSi). Next, we describe these measures, then we give the way to compute trust in sociability.

4.2.1. Social Position Measure (SPo). The social position of an agent a_j is computed based on its degree centrality which represents its social power. In a single-relation social network, it is computed as the degree of an agent (see for example [6]). In our work, we do not simply compute $SPo(a_j)$ as the number of relationships of agent a_j , but we also consider relationship types $\{R_i, 1 \leq i \leq n\}$ connecting a_j to the other agents as follows:

$$SPo(a_j) = \sum_{i=1}^r \sum_{a_l \in SA_j} U(\rho((a_j, a_l))) \times b^i(a_j, a_l) \quad (4.1)$$

where $b^i(a_j, a_l) = 1$ iff a_j and a_l are directly connected with an edge of relationship R_i , 0 otherwise.

An agent with a high degree centrality has an important structural position in the social network because the more interactions it has, the more prestigious it is in the social network. Therefore, we believe that agents with a high degree centrality have a leading position and are considered trustworthy by other agents in the network.

4.2.2. Social Proximity Measure (SPr). It is defined as the average cost of a path between two agents in the graph. Since there are different kind of relationships $\{R_i, 1 \leq i \leq n\}$ in the MRSN with preferences associated with them, we define SPr as follows:

Let $path = (a_k, \dots, a_j)$ be a path of length d between an agent a_k and an agent a_j , and $U(\rho((a_{l-1}, a_l)))$ be the cost of the edge $(a_{l-1}, a_l) \in path$ (see Def. 4).

$$SPr(a_k, a_j) = \frac{\sum_{l=1}^d U(\rho((a_{l-1}, a_l)))}{d} \quad (4.2)$$

In a single-relation social network, the shortest path between a pair of agents is the best path for social proximity [38]. In a multi-relation social network and according to Eq. 4.2, proximity between two agents depends not only on the length of the path but also on semantic aspects, that is the type of links that compose it. In some cases, a longer path may have a better cost if it is composed of relationships preferred by the user.

4.2.3. Social Similarity Measure (SSi). The social similarity between two agents is computed based on the comparison of their profiles and their social acquaintance sets. As demonstrated in [55], we believe that similarity between two agents goes beyond the similarity of their profiles and includes similarity of their neighborhoods. $SSi(a_k, a_j)$ is an aggregation of two measures, namely, *Neighborhood Similarity* (NS) and *Profile Similarity* (PS). First, we present NS and PS then we show how to aggregate them to compute SSi.

Neighborhood Similarity Measure (NS). As proved by Hang and Singh in [20], the number of common neighbors influences the trustworthy recommendation process. Thus, the higher the number of common neighbors between two agents, the higher the social intimacy degree between them. This is the case of Facebook which recommends friends based on the number of mutual friends between them. One of the fundamental principle in social psychology [41] is that an agent can trust more agents with intimate social relationships than agents with less intimacy. Following this principle, we define a measure called neighborhood similarity to find links between agents based on their social acquaintances as follows:

$$NS(a_k, a_j) = \sum_{i=1}^{|R|} U(\rho((a_{l-1}, a_l))) \times \delta^i(a_k, a_j) \quad (4.3)$$

with $\delta^i(a_k, a_j) = \frac{1}{1+jac^i}$ where $jac^i = \frac{y_i+z_i}{x_i+y_i+z_i}$ is the *Jaccard distance* between a_k and a_j according to the relationship R_i such as $x_i = |N_{R_i}(a_k) \cap N_{R_i}(a_j)|$, $y_i = |N_{R_i}(a_k)| - x_i$, $z_i = |N_{R_i}(a_j)| - x_i$.

Profile Similarity (PS). In social networks, an agent's profile is not only characterized by its acquaintances, but also by a set of personal information (e.g., gender, work) and interests (e.g., games, music, movies). We define a profile as follow:

- $\langle Profil \rangle ::= \langle item \rangle^+$
- $\langle item \rangle ::= \langle field \rangle^+$
- $\langle field \rangle ::= \langle value \rangle^*$

We use this information to compute trust between agents taking into account the similarity of their profiles. As shown in [55], in a real online community, a correlation exists between the degree of trust between agents and their profile similarity. The more two users trust each other, the more similar their profiles are. Based on these findings, we define a measure called profile similarity as follows:

$$PS(a_k, a_j) = \frac{1}{|I|} \times \sum_{i \in I} \beta_i \times S_i(a_k, a_j) \quad (4.4)$$

where $S_i(a_k, a_j)$ is the similarity between the i th items of a_k and a_j using Burnaby measure [13], I is the set of items in profiles and β_i is the weight attributed to the item i with $\sum_{i \in I} \beta_i = 1$. For more details concerning item similarity computation, readers are referred to our previous work [34].

The overall measure of social similarity, $SSi(a_k, a_j)$, between a requester agent a_k and an agent a_j is computed as the product of the two above measures:

$$SSi(a_k, a_j) = NS(a_k, a_j) \times PS(a_k, a_j) \quad (4.5)$$

4.2.4. Trust in Sociability Computation. ST between a requester agent a_k and an agent a_j is computed from SPo (Eq. 4.1), SPr (Eq. 4.2), and SSi (Eq. 4.5). Let M_j be a vector associated with each agent a_j such as $M_j = (SPo(a_j), SPr(a_k, a_j), SSi(a_k, a_j))$. Let M be a matrix obtained by merging the M_j vectors of all acquaintances a_j such as $M = (M_{jt}, a_j \in SA_k \text{ and } 1 \leq t \leq 3)$ in which each row M_j corresponds to an agent a_j , and each column corresponds to a social measure. To compute the value of the trust in sociability for each agent a_j , we use a Simple Additive Weighting technique that proceeds in two phases:

- Scaling phase: which aims to transform each measure value, of M_j vector, into a value between 0 and 1 according to the following formula.

$$M'_{jt} = \begin{cases} \frac{M_{jt} - M_t^{min}}{M_t^{max} - M_t^{min}} & \text{if } M_t^{max} - M_t^{min} \neq 0 \\ 1 & \text{if } M_t^{max} - M_t^{min} = 0 \end{cases} \quad (4.6)$$

where $M_t^{max} = \text{Max}(M_{jt}), \forall a_j$ and $M_t^{min} = \text{Min}(M_{jt}), \forall a_j$ are the maximal value (respectively the minimal value) of a column in the matrix M . Applying the formula 4.6 to M , we obtain a matrix $M' = (M'_{jt}, a_j \in SA_k \text{ and } 1 \leq t \leq 3)$, in which each row M'_j corresponds to an acquaintance a_j , and each column corresponds to a social measure.

- Weighting phase: which aims to give a social trust value to each acquaintance $a_j \in SA_k$. The overall social trust value is computed as follow:

$$ST(a_k, a_j) = \sum_{t=1}^3 \lambda_t \times M'_{jt}(a_k, a_j) \quad (4.7)$$

where $\lambda_t \in [0, 1]$ is the weight of the t -th social measure with $\sum_{t=1}^3 \lambda_t = 1$.

4.3. Trust in Expertise (ET). Trust in sociability checks whether an agent is socially trustworthy or not. However, it does not evaluate the relevance of its offered services. We think that building trust solely on the basis of a social-based component is insufficient to lead to a good service selection. According to [4], when an agent relies on another socially trustworthy agent there is still an amount of risk regarding the quality of its offered services. The quality of the selection process can be improved further by assessing providers quality of service called expertise. Based on this statement, we introduce another component of trust called trust in expertise ET that considers the non-functional properties of a service. Therefore, a good agent should be not only socially trustworthy but also sufficiently expert. Based on [25], we define a trust in expertise $ET(a_k, a_j, s_{jl})$ that an agent a_k has in a service s_{jl} offered by an agent a_j as the aggregation of three following non-functional attributes.

1. *Specialization (Sp)*: is the percentage of successful use of an agent's service s_{jl} compared to the other services it offers. It is defined as the ratio between the number of times that a service s_{jl} has been successfully completed [38] and the total number of successful executions of the agent a_j regardless of the used service.

$$Sp(s_{jl}) = \frac{Nb_{success}(s_{jl})}{\sum_{t=1}^{m_j} Nb_{success}(s_{jt})} \quad (4.8)$$

where $Nb_{success}(s_{jl})$ is the number of successful executions of s_{jl} and m_j is the number of services offered by agent a_j . This means that the more the service s_{jl} is sought for in the social network the more a_j is recognized as an expert in this field.

2. *Reliability (Re)*: is the probability that a service s_{jl} is operational at the time of invocation. It is computed as the rate between the number of successful executions $Nb_{success}(s_{jl})$ and the total number of functionality invocations $Nb_{invoc}(s_{jl})$.

$$Re(s_{jl}) = \frac{Nb_{success}(s_{jl})}{Nb_{invoc}(s_{jl})} \quad (4.9)$$

3. *Experience rating (Eval)*: is the rating of the service realization quality. After the execution of a service s_{jl} , an agent a_k gives an evaluation $\nu \in [0, 1]$ of this execution reflecting its experience feedback as a customer of this service.

Unlike the previous attributes, the evaluation of this attribute results from a subjective perception. Let $Eval(a_k, s_{jl})$ be the average of the experience ratings of s_{jl} for n uses by a_k .

$$Eval(a_k, s_{jl}) = \frac{\sum_{x=1}^n \nu_x}{n} \quad (4.10)$$

For the cold start, we introduce a trust bootstrapping phase. The goal for trust bootstrapping phase is to assign an initial trust value $Eval_{ini}$ to the new services (i.e., $n = 0$). We consider two strategies: negative and positive bootstrapping. In the first strategy, $Eval_{ini} \in [0, 0.5[$ whereas in the second strategy $Eval_{ini} \in [0.5, 1]$.

The overall trust in expertise $ET(a_k, a_j, s_{jl})$ that an agent a_k have in a service s_{jl} offered by an agent a_j is computed as follows:

$$ET(a_k, a_j, s_{jl}) = Sp(s_{jl}) \times Re(s_{jl}) \times Eval(a_k, s_{jl}) \quad (4.11)$$

4.4. Trust in Recommendation (RT). According to [19], two types of trust are needed to evaluate the reliability of an agent: trust in this agent and trust in its recommendations for a specific service. When an agent a_k receives a recommendation from an agent a_j , apart from the fact that it ensures that a_j is socially trustworthy, it prefers also that a_j recommendations are trustworthy [7]. We associate a trust value called trust in recommendation (RT) with a service recommendation made by an a_j to a_k . The estimation of the recommendation trustworthiness is done using previous experiences with the potential recommender during past compositions. We decompose trust in recommendation $RT(a_k, a_j, s_{pl})$ that an agent a_k gives to another agent a_j regarding the quality of its recommendations for a service s_{pl} offered by the provider a_p into two parts: the objective part, $[r_{kj}|s_{pl}] \in [0, 1]$, which indicates the performance of a_j in terms of the number of good recommendations for the service s_{pl} , and the subjective part, $[q_{kj}|s_{pl}] \in [0, 1]$, which reflects the satisfaction of the agent a_k for recommendations given by a_j concerning the service s_{pl} . Based on a formula given in [16], we define RT as follows:

$$RT(a_k, a_j, s_{pl}) = \begin{cases} 1 & \text{if } [r_{kj}|s_{pl}] = 0 \text{ or } [q_{kj}|s_{pl}] = 0 \\ ([r_{kj}|s_{pl}] + 1)^{[q_{kj}|s_{pl}]} - 1 & \text{otherwise} \end{cases} \quad (4.12)$$

As mentioned above, $[r_{kj}|s_{pl}]$ represents the ratio of the effective selection made by a_k among the total number of recommendations of a_j for the service s_{pl} offered by the provider a_p . Based on [37], $[r_{kj}|s_{pl}]$ is defined as follows:

$$[r_{kj}|s_{pl}] = \begin{cases} 1 & \text{if } Nbrec_{jk|s_{pl}} = 0 \\ \frac{Nbsel_{kj|s_{pl}}}{Nbrec_{jk|s_{pl}}} & \text{otherwise} \end{cases} \quad (4.13)$$

where $Nbrec_{jk|s_{pl}}$ and $Nbsel_{kj|s_{pl}}$ correspond to the number of times that a_j has recommended the service s_{pl} offered by a_p to a_k and the number of times that a_k has selected s_{pl} in the underlying composition, respectively.

However, $[q_{kj}|s_{pl}]$ indicates how much an agent a_k is satisfied with the recommendations of a_j concerning the service s_{pl} offered by the provider a_p . Given $Eval(a_k, s_{pl})$ (Eq. 4.10), the experience rating of s_{pl} given by a_k after its execution, $[q_{kj}|s_{pl}]$ is defined as the average of the different ratings made by a_k to s_{pl} after successful executions:

$$[q_{kj}|s_{pl}] = \begin{cases} 1 & \text{if } Nbsel_{kj|s_{pl}} = 0 \\ \frac{\sum_1^{Nbsel_{kj|s_{pl}}} Eval(a_k, s_{pl})}{Nbsel_{kj|s_{pl}}} & \text{otherwise} \end{cases} \quad (4.14)$$

5. Trust-based Service Discovery and Selection Approach Description. In this section, we describe the three steps that encompass our Trust-based Service Discovery and Selection approach (TSDS): service discovery, trust inference, and service selection. The service discovery step (Step 1) is an algorithm composed of three phases: trust in sociability computation as defined in Sect. 4.2, matching functionalities, and computing trust in recommendation as defined in Sect. 4.4. The output of this step as shown in Fig. 5.1 is a tree graph called Trust-Relation Social Network (TRSN) in which providers as well as recommenders are identified. Relationships between agents in TRSN are weighted by a pair of values representing trust in sociability and trust in recommendation between them. The second step, trust inference (Step 2), enables the service requester to evaluate trust it has in each of the discovered providers in the TRSN. It consists of propagating trust values from each discovered provider along the chain until the service requester. The latter retrieves the propagated trust values as an input and uses them to infer its trustworthiness to each discovered provider. The output of this step is a Requester-Centered Social Network (RCSN) in which a relationship between the requester and a provider is weighted by a trust value representing the inferred trust of the requester in the corresponding provider. The third step is the service selection one (Step 3). In this last step, the service requester ranks services of discovered providers depending on the value of their trust in expertise as defined in Sect. 4.3. Only services with trust in expertise value greater than a certain threshold will be selected.

Note that in both the discovery and the selection steps of our TSDS approach, different trust components are computed. Each trust component has a specific role in the process. Trust in sociability and trust in recommendation used in Step 1 act as a first filter leading to a set of trustworthy providers and recommenders. Trust in expertise used in Step 3 enables the selection of good services among those offered by the discovered providers. This separation allows us to avoid missed opportunities during the service discovery step. If we consider trust in expertise in Step 1, only trustworthy providers will be discovered during this step and a socially trustworthy agent that does not have a required service (i.e. a recommender) will not be discovered. However, in some cases, such a trustworthy agent may be well connected and thus, may recommend a good provider.

5.1. Step 1: Service Discovery. The aim of this step is to discover trustworthy providers through a distributed trust-based breadth-first search algorithm, namely Algo. 1. The inputs of this algorithm are the MRSN graph and a query $Q = (F, U, \alpha, \beta, \mu)$. To start the algorithm, the requester agent a_r assigns itself the requester role ($role_r = Req$), sets to 0 its distance $dist_r$ ² in the provider-recommender chain³, and initializes the set $PSet_r$ ⁴ to \emptyset (see Algo. 1 lines 2 – 4).

To start the tree construction, a_r needs to ensure that queried agents are socially trustworthy. To perform trust computation, a_r updates its Personal Interaction Table PIT_r (see Tab. 3.1) regarding current information of each of its social acquaintance $a_j \in SA_r$ (i.e., offered services S_j , social acquaintance SA_j and profile Pr_j). To do that, a_r sends a REQUEST message to each $a_j \in SA_r$, (see Algo. 1 lines 5 – 7).

The behavior of an agent a_k ⁵ is event-driven: the reception of a message triggers the execution of a particular procedure. Upon receipt of a REQUEST message, an agent a_k replies with an INFORM message containing information about its offered services S_k , its social acquaintance SA_k and its profile Pr_k (see Algo. 1 lines 29 – 31). As a result of receiving an INFORM message by an agent in SA_r , the requester increments its message counter and updates its Personal Interaction Table PIT_r (see Algo. 1 lines 32 – 35). Once all INFORM messages have been received, a_r begins trust computation phase in which it evaluates trust in sociability of each acquaintance a_j based on the social measure values described in Sect. 4.2. Agents with trust in sociability

² $dist_k$ represents the distance of an agent a_k in the provider-recommender chain to the requester agent a_r (initially equals $+\infty$).

³ A provider-recommender chain corresponds to a sequence of agents starting from the requester agent and leading to a provider in which all intermediate agents are either providers or recommenders.

⁴ $PSet_r$ is a data structure that contains for each discovered provider its offered services as well as trust values of the intermediate agents with their respective roles in the provider-recommender chain. This information is useful to infer trust that a_r has in each discovered provider.

⁵ The index k identifies the agent executing the code.

Algorithm 1 Distributed Trust-based Discovery Algorithm

Variables: $role_k \in \{\perp, Req, Pro, Rec, Stop\}$, $dist_k \in \{0, \dots, n\}$, $count_k \in \{0, \dots, |SA_k|\}$, LTA_k a set of trustworthy potential providers, LRA_k a set of trustworthy potential recommenders, τ_k is an array of stacks where each stack represents a provider-recommender chains of a discovered service, $PSet_k$ a set of stacks $\tau_k[]$ one per discovered service, $father_k$ the predecessor agent in the chain to the requester, $child_k$ a list of agents representing children of a_k in the tree.

```

1: procedure PROPAGATION( $F, U, \alpha, \beta, \mu$ )
2:   if ( $role_k == Req$ ) then
3:      $dist_k \leftarrow 0$ ;
4:      $PSet_k \leftarrow \emptyset$ ;
5:     for all  $a_j \in SA_k$  do
6:       Request( $a_k, a_j, \emptyset$ ); ▷ go to line 29
7:     end for
8:     wait ( $count_k == |SA_k|$ );
9:      $count_k \leftarrow 0$ ;
10:     $LTA_k \leftarrow \{a_j \in SA_k \mid ST(a_k, a_j) \geq \alpha\}$ ;
11:  end if
12:  if ( $role_k \in \{Req, Pro\}$ ) then
13:    for all  $a_j \in LTA_k$  do
14:      if ( $\exists s \in F$  and  $s_{jl} \in PIT_k.S_j \mid matching(s, s_{jl}) == True$ ) then
15:        Propagate( $a_k, a_j, (F, U, \alpha, \beta, \mu, dist_k)$ ); ▷ go to line 36
16:      else
17:        if ( $\exists s \in F \mid RT(a_k, a_j, s) \geq f_\beta(dist_k)$ ) then
18:          Propagate( $a_k, a_j, (F, U, \alpha, \beta, \mu, dist_k)$ ); ▷ go to line 36
19:        end if
20:      end if
21:    end for
22:  end if
23:  if ( $role_k == Rec$ ) then
24:    for all  $a_j \in LRA_k$  do
25:      Propagate( $a_k, a_j, (F, U, \alpha, \beta, \mu, dist_k)$ ); ▷ go to line 36
26:    end for
27:  end if
28: end procedure
29: procedure RECEIVE_REQUEST( $a_j, a_k, \emptyset$ )
30:   Inform( $a_k, a_j, (S_k, SA_k, Pr_k)$ ); ▷ go to line 32
31: end procedure
32: procedure RECEIVE_INFORM( $a_j, a_k, (S, SA, Pr)$ )
33:    $count_k \leftarrow count_k + 1$ ;
34:    $PIT_k.set(a_j, S, SA, Pr)$ ;
35: end procedure
36: procedure RECEIVE_PROPAGATE( $a_j, a_k, (F, U, \alpha, \beta, \mu, dist)$ )
37:   if ( $dist_k > dist + 1$ ) then
38:      $dist_k \leftarrow dist + 1$ ;
39:      $father_k \leftarrow a_j$ ;
40:     Inform( $a_k, father_k, child$ ); ▷ go to line 62
41:     for all  $a_j \in SA_k$  do
42:       Request( $a_k, a_j, \emptyset$ );
43:     end for
44:     wait ( $count_k == |SA_k|$ );
45:      $count_k \leftarrow 0$ ;
46:     if ( $\exists s \in F$  and  $s_{kl} \in S_k \mid matching(s, s_{kl}) == True$ ) then
47:        $role_k \leftarrow Pro$ ;
48:        $\tau_k[k].add(a_k, s_{kl})$ ;
49:        $LTA_k \leftarrow \{a_j \in SA_k \mid ST(a_k, a_j) \geq f_\alpha(dist_k)\}$ ;
50:       PROPAGATION( $F, U, \alpha, \beta, \mu$ ); ▷ go to line 12
51:     else
52:        $LRA_k \leftarrow \{a_j \in SA_k \mid ST(a_k, a_j) \geq f_\alpha(dist_k) \text{ and } \exists s \in F \text{ and } s_{jl} \in PIT_k.S_j \mid matching(s, s_{jl}) == True\}$ ;
53:       if ( $LRA_k \neq \emptyset$ ) then
54:          $role_k \leftarrow Rec$ ;
55:         PROPAGATION( $F, U, \alpha, \beta, \mu$ ); ▷ go to line 23
56:       else
57:          $role_k \leftarrow Stop$ ;
58:       end if
59:     end if
60:   end if
61: end procedure
62: procedure RECEIVE_INFORM( $a_j, a_k, child$ )
63:    $child_k.add(a_j)$ ;
64: end procedure

```

value greater than α threshold ⁶ are kept in the LTA_r set (see Algo. 1 lines 8 – 10).

An agent a_k has to check before propagating the query to a trustworthy agent $a_j \in LTA_k$ whether it could participate or not to the query resolution. Either by offering a required service or if not, by being a potential trustworthy recommender. In both cases, a_k sends to a_j a PROPAGATE message containing the user's query and its current distance (see Algo. 1 lines 12 – 22). We use trust in recommendation threshold $f_\beta(dist_k)$ during service discovery step to reduce the search space (i.e. visited agents) while ensuring recommendations given by trustworthy recommenders.

An agent a_k that receives a PROPAGATE message, compares the received distance value with its current distance value $dist_k$. If the received value is less than its current distance value then, a_k sets $dist_k$ to the new value (see Algo. 1 lines 37 – 38). This test condition allows to give the shortest trustworthy path tree that prevents cycles. Then, it sets its father value $father_k$ to the sender and sends to it an INFORM message containing a *child* string (see Algo. 1 lines 39 – 40). Upon receipt of an INFORM message containing a *child* string, the father adds to its $child_k$ list the identifier of the sender (see Algo. 1 lines 62 – 64). Thanks to this INFORM, each agent knows its children in the tree. After that, a_k updates its beliefs and matches required services against its own offered services S_k using the matching function (see Algo. 1 lines 41 – 46). The matching result could lead to two cases:

- a_k has a service that matches a required service. In this case, a_k sets its role to *provider* ($role_k = Pro$) and initializes its stack $\tau_k[k]$ ⁷ as follows: $\tau_k[k] = \{a_k, s_{kl}\}$ where a_k is its identifier and s_{kl} is its offered service. Next, a_k determines the LTA_k set (see Algo. 1 lines 47 – 48). To be in this set, an acquaintance $a_j \in SA_k$ must have a trust in sociability value $ST(a_k, a_j)$ greater than $f_\alpha(dist_k)$ (see Algo. 1 line 49). Thereafter, a_k executes the PROPAGATION procedure that sends a PROPAGATE message to all acquaintances in LTA_k (see Algo. 1 lines 50 and 12 – 22).
- a_k does not provide any of the required services, it determines LRA_k set containing its acquaintances which are not only trustworthy but also that offer one of the required services (see Algo. 1 line 52). If LRA_k set of a_k does not contain relevant acquaintances (i.e. either they are not enough trusty to be recommended or they do not offer a required service) then, it stops the query propagation and sets its role to *stopper* ($role_k = Stop$) (see Algo. 1 lines 56 – 58). Otherwise, a_k sets its role to *recommender* ($role_k = Rec$) and executes the PROPAGATION procedure by sending a PROPAGATE message to all acquaintances in LRA_k (see Algo. 1 lines 53 – 55 and 23 – 27). In this case, although a_k does not provide any required service, it participates in the discovery process while leading to relevant agents.

The output of the first step is a tree called Trust-Relation Social Network (TRSN) as shown in Fig. 5.1 in which the requester is the root and the providers (black nodes) and recommenders (white nodes) are identified. The different provider-recommender chains allows to build a directed and acyclic graph where the requester is the root and the distance of each discovered provider to the requester is the length of its chain. At this stage, providers are discovered on the basis of the functional aspect as well as of the degree of their sociability. So far, the non functional aspect expressed by the trust in expertise is not yet considered.

5.2. Step 2: Trust Inference. In social networks, an agent evaluates the trustworthiness of another based on direct interactions with it. With the absence of direct interactions, a *trust inference mechanism* is applied. This mechanism uses trust values along paths that connect two agents to come up with a single evaluation. In this step, we first show how trust values are propagated via intermediate agents to infer trust between non adjacent agents then, we describe how to build a new tree graph called a Requester-Centered Social Network (RCSN).

⁶ $f_x(dist_k) = (1-x) \times (1 - e^{-\frac{dist_k}{D}}) + x$ where D is the network diameter and $x \in \{\alpha, \beta\}$ is the function that updates trust threshold value depending on the chain length d (for more details see [35]). The initial values $f_\alpha(0)$ and $f_\beta(0)$ correspond respectively to α and β the preference of the requester's query. These values are incremented locally by a_k according to its $dist_k$. We believe that the longer the chain is, the higher the requester's sociability and recommendation value expectations are. Therefore, it is reasonable to adjust α and β values to the current search depth. This ensures high sociability and recommendation values for providers that are discovered far away from the requester.

⁷ τ_k is an array of stacks. Each stack represents a provider-recommender chain of a discovered provider containing its offered service and trust values of the intermediate agents leading to it with their respective roles.

Function 1 Upgrade(τ)

```

1: if ( $\tau.length() == 1$ ) then
2:    $role \leftarrow Pro$ ;
3: else
4:    $role \leftarrow top(\tau).getrole()$ ;
5: end if
6: if ( $role == Rec$ ) then
7:    $\tau.push(\{role_k, ST(a_k, a_j), RT(a_k, a_j)\})$ ;
8: else
9:    $\tau.push(\{role_k, ST(a_k, a_j), 1\})$ ;
10: end if

```

Trust Values Propagation. Recall that the output of the previous step is a tree where agents are either providers or recommenders and relationships between them are weighted by a pair of values representing their corresponding trust in sociability and trust in recommendation. In this second step, the requester a_r starts a trust inference algorithm, called Algo. 2, by sending to each of its children in TRSN an INFORM message containing an *infer* string (see Algo. 2 lines 1 – 5). An agent a_k that receives such a message, sends in turn each of its children an INFORM message (see Algo. 2 lines 6 – 9). Additionally, if a_k is a provider, then it sends its father in the tree an INFORM message with a copy of its $\tau_k[k]$ containing its identifier and one of its offered services that matches one of the required services (see Algo. 2 lines 10 – 12). Upon receipt of an INFORM message containing a stack τ from its child a_j (see Algo. 2 line 14), an agent a_k determines the *id* of the provider and applies an upgrade function (see Algo. 2 lines 15 – 17 and Func. 1) as follows. Depending on the role of a_j , a_k inserts a new element in $\tau_k[id]$. If a_j is a recommender (see Func. 1 lines 6 – 7), then a_k upgrades $\tau_k[id]$ with $\{role_k, ST(a_k, a_j), RT(a_k, a_j)\}$ where $role_k$ is its role, $ST(a_k, a_j)$ and $RT(a_k, a_j)$ are respectively its trust in sociability and trust in recommendation to its child a_j . If a_j is a provider, then a_k upgrades $\tau_k[id]$ with $\{role_k, ST(a_k, a_j), 1\}$. In this case, a_j being a provider, it does not generate recommendations and therefore $RT(a_k, a_j) = 1$ (see Func. 1 lines 8 – 10). Once the upgrading is done, if a_k is the requester agent ($role_k == Req$), then it adds this stack to its $PSet_r = PSet_r \cup \{\tau_k[id]\}$ (see Algo. 2 lines 18 – 19). Otherwise, a_k is an intermediate agent in the chain, then it applies the same mechanism by transmitting its upgraded $\tau_k[id]$ to its father and so forth until the stack reaches the requester agent (see Algo. 2 lines 20 – 23). An example of trust values propagation in the TRSN of Fig. 5.1 from the provider a_9 to the requester a_0 is given in Fig. 5.2.

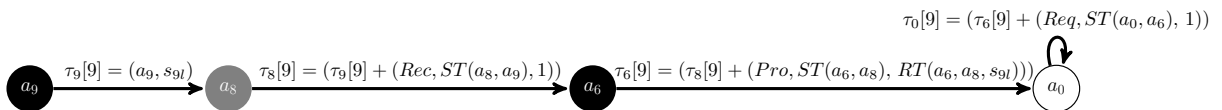


Fig. 5.2: A provider-recommender chain

At the end of Algo. 2, the requester set $PSet_r$ contains one stack for each discovered provider a_p . Each stack includes trust values (i.e. trust in sociability and trust in recommendation) of intermediate agents in the $chain = (a_r, a_k, a_{k+1}, \dots, a_p)$.

Trust values propagation is essential to infer trust that the requester has in a provider although it has no direct interaction with it. In the literature, several methods have been proposed for trust inference in social networks [19, 21, 33, 49, 48]. Most of these works [21, 33, 49] propose well defined operators of aggregation to infer trust between non adjacent agents stemming from different paths. In this work, we infer trust between non adjacent agents from the provider-recommender chain based on the probabilistic approach proposed in [48]. Compared to [48], our contribution consists of integrating trust in recommendation in the calculations depending on the role of intermediate agents in the chain. Using information stored in $PSet_r$ and the total probability law, trust of the requester a_r in a provider a_p along the chain $(a_r, a_k, a_{k+1}, \dots, a_{p-1}, a_p)$ is inferred as follows:

$$Trust(a_r, a_p) = P(a_r) \times P(a_k|a_r) \times P(a_{k+1}|a_k) \times \dots \times P(a_p|a_{p-1}) \quad (5.1)$$

where $P(a_r) = 1$ denotes the trustworthiness of a_r to itself. $P(a_k|a_r)$ denotes the trustworthiness of a_k from a_r 's point of view. This is an aggregation of trust in sociability and trust in recommendation such as $P(a_k|a_r) = ST(a_r, a_k) \times RT(a_r, a_k)$ where, $RT(a_r, a_k) = 1$ if $role_k = Pro$. At the end of this phase, from $PSet_r$ set and Eq. 5.1, a_r infers its trustworthiness in each discovered provider in the TRSN network.

RCSN Construction. The aim of this phase is to build a new network from the requester a_r point of view. Based on the trust value $Trust(a_r, a_p)$ for each discovered provider a_p , we build the Requester-Centered Social Network (RCSN) modeled by a directed and weighted tree graph $G' = (V', E')$, where V' is the set of providers and the requester, and E' is the set of edges. An edge $(a_r, a_p) \in E'$ is a social trust relationship between a_r and a_p and the weight of an edge (a_r, a_p) is the inferred trust value between them. Detailed results are stored in a data structure $PSet_r^*$ computed as follows:

$$\begin{aligned} \mathcal{F}_1 : PSet_r &\rightarrow PSet_r^* \\ \tau_r[p] &\mapsto (\tau_r[p], Trust(a_r, a_p)) \end{aligned}$$

where \mathcal{F}_1 is a function that takes as input each stack $\tau_r[p]$ of a discovered provider a_p and adds it to $Trust(a_r, a_p)$, its inferred trust.

5.3. Step 3: Service Selection. In the previous steps, the service requester a_r has discovered a set of trustworthy providers in the social network. This present step consists of sorting these discovered providers and select the best to perform the service selection. After computing the trust in expertise ET, a_r constructs an ordered set $PSel_r$ using the following function:

$$\begin{aligned} \mathcal{F}_2 : PSet_r^* &\rightarrow PSel_r \\ (\tau_r[p], Trust(a_r, a_p)) &\mapsto (\tau_r[p], Trust(a_r, a_p), ET(a_r, a_p, s_{pl})) \end{aligned}$$

where \mathcal{F}_2 is a function that takes as input each record of $PSet_r^*$ and adds it to the trust in expertise that a_r has in a_p . Then for each required service, a_r selects providers that have a trust in expertise value greater than μ as shown in Fig. 5.1.

6. Experimental setup and performance evaluation. In order to empirically evaluate TSDS, our multi-agent approach for trust-based service discovery and selection, we have developed a prototype using Java 1.7 and the Jade⁸ multi-agent platform. The MRSN graph data was stored in a GML format⁹. Several experiments have been performed, and all of them were run on a 3.1GHz Core(TM) i5-2400 running windows 7, with a 8Go of RAM. In the first series of experiments, we examine the performance of our service discovery step by assessing its *effectiveness* and *efficiency*. In the second series of experiments, we focus on the evaluation of the performance of our service selection step. The performance review is carried out through a comparative study versus two trust-based approaches, Bansal *et al.* [6] and Maaradji *et al.* [38], on two criteria, *utility* and *requesters' satisfaction*. The aim of this evaluation is to check whether or not our TSDS approach helps users to acquire good providers and so, allows them to gain better utility and satisfaction comparing to other approaches. Note that none of the aforementioned approaches have presented experimental results that evaluate their performance.

6.1. Experimental setup. The testbed environment for evaluating our approach is a social network of agents providing services (called providers) and agents using those services (called requesters). We focus on two network families: small-world and scale-free networks. Small-world networks are defined as a family of graphs

⁸ Telecom Italia Lab. JADE 4.3 <http://jade.tilab.com/>.

⁹ Graph Modeling Language, 1997, <http://www.fim.uni-passau.de/en/fim/faculty/chairs/theoretische-informatik/projects.html>

¹⁰ <http://snap.stanford.edu/data/egonets-Facebook.html>

exhibiting three properties: weak connectivity, strong clustering, and small diameter. Scale-free networks are networks following a power-law degree distribution, a model for describing the behavior of node degrees. Most nodes in such networks have few edges, but a few nodes have much higher degree.

In the first part (Sect. 6.2), we run simulations on synthetic graphs fitting the small-world properties (the maximal path length in each of these graphs is seven hops). To do that, we generate 5 graph instances where the number of agents is equal to 1000 and the number of edges is equal to 6000. We consider 3 categories of functionalities: transportation, accommodation and entertainment. In each of them we generate 5 different services. We equip randomly each agent with three different services, one from each category. Recall that a requester's query is defined as $(F, U, \alpha, \beta, \mu)$. For sake of simplicity, we suppose that requesters need only one service $F = \{s\}$. We vary the initial α 's value in $[0, 1]$, while β and μ are both set to 0.4.

In the second part (Sect. 6.3), we conduct experiments on a real dataset which is Facebook dataset¹⁰ containing 4039 agents and 88234 edges. This dataset has been proved to possess the small-world [52] and scale-free [24] characteristics of social networks. We use the dataset WSDream¹¹ of [54] which contains 5825 Web services distributed among all agents. Each agent is equipped with three different services. Similarly, for requester's query we suppose that requesters need only one service $F = \{s\}$. We set trust thresholds as follows: $\alpha = 0.6$, $\beta = 0.4$ and $\mu = 0.4$. In both datasets (real and synthetic), we consider three type of relationships $\{R_1, R_2, R_3\}$. The requester's preferences over relationship types are equal to $U(R_i) = \frac{1}{2^{i-1}}$ with $i \in \llbracket 1, 3 \rrbracket$. For non-functional values, we consider also the dataset WSDream of [54].

6.2. Performance evaluation of our service discovery step. Effectiveness consists of monitoring the average quality of providers' expertise in terms of trust in expertise for different trust in sociability threshold values. Efficiency consists of examining the distribution of the trustworthy discovered providers along the provider-recommender chain. The goal of these two experiments is to determine the best trust in sociability threshold as well as the provider-recommender chain length to enhance the service discovery results.

Effectiveness. This experiment evaluates the effectiveness of our service discovery step based on referral systems. It allows us to estimate in advance the quality of discovered services for different ranges of trust in sociability values. As experimental methodology, we consider the three following referral policies:

- **referAll:** an agent sends the query to all agents in its social acquaintances set. This is a special case of referencing in which trust is not considered. This is similar to GNUtella's search process [23] where queries are spread by flooding i.e. each agent forwards an incoming query to all of its neighbors if it doesn't have the required file.
- **referTrust:** only agents with sociability value above a predefined trust in sociability threshold are referred. This means, the higher the trust threshold is, the smaller the number of discovered providers is. This policy corresponds to our approach.
- **referBest:** an agent refers a single agent with the best sociability value among its acquaintances. This is similar to Freenet's routing for request messages [26], where each Freenet client forwards the request to another that is the likeliest to have the required information.

Figure 6.1 illustrates the evolution of the average quality of the expertise (averaged over all graph instances) versus the variation of the α 's value of different policies. We limit the length of chains to five similar to GNUtella's time-to-live value. During the simulation, each agent generates 5 queries resulting in 5000 queries all together over which we compute the average value of the trust in expertise of discovered providers. ReferAll policy corresponds to the case where α 's value is set to zero. ReferBest policy is independent of α and remains constant for all values. It is interesting to note that, among the three policies, the referAll policy performs the worst and its average quality is equal to 0.27. This result can be justified given that trust is not considered. Trustworthy as well as not trustworthy providers are discovered which would substantially decrease the average quality of the expertise. Furthermore, it is also interesting to note three important points about the evolution of the average quality of the expertise of referTrust policy versus the variation of α 's value. First, as shown in Fig. 6.1, the best average quality of the expertise corresponds to a α 's value equals to 0.6. Thanks to this finding, we can consider this value as a reference which could be recommended to the requester before launching the service discovery process. Second, when $\alpha = 1$, referBest and referTrust curves intersect. This is equivalent

¹¹<http://www.wsdream.net/dataset.html>

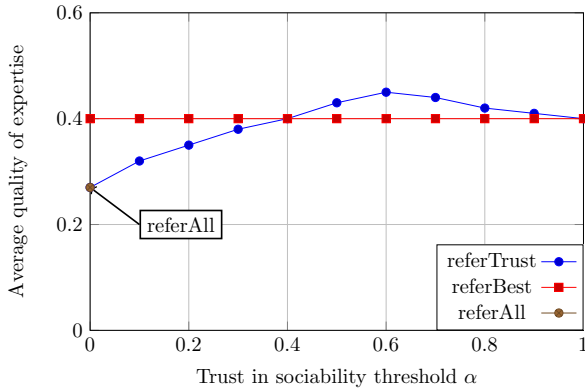


Fig. 6.1: Effectiveness of our service discovery step

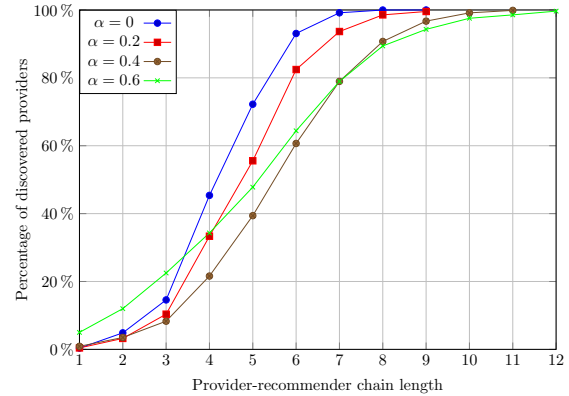


Fig. 6.2: Efficiency of our service discovery step

to keep in both policies a single agent, which is the most trustworthy one. Third, within the range of values $[0, 0.6]$, the referTrust policy curve has an increasing slope (i.e. providers with a low sociability also have low expertise). This indicates the existence of a correlation between the sociability and the expertise of an agent. With the increase of α 's value, less trustworthy providers are found as the average value of the expertise quality increases. However, when the α 's value is high, this correlation is no longer relevant. In this case, agents are too selective and few trustworthy agents are queried. This means that for high α 's values, some providers with good expertise are no longer discovered because there are no chain of trustworthy agents leading to them. This indicates that these providers are socially disconnected from the requester agent.

Efficiency. We study in this experiment the efficiency of the discovery step according to another experimental parameter which is the length of the provider-recommender chain. During the simulation, each agent generates 5 queries resulting in 5000 queries all together over which we compute the average number of the discovered providers for different chain length.

Approaches which evaluate trust in social networks offer computational models based on social relationships which are often inferred from long paths. To illustrate this point, we represent in Fig. 6.2 the distribution in cumulative percentage of the discovered providers (averaged over all graph instances) depending on the length of the provider-recommender chain. For different α 's values, we note that the number of the discovered providers in requesters' social acquaintances represents a small fraction (about 1%) of the total number of discovered providers. Some agents may not be good experts, but may be well connected and may recommend good providers hence the interest of referral systems in searching for providers. In our approach, sociability credits the ability to give good referrals. The propagation of the search is performed via navigation in the social network, which explains the increase of the number of discovered providers depending on chain length. It is also interesting to note that there is a trade-off between the length of the provider-recommender chain and the number of discovered providers. As a matter of fact, there is six times more chances (from 14% to 80%) to find a trustworthy provider in a chain length equal to 7 than in chain length equal to 3. However, the average number of discovered providers tends to level off once the chain length reaches a certain point (here equal to 8) regardless of trust threshold value. Concretely, when the chain length goes beyond a certain limit, the remaining providers are socially disconnected from the requester. Following these results, we can consider the length of the provider-recommender chain as a parameter that could be adjusted before launching the discovery process. Setting a maximum length of 7 for example gives us the possibility to control the scope of search which would substantially limit the computation cost.

6.3. Performance evaluation of our service selection step. In this section, we evaluate the performance and the benefit of our service selection step in terms of *utility* and *requesters' satisfaction* through a comparative study including three different approaches. The idea is to note the values of the different non-functional attributes of the selected services for each approach. Then, we compare the performance of our

three-step approach with (i) a one-step approach (i.e. Maaradji *et al.*) that aggregates trust in sociability and trust in expertise as a single value in service selection, (ii) another one-step approach (i.e. Bansal *et al.*) that uses only trust in sociability in service selection without any consideration of the non-functional aspect, and (iii) a random approach that selects services randomly and does not pay any consideration of trust. To do that, we perform experiments on real data (Facebook dataset) and we plot the averaged utility/satisfaction of selected services for all agents in the network while changing the requester agent at each round. In the experiments we test three different service selection strategies:

- Random Strategy: a requester chooses a provider randomly among potential providers. Thus, the service selection does not pay any consideration of trust.
- Basic Strategy: a requester makes its selection decision by evaluating the trustworthiness of the potential providers using Bansal *et al.* [6] or Maaradji *et al.* [38] approaches. Bansal *et al.* approach is based only on trust in sociability component using the degree centrality. Maaradji *et al.* approach considers both trust in sociability (i.e. social proximity) and trust in expertise components to evaluate the provider trustworthiness.
- Our inference Strategy: which implements TSDDS approach as described in Sect. 5. Similar to the Basic strategy, the selection decision is guided by trust. Compared to Maaradji *et al.* and Bansal *et al.* approaches, TSDDS approach intervenes at service selection step and at service discovery step.

In the social network, we define 4 sets of agents where each set of agents is equipped with a different strategy.

Utility. Utility is an objective criterion defined as the average of the values taken by the specialization or the reliability attributes of the selected services. In this experiment, we evaluate the utility of each strategy where each agent generates 10 queries during the simulation resulting in 40390 queries all together.

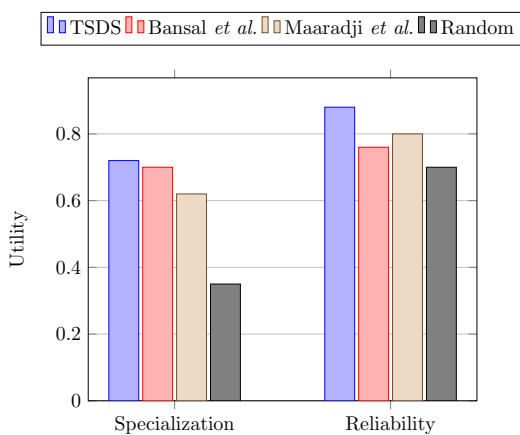


Fig. 6.3: Utility of our service selection step

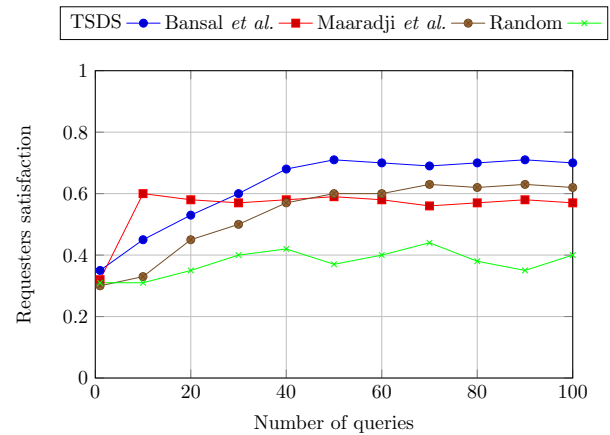


Fig. 6.4: Requesters' satisfaction of our service selection step

As we expected, Fig. 6.3 shows that Random strategy (i.e. agents that select services randomly without any consideration of trust) has the worst performance on all non-functional attributes. On the other hand, TSDDS, Bansal *et al.* and Maaradji *et al.* approaches are beneficial to requesters, helping them to obtain a significant utility i.e. services with better specialization and reliability values. This indicates that a selection strategy based on trust is more effective than without. We also note that our approach outperforms Bansal *et al.* and Maaradji *et al.* approaches on both attributes. This is because TSDDS approach is more expressive and richer as it evaluates not only sociability and expertise of an agent but also the recommendation quality of intermediate agents leading to this agent. This increases the quality of the selected services and thereafter the measured utility. The performance difference between TSDDS and Maaradji *et al.* approaches is explained by the fact that TSDDS approach differentiates between sociability and expertise when choosing an agent to interact with. This separation allows us to avoid missed opportunities at the service discovery step. A socially trustworthy agent that has low expertise will not be queried using the Maaradji *et al.* approach because the aggregation of the

two components may give a trust value below the threshold. In TSDS approach, this agent will be queried as it is socially trustworthy and even if it does not provide the required service, it can be helpful in recommending a good provider. Although Maaradji *et al.* approach is richer than Bansal *et al.* approach, since it considers both sociability (i.e. social proximity) and expertise in assessing agent's trustworthiness, we note that it is less efficient on specialization attribute. This can be explained by the fact that the specialization of a provider is naturally correlated to its degree which is the measure used by Bansal *et al.* approach. Therefore, the more an agent is specialist in a specific service, the more it is sought for in the social network. This increases the number of its interactions and subsequently its degree. These results justify the fact that the utility of Bansal *et al.* approach is better than Maaradji *et al.* approach for this attribute.

Requesters' satisfaction. In the second experiment, we perform the same comparative study but from a subjective point of view. Specifically, our focus will be on identifying the effect of TSDS and how it may impact requesters' satisfaction. Satisfaction is a subjective criterion that reflects the opinion of a requester regarding the behavior of a service after use. It corresponds to the average of the values taken by the attribute *Eval* for all selected services. For the initial values, we adopt a negative bootstrapping (see Sect. 4.3) to better observe the influence of the selection strategies on requesters's satisfaction. As shown in Fig. 6.4, the objective of this experiment is to examine the evolution of the requesters' satisfaction over time for each of the selection strategies: TSDS, Bansal *et al.*, Maaradji *et al.* and Random. To do that, each agent generates 100 queries (resulting in 403900 queries all together) which are sent sequentially in the social network. To monitor the progress of requesters' satisfaction, we note every 10 queries the average ratings of services after use. At the beginning of the simulation, we note that requesters' satisfaction behaves similarly for the three selection strategies. This is due to the fact that for a small number of queries, these values change slowly. However, after 10 queries we notice a significant difference in the satisfaction performance. Comparing the Bansal *et al.* approach with other groups of agents, results show that initially Bansal *et al.* approach is the most efficient. This can be explained by the fact that at the beginning of the simulation, the quality of all services is initialized with a low value (negative bootstrapping). Therefore, for a selection strategy as TSDS and Maaradji *et al.* using this attribute in trust evaluation, agents fail to find the good providers. However, these low quality values do not influence the evaluation of Bansal *et al.* approach because it is based solely on the degree centrality. Thus, providers which are selected by Bansal *et al.* with a high centrality degree have a good chance to provide good services which is the case. This confirms our intuition about the great influence of the sociability component in assessing trust. Another noticeable point is that satisfaction tends to level off after a finite number of queries and this for all selection strategies that use trust (i.e., TSDS, Bansal *et al.*, and Maaradji *et al.*). These groups of agents are able to learn gradually the quality of service, but at different speeds, the actual quality of service. For example, the TSDS agent group is the fastest group and it is able to learn after 50 queries. However, as we expected, satisfaction is fluctuating and gradually falls for a Random strategy that does not use trust. We also note that among all the selection strategies that use trust, TSDS is the most efficient because it maintains a stable and high requesters' satisfaction value. In addition, we note that after 45 queries the satisfaction in Maaradji *et al.* approach outperforms the satisfaction in Bansal *et al.* approach.

7. Conclusion and Future Work. In this work, we have presented an original approach for trustworthy service discovery and selection to satisfy users' needs in distributed environments like social networks. This approach is performed by a three-step process in which agents propagate the query as well as trust values in the social network using referral systems. Trustworthiness in providers and their services is evaluated over three measures namely sociability, expertise and recommendation. Sociability measure consists of analyzing the social network to evaluate the provider trustworthiness. Expertise measure quantifies the capability of the provider to offer good services. Recommendation measure evaluates the ability of an agent to give good recommendations. Based on a probabilistic approach proposed in [48], we infer trust between non adjacent agents while integrating trust in recommendation in computation. The conducted experiments have demonstrated that our model and approaches could yield more trustworthy results and recommendations than classical approaches by considering societal factors. The proposed distributed algorithms are efficient and can be applied to real social networks.

As future work, we would like to generalize, evaluate and test our approach on more complex networks such as multiplex networks. We intend also to explore the extension of our model to perform a service composition built upon a coalition formation of trust-based selected services.

REFERENCES

- [1] A. ABDUL-RAHMAN AND S. HAILES, *Supporting trust in virtual communities*, in Proceedings of the 33rd Hawaii International Conference on System Sciences-Vol. 6, IEEE Computer Society, 2000, pp. 6007–6016.
- [2] P. ADLER, *Market, Hierarchy, and Trust: The Knowledge Economy and the Future of Capitalism*, Organization Science, 12, 2001, pp. 215–234.
- [3] E. AL-MASRI AND Q. H. MAHMOUD, *Toward quality-driven web service discovery.*, IT Professional 10, 2008, pp. 24–28.
- [4] J. AL-SHARAWNEH, *Social Networks: Service Selection and Recommendation*, PhD thesis, University of Technology, Sydney, 2012.
- [5] J. AL-SHARAWNEH AND M.-A. WILLIAMS, *Abms: Agent-based modeling and simulation in web service selection*, in Management and Service Science, 2009, pp. 1–6.
- [6] S. K. BANSAL, A. BANSAL, AND M. B. BLAKE, *Trust-based dynamic web service composition using social network analysis*, in BASNA, 2010, pp. 1–8.
- [7] E. BERSCHIED AND H. T. REIS, *Attraction and close relationships*, The handbook of social psychology, 2, 1998, pp. 193–281.
- [8] D. BIANCHINI, V. D. ANTONELLIS, AND M. MELCHIORI, *Flexible semantic-based service matchmaking and discovery.*, World Wide Web, 11, 2008, pp. 227–251.
- [9] H. BILLHARDT, R. HERMOSO, S. OSSOWSKI, AND R. CENTENO, *Trust-based service provider selection in open environments*, in Proceedings of the 2007 ACM Symposium on Applied Computing, SAC '07, ACM, 2007, pp. 1375–1380.
- [10] S. BORIAH, V. CHANDOLA, AND V. KUMAR, *Similarity measures for categorical data: A comparative evaluation*, SIAM 30, 2008, pp. 243–254.
- [11] A. BOSE, R. NAYAK, AND P. BRUZA, *Improving web service discovery by using semantic models.*, in WISE, vol. 5175, Springer, 2008, pp. 366–380.
- [12] K. BRYSON, M. LUCK, M. JOY, AND D. T. JONES, *Applying agents to bioinformatics in geneveaver*, in CIA, vol. 1860, 2000, pp. 60–71.
- [13] T. BURNABY, *On a method for character weighting a similarity coefficient, employing the concept of information*, Mathematical Geology 2, 1970, pp. 25–38.
- [14] C. CASTELFRANCHI AND R. FALCONE, *Principles of trust for mas: Cognitive anatomy, social importance, and quantification*, in Proceedings of ICMAS, IEEE Computer Society, 1998, pp. 72–79.
- [15] S. CATANESE AND P. DE MEO AND E. FERRARA AND G. FIUMARA AND A. PROVETTI, *Extraction and Analysis of Facebook Friendship Relations*, in Computational Social Networks, Springer Verlag, 2011, pp. 291–324.
- [16] M. CHEN AND J. P. SINGH, *Computing and using reputations for internet ratings.*, in ACM Conference on Electronic Commerce, ACM, 2001, pp. 154–162.
- [17] I. DA SILVA AND A. ZISMAN, *A framework for trusted services*, in Proceedings of ICSOC, Berlin, Heidelberg, 2012, pp. 328–343.
- [18] M. GEORGEFF AND A. LANSKY, *Reactive reasoning and planning.*, in Proceedings of AAAI, 1987, pp. 677–682.
- [19] J. GOLBECK, *Generating predictive movie recommendations from trust in social networks*, in Proceedings of the 4th international conference on Trust Management, Springer-Verlag, 2006, pp. 93–104.
- [20] C.-W. HANG AND M. SINGH, *Trust-based recommendation based on graph similarity*, in Proceedings of TRUST10 at AAMAS, 2010.
- [21] C.-W. HANG, Y. WANG, AND M. P. SINGH, *Operators for propagating trust and their evaluation in social networks*, in Proceedings of AAMAS, vol. 2, 2009, pp. 1025–1032.
- [22] T. D. HUYNH, N. R. JENNINGS, AND N. R. SHADBOLT, *An integrated trust and reputation model for open multi-agent systems*, Autonomous Agents and Multi-Agent Systems vol. 13, 2006, pp. 119–154.
- [23] G. KAN, 8: *GNUtella*, in Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001, pp. 94–122.
- [24] J. KLEINBERG, *The small-world phenomenon: An algorithmic perspective*, in Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, STOC '00, 2000, ACM, pp. 163–170.
- [25] F. LALANNE, A. CAVALLI, AND S. MAAG, *Quality of experience as a selection criterion for web services*, in SITIS, 2012, pp. 519–526.
- [26] A. LANGLET, 9: *Freenet*, in Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001, pp. 123–132.
- [27] L. LI AND Y. WANG, *The roadmap of trust and trust evaluation in web applications and web services.*, in Advanced Web Services, A. Bouguettaya, Q. Z. Sheng, and F. Daniel, eds., Springer, 2014, pp. 75–99.
- [28] L. LI, Y. WANG, AND E.-P. LIM, *Trust-oriented composite service selection and discovery.*, in Proceedings of IC-SOC/ServiceWave, vol. 5900, 2009, pp. 50–67.
- [29] M. LI, Z. HUA, J. ZHAO, Y. ZOU, AND B. XIE, *Arima model-based web services trustworthiness evaluation and prediction.*, in Proceedings of ICSOC, Springer, 2012, pp. 648–655.
- [30] Z. LI AND Y. JIANG, *Cross-layers cascade in multiplex networks*, in Proceedings of AAMAS, 2014, pp. 269–276.
- [31] G. LIU, *Trust Management in Online Social Networks*, PhD thesis, Department of Computing Faculty of Science Macquarie University, 2013.
- [32] G. LIU AND Y. WANG, *Trust-oriented service provider selection in complex online social networks.*, in Advanced Web Services, A. Bouguettaya, Q. Z. Sheng, and F. Daniel, eds., Springer, 2014, pp. 363–380.
- [33] G. LIU, Y. WANG, AND M. A. ORGUN, *Trust transitivity in complex social networks.*, in AAAI, W. Burgard and D. Roth, eds., AAAI Press, 2011.
- [34] A. LOUATI, J. EL HADDAD, AND S. PINSON, *Trust-based service discovery in multi-relation social networks*, in Proceedings of ICSOC, 2012, pp. 664–671.

- [35] ———, *Towards agent-based and trust-oriented service discovery approach in social networks*, in TRUST14 at AAMAS, 2014, pp. 1–12.
- [36] N. LUHMANN, *Trust and Power*, John Wiley and Sons Ltd., 2012.
- [37] Z. MAAMAR, L. WIVES, B. YOUAKIM, E. SAID, B. KHOULOU, AND F. NOURA, *LinkedWS: A Novel Web Services Discovery Model Based on the Metaphor of Social Networks*, Simulation Modelling Practice and Theory, 2010, pp. 121–132.
- [38] A. MAARADJI, H. HAKIM, J. DAIGREMONT, AND N. CRESPI, *Towards a social network based approach for services composition*, IEEE International Conference on Communications, 2010, pp. 1–5.
- [39] P. MASSA AND P. AVESANI, *Trust-aware recommender systems*, in Proceedings of the ACM Conference on Recommender Systems, ACM, 2007, pp. 17–24.
- [40] B. NEVILLE, M. FASLI, AND J. PITT, *Utilising social recommendation for decision-making in distributed multi-agent systems.*, Expert Systems with Applications vol. 42, 2015, pp. 2884–2906.
- [41] M. ROWLAND, *Intimate Relationships*, McGraw-Hill College Press, 1979.
- [42] J. SABATER AND C. SIERRA, *Regret: a reputation model for gregarious societies*, in AAMAS, 2002, pp. 475–482.
- [43] ———, *Reputation and social network analysis in multi-agent systems*, in Proceedings of AAMAS, 2002, pp. 475–482.
- [44] J. SABATER AND L. VERCOUTER, *Trust and reputation in multiagent systems*, in Multiagent Systems, MIT Press, 2013, pp. 381–420.
- [45] C. SIERRA AND J. DEBENHAM, *Information-based reputation*, First International Conference on Reputation: Theory and Technology, 2009, pp. 5–19.
- [46] M. SZELL, R. LAMBIOTTE, AND S. THURNER, *Multirelational Organization of Large-scale Social Networks in an Online World*, in Proceedings of the National Academy of Science, vol. 107, 2010, pp. 13636–13641.
- [47] L.-H. VU, M. HAUSWIRTH, AND K. ABERER, *Qos-based service selection and ranking with trust and reputation management*, in OTM, Springer-Verlag, 2005, pp. 466–483.
- [48] Y. WANG, L. LI, AND G. LIU, *Social Context-Aware Trust Inference for Trust Enhancement in Social Network based Recommendations on Service Providers*, World Wide Web Journal (WWWJ), vol. 18 (1), 2015, pp. 159–184.
- [49] Y. WANG AND M. SINGH, *Trust representation and aggregation in a distributed agent system.*, in Proceedings of AAI, 2006, pp. 1425–1430.
- [50] Y. WANG AND J. VASSILEVA, *Toward Trust and Reputation Based Web Service Selection: A Survey*, ITSSA Journal, 2007.
- [51] Z. XU, P. MARTIN, W. POWLEY, AND F. H. ZULKERNINE, *Reputation-enhanced qos-based web services discovery.*, in ICWS, IEEE Computer Society, 2007, pp. 249–256.
- [52] J. WOHLGEMUTH, : *Small World Properties of Facebook Group Networks*, PhD thesis, University of Nebraska at Omaha, 2012.
- [53] B. YU AND M. SINGH, *Searching social networks*, in Proceedings of AAMAS, 2003, pp. 65–72.
- [54] Y. ZHANG, Z. ZHENG, AND M. R. LYU, *Wsexpress: A qos-aware search engine for web services*, in Proc. IEEE Int Conf. Web Services (ICWS'10), 2010, pp. 83–90.
- [55] C. ZIEGLER AND G. LAUSEN, *Analyzing correlation between trust and user similarity in online communities*, in Proceedings of Second International Conference on Trust Management, Springer-Verlag, 2004, pp. 251–265.

Edited by: Hoa Dam

Received: April 15, 2015

Accepted: December 31, 2015

