Multiple QoT Constrained Social Trust Path Selection in Complex Social Networks

Guanfeng Liu and Yan Wang
Department of Computing
Macquarie University
Sydney, NSW, Australia, 2109
{*guanfeng.liu, yan.wang*}*@mq.edu.au*

Duncan S. Wong
Department of Computer Science
City University of Hong Kong
Hong Kong, China
duncan@cityu.edu.hk

***Abstract*—In recent years, online social networks with numerous participants have been used as the means for rich activities, where trust is one of the most important indications for participants' decision making, demanding the evaluation of the trustworthiness of a target participant along a certain social trust path from a source participant. However, there are usually many social trust paths between participants. Thus, a challenging problem is how to select the optimal one from massive social trust paths yielding the most trustworthy trust evaluation result based on participants trust evaluation criteria.**

To address this issue, in this paper, we first propose a new Multiple QoT Constrained Social Trust Path (MQCSTP) selection model which considers both *adjacent constraints* and *end-to-end constraints*, based on a novel concept Quality of Trust (QoT) and a novel complex social network structure. We then model the MQCSTP selection as the classical NP-Complete Multi-Constrained Optimal Path (MCOP) selection problem. For solving this problem, we propose an effective and efficient heuristic algorithm, called H_MQCSTP. The results of our experiments conducted on a real dataset of online social networks illustrate that the proposed method outperforms existing models in both efficiency and the quality of delivered solutions.

keywords: Social network; trust propagation; social trust path

I. Introduction

In recent years, social networking websites, such as MySpace[1] and Facebook[2] have been attracting a large number of participants. These websites have been used as the means for diverse activities. For example, according to a survey on 2600 hiring managers in 2008 by CareerBuilder (careerbuilder.com, a popular job hunting website), 22% of those managers used social networking sites to investigate potential employees. The ratio increased to 45% in June 2009 and 72% in January 2010. In addition, Microsoft has developed a dynamic *CRM* (Customer Relationship Management) system[3], which allows business professionals to monitor and analyze customers' conversations on social networking sites to improve their products and services.

Social network consists of nodes representing participants and links corresponding to real-world interactions or online interactions between nodes (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 one participant to the other based on the quality of the movies recommended by the latter at FilmTrust [6] which is a social networking site for movie recommendations. As multiple trust paths may exist between two nonadjacent participants, such as the trust path $A \rightarrow B \rightarrow E \rightarrow M$ and $A \rightarrow D \rightarrow M$ in Fig. 1. The source participant can evaluate the trustworthiness of the target participant 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* [6, 8]. For example, in Fig. 1, if A is an employer and M is an employee candidate in the social network, A can evaluate the trustworthiness of M along the social trust paths from A to M.

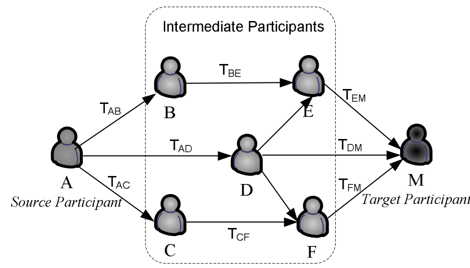

Figure 1. Social network

In large-scale social networks, there may be over many thousands of social trust paths between a source participant and a target participant [11]. Evaluating the trustworthiness of a target participant along all these social trust paths leads to huge computation time [2]. 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*. [15] 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 participant is selected as the optimal one. This method neglects *trust information* between participants. In another reported work [8], the path with the maximal propagated trust value is selected as the most trustworthy social trust path. But these methods neglect *social relationships* between adjacent participants (e.g., the relationship between an employer and an employee) and the *recommendation roles* of a participant (e.g., a supervisor as a referee in his postgraduate's job application), both of which have significant influence on trust propagation [1, 23] and can be obtained by using data mining techniques in social networks [21, 25, 28]. In addition, in social networks, a source participant may have different purposes in evaluating the trustworthiness of the target participant, such as hiring employees or introducing products. Therefore, a source participant should be able to set certain constraints on the above factors in trust propagation, which can help the source participant select the optimal social trust path that can yield the most trustworthy trust propagation

[1]www.myspace.com
[2]www.facebook.com
[3]http://crm.dynamics.com/

978-0-7695-4745-9/12 $26.00 © 2012 IEEE
DOI 10.1109/TrustCom.2012.210

result. However, the above social trust path selection methods neither consider these important impact factors nor support these selection criteria.

In our previous work, we have proposed the social trust path selection models [17, 20] to select one or K optimal social trust paths. In these models, the above impact factors and the constraints of these factors are considered. Although their performance is good in path selection, they only support the *end-to-end* constraints (i.e., the constraint of the aggregated values of these factors in a path). But in applications, a source can have constraints of the social relationship and/or trust between any two adjacent participants, and/or the recommendation role of each of the intermediate participants in a path. For example, in Fig. 1, source A may have constraints of the trust value between any two adjacent participants and the recommendation role of each of the intermediate participants in the selected path from A to M to reflect A's path selection criteria. We term these constraints as *adjacent constraints*, which are not supported by any existing methods. Therefore, in order to deliver more reasonable trust evaluation results, it is necessary to consider these *adjacent constraints* of impact factors in social trust path selection.

In this paper, we first present the structure of complex social networks taking *trust*, *social relationships* and *recommendation roles* into account, and a concept, *Quality of Trust* (QoT). We then propose a Multiple QoT Constrained Social Trust Path Selection (MQCSTP) model which considers both *Adjacent QoT Constraint* (AQC) and *End-to-End QoT Constraint* (EEQC). As the MQCSTP selection problem can be modeled as a classical Multi-Constrained Optimal Path (MCOP) selection problem, which is NP-Complete [10], and the existing approximation algorithms [3, 10, 14, 30] for solving the MCOP selection problem can not scale to large realistic social networks and thus can not deliver good performance, we propose an efficient heuristic algorithm, H_MQCSTP for solving MQCSTP selection problem. Finally, we have conducted experiments on a real online social network dataset, that is, the *Enron* email corpus[4]. Experimental results illustrate H_MQCSTP outperforms existing models in both efficiency and the quality of delivered solutions..

II. Related work

The studies of social network properties can be traced back to 1960's when the *small-world* characteristic in social networks were validated by Milgram [22], through illustrating that the average path length between two Americans was about 6 hops in an experiment of mail sending. In recent years, Mislove *et al*. [24] analyzed several popular social networks including Facebook, MySpace, Flickr and Orkut, and validate 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.

Trust is a critical factor for the decision-making of participants in online social networks [12]. In the studies of trust propagation in social networks. Golback *et al*. [6] propose a trust propagation mechanism, where the trustworthiness of a target participant is calculated based on averaging trust values along all social trust paths between a participant and the target one. In addition, Guha *et al*. [7] propose a trust propagation model, where the number of hops in trust propagation is considered in calculating the propagated trust values of a target participants based on all social trust paths between a source participant and the target one. Furthermore, Bi *et al.* [3] propose a trust propagation method in an email based social network, where each node represents an email sender or receiver who has a global reputation and local trust values to other participants. If there are m social trust paths, and n nodes in each social trust path between a source participant (e.g., A) and the target one (e.g., M), and the global reputation value of node i is T_i, then the propagated trust value between A and M based on trust path j is $E_W(j) = \frac{\sum_{i=1}^{n} W_i \cdot T_i}{n}$. The weight of node i is defined as $W_i = 2^{-(hop(i,A))}$, where A is the truster and $hop(i, A)$ is the number of hops (direct edges) from node i to A. The aggregated trust value based on m social trust paths is $LT_{AM} = \frac{\sum_{j=1}^{m} W'_j \cdot E_W(j)}{m}$. In the studies of trust management of recommendation system in social networks, Walter *et al*. [26] propose a trust model in a recommendation system based on social networks where a participant can give a trust value to a recommender based on the recommendation behavior. This trust value is visible and regarded as a reference for other participants to select recommendations. Jamali *et al.* [9] propose a random walk model in a trust-based social network consisting of sellers and buyers. In their model, a buyer performs several random walks with a fixed number of hops to find the ratings to a seller. The degree of confidence of the seller is calculated based on the number of random walk hops, ratings and the number of random walk paths.

As pointed in social science theories [1, 23], both *social relationships* (e.g., the relationship between a buyer and a seller) and *recommendation roles* (e.g., the supervisor as a referee in a job application) have significant influence on trust relation establishment, and can be obtained from online social networks by using data mining techniques [21, 25]. However, existing models and applications in the field of social networks neglect these factors. In addition, in these trust management methods, all social trust paths between a source participants and the target one are selected to evaluate the trustworthiness of a target participants, which leads to a huge computation time [2] and thus do not fit large-scale social networks.

In the literature, there are only a few works addressing the social path selection problem in social networks. *SmallBlue* [15] is an online social network constructed for IBM staff. In this system, between a source participant and a target participant, up to 16 social paths with no more than 6 hops are selected and the shortest one is taken as the optimal path. However, in this method, some significant influence factors including *trust*, *recommendation roles* and *social relationships* are not taken into account in social path selection. Hang *et al*. [8] propose a social trust path selection method in online social networks, where the social trust path with the highest belief (i.e., the maximum of propagated trust values) is selected as the optimal one that yields the most trustworthy results of trust propagation between a source participant and the target participant. In their model, although trust information is taken into consideration in social trust path selection, other two important factors (i.e., social relationship and recommendation role) are not considered. In addition, the above social trust path selection methods do not support the selection criteria specification by source participants in different applications. In [17, 20], we have proposed the models for one or K optimal social trust path selection, where the impact factors and end-to-end constraints are considered. However, all existing methods including our previous models do not support the *adjacent constraints* which is one of the most important path selection

[4]http://www.cs.cmu.edu/enron/

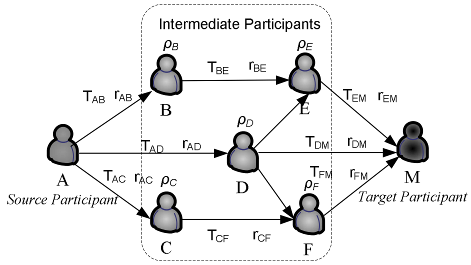

Figure 2. Complex social network

criteria from a source participant.

III. Complex Social Networks

The existing first and second generation of online social network structures can not illustrate the complex social information of social networks in real-world [16]. To address this issue, we first present the complex social network structure, depicted in Fig. 2. It comprises the attributes of three impact factors. They are *trust*, *social intimacy degree* and *role impact factor*, which influence the trustworthiness of trust propagation and hence the decision making of a source participant. Here we give a brief introduction about the structure. The more details can be found in our previous work [19].

1) Trust.: In the context of this paper, trust between participants in social networks is 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 [23], a participant can trust more the participants with whom he/she has more 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 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.: Rich activities of participants in social networks can be categorized into different domains (e.g., hiring employees or product sale) based on their characteristics [27]. As illustrated in social psychology [1], in a certain domain of interest, recommendations from a domain expert are more credible than that from a beginner. Let $\rho_A \in [0, 1]$ denote the *Role Impact Factor* (RIF), illustrating the impact of participant A's recommendation role on trust propagation in a certain domain. $\rho_A = 1$ 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 comprehensive social relationships and 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*. [21], through mining the subjects and contents of emails in *Enron* Corporation[4], the social relationship between each email sender and receiver can be discovered and their roles can be obtained. Then the corresponding SID and RIF value can be calculated based on probabilistic models. In addition, in academic social networks formed by large databases of Computer Science literature (e.g, DBLP or ACM Digital Library), the social relationships between two scholars (e.g., co-authors or supervisor and his/her students) and the role of scholars (e.g., professor in the field of data mining) can be mined from publications and their homepages. The SID and RIF values can be calculated by applying the PageRank model [25]. The detailed mining method is out of the scope of this paper.

IV. Multiple QoT Constrained Social Trust Path Selection

A. Quality of Trust (QoT)

In Service-Oriented Computing (SOC), QoS consists of a set of attributes, used to illustrate the ability of services to guarantee a certain level of performance [5]. Similar to the QoS, we present a concept, *Quality of Trust* [19].

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.

B. QoT Constraint

Activities in social networks can be divided into different domains [19], such as *hiring employees* or *product sale*. In different domains, a source participant can have different preferences in evaluating the trustworthiness of the target participant. Therefore, a source participant should be able to set certain constraints of QoT attributes, which can help the source participant select the optimal social trust path, satisfying the requirements in different domains. For this purpose, a source participant can specify two types of QoT constraints, i.e., *Adjacent QoT Constraint (AQC)* and *End-to-End QoT Constraint (EEQC)*.

1) Adjacent QoT Constraint (AQC): Adjacent QoT Constraint (AQC) is the constraint of QoT attributes (i.e., T, r and ρ) between any two adjacent participants in a social trust path. Let $Q^{\mu(AQC)}_{AM}$ ($\mu \in \{T,\ r,\ \rho\}$) denote the adjacent QoT constraint for the path between the source participant A and the target participant M. $Q^{\mu(AQC)}_{AM} > \lambda_\mu$ ($0 < \lambda_\mu < 1$) means that the value of QoT attribute μ between any two adjacent participants should be larger than λ_μ in a selected social trust path. In our model, a source participant can specify different AQCs for social trust path selection in different domains. For example, in *hiring employees*, A, a retailer manager specifies AQCs as $Q^{T(AQC)}_{AM} > 0.3$, $Q^{r(AQC)}_{AM} > 0.3$ and $Q^{\rho(AQC)}_{AM} > 0.8$. But when looking for new customers for *selling products*, A can specify $Q^{r(AQC)}_{AM} > 0.8$, if he/she believes the social relationships between participants are more important.

2) End-to-End QoT Constraint (EEQC): In service invocations, service consumer can set multiple end-to-end constraints for the attributes of QoS to satisfy their requirements (e.g., cost, delay and availability) of services. Different requirements have different constraints (e.g., total cost<\$20, delay<5s and availability>70%). In our model, a source participant 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. Let $Q^{\mu(EEQC)}_{AM}$ ($\mu \in \{T,\ r,\ \rho\}$) denote the End-to-End QoT Constraint (EEQC) between the source participant A and the target participant M. $Q^{\mu(EEQC)}_{AM} > \lambda_\mu$ ($0 < \lambda_\mu < 1$) means the value of QoT attribute μ between A and M should be larger than λ in a selected social trust path. Similar with AQC, a source participant can also specify different EEQCs for social trust path selection in different domains. For example, in *hiring employees*, A can set EEQCs as $Q^{T(EEQC)}_{AM} > 0.3$, $Q^{r(EEQC)}_{AM} > 0.3$ and $Q^{\rho(EEQC)}_{AM} > 0.8$. But when looking for new customers for *selling products*, A can specify

$Q_{AM}^{r(EEQC)} > 0.8$, if he/she believes the social relationships between participants are more important.

C. QoT Attribute Aggregation

When specifying end-to-end QoT constraints of social trust path selection, we need to know the aggregated value of each QoT attribute in a social trust path.

1) Trust Aggregation: If there are n participants $a_1, ..., a_n$ in order in a social trust path (denoted as $p(a_1, ..., a_n)$), the aggregated trust value is calculated based on Eq (1). This strategy has been widely used in the literature as a feasible trust aggregation method [26].

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

Different from existing works [8, 15], this aggregated trust value will be combined with social intimacy degree and role impact factor to identify the optimal social trust path that yields the most trustworthy result of trust propagation,

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 [13]. In addition, 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 [23]. Namely, the attenuation of social intimacy degree is non-linear in social networks. The aggregated r value in path $p(a_1, ..., a_n)$ can be calculated by Eq.(2) whose function image is a hyperbolic curve, fitting the characteristic of social intimacy attenuation.

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

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

3) Role Impact Factor Aggregation: As the recommendation roles do not have the property of transitivity [1], in this paper, we average the RIF values of intermediate recommending participants in a social trust path $p(a_1, ..., a_n)$ as the aggregated value based on Eq. (3).

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

D. Utility Function

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

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

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 adjacent and end-to-end QoT constraints, and yields the best utility with the weights specified by the source participant.

V. Social Trust Path Selection Algorithm

The optimal social trust path selection with multiple adjacent and end-to-end QoT constraints can be modelled as the classical Multi-Constrained Optimal Path (MCOP) selection problem which is an NP-Complete problem [10]. In this section, we first analyze some existing approximation algorithms for the MCOP selection problem and then propose an efficient heuristic algorithm, H_MQCSTP, for Multiple QoT Constrained Social Trust Path selection.

A. Existing Approximation Algorithms for MCOP

In the literature, several algorithms have been proposed to solve the MCOP selection problem.

Korkmaz *et al*. propose a heuristic algorithm, H_MCOP [10]. In this algorithm, both multi-constraint values and QoS attribute values are aggregated based on Eq. (5).

$$g_\lambda(p) \triangleq (\frac{q_1(p)}{Q_1})^\lambda + (\frac{q_2(p)}{Q_2})^\lambda + ... + (\frac{q_m(p)}{Q_m})^\lambda \tag{5}$$

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 adopts Dijkstra's shortest path algorithm [4] 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 $g_\lambda(p) > m$, indicating 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 further adopts 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 consequently 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 [10]. Consequently, based on it, in the field of Service-Oriented Computing (SOC), Yu *et al.* [30] propose an approximation algorithm, 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 selected from two adjacent service sets have a link with each other and thus all the paths from a source node to an intermediate node can be enumerated when necessary, avoiding an exhaustive searching. 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 [31, 32] adopt the integer linear programming method to solve the service selection problem with multi-QoS constraints. But in [30] they have been proved having low efficiency in finding a near-optimal solution in large-scale networks.

B. H_MQCSTP

In this section, we propose an efficient heuristic algorithm H_MQCSTP which contains some novel heuristic search strategies for selecting the optimal social trust path with multiple

Table I
NOTATIONS USED IN PSEUDOCODE

Notations	Representation
$Dist(v).r$ and $Dist(v).\rho$	The aggregated QoT attribute values of the identified social trust path from v_t to v in the *Backward_Search* procedure.
$Dist(v).\delta$	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.
$M(v_x, v_y).r$, $M(v_x, v_y).\rho$	The SID between v_x and v_y, and the RIF 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 *Forward_Search* procedure, e.g., $pre_x(v'') = v'$ represents in the shortest path from v_t to v'', v' is the preceding node of v''.
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.
$v.r$ and $v.\rho$	The aggregated QoT attribute values of the identified social trust path from v_s to v in the *Forward_Search* procedure.

adjacent and *end-to-end* QoT constraints in complex social networks.

In H_MQCSTP, 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 *potential solution* which satisfies the EEQCs in the sub-network between v_s and v_t, and record the aggregated QoT attributes (i.e., r and ρ) of the identified path from v_t to each intermediate node v. If a potential solution exists, we then adopt the *Forward_Search* procedure to search the sub-network from v_s to v_t to deliver a near-optimal solution.

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

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

Backward_Search: In the backward search from v_t to v_s, H_MQCSTP identifies the path p_s from v_t to v_s with the minimal δ based on the Dijkstra's shortest path algorithm [4]. 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 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 potential solution that satisfies the EEQCs 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 potential 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 potential solution that satisfies the end-to-end QoT constraints. Then, $\delta(p_s) \leq \delta(p_*)$. Assume p_s is not a potential solution, then $\exists \varphi \in \{r, \rho\}$ that $\varphi_{p_s} < Q_{v_s,v_t}^{\varphi(EEQC)}$. Hence, $\delta(p_s) > 1$. Since p_* is a potential 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 potential solution. □

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

Forward_Search: If there exists a potential solution in the sub-network, a heuristic forward search is executed from v_s to v_t. This process adopts the information provided by the above *Backward_Search* to identify whether there is another path p_t which satisfies both AQCs and EEQCs. In this procedure, H_MQCSTP first searches the path with the maximal $\mathcal{F}$ value from v_s. Assume node $v_m \in \{neighboring\ nodes\ of\ v_s\}$ is selected based on the Dijkstra's shortest path algorithm. H_MQCSTP 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 $r_{p_{fm}} = (r_{p_m} * r_{p'_m})/h^{\alpha}$ ($\alpha \geq 1$ is the argument for controlling the attenuation speed of r) and $\rho_{p_{fm}} = (\rho_{p_m} + \rho_{p'_m})/(h-1)$. According to whether p_{fm} is feasible, H_MQCSTP adopts the following searching strategies.

Situation 1: If each QoT attribute between v_s and v_m, and each aggregated QoT attribute of p_{fm} can satisfy the corresponding AQC and EEQC, then H_MQCSTP 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 QoT attribute between between v_s and v_m, or any aggregated QoT attribute of p_{fm} does not satisfy the corresponding AQC or EEQC, then H_MQCSTP does not search the path from v_m and the link $v_s \rightarrow v_m$ is deleted from the sub-network. Subsequently, H_MQCSTP 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 if the social trust path p_s identified by the *Backward_Search* procedure is a feasible solution which satisfies both ACQs and EEQCs, the social trust path p_t identified by the *Forward_Search* procedure can not be worse than p_s. Namely, $\mathcal{F}(p_t) \geq \mathcal{F}(p_s)$.

Theorem 2: With a social trust path p_s identified by the *Backward_Search* procedure and a social trust path p_t identified by the *Forward_Search* procedure in H_MQCSTP, 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 $v_s, v_1, ..., v_n, v_t$. In the *Forward_Search* procedure, H_MQCSTP 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 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., $v_1, ..., v_n$) 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. □

The process of H_MQCSTP 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 $Dist(v_x).\delta = \infty$ and $Dist(v_t).\delta = 0$. 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 out 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 $v_b \in \{neighboring\ nodes\ of\ v_a\}$, calculate δ value of the identified social trust path form 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

Algorithm 1: H_MQCSTP

Data: M, $Q^{T(EEQC)}_{v_s,v_t}$, $Q^{r(EEQC)}_{v_s,v_t}$, $Q^{\rho(EEQC)}_{v_s,v_t}$, v_s, v_t
Result: p_t, $\mathcal{F}(p_t)$
1 **begin**
2 $\quad p_s = \emptyset$, $p_t = \emptyset$
3 $\quad$ **Backward_Search** $(M, Q^{T(EEQC)}_{v_s,v_t}, Q^{r(EEQC)}_{v_s,v_t}, Q^{\rho(EEQC)}_{v_s,v_t}, v_s, v_t)$
4 $\quad$ **if** $\delta(p_s) > 1$ **then**
5 $\quad\quad$ **Return** no feasible solution
6 $\quad$ **else**
7 $\quad\quad$ **Forward_Search** $(M, Dist(v).T, Dist(v).r, Dist(v).\rho, Q^{T(EEQC)}_{v_s,v_t}, Q^{r(EEQC)}_{v_s,v_t}, Q^{\rho(EEQC)}_{v_s,v_t}, Q^{T(AQC)}_{v_s,v_t}, Q^{r(AQC)}_{v_s,v_t}, Q^{\rho(AQC)}_{v_s,v_t}, v_s, v_t)$
8 $\quad\quad$ **Return** p_t and $\mathcal{F}(p_t)$
9 **end**

Algorithm 2: Backward_Search

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

Algorithm 3: Forward_Search

Data: M, $Dist(v).T$, $Dist(v).r$, $Dist(v).\rho$, $Q^T_{v_s,v_t}$, $Q^r_{v_s,v_t}$, $Q^{\rho}_{v_s,v_t}$, v_s, v_t
Result: p_t, $\mathcal{F}(p_t)$
1 **begin**
2 $\quad$ **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 $\quad$ **Add** v_s into S_y
4 $\quad$ **while** $S_y \neq \emptyset$ **do**
5 $\quad\quad v_i.\mathcal{F}' = min(v_i^*.\mathcal{F}') \ (v_i^* \in S_y)$
6 $\quad\quad$ **for** *each* $v_j \in adj[v_i]$ **do**
7 $\quad\quad\quad h'$ is the number of the hops of the foreseen path from v_s to v_t via v_j
8 $\quad\quad\quad temp_T = v_i.T * M(v_i, v_j).T * Dist(v_j).T$
$\quad\quad\quad temp_r = v_i.r * M(v_i, v_j).r * Dist(v_j).r$
$\quad\quad\quad temp_{\rho} = v_i.\rho + M(v_i, v_j).\rho + Dist(v_j).\rho$
9 $\quad\quad\quad$ **if** $(temp_T > Q^{T(EEQC)}_{v_s,v_t})$ *And* $(temp_r/h'^{\alpha} > Q^{r(EEQC)}_{v_s,v_t})$ *And* $(temp_{\rho}/(h' - 1) > Q^{\rho(EEQC)}_{v_s,v_t})$ *And* $M(v_i, v_j).T > Q^{T(AQC)}_{v_s,v_t}$ *And* $M(v_i, v_j).r > Q^{r(AQC)}_{v_s,v_t}$ *And* $M(v_i, v_j).T > Q^{\rho(AQC)}_{v_s,v_t}$ **then**
10 $\quad\quad\quad\quad$ **if** $v_j \notin S_y$ **then**
11 $\quad\quad\quad\quad\quad$ Put v_j into S_y
12 $\quad\quad\quad\quad\quad pre_y(v_j) = v_i$
13 $\quad\quad\quad\quad$ **else if** $\mathcal{F}'(p_j) < v_j.\mathcal{F}'$ **then**
14 $\quad\quad\quad\quad\quad v_j.\mathcal{F}' = \mathcal{F}'(p_j)$
15 $\quad\quad\quad\quad\quad v_j.T = v_i.T * M(v_i, v_j).T$
16 $\quad\quad\quad\quad\quad v_j.r = v_i.r * M(v_i, v_j).r$
17 $\quad\quad\quad\quad\quad v_j.\rho = v_i.\rho + M(v_i, v_j).\rho$
18 $\quad\quad\quad\quad\quad$ Put v_j into S_y
19 $\quad\quad\quad\quad\quad pre_y(v_j) = v_i$
20 $\quad\quad$ **Remove** v_i from S_y
21 $\quad p_t \leftarrow Pre_y(v_t)$ to $Pre_y(v_s)$
22 $\quad$ **Return** p_t and $\mathcal{F}(p_t))$
23 **end**

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 18 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 19 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 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 $v_j \in \{neighboring\ nodes\ of\ v_i\}$, 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 r_{p_j} and ρ_{p_j}. If each QoT attribute between v_s and v_j, and each aggregated QoT attribute can satisfy the corresponding AQC and EEQC, then replace the stored $1/\mathcal{F}(p_j)$ with the current $1/\mathcal{F}(p_j)$ at v_j and set $pre_y(v_j) = v_i$. Otherwise, set $M(v_i, v_j).r = 0$ 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 $pre_y(v_t)$ (*lines 8 to 9 in Algorithm 1 and lines 20 to 23 in Algorithm 3*).

H_MQCSTP consumes twice the execution time of Dijkstra's shortest path algorithm [4]. The time complexity of H_MQCSTP is $O(NlogN + E)$, 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_MQCSTP 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 VI-B*).

VI. Experiments

A. Experiment Settings

The *Enron* email dataset[4] has 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 [18, 21, 29]. In addition, as we explained in *section III-3* 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 [21]. Therefore, in contrast to other real social network datasets (e.g., Epinions[5] and FilmTrust[6]), the *Enron* email dataset fits complex social network structure better. Thus, to validate our proposed algorithm, we select the *Enron* email corpus[4] with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) as the dataset for our experiments.

As we analyzed in *section V-A*, H_MCOP is the most promising algorithm for the MCOP selection. Based on it, several approximation algorithms [14, 30] 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_MQCSTP, we compare it with H_MCOP [10] in both execution time and the utilities of identified social trust paths (*see section VI-B*). 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 $Q^{(EEQC)} = \{Q^{T(EEQC)} > 0.05,\ Q^{r(EEQC)} > 0.001,\ Q^{\rho(EEQC)} > 0.3\}$ and the adjacent QoT constraints are set as $Q^{(AQC)} = \{Q^{T(AQC)} > 0.1,\ Q^{r(AQC)} >$

[5]http://epinions.com/
[6]http://trust.mindswap.org/filmtrust/

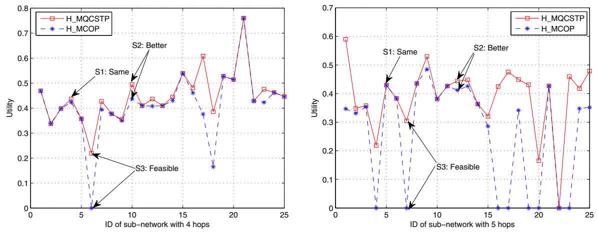

Figure 3. Path utility of sub-networks with 4 and 5 hops

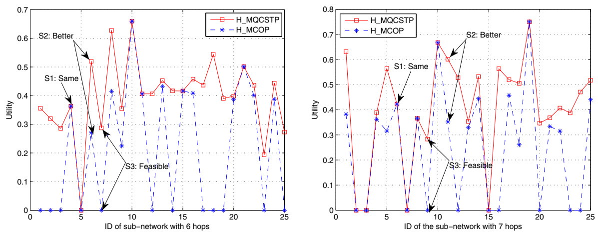

Figure 4. Path utility of sub-networks with 6 and 7 hops

$0.05,\ Q^{\rho(AQC)} > 0.1\}$. 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 $\omega_\rho = 0.5$.

Both H_MQCSTP 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.

B. Performance in Social Trust Path Selection

Table II
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

In this experiment, 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 dataset[4]. 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 II lists the properties of the simplest and the most complex 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). With each sub-network, we repeat the experiment 5 times for each of H_MQCSTP and H_MCOP. The results are plotted from Fig. 3 to Fig. 6 where the execution time of each of H_MQCSTP and H_MCOP is averaged based on the 5 independent runs.

Results (Utility). From Fig. 3 to Fig. 4, we observe that in any case, our H_MQCSTP does not yield any utility worse than that of H_MCOP (e.g., $S1$ in Fig. 3 to Fig. 4) while in most sub-networks (i.e., 61% of total sub-networks), the utilities of social trust paths identified by H_MQCSTP are better than those of H_MCOP (e.g., $S2$ in Fig. 3 to Fig. 4). The sum of

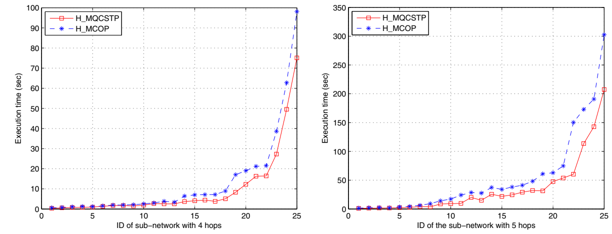

Figure 5. Execution time of sub-networks with 4 and 5 hops

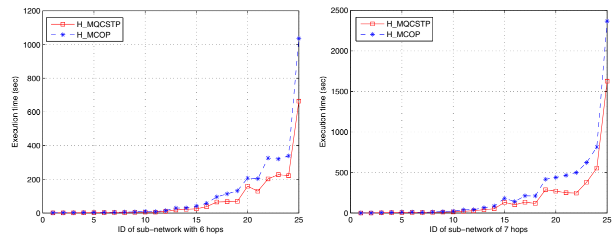

Figure 6. Execution time of sub-networks with 6 and 7 hops

utilities computed by H_MQCSTP and H_MCOP in the sub-networks with each group of hops is listed in Table III. From Table III, we can see that the sum of utilities of our proposed heuristic algorithm is 10.87% more than that of H_MCOP in 4 hops sub-networks, 14.88% more in 5 hops, 18.87% more in 6 hops and 16.42% more in 7 hops.

Analysis (Utility). From the above results, we can see that H_MQCSTP 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_MQCSTP 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 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 to Fig. 4). In contrast, as illustrated by *Theorem 1*, H_MQCSTP can identify a feasible solution if it exists (e.g., $S3$ in Fig. 3 to Fig. 4). In addition, as illustrated by *Theorem 2*, H_MQCSTP can identify a near-optimal social trust path satisfying both AQCs and EEQCs if it exists. Therefore, in this case, the quality of the social trust path identified by H_MQCSTP is better than H_MCOP.

Results (Execution Time). From Fig. 5 to Fig. 6, we observe that the execution time of H_MQCSTP is less than that of H_MCOP in all sub-networks. The total execution time of each of H_MQCSTP and H_MCOP in each group of hops is listed in Table III. From Table III, we can see that the total execution time of our proposed heuristic algorithm is only 72.22% of that of H_MCOP in 4 hops sub-networks, 64.24% in 5 hops, 65.04% in 6 hops and 64.19% in 7 hops.

Analysis (Execution Time). From the above results, we can see that H_MQCSTP is much more efficient than H_MCOP. The reasons are twofold. Firstly in the *Forward_Search* procedure, H_MQCSTP does not calculate g_λ $(\lambda > 1)$ which consumes a large amount of execution time when $\lambda \to \infty$ [10]. Secondly, in the searching process, when any QoT attribute between any adjacent participants in a selected path from v_s to v_y $(v_y \neq v_t)$, or aggregated QoT attribute of that path does

Table III
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_MQCSTP	11.2014	9.7113	9.9469	10.1747	245.8564	871.8128	1.9528e+003	4.3005e+003
H_MCOP	10.3047	6.5274	6.6006	6.1979	340.4162	1.3571e+003	3.0024e+003	6.6996e+003
difference	10.87% more	14.88% more	18.87% more	16.42% more	27.78% less	35.76% less	34.96% less	35.81% less

not satisfy the corresponding AQC or EEQC, 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_MQCSTP is superior to H_MCOP in both the execution time and the quality of selected social trust path.

VII. CONCLUSIONS

In this paper, we have presented a complex social network structure that takes trust, social relationship and recommendation roles into account, reflecting the real-world situations better. In addition, we proposed a multiple QoT constrained social trust path selection model in complex social networks. Furthermore, for selecting the optimal social trust path with both AQCs and EEQCs, which is an NP-Complete problem, we have also proposed H_MQCSTP, an efficient heuristic algorithm. The results of experiments conducted on a real dataset of social networks demonstrate that H_MQCSTP significantly outperforms existing methods in both execution time and optimal social trust path selection.

In our future work, we plan to incorporate our models and algorithms in a new generation of social network based recommendation systems. In this system, our proposed method will be applied, for instance, to help a customer identify the most trustworthy one from all sellers selling the product preferred by the customer.

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] S. Baase and A. Gelder. *Computer Algorithms Introduction to Design and Analysis*. Addision Wesley, 1999.

[3] J. Bi, J. Wu, and W. Zhang. A trust and reputation based anti-spim method. In *INFOCOM'08*, pages 2485–2493.

[4] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.

[5] L. Franken. Quality of service management: A model-based approach. *PhD Thesis, Centre for Telematics and Information Technology*, 1996.

[6] J. Golbeck and J. Hendler. Inferring trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, 6(4):497–529, 2006.

[7] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW'04*, pages 403–412.

[8] C. Hang, Y. Wang, and M. Singh. Operators for propagating trust and their evaluation in social networks. In *AAMAS'09*, pages 1025–1032.

[9] M. Jamali and M. Ester. Trustwalker: A random walk model for combinging trust-based and item-based recommendation. In *KDD'09*, pages 29–42.

[10] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *INFOCOM'01*, pages 834–843.

[11] J. Kunegis, A. Lommatzsch, and C. Bauckhang. The slashdot zoo: Mining a social network with negative edges. In *WWW'09*, pages 741–750.

[12] 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.

[13] G. Levinger. Development and change. *Close Relationships*, pages 315–359, 1983.

[14] L. Li, J. Wei, and T. Huang. High performance approach for multi-qos constrained web services selection. In *ICSOC'07*, pages 283–295.

[15] 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.

[16] G. Liu, Y. Wang, and L. Li. Trust management in three generations of web-based social networks. In *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pages 446–451, 2009.

[17] G. Liu, Y. Wang, M. Orgun, and E.-P. Lim. A heuristic algorithm for trust-oriented service provider selection in complex social networks. In *SCC*, pages 130–137, 2010.

[18] G. Liu, Y. Wang, and M. A. Orgun. Optimal social trust path selection in complex social networks. In *AAAI'10*.

[19] G. Liu, Y. Wang, and M. A. Orgun. Quality of trust for social trust path selection in complex social networks. In *AAMAS'10*, pages 1575–1576.

[20] G. Liu, Y. Wang, and M. A. Orgun. Finding k optimal social trust paths for the selection of trustworthy service providers in complex social networks. In *ICWS'11*, pages 41–48, 2011.

[21] 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.

[22] S. Milgram. The small world problem. *Psychology Today*, 1967.

[23] R. Miller, D. Perlman, and S. Brehm. *Intimate Relationships*. McGraw-Hill College, 4th edition, 2007.

[24] 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.

[25] J. Tang, J. Zhang, L. Yan, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[26] 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.

[27] Y. Wang and V. Varadharajan. Role-based recommendation and trust evaluation. In *IEEE EEE'07*, pages 278–295.

[28] S. Yang, J. Zhang, and I. Chen. Web 2.0 services for identifying communities of practice. In *SCC'07*, pages 130–137.

[29] S. Yoo, Y. Yang, F. Lin, and I. Moon. Mining social networks for personalized email prioritization. In *KDD'09*, pages 967–976.

[30] 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.

[31] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Sheng. Quality driven web services composition. In *WWW'03*, pages 411–421.

[32] 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.