

A Heuristic Algorithm for Trust-Oriented Service Provider Selection in Complex Social Networks

Guanfeng Liu, Yan Wang and Mehmet A. Orgun
 Department of Computing
 Macquarie University
 NSW 2109, Australia
 {gliu, yanwang, mehmet}@science.mq.edu.au

Ee-Peng Lim
 School of Information System
 Singapore Management University
 80 Stamford Road, Singapore
 eplim@smu.edu.sg

Abstract— In a service-oriented online social network consisting of service providers and consumers, a service consumer can search trustworthy service providers via the social network. This requires the evaluation of the trustworthiness of a service provider along a certain social trust path from the service consumer to the service provider. However, there are usually many social trust paths between participants in social networks. Thus, a challenging problem is which social trust path is the optimal one that can yield the most trustworthy evaluation result. In this paper, we first present a novel complex social network structure and a new concept, Quality of Trust (QoT). We then model the optimal social trust path selection with multiple end-to-end QoT constraints as a Multi-Constrained Optimal Path (MCOP) selection problem which is NP-Complete. For solving this challenging problem, we propose an efficient heuristic algorithm, H_OSTP. The results of our experiments conducted on a large real dataset of online social networks illustrate that our proposed algorithm significantly outperforms existing approaches.

I. INTRODUCTION

Online social networking sites have been attracting a large number of participants and are being used as the means for a variety of rich activities. For example, participants carry out business, and share photos and movies on the first generation (e.g., ebay.com) and second generation (e.g., facebook.com) social networking sites respectively [16]. In service-oriented environments, social networks can be used as the means for service consumers to look for trustworthy service providers who are unknown prior to invoking services, with the assistance of information from other participants. For example, at FilmTrust¹, which is a social networking site for movie recommendations, a recommendation receiver can evaluate the trustworthiness of a recommender via the social network between them. For another example, if a social network consists of lots of buyers and sellers, it can be used by a buyer to find the most trustworthy/reputable seller who sells the product preferred by the buyer [7].

In social networks, each node represents a participant and each link between participants corresponds to real-world interactions or online interactions between them (e.g., $A \rightarrow B$ and $A \rightarrow C$ in Fig. 1). One participant can give a trust value to another based on their interactions. For example, a trust rating can be given by a participant to another based on the quality of the movies recommended by the latter at

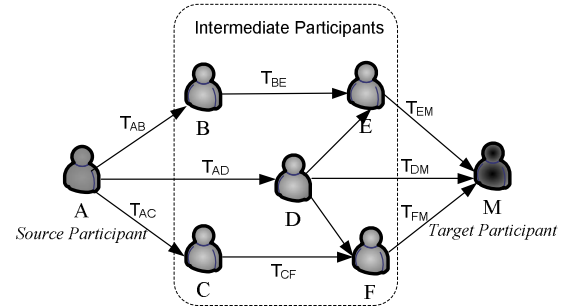


Figure 1: Social network

FilmTrust¹. As each participant usually interacts with many other participants, multiple trust paths may exist between two participants who have no direct links with each other (for example, there are five trust paths from A to M in Fig. 1). If a trust path links two nonadjacent participants (i.e., when there is no direct link between them), the source participant can evaluate the trustworthiness of the target one based on the trust information between the intermediate participants along the path. This process is called *trust propagation* and the path with trust information linking the source participant and the target one is called a *social trust path* [4], [6]. For example, in Fig. 1, if A is a buyer and M is a seller in the social network, A can evaluate the trustworthiness of M along the social trust paths from A to M . We term A as the *source participant* and M as the *target participant*.

In large-scale social networks, there are over tens of thousands of social trust paths between a source participant and the target one [9]. A challenging problem is that among multiple paths, which one is the optimal yielding the most trustworthy result of trust propagation. In the literature, Lin *et al.* [14] propose an optimal social path selection method, where all links are assigned the same weight and the shortest path between the source participant and the target one is selected as the optimal one, neglecting *trust information* between participants. In another reported work [6], the path with the maximal propagated trust value is selected as the most trustworthy social trust path. However, *social relationships* between adjacent participants (e.g., the relationship between a buyer and a seller) and the *recommendation roles* of a participant (e.g., a supervisor as a referee in his postgraduate's job application) have significant influence on trust propagation [1], [19] and

¹<http://trust.mindswap.org/filmtrust/>

can be obtained by using data mining techniques in social networks [17]. Unfortunately, these factors have not been considered in any existing trust propagation and social trust path selection method.

In this paper, we aim to solve the optimal social trust path selection problem in complex social networks. Our main contributions are summarized as follows.

(1) We first present the structure of complex social networks taking *trust information*, *social relationships* and *recommendation roles* of participants into account. In addition, we also introduce a new concept, *Quality of Trust* (QoT), taking the above three factors as attributes. Furthermore, we model the optimal social trust path selection problem as a Multi-Constrained Optimal Path (MCOP) selection problem, which is NP-Complete [8] (*see section II*).

(2) Since existing approximation algorithms [8], [13], [24] for solving the MCOP selection problem do not scale to large social networks and thus can not deliver good performance, we propose an efficient heuristic algorithm, H_OSTP for solving the optimal social trust path selection problem (*see section III*).

(3) We have conducted experiments on a real online social network dataset, *Enron* email corpus². The experimental results show that H_OSTP performs well in both efficiency and the quality of selected social trust paths. (*see section IV*).

The paper is organized as follows. Section II presents the novel social trust path selection model in complex online social networks. Section III introduces our proposed heuristic algorithm, H_OSTP. In the above two sections, before presenting our proposed model, we briefly introduce some related works. Section IV presents the experimental results and analysis. Finally, section V concludes this paper.

II. SOCIAL TRUST PATH SELECTION IN COMPLEX SOCIAL NETWORKS

In this section, we first introduce the complex social network structure and then propose a novel social trust path selection model with end-to-end *Quality of Trust* (QoT) constraints.

A. Complex Social Networks

Several trust management methods have been proposed for online social networks [10]. Golback *et al.* [4] propose a trust inference mechanism for the trust relation establishment between a source participant and the target one based on averaging trust values along the social trust paths. Jamali *et al.* [7] propose a random walk model in a trust-based social network consisting of sellers and buyers. These trust management strategies are based only on ratings given by participants. Again, as pointed in social science theories [1], [19], *social relationships* and *recommendation roles* both have significant influence on participants' decision making. To address these issues, we have proposed a complex social network structure that comprises of the attributes of three impact factors of *trust*, *social intimacy degree* and *role impact factor*, as shown in Fig. 2. These three factors naturally influence the trustworthiness of trust propagation and hence the decision making of a source participant. For completeness, we give

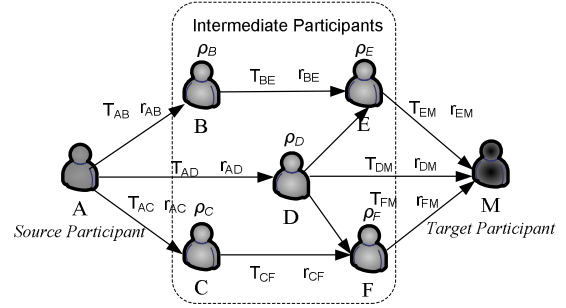


Figure 2: Complex social network

a brief introduction to the complex social network structure. Further details can be found in our previous works [15], [16].

1) *Trust*: In the context of this paper, trust between participants in social networks can be defined as “Trust is 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.” Let $T_{AB} \in [0, 1]$ denote the trust value that participant A assigns to participant B. If $T_{AB} = 0$, it indicates that A completely distrusts B while $T_{AB} = 1$ indicates A completely believes B’s future action can lead to the expected outcome.

2) *Social Intimacy Degree*: As illustrated in social psychology [19], a participant can trust more the participants with whom he/she has intimate social relationships than those with whom he/she has less intimate social relationships. Let $r_{AB} \in [0, 1]$ denote the *Social Intimacy Degree* (SID) between participant A and participant B in online social networks. $r_{AB} = 0$ indicates that A and B have the least intimate social relationship while $r_{AB} = 1$ indicates they have the most intimate social relationship.

3) *Role Impact Factor*: In a certain domain of interest such as opinions on movies or electronic goods, the recommendations of a domain expert may be more credible than that from a novice. In our model, the impact of recommendations in trust calculations is considered by introducing the notion of the role impact factor. Let $\rho_A \in [0, 1]$ denote the *Role Impact Factor* (RIF), illustrating the impact of participant A’s recommendation role on trust propagation. Here $\rho_A = 1$ indicates that A is a domain expert while $\rho_A = 0$ indicates that A has no knowledge in the domain.

Though it is difficult to build up social relationships and comprehensive role hierarchies in all domains, it is feasible to build them up in a particular application. For example, in the work by McCallum *et al.* [17], through mining the subjects and contents of emails in *Enron* Corporation², the social relationship between each email sender and receiver can be discovered and their roles can be known. Then the corresponding SID and RIF value can be calculated based on probabilistic models. In another reported work [22], the SID and RIF values are specified by participants when they interact with each other in a small social network formed by the staff of an university.

²<http://www.cs.cmu.edu/enron/>

B. Quality of Trust (QoT)

In Service-Oriented Computing (SOC), QoS consists of a set of attributes (e.g., cost, delay and availability), used to illustrate the ability of services to guarantee a certain level of performance [3]. Similar to the QoS, we present a new concept, *Quality of Trust* [15].

Definition 1: Quality of Trust (QoT) is the ability to guarantee a certain level of trustworthiness in trust propagation along a social trust path, taking trust (T), social intimacy degree (r), and role impact factor (ρ), as attributes.

In our model, depending on their requirements of services, the source participants can set multiple end-to-end constraints for QoT attributes (i.e., T , r and ρ) as the requirements of trust propagation in a social trust path. In addition, a participant can set different end-to-end QoT constraints in the social trust path selection of different domains.

C. QoT Attribute Aggregation

For satisfying the different requirements of source participants in social trust path selection, we first need to know the aggregated value of each QoT attribute in every social trust path between a source participant and the target participant. The aggregated values of all the QoT attributes are then combined in a utility function defined over social trust paths, and then the path with the best utility value is selected as the optimal social trust path.

1) *Trust Aggregation*: The trust values between a source participant and the target participant in a social path can be aggregated based on trust transitivity (i.e., if A trusts B and B trusts C , then A trusts C to some extent) [4]. In our model, we adopt the strategy proposed in [12], [21], where if there are n participants a_1, \dots, a_n in order in a social trust path (denoted as $p(a_1, \dots, a_n)$), the aggregated trust value is calculated as in Eq. (1). This strategy has been widely used in the literature as a feasible trust aggregation method [12], [21].

$$T_{p(a_1, \dots, a_n)} = \prod_{(a_i, a_{i+1}) \in p(a_1, \dots, a_n)} T_{a_i a_{i+1}} \quad (1)$$

2) *Social Intimacy Degree Aggregation*: Firstly, social intimacy between participants is attenuated with the increasing number of hops between them in a social trust path [11]. Secondly, in the real-world, the intimacy degree is attenuated fast when it is approaching one. In contrast, the intimacy degree is attenuated slowly when it is approaching zero [19]. In other words, the attenuation of social intimacy degree is non-linear in social networks. The aggregated r value in path $p(a_1, \dots, a_n)$ can be calculated by Eq.(2) whose function image is a hyperbolic curve, fitting the characteristic of social intimacy attenuation.

$$r_{p(a_1, \dots, a_n)} = \frac{\prod_{(a_i, a_{i+1}) \in p(a_1, \dots, a_n)} r_{a_i a_{i+1}}}{\theta^\alpha} \quad (2)$$

where θ is the number of hops of path $p(a_1, \dots, a_n)$, $\alpha \geq 1$ is used to control the attenuation speed.

3) *Role Impact Factor Aggregation*: We average the RIF values of intermediate recommending participants in a social

trust path $p(a_1, \dots, a_n)$ as the aggregated value:

$$\rho_{p(a_1, \dots, a_n)} = \frac{\sum_{i=2}^{n-1} \rho_{a_i}}{n-2} \quad (3)$$

D. Utility Function

In our model, the utility measures the trustworthiness of social trust paths. The utility function (denoted as \mathcal{F}) takes the QoT attributes T , r and ρ as arguments in Eq. (4)

$$\mathcal{F}_{p(a_1, \dots, a_n)} = \omega_T * T_{p(a_1, \dots, a_n)} + \omega_r * r_{p(a_1, \dots, a_n)} + \omega_\rho * \rho_{p(a_1, \dots, a_n)} \quad (4)$$

where ω_T , ω_r , and ω_ρ are the weights of T , r and ρ respectively; $0 < \omega_T, \omega_r, \omega_\rho < 1$ and $\omega_T + \omega_r + \omega_\rho = 1$.

The goal of optimal social trust path selection is to select the path that satisfies multiple end-to-end QoT constraints and yields the best utility with the weights specified by the source participant.

III. SOCIAL TRUST PATH SELECTION ALGORITHM

The optimal social trust path selection with multiple end-to-end QoT constraints can be modelled as the classical Multi-Constrained Optimal Path (MCOP) selection problem, which is shown to be NP-Complete [8]. In the literature, several approximation algorithms have been proposed to solve the MCOP selection problem [8], [24], [25], [26]. In this section, we first analyze some of those algorithms and then propose an efficient Heuristic algorithm for Optimal Social Trust Path selection.

A. Existing Algorithms

In an earlier work, Korkmaz *et al.* proposed a heuristic algorithm, called H_MCOP [8]. In this algorithm, both multi-constraint values and QoS attributes values are aggregated based on Eq. (5).

$$g_\lambda(p) \triangleq \left(\frac{q_1(p)}{Q_1}\right)^\lambda + \left(\frac{q_2(p)}{Q_2}\right)^\lambda + \dots + \left(\frac{q_m(p)}{Q_m}\right)^\lambda \quad (5)$$

where $\lambda \geq 1$; $q_i(p)$ is the aggregated value of the i^{th} QoS attribute of path p ; Q_i is the i^{th} QoS constraint of path p .

Firstly, H_MCOP employs Dijkstra's shortest path algorithm [2] to find the path with the minimum g_λ from the target to the source when $\lambda=1$, and stores q_i^v which is the aggregated value of the i^{th} QoS attribute from the target node to intermediate node v . Secondly, from Eq. (5), if any QoS attribute does not satisfy the corresponding QoS constraint in path p , then $g_\lambda(p) > m$, which indicates that no feasible solution exists in the network. This process investigates whether a feasible solution exists in the network. If $g_\lambda(p) \leq m$, the algorithm again employs Dijkstra's shortest path algorithm to search the path with the minimum cost and calculates $q_i^{v'}$ which is the aggregated value of the i^{th} QoS attribute from the source node to intermediate node v . In this process, the aggregated i^{th} QoS attribute value of each node is calculated as $q_i^{v'} + q_i^v$. If $q_i^{v'} + q_i^v$ satisfies the QoS constraint Q_i , then the algorithm continues to search the path with the minimum cost from v to the target node. Otherwise, it stops searching the path with the minimum cost and starts searching the path with the minimum g_λ ($\lambda > 1$).

In this process, if the identified path with the minimum cost is a feasible solution, it is the optimal one.

H_MCOP is one of the most promising algorithms in solving the MCOP selection problem as it outperforms prior existing algorithms in both efficiency and the quality of delivered solutions [8]. In the field of Service-Oriented Computing (SOC), Yu *et al.* [24] propose an approximation algorithm based on H_MCOP, called MCSP_K, which keeps only K paths from a source node to each intermediate node, aiming to reduce the search space and execution time. In their service candidate graph, each node represents a service and all services are categorized into different service sets based on their functionality. Any two nodes in adjacent service sets have a link with each other and thus all paths from a source node to an intermediate node can be enumerated when necessary, avoiding an exhaustive search. But if a network does not have such a typical structure, MCSP_K has to search all paths from a source to each intermediate node and hence the time complexity becomes exponential. Therefore, it does not fit large-scale social networks.

Some other algorithms [25], [26] adopt the integer linear programming method to solve the service selection problem with multi-QoS constraints. However, in [24], they have been proved having low efficiency in finding a near-optimal solution in large-scale networks.

B. H_OSTP

In this section, we propose an efficient heuristic algorithm, H_OSTP, for the optimal social trust path selection with end-to-end QoT constraints in complex social networks. In H_OSTP, we first adopt the *Backward_Search* procedure from the target (denoted as v_t) to the source (denoted as v_s) to investigate whether there exists a feasible solution in the sub-network between v_s and v_t , and record the aggregated QoT attributes (i.e., T , r and ρ) of the identified path from v_t to each intermediate node v . If a feasible solution exists, we then adopt the *Forward_Search* procedure to search the network from v_s to v_t to deliver a near-optimal solution.

In social trust path selection, if a path satisfies multiple QoT constraints, it means that each aggregated QoT attribute (i.e., T , r or ρ) of that path should be larger than the corresponding QoT constraint. Therefore, we propose an objective function in Eq. (6) to investigate whether the aggregated QoT attributes of a path can satisfy the QoT constraints. From Eq. (6), we can see that if any aggregated QoT attribute of a social trust path does not satisfy the corresponding QoT constraint, then $\delta(p) > 1$. Otherwise $\delta(p) \leq 1$.

$$\delta(p) \triangleq \max\left\{\left(\frac{1-T_p}{1-Q_p^T}\right), \left(\frac{1-r_p}{1-Q_p^r}\right), \left(\frac{1-\rho_p}{1-Q_p^\rho}\right)\right\} \quad (6)$$

Backward_Search: In the backward search from v_t to v_s , H_OSTP identifies the path p_s from v_t to v_s with the minimal δ based on the Dijkstra's shortest path algorithm [2]. In the searching process, at each node v_k ($v_k \neq v_t$), the path from v_t to v_k with the minimal δ (denoted as p_k) is identified and T_{p_k} , r_{p_k} and ρ_{p_k} are recorded. According to the following *Theorem 1*, the *Backward_Search* procedure can investigate

whether there exists a feasible solution in the sub-network.

Theorem 1: In the *Backward_Search* procedure, the process of identifying the path with the minimal δ can guarantee to find a feasible solution if one exists in a sub-network.

Proof: Let p_s be a path from v_t to v_s with the minimal δ , and p_* be a feasible solution. Then, $\delta(p_s) \leq \delta(p_*)$. Assume p_s is not a feasible solution, then $\exists \varphi \in \{T, r, \rho\}$ that $\varphi_{p_s} < Q_{v_s, v_t}^\varphi$. Hence, $\delta(p_s) > 1$. Since p_* is a feasible solution, then $\delta(p_*) \leq 1$ and $\delta(p_s) > \delta(p_*)$. This contradicts $\delta(p_s) \leq \delta(p_*)$. Therefore, p_s is a feasible solution. \square

The *Backward_Search* procedure can always identify the path with the minimal δ . If $\delta_{min} > 1$, it indicates there is no feasible solution in the sub-network. If $\delta_{min} \leq 1$, it indicates there exists at least one feasible solution and the identified path is a feasible solution.

Forward_Search: If there exists a feasible solution in the sub-network, a heuristic forward search is executed from v_s to v_t . This process uses the information provided by the above *Backward_Search* to identify whether there is another path p_t which is better than the above returned path p_s (i.e., $\mathcal{F}(p_t) > \mathcal{F}(p_s)$). In this procedure, H_OSTP first searches the path with the maximal \mathcal{F} value from v_s . Assume node $v_m \in \{\text{neighboring nodes of } v_s\}$ is selected based on the Dijkstra's shortest path algorithm. H_OSTP calculates the aggregated QoT attribute values of the path from v_s to v_m (denoted as path p_m). Let p'_m denote the path from v_m to v_t identified in the *Backward_Search* procedure, then a *foreseen path* from v_s to v_t via v_m (denoted as $p_{fm} = p_m + p'_m$) can be identified. Let h denote the number of hops of path p_{fm} . The aggregated QoS attribute values of p_{fm} can be calculated as $T_{p_{fm}} = T_{p_m} * T_{p'_m}$, $r_{p_{fm}} = (r_{p_m} * r_{p'_m})/h^\alpha$ ($\alpha \geq 1$ is the argument for controlling the attenuation speed of r) and $\rho_{p_{fm}} = (\rho_{p_m} + \rho_{p'_m})/(h-1)$. According whether p_{fm} is feasible, H_OSTP adopts the following searching strategies.

Situation 1: If each aggregated QoT attribute of p_{fm} satisfies the corresponding end-to-end QoT constraint, then H_OSTP chooses the next node from v_m with the maximal \mathcal{F} value which is calculated based on the Dijkstra's shortest path algorithm.

Situation 2: If any aggregated QoT attribute of p_{fm} does not satisfy the corresponding end-to-end QoT constraint, then H_OSTP does not search the path from v_m and the link $v_s \rightarrow v_m$ is deleted from the sub-network. Subsequently, H_OSTP performs the *Forward_Search* procedure to search the path from v_s in the sub-network without the link $v_s \rightarrow v_m$.

The following *Theorem 2* illustrates that the social trust path p_t identified by the *Forward_Search* procedure can not be worse than the feasible social trust path p_s identified by the *Backward_Search* procedure. Namely, $\mathcal{F}(p_t) \geq \mathcal{F}(p_s)$.

Theorem 2: With the social trust path p_s identified by the *Backward_Search* procedure and the social trust path p_t identified by the *Forward_Search* procedure in H_OSTP, if p_s is a feasible solution, then p_t is feasible and $\mathcal{F}(p_t) \geq \mathcal{F}(p_s)$.

Proof: Assume that path p_s consists of $n+2$ nodes $v_s, v_1, \dots, v_n, v_t$. In the *Forward_Search* procedure, H_OSTP searches the neighboring nodes of v_s and chooses v_1 from these nodes when a foreseen path from v_s to v_t via v_1 is

Algorithm 1: H_OSTP

Data: $M, Q_{v_s, v_t}^T, Q_{v_s, v_t}^r, Q_{v_s, v_t}^\rho, v_s, v_t$
Result: $p_t, \mathcal{F}(p_t)$

```

1 begin
2    $p_s = \emptyset, p_t = \emptyset$ 
3   Backward_Search ( $M, Q_{v_s, v_t}^T, Q_{v_s, v_t}^r, Q_{v_s, v_t}^\rho, v_s, v_t$ )
4   if  $\delta(p_s) > 1$  then
5     Return no feasible solution
6   else
7     Forward_Search ( $M, \text{Dist}(v).T, \text{Dist}(v).r,$ 
8        $\text{Dist}(v).\rho, Q_{v_s, v_t}^T, Q_{v_s, v_t}^r, Q_{v_s, v_t}^\rho, v_s, v_t$ )
9     Return  $p_t$  and  $\mathcal{F}(p_t)$ 
9 end
```

feasible and the current path from v_s to v_1 has the maximal \mathcal{F} . This step is repeated at all the nodes between v_1 and v_n until a social trust path p_t is identified. If at each search step, only one node (i.e., v_1, \dots, v_n) has a feasible foreseen path, then p_t is the only feasible solution in the sub-network between v_s and v_t . According to *Theorem 1*, then $p_t = p_s$. Thus, $\mathcal{F}(p_t) = \mathcal{F}(p_s)$. Otherwise, if $p_t \neq p_s$, It can lead to $\mathcal{F}(p_t) > \mathcal{F}(p_s)$ by maximizing the \mathcal{F} value in all candidate nodes which have feasible foreseen paths based on the Dijkstra's shortest path algorithm. Therefore, *Theorem 2* is correct. \square

If there exists only one feasible solution in the sub-network, it can be identified by both the *Backward_Search* procedure and the *Forward_Search* procedure, and it is the optimal solution. Otherwise, if there exists more than one feasible solutions in the sub-network, then the solution identified by the *Forward_Search* procedure is near-optimal or optimal, which is better than the one identified by the *Backward_Search* procedure.

Next, we introduce the notations used in H_OSTP.

Notations:

- $\text{Dist}(v).T, \text{Dist}(v).r$ and $\text{Dist}(v).\rho$: the aggregated values of the identified social trust path from v_t to v in the *Backward_Search* procedure.
- $\text{Dist}(v).\delta$: the δ value of the identified social trust path from v_t to v in the *Backward_Search* procedure.
- M : an adjacency matrix that represents the sub-network between v_s to v_t .
- $M(v_x, v_y).T, M(v_x, v_y).r$ and $M(v_x, v_y).\rho$: the trust value, social intimate degree between v_x and v_y , and the role impact factor of v_y .
- p_s and p_t : the paths identified by the *Backward_Search* procedure and the *Forward_Search* procedure respectively.
- pre_x : an array stores the ordered nodes in the shortest path from v_t to each node in the *Backward_Search* procedure.
- pre_y : an array stores the ordered nodes in the shortest path from v_s to each node in the *Forward_Search* procedure. For example, $pre_x(v'') = v'$ represents in the shortest path from v_t to v'' , v' is the preceding node of v'' .

Algorithm 2: Backward_Search

Data: $M, Q_{v_s, v_t}^T, Q_{v_s, v_t}^r, Q_{v_s, v_t}^\rho, v_s, v_t$
Result: $\delta(p_s), \text{Dist}(v).T, \text{Dist}(v).r, \text{Dist}(v).\rho$

```

1 begin
2   Set  $v_x.\delta = \infty (v_x \neq v_t), v_t.\delta = 0, S_x = \emptyset$ 
3   Add  $v_t$  into  $S_x$ 
4   while  $S_x \neq \emptyset$  do
5      $v_a.\delta = \min(v_a^*.\delta) (v_a^* \in S_x)$ 
6     for each  $v_b \in \text{adj}[v_a]$  do
7        $h$  is the number of hops of the path from  $v_t$  to  $v_b$ 
8        $\delta(p_b) = \max[(1 - v_b.T * M(v_a, v_b).T / (1 - Q_{st}^T), (1 - v_b.r * M(v_a, v_b).r / h^\alpha) / (1 - Q_{st}^r), (1 - (v_b.\rho + M(v_a, v_b).\rho) / (h - 1) / (1 - Q_{st}^\rho))]$ 
9       if  $v_b \notin S_x$  then
10         Put  $v_b$  into  $S_x$ 
11          $pre_x(v_b) = v_a$ 
12       else if  $\delta(p_b) < \text{Dist}(v_b).\delta$  then
13          $\text{Dist}(v_b).\delta = \delta(p_b)$ 
14          $\text{Dist}(v_b).T = v_a.T * M(v_a, v_b).T$ 
15          $\text{Dist}(v_b).r = v_a.r * M(v_a, v_b).r$ 
16          $\text{Dist}(v_b).\rho = v_a.\rho + M(v_a, v_b).\rho$ 
17         Put  $v_b$  into  $S_x$ 
18          $pre_x(v_b) = v_a$ 
19     Remove  $v_a$  from  $S_x$ 
20    $p_s \leftarrow pre_x(v_s)$  to  $pre_x(v_t)$ 
21   Return  $p_s$  and  $\delta(p_s)$ 
22 end
```

- S_x and S_y : the sets of expanding node candidates in *Backward_Search* and *Forward_Search* respectively.
- $v.\mathcal{F}$: the utility of the identified social trust path from v_s to v in the *Forward_Search* procedure.
- $v.T, v.r$ and $v.\rho$: the aggregated QoT attributes values of the identified social trust path from v_s to v in the *Forward_Search* procedure.

The process of H_OSTP is as follows.

Step 1: Start the *Backward_Search* procedure. Add v_t into S_x . At each node $v_x (v_x \neq v_t)$ in the sub-network, set $\text{Dist}(v_x).\delta = \infty$ and $\text{Dist}(v_t).\delta = 0$. Select the node v_a from S_x , where the δ value of the path from v_t to v_a (denoted as p_a) is the minimum of all δ of the paths from v_t to v_a^* ($v_a^* \in S_x$) (lines 1-3 in Algorithm 1 and lines 1 to 5 in Algorithm 2).

Step 2: At each $v_b \in \{\text{neighboring nodes of } v_a\}$, calculate δ value of the identified social trust path from v_t to v_b (denoted as p_b). If $v_b \notin S_x$, add v_b into S_x . Otherwise, if the current δ of v_b less than the previous δ value recorded at v_b , then replace the stored δ with the current δ and record T_{p_b}, r_{p_b} and ρ_{p_b} at v_b . Add v_b into S_x and set $pre_x(v_b) = v_a$ (lines 1-3 in Algorithm 1 and lines 6 to 19 in Algorithm 2).

Step 3: Remove v_a from S_x . If $S_x \neq \emptyset$, then go to Step 1. Otherwise return p_s through searching $pre_x(v_s)$. If $\delta(p_s) \leq 1$, go to Step 3. Otherwise terminate (i.e., there is no feasible solution in the sub-network) (lines 4 to 5 in Algorithm 1 and lines 20 to 22 in Algorithm 2).

Step 4: Start the *Forward_Search* procedure. Add v_s into S_y . At each node $v_y (v_y \neq v_s)$ in the sub-network, set $v_y.\mathcal{F} = 0$, and $v_s.\mathcal{F} = \infty$. Select the node v_i from S_y , where the $1/\mathcal{F}$ value of the path from v_s to v_i (denoted as p_i) is the minimum

Algorithm 3: Forward_Search

Data: $M, \text{Dist}(v).T, \text{Dist}(v).r, \text{Dist}(v).\rho, Q_{v_s, v_t}^T, Q_{v_s, v_t}^r, Q_{v_s, v_t}^\rho$

Result: $p_t, \mathcal{F}(p_t)$

```

1 begin
2   Set  $\mathcal{F}' = 1/\mathcal{F}, v_y.\mathcal{F}' = \infty (v_y \neq v_s), v_s.\mathcal{F}' = 0, S_y = \emptyset$ 
3   Add  $v_s$  into  $S_y$ 
4   while  $S_y \neq \emptyset$  do
5      $v_i.\mathcal{F}' = \min(v_i^*.\mathcal{F}') (v_i^* \in S_y)$ 
6     for each  $v_j \in \text{adj}[v_i]$  do
7        $h'$  is the number of the hops of the foreseen path
        from  $v_s$  to  $v_t$  via  $v_j$ 
8        $\text{temp}_T = v_i.T * M(v_i, v_j).T * \text{Dist}(v_j).T$ 
        $\text{temp}_r = v_i.r * M(v_i, v_j).r * \text{Dist}(v_j).r$ 
        $\text{temp}_\rho = v_i.\rho + M(v_i, v_j).\rho + \text{Dist}(v_j).\rho$ 
9       if  $\text{temp}_T \geq Q_{st}^T$  and  $\text{temp}_r/h'^\alpha \geq Q_{st}^r$  and
        $\text{temp}_\rho/(h' - 1) \geq Q_{st}^\rho$  then
10        if  $v_j \notin S_y$  then
11          Put  $v_j$  into  $S_y$ 
12           $\text{pre}_y(v_j) = v_i$ 
13        else if  $\mathcal{F}'(p_j) < v_j.\mathcal{F}'$  then
14           $v_j.\mathcal{F}' = \mathcal{F}'(p_j)$ 
15           $v_j.T = v_i.T * M(v_i, v_j).T$ 
16           $v_j.r = v_i.r * M(v_i, v_j).r$ 
17           $v_j.\rho = v_i.\rho + M(v_i, v_j).\rho$ 
18          Put  $v_j$  into  $S_y$ 
19           $\text{pre}_y(v_j) = v_i$ 
20      Remove  $v_i$  from  $S_y$ 
21       $p_t \leftarrow \text{Pre}_y(v_t)$  to  $\text{Pre}_y(v_s)$ 
22      Return  $p_t$  and  $\mathcal{F}(p_t)$ 
23 end

```

in all $1/\mathcal{F}$ values of the paths from v_s to v_i^* ($v_i^* \in S_y$) (lines 6 to 7 in Algorithm 1 and lines 1 to 5 in Algorithm 3).

Step 5: At each $v_j \in \{\text{neighboring nodes of } v_i\}$, calculate \mathcal{F} value of the identified path from v_s to v_j (denoted as p_j). If the current $1/\mathcal{F}(p_j)$ is less than the value recorded at node v_j , then calculate each aggregated QoT attribute value T_{p_j}, r_{p_j} and ρ_{p_j} . If each aggregated QoT value can satisfy the corresponding QoT constraint, then replace the stored $1/\mathcal{F}(p_j)$ with the current $1/\mathcal{F}(p_j)$ at v_j and set $\text{pre}_y(v_j) = v_i$. Otherwise, set $M(v_i, v_j).T = 0, M(v_i, v_j).r = 0$ and $M(v_i, v_j).\rho = 0$ (lines 6 to 7 in Algorithm 1 and lines 6 to 19 in Algorithm 3).

Step 6: Remove v_i from S_x . If $S_y \neq \emptyset$, then go to Step 5. Otherwise, return p_t through searching array $\text{pre}_y(v_t)$ (lines 8 to 9 in Algorithm 1 and lines 20 to 23 in Algorithm 3).

H_OSTP consumes twice the execution time of Dijkstra's shortest path algorithm. The time complexity of H_OSTP is $O(N \log N + E)$, where N is the number of nodes in the sub-network between v_s and v_t , and E is the number of links in the sub-network. H_OSTP has the same time complexity with H_MCOP. But our proposed heuristic algorithm has better searching strategies than H_MCOP and thus outperforms it in both efficiency and the quality of selected social trust paths (see a more detailed analysis in section IV-B).

IV. EXPERIMENTS

The studies of social network properties can be traced back to 1960's when the *small-world* characteristic in social

networks was validated by Milgram [18], through illustrating that the average path length between two Americans was about 6 hops in an experiment of mail sending. More recently, Mislove *et al.* [20] analyzed several popular social networks including Facebook, MySpace, Flickr and Orkut, and validated the small-world and *power-law* (i.e. in a social network, the probability that a node has degree k is proportional to k^{-r} , $r > 1$) characteristics of online social networks by using data mining techniques. The *Enron* email dataset² has also been proved to possess the small-world and power-law characteristics of social networks and thus it has been widely used in the studies of social networks [5], [17], [23]. In addition, as we explained in section II-A3 the social intimate degree between participants and the role impact factor of participants can be calculated through mining the subjects and contents of emails in the *Enron* email dataset [17]. Therefore, in contrast to other real social network datasets (e.g., Epinions³ and FilmTrust⁴), the *Enron* email dataset fits our proposed complex social network structure better. Thus, to validate our proposed algorithm, we select the *Enron* email corpus² with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) as the dataset for our experiments.

A. Experiment Settings

As discussed in section III-A, H_MCOP is the most promising algorithm for the MCOP selection problem. Based on it, several approximation algorithms [13], [24] have been proposed for the quality-driven service selection in the field of SOC. But they do not fit the structure of large-scale complex social networks. Thus, to study the performance of our proposed heuristic algorithm H_OSTP, we compare it with H_MCOP [8] in both execution time and the utilities of identified social trust paths (see section IV-B).

Both H_OSTP and H_MCOP are implemented using Matlab R2008a running on an IBM ThinkPad SL500 laptop with an Intel Core 2 Duo T5870 2.00GHz CPU, 3GB RAM, Windows XP SP3 operating system and MySQL 5.1.35 database.

In our experiments, the T, R and ρ values are randomly generated. The argument for controlling the attenuation speed is set as $\alpha = 1.5$. The end-to-end QoT constraints specified by a source participant are set as $Q = \{Q^T \geq 0.05, Q^r \geq 0.001, Q^\rho \geq 0.3\}$ and the weights of attributes in the utility function specified by the source participant are set as $\omega_t = 0.25, \omega_r = 0.25$ and $\omega_\rho = 0.5$.

In order to evaluate the performance of our proposed heuristic algorithm in the sub-networks of different scales and structures, we first randomly select 100 pairs of source and target participants from the *Enron* email dataset². We then extract the corresponding 100 sub-networks between them by using the exhaustive searching method. Among them, the maximal length of a social trust path varies from 4 to 7 hops following the *small-world* characteristic. These sub-networks are grouped by the number of hops. In each group they are ordered by the number of nodes of them. Table I list the properties of the simplest and the most complex sub-networks

³<http://epinions.com/>

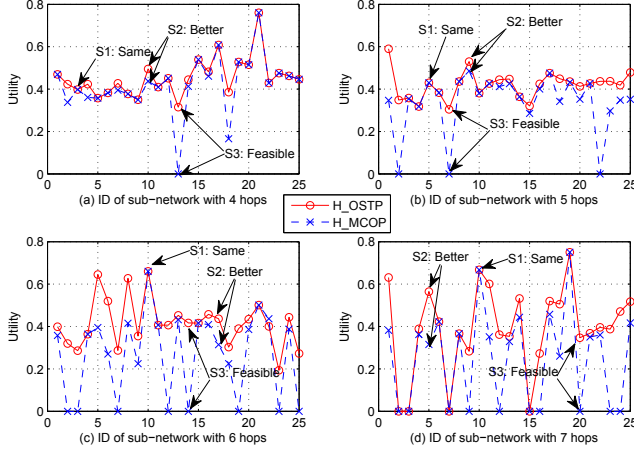


Figure 3: The comparison in path utilities of sub-networks

in each group of hops. In the simplest case, the sub-network has 33 nodes and 56 links (4 hops), while in the most complex case, the sub-network has 1695 nodes and 11175 links (7 hops).

B. Performance in Social Trust Path Selection

With each sub-network extracted from the Enron email corpus, we repeat the experiment 5 times for each of H_OSTP and H_MCOP. The results are plotted in Fig. 3 and 4 where the execution time of each of H_OSTP and H_MCOP is averaged based on the 5 independent runs.

Table I: The properties of the simplest and the most complex sub-networks in each group of hops

Hops	The simplest sub-network			The most complex sub-network		
	ID	Nodes	Links	ID	Nodes	Links
4	1	33	56	25	393	1543
5	1	49	90	25	680	2670
6	1	48	74	25	1300	6396
7	1	40	64	25	1695	11175

Results (Utility). From Fig. 3, we can observe that in any case, our H_OSTP does not yield any utility worse than that of H_MCOP (e.g., S1 in Fig. 3 (a) to (d)) while in most sub-networks (i.e., 59% of total sub-networks), the utilities of social trust paths identified by H_OSTP are better than those of H_MCOP (e.g., S2 in Fig. 3 (a) to (d)). The sum of utilities computed by H_OSTP and H_MCOP in the sub-networks with each group of hops is listed in Table II. From Table II, we can see that the sum of utilities of our proposed heuristic algorithm is 10.78% more than that of H_MCOP in 4 hops sub-networks, 12.37% more in 5 hops, 15.75% more in 6 hops and 15.57% more in 7 hops.

Analysis (Utility). From the above results, we can see that H_OSTP can yield a better social trust path than H_MCOP in most cases. This is because when a social trust path with the maximal utility is a feasible solution in a sub-network, both H_MCOP and H_OSTP can identify it as the optimal solution. Thus, they can identify the same social trust path with the same utility. However, when the social trust path with the maximal

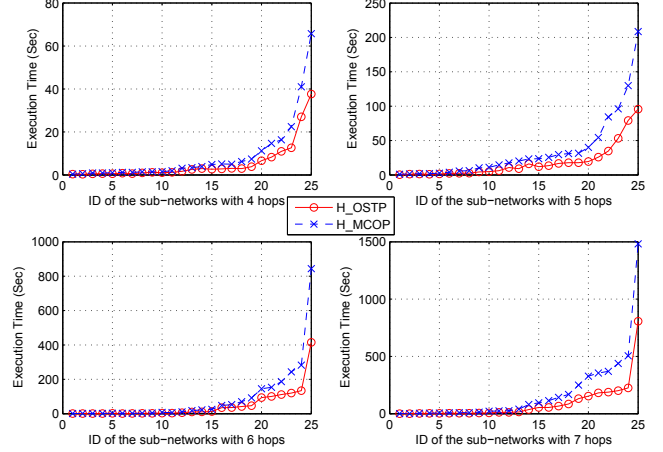


Figure 4: The comparison in execution time

utility is not a feasible solution, H_MCOP stops searching the path with the minimum cost and consequently start searching the social trust path with the minimum g_λ ($\lambda > 1$). This heuristic search strategy can hardly find a near-optimal solution and sometimes returns an infeasible one even when a feasible solution exists (e.g., S3 in Fig. 3 (a) to (d)). In contrast, as illustrated by *Theorem 1*, H_OSTP can identify a feasible solution if it exists (e.g., S3 in Fig. 3 (a) to (d)). In addition, as illustrated by *Theorem 2*, H_OSTP can identify a near-optimal social trust path satisfying the end-to-end QoT constraints if it exists. Therefore, in this case, the quality of the social trust path identified by H_OSTP is better than H_MCOP.

Results (Execution Time). From Fig. 4, we can observe that the execution time of H_OSTP is less than that of H_MCOP in all sub-networks. The total execution time of each of H_OSTP and H_MCOP in each group of hops is listed in Table II. From Table II, we can see that the total execution time of our proposed heuristic algorithm is only 60.06% of that of H_MCOP in 4 hops sub-networks, 51.33% in 5 hops, 53.56% in 6 hops and 50.29% in 7 hops.

Analysis (Execution Time). From the above results, we can see that H_OSTP is much more efficient than H_MCOP. The reasons are twofold. Firstly in the *Forward_Search* procedure, H_OSTP does not calculate g_λ ($\lambda > 1$) which consumes a large amount of execution time when $\lambda \rightarrow \infty$ [13]. Secondly, in the searching process, when any aggregated QoT attribute of a selected path from v_s to v_y ($v_y \neq v_t$) does not satisfy the corresponding QoT constraint, node v_y is not regarded as a candidate to be selected in the next searching step, which can reduce the search space and thus significantly save the execution time.

Through the above experiments conducted in sub-networks with different scales and structures, we can see that overall H_OSTP is superior to H_MCOP in both the execution time and the quality of selected social trust path.

V. CONCLUSIONS

In this paper, we have presented a complex social network structure that takes trust information, social relationship and recommendation roles into account, reflecting the real-world

Table II: The comparison of utility and execution time

Algorithms	The sum of utility				The sum of execution time (sec)			
	4 hops	5 hops	6 hops	7 hops	4 hops	5 hops	6 hops	7 hops
H_OSTP	11.3515	10.4770	10.3937	9.7074	133.9208	449.6327	1.1924e+003	2.2585e+003
H_MCOP	10.5265	8.4712	6.6006	6.2363	222.9832	875.9788	2.2262e+003	4.4913e+003
difference	10.78% more	12.37% more	15.75% more	15.57% more	39.94% less	48.67% less	46.44% less	49.71% less

situations better. For selecting the optimal social trust path with end-to-end QoT constraints in complex social networks, which is an NP-Complete problem, we have also proposed H_OSTP, an efficient heuristic algorithm. The results of experiments conducted on a real dataset of social networks demonstrate that H_OSTP significantly outperforms existing methods in both execution time and optimal social trust path selection.

In our future work, we plan to develop a new trust-oriented social service search engine, which maintains a database of participants and the complex social network containing them. In this system, our proposed method will be applied, for instance, to help a buyer identify the most trustworthy one from all sellers selling the product preferred by the buyer.

ACKNOWLEDGEMENTS

This research has been supported in part by an Australian Research Council (ARC) Discovery grant (DP1094799) and a Macquarie University Research Excellence Scholarship (MQRES).

REFERENCES

- [1] P. S. Adler. Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, 12(12):215–234, 2001.
- [2] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.
- [3] L. Franken. Quality of service management: A model-based approach. *PhD Thesis, Centre for Telematics and Information Technology*, 1996.
- [4] J. Golbeck and J. Hendler. Inferring trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, 6(4):497–529, 2006.
- [5] J. Goldstein, A. Kwasinski, P. Kingsbury, R. E. Sabin, and A. McDowell. Annotating subsets of the enron email corpus. In *Proceedings of the Third Conference on Email and Anti-Spam*, 2006.
- [6] C. Hang, Y. Wang, and M. Singh. Operators for propagating trust and their evaluation in social networks. In *AAMAS’09*, pages 1025–1032.
- [7] M. Jamali and M. Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *KDD’09*, pages 29–42.
- [8] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *INFOCOM’01*, pages 834–843.
- [9] J. Kunegis, A. Lommatzsch, and C. Bauckhang. The slashdot zoo: Mining a social network with negative edges. In *WWW’09*, pages 741–750.
- [10] U. Kuter and J. Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence model. In *AAAI’07*, pages 1377–1382.
- [11] G. Levinger. Development and change. *Close Relationships*, pages 315–359, 1983.
- [12] L. Li, Y. Wang, and E. Lim. Trust-oriented composite services selection and discovery. In *ICSOC’09*, pages 50–67.
- [13] L. Li, J. Wei, and T. Huang. High performance approach for multi-qos constrained web services selection. In *ICSOC’07*, pages 283–295.
- [14] C. Lin, N. Cao, S. Liu, S. Papadimitriou, J. Sun, and X. Yan. Smallblue: Social network analysis for expertise search and collective intelligence. In *ICDE’09*, pages 1483–1486.
- [15] G. Liu, Y. Wang, and M. Orgun. Quality of trust for social trust path selection in complex social networks. In *AAMAS’10*.
- [16] G. Liu, Y. Wang, and M. Orgun. Trust inference in complex trust-oriented social networks. In *IEEE CSE’09*, pages 996–1001.
- [17] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1):249–272, 2007.
- [18] S. Milgram. The small world problem. *Psychology Today*, 1967.
- [19] R. Miller, D. Perlman, and S. Brehm. *Intimate Relationships*. McGraw-Hill College, 4th edition, 2007.
- [20] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *ACM IMC’07*, pages 29–42.
- [21] F. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agent Multi-Agent System Journal*, 16(1):57–74, February 2008.
- [22] S. Yang, J. Zhang, and I. Chen. Web 2.0 services for identifying communities of practice. In *SCC’07*, pages 130–137.
- [23] S. Yoo, Y. Yang, F. Lin, and I. Moon. Mining social networks for personalized email prioritization. In *KDD’09*, pages 967–976.
- [24] T. Yu, Y. Zhang, and K. Lin. Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Transactions on the Web*, 1(1), 2007.
- [25] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Sheng. Quality driven web services composition. In *WWW’03*, pages 411–421.
- [26] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web service composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, 2004.