Trust Management in Online Social Networks

by

Guanfeng Liu

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy
in the
Department of Computing
Faculty of Science
Macquarie University

Supervisor: A/Prof. Yan Wang
Associate Supervisor: Prof. Mehmet A. Orgun

2013

Statement of Candidate

I certify that the work in this thesis entitled **"Trust Management in Online Social Networks"** has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree to any other university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged.

In addition, I certify that all information sources and literature used are indicated in the thesis.

Guanfeng Liu

17 June 2013

To my parents,

Zhenhua Liu and Yalin Sun,

who dedicated all their life to me

Abstract

Online Social Networks (OSNs) have attracted many participants, and they have been used as a means for a rich variety of activities, such as movie recommendations and product recommendations. In these activities, trust is one of the most important factors for participants' decision-making. Therefore, it is necessary and significant to evaluate the trust between two participants who have no direct interaction. This thesis aims to provide effective and efficient trust management methods to compute reasonable trust evaluation results, which can be divided into the following three contributions.

The first contribution of the work is to study trust-oriented social network structures and solve the trust network extraction problem. To reflect the social networks in the real world, we propose a complex trust-oriented social network structure, which contains social contextual information that has significant influence on trust evaluation. In addition, the trust network from a truster to a trustee without direct interactions is extracted prior to performing trust evaluation. To extract a trust network that can deliver trustworthy trust evaluation results, which is an NP-Complete problem, we propose an approximation algorithm, called SCAN, and two new heuristic algorithms, called SCAN-K and H-SCAN-K.

The second contribution of the work is to address the optimal social trust path selection problem. To deal with the NP-Complete optimal social trust path selection problem with multiple constraints, a novel approximation algorithm, called MONTE_K, and two novel heuristic algorithms, called H_OSTP and MFPB-HOSTP, have been proposed. In addition, we propose a heuristic algorithm, called H-OSTP-K, for K optimal social trust paths selection.

The third contribution of the work is to study trust transitivity in OSNs. In order to compute a reasonable propagated trust value along a social trust path, a general

concept, called Quality of Trust Transitivity (QoTT) and a novel Multiple QoTT Constrained Trust Transitivity (MQCTT) model have been proposed.

For the proposed approaches, extensive experiments have been conducted on real datasets or real scenarios. The experimental resulats have demonstrated the proposed methods are superior to existing approaches in terms of the utility of delivered results and efficiency.

Acknowledgments

First of all, I would like to express my sincere appreciation to my supervisor A/Prof. Yan Wang and my associate supervisor Prof. Mehmet A. Orgun for their kindness and patience. They have led me in the correct direction over the past few years not only with their knowledge and experience, but also with thoughtfulness about a young man's personal growth. Literally, without their continuous support and endless guidance, this work would not have been possible. It is my great fortune to have them as my supervisors at Macquarie University.

I also would like to thank Prof. Ee-Peng Lim for his help in various stages of my work. This academic journey would not have been so rewarding without his kindness and wisdom.

In addition, my colleagues have helped me to develop this work. I wish to express my thanks to Lei Li, Joe Zou and Haibing Zhang for providing a friendly and enjoyable environment during these years.

Many thanks to the staff in the Department of Computing for their administrative help. I would like to thank Sylvian Chow, Melina Chan, Donna Hua and Jackie Walsh for their warm support and help.

Most important of all, I would like to thank my family. My parents, Zhenhua Liu and Yalin Sun, have always been there for me. Their love, support and encouragement have been the foundation for my life. I wish to thank them for all the opportunities they have made available to me, and for the support they have given me during my life. Thanks to my wife, Tong Tong, for her love, understanding and support. Without their love, unwavering support and inspiration, this work could never have been accomplished.

Publications

[1] **Guanfeng Liu**, Yan Wang, Mehmet A. Orgun and Ee-Peng Lim, Finding the Optimal Social Trust Path for the Selection of Trustworthy Service Providers in Complex Social Networks, IEEE Transactions on Services Computing (TSC) (regarded as the best journal in the field of services computing), Vol. 6, No. 2, 2013, pp. 152-167.

[2] **Guanfeng Liu** and Yan Wang, Trust-Oriented Service Provider Selection in Complex Online Social Networks, A. Bouguettaya et.al (Ed.), Handbook on Web Services, Springer, accepted on 03 Sep., 2012.

[3] **Guanfeng Liu**, Yan Wang and Mehmet A. Orgun, Social Context-Aware Trust Network Discovery in Complex Contextual Social Networks, Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12) (**ERA rank A conference**[1]), 22-26 August, 2012, Toronto, Canada, pp. 101-107.

[4] **Guanfeng Liu**, Yan Wang, Mehmet A. Orgun and Huan Liu, Discovering Trust Networks for the Selection of Trustworthy Service Providers in Complex Contextual Social Networks, IEEE The 9th International Conference on Web Services (IEEE ICWS 2012) (**ERA rank A conference**), 24-29 June, 2012, Honolulu, Hawii, USA, pp. 384-391.

[5] **Guanfeng Liu**, Yan Wang and Duncan S. Wong, Multiple QoT Constrained Social Trust Path Selection in Complex Social Networks, The 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom-2012) (**ERA rank A conference**), 25-27 June, 2012, Liverpool, UK, pp. 624-631.

[1] refer to http://core.edu.au/index.php/categories/conference rankings

[6] **Guanfeng Liu**, Yan Wang and Mehmet A. Orgun, Trust Transitivity in Complex Social Networks, Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-11) (**ERA rank A conference**), 7-11 August, 2011, San Francisco, California, USA, pp. 1222-1229.

[7] **Guanfeng Liu**, Yan Wang and Mehmet A. Orgun, Finding K Optimal Social Trust Paths for the Selection of Trustworthy Service Providers in Complex Social Networks, The 9th International Conference on Web Services (IEEE ICWS 2011) (**ERA rank A conference**), 4-9 July, 2011, Washington DC, USA, pp. 41-48.

[8] **Guanfeng Liu**, Yan Wang and Mehmet A. Orgun, Optimal Social Trust Path Selection in Complex Social Networks, The Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010) (**ERA rank A conference**), 11-15 July, 2010, Atlanta, Georgia, USA, pp. 1391-1398.

[9] **Guanfeng Liu**, Yan Wang and Mehmet A. Orgun, Quality of Trust for Social Trust Path Selection in Complex Social Networks, The 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS2010) (**ERA rank A conference**), 10-14, May 2010, Toronto, Canada, pp. 1575-1576.

[10] **Guanfeng Liu**, Yan Wang, Mehmet A. Orgun and Ee-Peng Lim, A Heuristic Algorithm for Trust-Oriented Service Provider Selection in Complex Social Networks, IEEE The 7th International Conference on Services Computing (IEEE SCC2010) (**ERA rank A conference**, **the Best Paper Award**), 5-10, July 2010, Miami, Florida, USA, pp. 130-137.

[11] **Guanfeng Liu**, Yan Wang and Mehmet A. Orgun, Trust Inference in Complex Trust-oriented Social Networks, 2009 International Conference on Computational Science and Engineering, 29-31, August, 2009, Vancouver, Canada, pp. 996-1001.

[12] **Guanfeng Liu**, Yan Wang and Lei Li, Trust Management in Three Generations of Web-Based Social Networks, International Workshop on Cyber Physical and Social Computing (CPSC09), in conjunction with ATC-09, 7-9 July, 2009, Brisbane, Australia, pp. 446-451.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The concept of social networks emerged in late 1800s. Ferdinand [35] and Emile [36] proposed the idea of social networks in their theories and research of social groups in 1887 and 1893 respectively. Major developments in the research of social networks occurred in the 1930s by research scientists in the disciplines of Psychology, Anthropology, and Mathematics [118, 119]. For example, in the discipline of Psychology, Moreno [119] systematically recorded and analysed the social interactions between people in small groups, especially classrooms and work groups. In addition, in the discipline of Sociology, Parsons [108] studied the social structure by analysing the social relationships between people. Furthermore, based on Parsons' theory, the work of sociologist Blau [13] provides a strong impetus for analysing the relational ties of social units with his work on social exchange theory.

With the development of Internet and Web technology, Online Social Networks (OSNs), such as Facebook (facebook.com) and Twitter (twitter.com), have attracted many participants. According to statistics provided by *The eBusiness Knowledgebase* (www.ebizmba.com, a Web statistic company) on 05 December 2012, the top 10 popular OSNs and the approximate number of the monthly unique visitors to them are listed in Table 1.1. From the table, we can see that, for the most popular OSN, Facebook, there are approximately 750,000,000 unique visitors visiting the Website in a month. In recent years, social networking sites have been used as a means for a variety of activities. For example, according to a survey of 2600 hiring managers in 2009 by CareerBuilder (careerbuilder.com, a popular job hunting website), 45% of those managers

used social networking sites to investigate potential employees. The ratio increased to 72% in January 2010. In addition, at FilmTrust (trust.mindswap.org/FilmTrust/), an OSN for movie recommendations, participants can rate movies and make movie recommendations. Furthermore, by connecting with OSNs (e.g., Facebook and Twitter) at some e-commerce websites like ThisNext (thisnext.com) and eBay (ebay.com), a buyer can recommend the products available on these e-commerce websites to his/her friends who participate in the OSNs. In these activities, trust is one of the most important factors for participants' decision making. However, most of participants do not have previous direct interactions, and thus require approaches and mechanisms for evaluating the trustworthiness between participants who are unknown to each other.

Table 1.1: Top 10 popular OSNs

Ranking	Name	URL	Unique Monthly Visitors
#1	Facebook	facebook.com	750,000,000
#2	Twitter	twitter.com	250,000,000
#3	Linkedin	linkedin.com	110,000,000
#4	Myspace	myspace.com	70,500,000
#5	Google Plus+	plus.google.com	65,000,000
#6	DeviantArt	deviantart.com	25,500,000
#7	LiveJournal	livejournal.com	20,500,000
#8	Tagged	tagged.com	19,500,000
#9	Orkut	orkut.com	17,500,000
#10	Pinterest	pinterest.com	15,500,000

An Online Social Network (OSN) can be represented as a graph, where each node represents a participant and each link between two nodes corresponds to a real-world or online interaction (e.g., $A \rightarrow B$ and $B \rightarrow C$ in Fig. 1.1). For adjacent participants (i.e., those nodes with a directed link between them), the trust value between them could be explicitly given by one to another based on their direct interaction (e.g., at FilmTrust, a participant can recommend movies to his/her friends, and the corresponding recommendee can give a trust rating (i.e., 1 to 10) to the recommender based on the quality of the movie recommendation(s)). In OSNs, as each participant usually interacts with many others, multiple trust paths may exist between nonadjacent participants

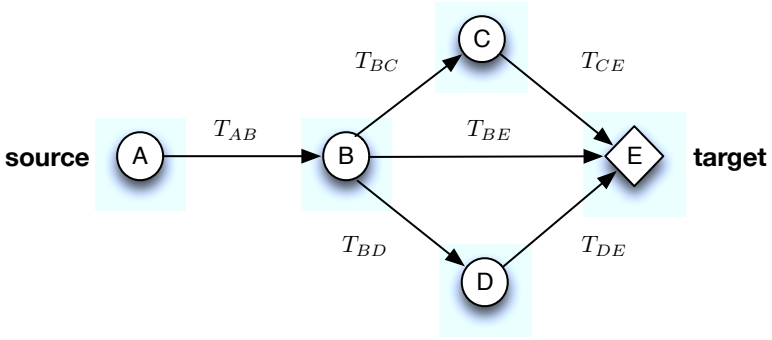

Figure 1.1: A trust-oriented social network

from the source participant (e.g., A) to the target participant (e.g., E) (For example, path $A \rightarrow B \rightarrow C \rightarrow E$ and $A \rightarrow B \rightarrow D \rightarrow E$ in Fig. 1.1). If there exists at least one social trust path linking two unknown participants (e.g., A and E are linked by three social trust paths), there exists a *social connection* between them. All such social trust paths form a *trust network* from a source to a target (e.g., the trust network from A to E in Fig. 1.1).

This thesis will focus on the three significant challenging problems of trust network extraction, trust path selection and trust transitivity in the trust management of OSNs.

1.1 Challenges in the Trust Management of OSNs

1.1.1 Trust Network Extraction

In the social network depicted in Fig. 1.1, suppose A is looking for a badminton coach and E is a badminton coach. In such a situation, as indicated in the theory of Social Psychology [23, 91] and Computer Science [48, 78], A can evaluate the trustworthiness of E based on the trust network from A to E by using trust transitivity methods (e.g., A trusts B, and B trusts C, then A can trusts C to some extent. This is also true in a long path from a source to a target) [48, 62]. Therefore, in OSNs, such a trust network is fundamental and critical for trust evaluation between two nonadjacent participants, as it contains some important intermediate participants, the trust relations

between those participants and the social context. All of them have important influences on the trust evaluation between two unknown participants in OSNs.

In the literature, there have been several existing trust evaluation approaches for trust evaluation between two nonadjacent participants [48, 83, 85]. However, they all assume that the trust network between the two nonadjacent participants has been identified. Therefore, given any two participants who have no direct interactions in a large-scale social network, extracting the trust network between them becomes a fundamental and essential step before performing any trust propagation methods. Such a task is called *trust network extraction*.

In the graph formed by social interactions, cycles (e.g., path $A \rightarrow B \rightarrow C \rightarrow A$) usually exist due to the complex interactions among multiple participants [103]. Extracting a sub network from a cyclic network has been proved to be an NP-Complete problem [6]. Therefore, it is computationally infeasible to extract all the social trust paths from a source and a target in a large-scale trust network. So, it is a significant challenge to extract a trust network with *higher quality* that contains more important intermediate nodes and social contexts and using such a trust network, one can deliver more reasonable trust evaluation results with high efficiency.

In the literature, some resource discovery methods, developed for peer-to-peer (P2P) networks, can be used for trust network extraction in OSNs. Those methods include the Time To Live Breadth First Search (TTL-BFS) method [19, 37], the Random Walk Search (RWS) method [45, 46] and the High Degree Search (HDS) method [2]. But these methods do not consider the social context in social networks, including *social relationships* between participants (e.g., the one between an employer and an employee), the *social position* (e.g., a professor in service computing research), the *preference* (e.g., like to play badminton) and the residential location (e.g., living in Sydney, Australia) of participants. As indicated in Social Psychology [16, 75, 135], all the above social contextual information has significant influence on both social interactions and trust evaluation, and they can be discovered by using data mining techniques [96, 123]. In addition, a source may have different purposes and needs when evalu-

ating the trustworthiness of a target, e.g., looking for a potential employee or looking for a movie recommendation. Thus, in order to obtain a more reasonable and trustworthy trust evaluation result, a source participant may specify some constraints on social contexts as his/her trust evaluation criteria in trust network extraction. However, this feature has not been supported in any of the existing methods.

1.1.2 Social Trust Path Selection

In an extracted large-scale trust-oriented social network, there could be tens of thousands of social trust paths between a source participant and the target one [70]. Evaluating the trustworthiness of the target participant based on all these social trust paths can incur huge computational time. Alternatively, we can search an optimal path yielding the most trustworthy trust propagation result from multiple paths. This is called the optimal social trust path selection problem that is known to be a challenging research problem in OSNs [78].

In the literature, Lin *et al.* [77] proposed an optimal social path selection method. In their model, the shortest path between the source participant and the target one is selected as the optimal one. This method neglects *trust information* between participants. In another work [53], the path with the maximal propagated trust value is selected as the most trustworthy social trust path. However, this work does not consider the social contextual information introduced in the above *Section 1.1.1*, which has significant influence on trust propagation [3, 102]. In addition, a source participant may have different social trust path selection criteria (e.g., with more focus on the recommendation roles of participants in employment and/or with more focus on the social relationships between participants in making friends) and should be able to set certain constraints on social context in trust propagation. This can help the source participant select the optimal social trust path that yields the most trustworthy trust propagation result. However, such a capability is not supported by any existing method.

1.1.3 Trust Transitivity

After extracting the trust network and selecting the trustworthy social trust paths from the trust network, the computation of the value of trust for the target participant requires an understanding of how trust is propagated along a social trust path, which is a critical and challenging problem in OSNs [48, 51]. In the literature, several trust transitivity models have been proposed [48, 50, 51, 113, 124], but they have the following drawbacks.

Firstly, similar to the trust evaluation and trust path selection methods discussed in Sections 1.1.1 and 1.1.2, these existing trust transitivity models do not fully consider those important social contextual information, i.e., social relationship, social position and preference that have significant influence on trust transitivity [3, 76, 102]. Secondly, although different trust evaluation criteria can influence trust transitivity results, the specification of such criteria is not supported by any existing method either. Finally, trust transitivity formalised in existing models does not follow the nature of trust decay illustrated in social psychology, namely, trust decays slowly in a certain number of early hops (specified by a source participant) from a source participant, and then decays fast until the trust value approaches the minimum [44, 61].

1.2 Contributions of the Work

Extracting the trust network between a source and a target is the first step for the trust management in OSNs, as it is fundamental to performing any trust path selection and trust evaluation methods. Based on the solution of trust network extraction, to effectively and efficiently evaluate the trust between two unknown participants, we need to select those trustworthy social trust paths and perform trust transitivity computation to deliver reasonable trust values.

In order to address the above significant and challenging problems in the trust management of OSNs, this thesis makes three major contributions.

1. The first contribution of this thesis is trust network extraction in large-scale trust-oriented social networks.

   (a) A novel complex trust-oriented contextual social network structure is proposed. This new structure contains complex social contextual information, including social relationships, social positions, preferences, residential locations. It can reflect the social networks in the real world better because the above mentioned important social contextual information in the human society is modeled in the new structure.

   (b) We propose a new general concept, called QoTN (Quality of Trust Network) which illustrates the trustworthiness of an extracted trust network. Then we propose a social context-aware trust network extraction method with QoTN constraints.

   (c) To address the NP-Complete trust network extraction problem, we propose an approximation algorithm, called SCAN, by adopting the Monte Carlo method [43] and several optimization strategies. In addition, we propose two heuristic algorithms, called H-SCAN and H-SCAN-K based on the K-Best-First Search (KBFS) method [34], bidirectional search (i.e., search from both the source node and the target node simultaneously) [57] and our proposed optimization and heuristic search strategies.

   Experiments conducted on real social network datasets illustrate that on average, our methods can extract trust networks with higher quality and consumes less execution time than the existing methods.

2. The second contribution of this thesis is social trust path selection.

   (a) In service invocations, users 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%). Similar to QoS, we propose a novel concept Quality of Trust (QoT) that illustrates the trustworthiness of the identified social trust paths. In our model, source participants can also specify end-to-end QoT constraints to reflect their trust path selection criteria. Then the multiple QoT constrained optimal social trust path selection problem is modeled as the classical Multi-Constrained Optimal Path (MCOP) selection problem, which is proved to be NP-Complete in [67].

(b) To address the NP-Complete social trust path selection problem, we propose an efficient approximation algorithm, MONTE_K, based on the Monte Carlo method [43], and two heuristic algorithms, H_OSTP and MFPB-HOSTP based on the Dijkstra's shortest path algorithm [31] and our proposed novel search strategies for optimal social trust path selection.

In addition, people are willing to believe what they have been told most often by others [68]. Therefore, in order to obtain a more reasonable trust evaluation result of a target participant, a source participant may refer to multiple social trust paths. We propose a new efficient Heuristic algorithm for the K Optimal Social Trust Path selection, called H-OSTP-K to select the top K trustworthy social trust paths in a large-scale trust-oriented social network. Experiments conducted on real online social network datasets demonstrate the superior performance of our proposed algorithms over the existing ones.

3. The third contribution of this thesis is a new model of trust transitivity in trust-oriented contextual OSNs.

(a) We propose a general concept, Quality of Trust Transitivity (QoTT), to illustrate the ability of a social trust path to guarantee a certain level of quality in trust transitivity.

(b) Based on the properties of trust illustrated in social psychology, we then propose a new Multiple QoTT Constrained Trust Transitivity (MQCTT) model.

Experiments conducted on real social network datasets demonstrate that the proposed trust transitivity model follows both the principles in social psychology and the properties of trust, and thus it obtains more reasonable trust values than existing methods.

1.3 Roadmap of the Thesis

This thesis is structured as follows.

Chapter 2 provides a comprehensive literature review of trust, social network properties and trust management in OSNs.

Chapter 3 proposes a complex trust-oriented social network structure that contains more social information that has significant influence on trust management. This structure can reflect the social networks in the real world better. This chapter is based on the paper published at AAMAS 2010 [80] [1].

Chapter 4 proposes a new concept Quality of Trust Network (QoTN). In addition, to solve the NP-Complete QoTN constrained trust network extraction problem, we propose an approximation algorithm called SCAN, and two heuristic algorithms, SCAN-K and H-SCAN-K. Experiments conducted on real social network datasets illustrate that the proposed methods can deliver high quality trust networks in less execution time than the existing methods. This chapter is based on the papers published in AAAI 2012 and IEEE ICWS 2012 [81, 85].

Chapter 5 proposes a new general concept of Quality of Trust (QoT). In addition, to solve the NP-Complete QoT constrained optimal social trust path selection problem, we propose an approximation algorithm, MONTE_K based on the Monte Carlo method [43], and two heuristic algorithms, H_OSTP and MFPH-HOSTP, based on

[1] For details of the publication, please refer to page ix.

Dijkstra's shortest path algorithm [31] and our novel heuristic search strategies. Experiments conducted on real social network datasets illustrate the proposed algorithms outperform the existing methods in both the quality of the identified social trust path and the execution time. The chapter is based on the papers published in AAAI 2010 and IEEE SCC 2010, and the paper accepted by IEEE Transactions on Services Computing in 2011 [78, 83, 84, 86].

Chapter 6 proposes a heuristic algorithm, H-OSTP-K, based on Dijkstra's shortest path algorithm [31] and our novel heuristic search strategies for the NP-Complete QoT constrained K optimal social trust paths selection. Experiments conducted on real social network datasets illustrate that the proposed algorithm outperforms the existing methods in both the quality of the identified K social trust paths and the execution time. The chapter is based on the paper published in ICWS 2011 [85].

Chapter 7 proposes a new concept Quality of Trust Transitivity (QoTT), and proposes a Multiple QoTT Constrained Trust Transitivity (MQCTT) model. The experimental results demonstrate that the proposed MQCTT model follows the properties of trust and the principles illustrated in social psychology, and thus can compute more reasonable trust values than the existing methods that consider neither the impact of social information nor the properties of trust in trust transitivity. The chapter is based on the paper published in AAAI 2011 [82].

Finally, Chapter 8 concludes the work in this thesis and discusses some directions for future research opportunities.

Chapter 2

Literature Review

Online Social Networks (OSNs) are becoming more and more popular, and have been used as a means for a variety of activities, where trust between participants has significant influence on their decision-making. In the literature, many scholars in both Social Psychology and Computer Science have studied 1) social network properties, 2) trust properties, and 3) trust evaluation and trust propagation methods in OSNs. In this chapter, we review the literature on the above three aspects organized as follows:

- Section 2.1 introduces the social network properties, and presents a new categorisation of OSNs.
- Section 2.2 introduces the different definitions of trust, the general trust properties and the influence of trust on human communities.
- Section 2.3 introduces the existing studies on different aspects of trust in OSNs, including trust evaluation, trust network extraction, trust path selection and trust transitivity.

2.1 Online Social Networks (OSNs)

According to the description of a social network in the discipline of Social Science [127], a social network is a social structure made up of a set of actors (such as individuals or organizations) and the dyadic ties between these actors. The social network perspective provides a clear way of analyzing the structure of whole social entities.

In this section, social network properties and a new categorisation of OSNs will be introduced.

2.1.1 Social Network Properties

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 [101], through illustrating that the average path length between two Americans was about 6 hops in an experiment of mail sending. In addition, the influences of small-world characteristic on human interactions was further analyzed by Pool *et al*. [110] in the 1970's.

Associativity is a bias in favor of connections between network nodes with similar characteristics [105]. Mcpherson *et al*., [98] validated the *associativity* characteristic in social networks. Namely, in social networks, individuals commonly choose to associate with others of similar age, nationality, location, race, income, educational level, religion, or language as themselves.

In graph theory, a *clustering coefficient* is a measure of the degree to which nodes in a graph tend to cluster together. Namely, in a network with a high clustering coefficient if A has a connection with B and C, then the probability that B has a connection with C is high. In general, a social network has a *high clustering coefficient*. Namely, most of the people we know may also know one another in the social network of the real-world scenarios, which has been validated by [55].

In recent years, as online social networks have been gaining more popularity, sociologists and computer scientists have started to investigate their characteristics. In [103], Mislove *et al*. analyzed several popular social networks including Facebook[1], MySpace[2] and Flickr[3], and validated the *small-world* and *power-law* characteristics (i.e., in a social network, the probability that a node has degree k is proportional to k^{-r}, $r > 1$) of online social networks using data mining techniques. Also using data

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

mining techniques, Mccallum *et al.* [96] discovered the social roles (e.g., *a chief financial officer* or *in-house lawyer*) and social relationships (e.g., *partnership in a funding application*) in an email based online social network of *Enron* Corporation (cs.cmu.edu/enron/). Guo *et al.* [52], further analyzed the influence of social interactions between buyers on the purchase decisions made by a buyer in buying products in online shopping websites.

2.1.2 The New Categorisation of OSNs

In the discipline of Computer Science, there is not any unified definition of what is an online social network. Golbeck et al. [48] propose the criteria of Web-Based Social Networks (WBSNs) as follows: 1) WBSNs could be accessible over the web with a web browser; 2) Users of WBSNs must explicitly state their relationships with other people; 3) The WBSN system has explicit built-in support for users to make social connections, and 4) Each relationship is visible and browseable to users. Boyd et al. [14] propose the definition of social networking sites as "Web-based services that allow individuals to 1) construct public or semi-public profiles within a bounded system; 2) articulate a list of other users with whom they share connections; and 3) view and traverse their list of connections and those made by others with the system". Clearly, Facebook (facebook.com) and MySpace (myspace.com) are in accordance with these definitions. However, many other Websites, like YouTube (youtube.com), eBay (ebay.com), Blogs and online forums, where people can share their experience and carry out business do not follow these criteria. The relationships between participants on this type of Websites are implicit. Thus, it is still a puzzling problem whether these Websites belong to the scope of WBSNs. Below, based on different socialities of the participants in OSNs, a new categorisation of OSNs is introduced as below.

2.1.2.1 The Current Generation of Functional Websites

The current generation of functional websites, like eBay (ebay.com), support rich functionality but not contain explicit social relationships. For example, eBay supports e-commerce activities and buying-selling relations, however, it does not care about social relationships like a supervisor and his/her students, and a father and his son among the set of buyers and sellers. We summarize the characteristics of these functional websites as follows.

1. They have weak sociality where the relationships between participants are implicit; and participants do not keep their friendship lists and thus they can not make new friends with friends of friends.

2. They have rich functionality, such as email, blogs, e-commerce, and video and photo sharing etc.

2.1.2.2 The Current Generation of OSNs

As the sociality of the above websites is too weak for people to make rich social interactions, the current generation of OSNs, such as MySpace and Facebook, emerged in 2003 and 2005 respectively. They can explicitly express simple social relations, but the functionality is still limited to a very small scope, like information sharing. We summarize the characteristics of the current generation of OSNs as below.

1. They have medium sociality where the social relationships between participants are explicit and binary (friendship or non-friendship) which can be specified by participants; and participants can make new friends with a friend's friends, which is stronger than that of current functional websites.

2. They provide a platform where participants can make new friends and conduct some simple activities (e.g., sharing photos and videos) that are not as rich as those in current functional websites.

2.1.2.3 The Future Generation of OSNs based Functional Websites

We can envisage that in the near future social networks can provide the backbone to extend a number of traditional systems. For example, a traditional e-commerce system can have a social network of its buyers, and the friends' friends of buyers. Likewise, the traditional CRM (Customer Relation Management) systems can be extended to be supported by a social network of customers and other people with relations to these customers. Thus, the new generation of social network systems can be expected to support both rich social relations and rich functionality. In these systems, it would be easier to introduce products (e.g., by a retailer) or good sellers (e.g., by a buyer) to buyers, and the friends' friends of buyers. We summarize the characteristics of the new generation of OSNs based functional websites as below.

1. They have strong sociality where the social relationships are explicit and complex rather than binary (friendship or non-friendship) as in current generation OSNs.

2. They provide a platform where participants can conduct rich activities, such as, e-commerce, CRM system, recommendation systems.

2.2 What is Trust?

The Oxford Reference Dictionary (oxfordreference.com) states that trust is "the firm belief in the reliability or truth or strength of an entity". Based on the definition, a trustworthy entity will typically have a high reliability, and a trustworthy person will tell the truth and be honest with respect to interactions.

Actually, trust is a complex subject that relates to different aspects of elements, such as belief in honesty, truthfulness, competence, and reliability of the trusted person or services. In human society, trust depends on many factors, such as the past experiences with a person, the experiences with the friends of the person, preference to trust that is linked to psychological factors impacted by a lifetime of history and

events rumor, and the benefit by establishing the trust [66, 111]. As there are complex factors behind a trust relation, there is no consensus definition in the literature on what trust is [66, 111], and trust cannot be easily modeled in a computational system [48]. In this section, the trust definitions based on different aspects of views, general trust properties, and the influence of trust on human activities will be introduced.

2.2.1 Definitions of Trust

Trust plays a role across many disciplines, including sociology, psychology, economics, political science, history, philosophy, and computer science. Thus, there is not a uniform definition of trust. The work in each discipline has attempted to define the concept. In the literature, trust has been defined in many ways, as described below.

From the perspective of social psychology, Deutsch [29] proposed a widely used trust definition. He states that trusting behavior occurs when a person encounters a situation where he/she perceives an ambiguous path. The result of following the path can be good or bad, and the occurrence of the good or bad result is contingent on the action of another person. In addition, the negative impact of the bad result is greater than the positive impact of the good result. Based on Deutsch's definition, Sztompka [122] proposed a general definition of trust as "Trust is a bet about the future contingent actions of others." There are two main components of this definition: belief and commitment. First, a person believes that the trusted person will act in a certain way. The belief alone, however, is not enough to say there is trust. Trust occurs when that belief is used as the foundation for making a commitment to a particular action. These two components are also present in the core of Deutsch's definition: a person will commit to take the ambiguous path if he/she believes that the trusted target person will take the action that will lead to a good outcome.

From the perspective of sociology and history, Seligman [120] proposed a trust definition as "trust enters into social interaction in the interstices of systems, when for one reason or another systematically defined role expectations are no longer viable. If

people play their roles according to role expectations, we can safely conduct our own transaction accordingly". The problem of trust (distrust) emerges only in cases where there is "role negotiability", i.e., there is a gap between roles and role expectations [120]. In the study of Online Social Networks, Golbeck *et al.* [48] defined trust as "trust in a person is a commitment to an action based on a belief that the future action of that person will lead to a good outcome". This view of trust is similar with the above definitions given by Deutsch and Sztompka respectively, as they all regard "belief" and "commitment" as two main components of trust.

From the perspective of economics, the European Commission Joint Research Centre [59] defined trust as "Trust is the property of a business relationship, such that reliance can be placed on the business partners and the business transactions developed with them". This view of trust is from a business management perspective, which illustrates what must be done to enable trust in business. In economics trust is often conceptualized as reliability in transactions [92].

Based on the above different trust definitions, we can see that trust is really a composition of many different attributes, including reliability, dependability, honesty, truthfulness, security, competence, and timeliness, which depends on the environment in which trust is being specified.

2.2.2 Properties of Trust

Having reviewed the definitions of trust, this section introduces the general trust properties. These properties were indicated by social scientists based on their long-term observations of large amounts of human activities. Thus, they are significant in the study of trust management.

2.2.2.1 Context Dependency

Oxford Reference Dictionary (oxfordreference.com) states that "Context is the circumstances that form the setting for an event, statement, or idea, and in terms of which it

can be fully understood". This is a general definition, where all the information related to the circumstances of an entity is regarded as the context. More specifically, in Computer Science, a widely accepted definition was proposed by Dey *et al.*, [30] as "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves."

As indicated in Social Psychology [106], trust is highly context dependent. In a society, a person's trust in another person varies with the changes in contexts, as a recommender may have a different level of expertise in different domains [3, 126]. McKnight *et al.* [97] have proposed interpersonal and personal trust as one of topological categories on trust, namely, one person trusts another person in a specific context. For example, Alice may trust Bob as a mechanic in the specific context of servicing her car but probably not in the context of babysitting her children. Similarly, in the discipline of Computer Science, Marsh [93] is the pioneer to propose the concept of situational trust, which is described in an example as "Whilst I may trust my brother to drive me to the airport, I most certainly would not trust him to fly the plane". Namely, the same person, but in a different context, will require different considerations with regard to trust.

Therefore, the calculation of trust needs to consider the contextual information that has significant influence on trust evaluation.

In OSNs, social contexts include social relationships (e.g., the relationship between a father and his son), social positions (e.g., a professor in the data mining research area), preferences (e.g., like play badminton) and residential locations (e.g., living in Sydney, Australia) etc [85]. As indicated in Social Psychology [3, 76, 102], these social contexts have significant influence on trust evaluation in OSNs.

Although it is difficult to build up comprehensive social relationships, social positions, preferences and residential locations in all domains, it is feasible to build them up in some specific social communities by using data mining techniques. For example, in the email based social networks (i.e., a network is formed email communication be-

tween participants), through mining the subjects and contents of the emails in Enron Corporation (cs.cmu.edu/enron/), the social relationship between each pair of email sender and receiver (e.g., a CEO and his/her assistant) can be discovered and their roles can be known [96]. In addition, in the academic social networks formed by large databases of Computer Science literature, the social relationships between scholars (e.g., co-authors, a supervisor and his/her students) can be mined from publications (e.g., from those listed on DBLP) and the role of a scholar (e.g., a professor in the field of data mining) can be mined from their homepages [123]. Furthermore, on Facebook (www.facebook.com), the preference and the residential location of a participant can be mined from their profiles [103].

In addition to mining the social relationships and social roles, these values could be explicitly specified by participants when they join in some OSNs. For example, regarding the community impact factor, at LinkedIn (www.linkedin.com), a user can specify his/her social positions (e.g., a senior C++ programmer at IBM). If the user becomes a recommender, this social position information can illustrate his/her community impact factor in the recommendation of a specified domain. In another example of a social network consisting of the staff members in a University [130], the social positions of a user can also be specified, illustrating the user's community impact factor in the recommendations or collaborations of a specific domain. Furthermore, at SmallBlue [77], an online social network created for IBM staff, the social position of each of the participants (e.g. a project manager or a senior PHP developer) can be explicitly specified when he/she joins into this social network.

2.2.2.2 Personalization

As illustrated in Social Psychology [54, 91], trust is a subjective phenomenon that is defined by the psychological experiences of the individual who bestows it, reflecting subjective attitudes that affect participants' thinking based on subjective evaluation criteria that can vary in different domains. Trust is inherently a personal opinion. Two people often have very different opinions about the trustworthiness of the same person.

For example, suppose that Alice and Bob are two customers and Cathy is a travel agent. Alice can trust Cathy and Bob can distrust Cathy based on their different personal preference of the services provided by Cathy and their different trust evaluation criteria. Another good example given by Golbeck [48] is from the United States; when asked, "do you trust the current President to effectively lead the country?" the population will be split; some will trust him very highly, and the others will have very little trust in his abilities.

Therefore, people can have a conflict of interests, priorities, opinions, and different trust criteria in different domains [48]. So when and how much we trust people will vary. Namely, there is not a universal measure of the trustworthiness of a person. In trust calculations, the individuals personalization should be considered to reflect their interests and opinions.

2.2.2.3 Asymmetry

Trust asymmetry, means trust is not necessarily the same in both directions between two users. Since individuals have different experiences, psychological backgrounds, and histories, it is understandable why two people may trust each other by different amounts.

Trust is mutual in that each party has some trust for the other. But in real communities, there are often differences in how much they trust one another [54]. For example, Alan trusts his manager Billy, but Billy may not trust Alan to the same degree of trust. This is shown as a directed arrow from one user or node to another in a trust diagram to indicate the direction of trust so that we know we are referring to the trust from Alan to Billy, or from Billy to Alan. Another example is that research students usually say they trust their supervisors more than the supervisors trust them. This can be seen in a variety of hierarchies [132].

This phenomenon is called "one-way trust" [25, 54]. Namely, under certain circumstances, one person may trust the other, but there is no reciprocal trust. This feature of trust should be considered in trust evaluation.

2.2.2.4 Transitivity

Trust transitivity means, for example, that if Alice trusts Bob who trusts Eric, then Alice will also trust Eric to some extent. This assumes that Bob actually tells Alice that he trusts Eric, which is called a recommendation [23]. Trust is not perfectly transitive in the mathematical sense. Namely, if Alice highly trusts Bob, and Bob highly trusts Eric, it does not always and exactly follow that Alice will highly trust Eric. Generally, when encountering an unknown person, it is common for people to ask trusted friends for opinions about how much to trust this new person [48]. In trust transitivity, trust decays with the increase of transitivity hops along a social trust path [23]. The general decay of trust is nonlinear [62, 91] and can be divided into three phases.

1. Phase 1: (Slow Decay Phase) In this phase, trust decays slowly in transitivity along a social trust path from a source participant within a certain number of hops. This is because the source participant may consider the familiarity with the trustee to extend no more than a certain number of transitivity hops.

2. Phase 2: (Fast Decay Phase) With the increase of transitivity hops, the trust decay speed increases in trust transitivity until the trust value approaches the minimum. This is because that in this phase, the trustee is becoming stranger to the source participant than the case in Phase 1.

3. Phase 3: When the trust value between the source participant and the trustee is approaching the minimum, the trust decay speed changes from fast to slow. This is because in this phase, the trustee has become a complete stranger to the source participant.

In addition, trust transitivity needs certain constraints of context [23, 63]. For example, if Alice trusts Bob in the domain of *teaching C++*, and Bob trusts Eric in the domain of *repairing a car*, then the trust can not be transitive from Alice to Eric via Bob in the domain of *teaching C++*. However, if Alice also trusts Bob in *repairing a car* (in the same domain with Bob trusts Eric), then trust can be transitive from Alice

to Eric in this domain. This same argument can be extended to longer chains (social trust path) of trust.

2.2.3 The Influence of Trust on Human Activities

Trust has been an important element in interpersonal relationships in many fields recognized by many research scientists. As illustrated in Social Science [25, 41, 42], both functioning societies and online communities rely heavily on trust among their members.

Organizations increasingly are recognizing the importance of trust in the workplace. Trust is considered a fundamental ingredient for motivating productive working relationships and driving a competitive business advantage [15, 115, 128]. For example, trust facilitates strategic collaboration and cooperation [32], citizenship behavior [28, 65, 95], and conflict resolution [107]. Trust also is related to employee attitudes, such as job satisfaction [4, 114] and organizational commitment [25], as well as criterion measures, such as justice perceptions [17] and customer satisfaction [22, 121].

As indicated in social psychology [9, 38], in the reality of our society, a person prefers the recommendations from his/her trusted friends to those from others. In addition, in the discipline of Computer Science, based on statistics, Bedi *et al.*, [8] has demonstrated that given a choice between recommendations from trusted friends and those from recommender systems, trusted friends' recommendations are preferred in terms of quality and usefulness. Furthermore, in several recent studies, some researchers [21, 26] have investigated how and to what extent a participant's service selection behavior (e.g., installing a specific application software) impacts on his/her friends' decision-making in service selection. These studies have indicated that the recommendations from trusted friends have significant influence on service or target selection, not only in the society in the real world, but also in online social networks.

Although a complete social network based trust-oriented service recommendation system does not yet exist, it has become an important research topic in recent years.

Some researchers [52, 90] have proposed several models to provide more accurate recommendations of products and/or services by taking some social context information into consideration. In these studies, social trust network extraction, social trust path selection and trust transitivity are critical problems.

2.3 Trust Related Issues in OSNs

In a social network of real-world scenarios, people conduct lots of different activities where two persons may have no previous interactions and they do not know each other. In such a situation, as introduced above, trust becomes one of the most important indications for people's decision making to guide their activities. The same situation also happens in OSNs as the online social networking platforms have also been used for a variety of activities, e.g., employment, recommendation system, CRM system etc. Therefore, it is necessary and significant to evaluate trust between two unknown participants in OSNs. As trust management is a significant research area in OSNs, in the literature, many trust models and trust evaluation methods have been proposed in the study of OSNs. Based on the different aspect of trust problems they addressed, these existing works can be divided into the categories of as 1) trust evaluation, 2) trust network extraction, 3) trust path selection, and 4) trust transitivity.

2.3.1 Trust Evaluation in OSNs

Trust is a critical factor in the decision-making of participants in online social networks [71]. In order to evaluate the trust between two unknown participants in OSNs, in this field, several trust management methods have been proposed.

In the studies of trust propagation, Golback *et al*. [47] firstly extended the Friend of A Friend (FOAF) vocabulary (foaf-project.org) by adding a property where users can specify how much they trust one another based on their past interactions. The new feature has been adopted on FilmTrust, where a participant can specify a trust rating

that range from 1 (very little trust) to 10 (very high trust) to another participants based on the quality of the participant's movie recommendation. Then, they proposed a trust inference mechanism for establishing the trust relation between a source participant and the target one who have no direct interactions [48]. In their model, the attitudes of those neighbors that have been trusted by the source (have been given a high trust value by the source) are considered in the computation of the inferred trust values of those nodes which are neighbors of those neighboring nodes. Their model averages those trust values given by the neighbors of the source to their neighbors and this process will be repeated till the target node. This trust inference model has also been further adopted into FilmTrust. Guha *et al*. [51] proposed a trust propagation model to infer trust and distrust between two participants who have no direct interactions. In their model, the number of propagation hops, and the trust situations of the intermediate nodes in the social trust paths between the source and the target are considered. They adopted a weighted sum method to aggregate each of the above parts that have influence on trust propagation to compute the inferred trust value.

In the studies of trust-oriented recommendation systems, Walter *et al*. [124] proposed a recommendation system in a social network. In their model, a participant can give a trust value to a recommender based on the recommendation behavior of participants. This trust value is visible and regarded as a reference for other participants to select recommendations. The participants compute the trust value of a target participant based on multiplying the trust value between any intermediate participants in the social trust paths between a source and the target. Jamali *et al.* [58] proposed a random walk model in an OSN consisting of sellers and buyers. In their model, a buyer performs several random walks with a fixed number of hops along a path from this buyer in the social network to find the ratings given by the ending participant to a seller who sells products preferred by the buyer. The degree of confidence on the seller is calculated based on the number of random walk paths, hops and ratings of the seller in each path.

The above trust evaluation methods consider the trust values between intermedi-

ate nodes in the social trust paths, which can help establish trust relation between two unknown participants. However, all the above strategies are solely based on trust values given by participants. The social context including *social relationships* (e.g., the relationship between a buyer and a seller, or the one between an employer and an employee), *social positions* (e.g., the supervisor as a referee in a job application) and *preference* (e.g., like playing badminton) are not taken into account in these trust evaluation methods. As pointed out in Social Science theories [3, 102, 76], such social information has significant influence on trust evaluation, and can impact on participants' decision-making. Thus, the existing methods cannot be expected to deliver a reasonable trust evaluation result without considering social information.

2.3.2 Trust Network Extraction in OSNs

Participants usually have interactions with each other and they give trust ratings (values) to each other based on their interactions. Then a trust network from a source to a target (e.g., the trust network from A to E in Fig. 1.1 in *Chapter 1*) can be formed. The trust network contains some important intermediate participants, the trust relations between them and the social context underlying their past interactions. This information has important influences on trust relationships and trust evaluation. Thus, the trust network provides a basis for evaluating the trustworthiness of the target.

Section 2.3.1 has introduced some existing trust evaluation approaches that evaluate the trust value between any two nonadjacent participants. However, these methods all assume the trust network between the two participants have been identified [48, 79, 83] as the basis of their trust evaluation models. Namely, extracting such a contextual trust network between two nonadjacent participants is an essential step before performing any trust evaluation between them.

In the graph formed by social interactions, cycles (e.g., $A \rightarrow B \rightarrow C \rightarrow A$ in Fig. 1.1) usually exist due to the interactions among multiple participants [103]. Extracting a sub network from a cyclic network has been proved to be an NP-Complete problem

[6]. To the best of our knowledge, there are no approximation algorithms proposed for the NP-Complete trust network extraction in OSNs in the literature. However, the resource discovery problem in P2P networks has similar properties as the trust network extraction problem. Thus, some search strategies have been developed for the resource discovery problem and they can be applied in trust network extraction. These strategies can be classified into the following three categories.

2.3.2.1 Flooding-Based Search (FBS)

The Flooding-Based Search (FBS) mechanism typically searches the network from the source by using the Breadth First Search (BFS) strategy to find the target resource in a P2P network. It has been applied into Gnutella (rfc-gnutella.sourceforge.net), a large P2P network where individuals can exchange files over the Internet directly without going through a Web site (e.g. it has been used as a means to download music files from or share them with other Internet users). FBS sends a query of finding the target resource to each of the neighboring nodes in the network, which can make FBS method inherently unscalable in a large-scale network. As the large amount of queries FBS forwarded can consume huge computation time, the Time To Live Breadth First Search (TTL-BFS) method [19, 37] was proposed. In TTL-BFS, the Time To Live (TTL) is usually set to be an integer to indicate the time of the BFS. During TTL-BFS, TTL value is decreased by 1 or Vr $(0 < Vr < 1)$ when the depth of search is increased by 1. In this process, if the target resource is found, then the search terminates. Otherwise, TTL-BFS continues with BFS till $TTL = 0$.

2.3.2.2 Random Walk Search (RWS)

A Random Walk is a mathematical formalization of a path that consists of a succession of random steps. Random Walk Search (RWS) is a popular alternative to FBS for locating resources in P2P networks [45, 46].

In RWS, firstly, the source node is regarded as the "querying" node that needs to

locate a resource. The querying node randomly selects up to K (K is no greater than the number of the querying node's neighbors) neighboring nodes to send queries. Each of these K queries is called a random walker. Each random walker has a time to live (TTL) value that is initiated with some value $T > 0$ that limits the number of times the random walker is forwarded. When an intermediate node receives a random walker, it checks to see if it has the resource. If the intermediate node does not have the resource, it checks the TTL value, and if $T > 0$, it decrements T by 1 and forwards the query to a randomly chosen neighbor, otherwise, if $T = 0$ the query is not forwarded and RWS terminates. On the other hand, if the intermediate node has the resource, the query is not forwarded and a reply is sent to the querying node.

2.3.2.3 High Degree Search (HDS)

In High Degree Search (HDS) [2], firstly, the source node sends a query to all its neighbors based on the BFS method to determine whether they contain the resources or not. If none of the neighbors contains the resources, then HDS broadcasts the search messages along directions of the nodes with the highest degree according to the DFS method, and sets the state of the node to indicate the node has been accessed. If it does not find the resource along directions of the nodes with the highest degree, the search message will return the precursor node and broadcast along its neighbor node with the second highest degree. This procedure will stop until the search steps increase to a predefined threshold or all the nodes in the network have been accessed. In extreme cases, when the outdegree of the node are all the same, the algorithm degenerates into the standard BFS algorithm. When all the outdegree of the node are different, the algorithm degenerates into the standard DFS algorithm.

Analysis: The above search methods have been validated in many P2P networks and they have good performance in resource discovery in P2P networks. However, P2P networks do not contain social contextual information, including social relationship, social position and preference, etc. Thus, these existing methods do not consider the social context in target resource discovery but the social context has a significant in-

fluence on social interactions and trust evaluation in OSNs. In addition, as introduced in Section 2.3.1, different people may have different trust evaluation criteria, and thus the trust evaluation criteria specification should be considered in trust network extraction. But this feature is not supported by these existing methods for resource discovery in P2P networks. Thus, these existing methods cannot be expected to extract a high quality trust network to deliver a trustworthy trust evaluation result.

2.3.3 Trust Path Selection in OSNs

As shown in Fig. 1.1, a trust network can contain many social trust paths (e.g., path $A \rightarrow B \rightarrow C \rightarrow E$ and path $A \rightarrow B \rightarrow D \rightarrow E$). Evaluating trust based on all the social trust paths in a trust network can lead to huge computation time, which makes it inapplicable in large-scale social networks. In the literature, a few works have been proposed to address the social path selection problem in such networks.

SmallBlue [77] is an OSN created for IBM staff. It also provides information analytics services that automatically visualizes social networks, helps participants manage and expand their social capital, and enables participants to find people with specific knowledge. In this system, if a source would like to find a target (e.g., a C++ programmer), it considers up to 16 social paths between them with the path length of no more than 6 hops, among which, the shortest one is taken as the optimal path. In this method the shortest path can mostly affect the decision-making of the source. But the trust situation between the intermediate nodes in a social path is neglected which has significant influence on participants decision making. Hang *et al.* [53] proposed a social trust path selection method in online social networks, where trust between participants is considered in the path selection. In their model, the aggregated belief value (trust value) of a social trust path is computed by multiplying the trust value between any two intermediate nodes in the path. Among all the social trust paths, the one with the highest aggregated belief (i.e., the maximum of aggregated trust value) is selected as the optimal path that yields the most trustworthy result of trust propagation between

a source participant and the target participant. This model considers trust information, but does not have any concern for participants different trust evaluation criteria. Wang *et al*. [125] proposed a social trust path selection method where a source participant can specify a threshold. Their method first aggregated trust values given to each of the recommenders (i.e., the intermediate nodes) in the network between a source participant and the target participant. If the aggregated trust value of a recommender is greater than the threshold specified by the source participant, the recommender is kept in the trust network. Otherwise, the recommender (the node) is deleted from the trust network. After this process of node deletion, the rest of the social trust paths are kept for trust evaluation.

These existing methods select the optimal social trust path(s) from a large volume of paths based on different selection criteria, which indeed reduces the computation complexity of the trust evaluation between two unknown participants. However, in the above methods, the social information including *social relation*, *social position* and *preference* of participants are not taken into account in path selection. In addition, a source participant can have different purposes in evaluating the trustworthiness of the target participants (e.g., employment or buying products). Their different trust evaluation criteria in different applications should be reflected by specifying certain constraints of the above social information for social trust path selection. Thus, although trust information is taken into consideration in some of the existing trust path selection methods, they cannot be expected to select the trustworthy trust paths without considering social information and complex trust criteria specification.

2.3.4 Trust Transitivity in OSNs

The trust transitivity property has been validated in both Social Psychology [23] and Computer Science [63, 48]. As introduced in Section 2.2.2, under the same context, if A trusts B and B trusts C, then A can trust C to some extent. For example, Alice needs to have her car serviced, and she asks Bob for his advice about where to find a

good car mechanic in town. Bob is thus trusted by Alice to know a good car mechanic and to tell his honest opinion about that, where as Bob actually trusts the car mechanic.

After extracting the trust network and finding those trustworthy social trust paths from the trust network, to deliver a reasonable trust value in the trust management of OSNs, we need to know how trust is transitive along a social trust path. As this is a significant and challenging problem in the study of trust in OSNs, some trust transitivity models in OSNs have been proposed in the literature, and these existing models can be classified into three categories based on the types of trust transitivity strategies they adopted. These strategies are 1) multiplication strategy, 2) averaging strategy, and 3) confidence-based strategy. This section discusses each strategy and analysis the disadvantages of the existing models.

In the *first category*, the trustworthiness of a target participant is computed as the *multiplication* of the trust values between any two adjacent participants along a social trust path. For example, if A trusts B with T_{AB} and B trusts C with T_{BC} ($T_{AB},\ T_{BC} \in [0, 1]$), then A trusts C with $T_{AC} = T_{AB} * T_{BC}$. This strategy has been used in many existing models. For example, Walter *et al*., [124] proposed a social network based recommendation system, where they adopted this type of a trust transitivity model to compute the trustworthiness of a target recommender along the social trust path from a recommendee to the recommender. In addition, Lei *et al*., [73] proposed a composite service trust evaluation method, where they adopted this type of trust transitivity model to compute the aggregated trust value of a composite service along a service composition path.

In the *second category*, the trustworthiness of a target participant is computed based on averaging the trust values between any two adjacent participants along a social trust path. i.e., $T_{AC} = (w_i \cdot T_{AB} + w_j \cdot T_{BC})/2$, where w_i and w_j are the weights of T_{AB} and T_{BC} respectively, and $w_i + w_j = 1$. In the literature, Gary *et al*., [50] proposed a trust-based admission control model. In their model, they adopted this type of trust transitivity model to evaluate the trust of unknown participants and the evaluated trust values are used to participants access control. In addition, Golbeck

et al., [48] proposed a trust inference method, where they adopted this type of trust transitivity model to compute the inferred trust value of the target participant. They further adopted their trust inference model into FilmTrust.

In the *third category*, the confidence between participants is considered in trust transitivity, i.e., T_{AC} is calculated based on T_{AB}, T_{BC} and the confidence of A on T_{BC} (denoted as C_A) . C_A is computed based on the preference similarity between A and B, and it is proportional to the latter. In the literature, Guha *et al*., [51] have proposed a trust and distrust propagation model in OSNs. In their model, in addition to the trust between any two intermediate participants in a social trust path, the individual trust of each intermediate participant was also considered as the confidence in trust transitivity. The confidence value was combined with the trust value between intermediate participants by using the weighted sum method to compute the trust value of the target. In addition, Kuter *et al*., [71] have proposed a trust inference model, where the confidence of an intermediate participant was considered. In their model, the confidence values were given by domain experts, and they were used in probabilistic models to compute the probability of the trustworthiness of the target under the given confidence values of the intermediate nodes and the confidence values of the trust between them.

These existing trust transitivity models provide some feasible methods to evaluate the trustworthiness of the target along a social trust path. However, they have the following drawbacks that lead to inaccuracy and unreasonable trust values delivered by the existing models. Firstly, they do not follow the nature of trust decay illustrated in social psychology [44, 61]. Secondly, social psychology [3, 23] also illustrates that trust is not transitive in all situations. For example, Alice trusts Bob (*a football player*) in *playing soccer* and Bob trusts Tom (*a car mechanic*) in *repairing a car*. In such a situation, Alice may not trust Tom in playing soccer. Namely, participants have different social positions (e.g., *a football player* or *a car mechanic*) in different domains (e.g., *playing soccer* or *repairing a car*), which impact on trust transitivity. But existing methods do not consider this impact factor. Moreover, the *social relationships* between participants have significant influence on trust transitivity [102]. However,

they are not considered in existing trust transitivity models either. Finally, a source participant should be able to set certain constraints of the above impact factors as criteria for the trust transitivity in different domains [91, 126]. But this is not supported by existing methods.

2.4 Conclusions

This chapter has provided a general overview of the research on social networks, trust properties and trust management in OSNs. We have first presented the social network properties indicated by social scientists that need to be followed in trust management. Then we have presented a new categorisation of OSNs based on their different socialities. In addition, we have presented trust definitions and trust properties that are indicated by social scientists based on their long-term observation of a large number of human activities. Therefore, these characteristics of trust should be considered in trust management. Furthermore, we have analysed the advantages and disadvantages of the existing studies of different aspects of trust management in OSNs, including trust evaluation, trust network extraction, trust path selection and trust transitivity.

Chapter 3

Contextual Trust-Oriented Social Networks

In Chapter 1, we have introduced the existing social network structure as shown in Fig. 1.1, which only contains the trust information between two participants. However, social networks contain complex social information, including social relationships, social positions, residential locations and preferences. But such social information has not been included in any existing social network structure. This chapter presents a complex contextual trust-oriented social network structure, where not only trust but also the complex social contextual information are taken into account in modelling, better reflecting the social networks in reality. This structure is the basis for all the trust management and evaluation methods proposed in this thesis.

3.1 Social Context

As illustrated in Social Science [7], social context is the social environment of individuals, including the culture in which he/she was educated and/or lives in, and the people and institutions with whom the person interacts. In Computer Science, researchers have proposed the definitions of social context in some specific social networks. For example, Yang *et al.* [131] define the social context in micro-blog systems as "compared with traditional contexts that are defined based on textual information, social context in micro-blog systems need incorporate various dynamic social relationships,

such as the follower-followee relationships between users, retweeting relationships and replying relationships between tweets". In addition, in recommender systems, Ma *et al.* [89] gave the definition of social context as "social context information including users' social trust network, tags issued by users or information about the interests of users or properties of items."

In this thesis, the definition of the social context in a general social network is proposed as below:

Definition 1: *Social Context* (denoted as SC) is the social environment of a participant in social networks, which is divided into the *independent social environment* and the *dependent social environment*. Independent social environment contains the independent social properties associated with one person, like *social position*, *residential location*, and *preference*. Dependent social environment includes the *social relationship* between participants, the *indegree* and *outdegree* of a participant.

3.1.1 Independent Social Environments

3.1.1.1 Social Position

Social position is the position of an individual in a given community and culture. A person can have several social positions in different domains [3]. For example, a researcher can be a professor and the head of a department in a university. Let $SP_A^{D_i}$ denote A's social position in domain i.

3.1.1.2 Preference

In Social Psychology [75], preferences could be conceived of as an individual's attitude towards a set of objects, typically reflected in an explicit decision-making process. A person can have different preferences in different domains. For example, a researcher prefers doing collaboration with others who have the same research interests with him/her, and the researcher may like playing badminton as well. Let $PF_A^{D_i}$ denote A's preference in domain i.

3.1.1.3 Residential Location

The residential locations of participants are the places where people live. Let RL_A denote the residential location where A lives.

3.1.2 Dependent Social Environments

3.1.2.1 Social Relationship

As indicated in Social Science [102], two participants can have more than one type of social relationships. For example, A and B are colleagues. They can also be the members of a badminton club. Let $SR_{A,B}^{TY_j}$ $(j \in [1, 2, ..., n])$ denote the n types of social relationships between participants A and B.

3.1.2.2 Indegree

In social networks, the indegree of a participant is the number of participants who have *social interactions* with him/her. A large indegree of a participant in an OSN indicates the participant is well known in the community. Moreover, the recommendation from such a participant with a larger indegree is more credible and this has been validated in Social Science theories [112]. Let $deg^-(A)$ denote the indegree of A.

3.1.2.3 Outdegree

In social networks, the outdegree of a participant is the number of other participants with whom this participant has *social interactions*. The larger the outdegree of a participant in an OSN, the more opportunities he/she has a social connection with others in the community (i.e., connect with unknown participants with the direct social interactions via intermediate nodes). If a node has a higher outdegree, it indicates that the node has more neighboring nodes. Thus, it is more likely for such a node to be connected with the target node via its neighbors and its neighbors' of neighbors. Let $deg^+(A)$ denote the outdegree of A.

As analysed above, let $NE(A) = \{B_1, ..., B_n\}$ be the set of all the neighbors of A; then the social context of A in a social network for a given domain D_i can be denoted as $\mathcal{SC}(\mathcal{A}) = \{\{SP_A^{D_i}, PF_A^{D_i}, RL_A, SR_{A,B_j}^{TY_j}, deg^-(A), deg^+(A)\} | (B_j \in NE(A), j \in [1, 2, ..., n]\}$.

3.2 Social Contextual Impact Factors

As indicated in Social Psychology [3, 75, 102], social contexts have significant influence on trust evaluation. Then based on the social contexts in social environments, several social context impact factors are proposed as follows.

3.2.1 Trust

As introduced in Section 2.2.1, trust is a complex subject, and lots of trust definitions have been proposed in different disciplines. In this thesis, we propose the definition of trust as below,

Definition 2: Trust is the belief of one participant in another, based on their interactions, with the extent to which the future action to be performed by the latter will lead to an expected outcome.

As pointed out in [91, 126], the trust value between two people can be different in different domains. For example, A trusts B in teaching C++, but A may not trust B in repairing a car. In our model, let $T_{AB}^{D_i} \in [0, 1]$ (e.g., $T_{AB}^{D_i} = 0.5$ in the closed interval between 0 and 1) denote the trust value that A assigns to B in domain i. If $T_{AB}^{D_i} = 0$, it indicates that A completely distrusts B in domain i, while $T_{AB}^{D_i} = 1$ indicates that A completely believes B's future action can lead to the expected outcome in that domain.

3.2.2 Social Intimacy Degree

As illustrated in Social Psychology [5, 16], a participant can trust and have more social interactions with the participants with whom he/she has more intimate social re-

lationships than those with whom he/she has less intimate social relationships. Let $SI_{AB} \in [0, 1]$ (e.g., $SI_{AB} = 0.5$ in the closed interval between 0 and 1) denote the *Social Intimacy Degree* between A and B in online social networks. $SI_{AB} = 0$ indicates that A and B have no intimate social relationship while $SI_{AB} = 1$ indicates they have the most intimate social relationship.

As introduced in *Section 2.2.2.1*, in email based social networks, through mining the subjects and contents of the emails, such as those in Enron Corporation (cs.cmu.edu/enron/), the social relationship between each pair of email sender and receiver (e.g., a CEO and his/her assistant) can be discovered. In addition, in the academic social networks formed by large databases of Computer Science literature, the social relationships between scholars (e.g., co-authors, a supervisor and his/her students) can be mined from their publications (e.g., from DBLP). Based on the mined social relationships, the corresponding social intimacy degree between participants can be computed by using probabilistic models [96] or the PageRank model [123].

3.2.3 Community Impact Factor

Rich activities of participants in social networks can be categorized into different domains (e.g., hiring employees or product sales) based on their characteristics [126]. As illustrated in Social Psychology [3, 27], in a certain domain of interest, an expert's recommendation is more credible than that from a beginner. In addition to expertise, as illustrated in Cognitive Science [69] and Computer Science [112], a well-known person (i.e., a large indegree of the node) is more credible than that of a person who is interacted by less people. Therefore, let $CIF_A^{D_i} \in [0, 1]$ (e.g., $CIF_A^{D_i} = 0.5$ in the closed interval between 0 and 1) denote the *Community Impact Factor* of A, illustrating the community impact of participant A in domain i, which is determined by the expertise of A and the number of social interactions with A (i.e., $deg^-(A)$) in domain i. $CIF_A^{D_i} = 1$ indicates that A is a domain expert and has the greatest impact in domain i while $CIF_A^{D_i} = 0$ indicates that A has no knowledge and has the least impact in that

domain.

As introduced in *Section 2.2.2.1*, in the email based social networks, the social position, like a CEO or his/her assistant can be discovered through mining the subjects and contents of the emails. In addition, in the academic social networks, the social position of a scholar (e.g., a professor in the field of data mining) can also be mined from their homepages. Then, based on mined social position information, the corresponding community impact factors can also be calculated by adopting probabilistic models [96] or PageRank model [123].

3.2.4 Preference Similarity

As illustrated in Social Psychology [88, 136], a participant A can trust and have more social interactions with another participant B, with whom A has more similar preferences (e.g., both of them like playing badminton) than others, with whom he/she has fewer similar preferences. Let $PS_{AB}^{D_i} \in [0,1]$ (e.g., $PS_{AB}^{D_i} = 0.5$ in the closed interval between 0 and 1) denote the *Preference Similarity* between A and B in domain i. When $PS_{AB}^{D_i} = 0$, A and B have no similar preference in domain i. When $PS_{AB}^{D_i} = 1$, they have the same preference in that domain.

On Facebook, as the preference of a participant can be found from his/her profile, the preference similarity between two participants can be mined based on their profiles [103].

3.2.5 Residential Location Distance

As illustrated in Social Psychology [7, 44], a participant A can trust more on and have more social interactions with B if A's residential location is closer to B than others whose residential location is far away. Let $RLD_{AB} \in [0,1]$ (e.g., $RLD_{AB} = 0.5$ in the closed interval between 0 and 1) denote the *Residential Location Distance* between A and B. When $RLD_{AB} = 1$, the residential locations of A and B are the same. When $RLD_{AB} = 0$, it indicates that the residential location between them has the largest

distance.

On Facebook, the residential location of a participant can also be found from his/her profile, and then the corresponding residential location distance between two participants can also be calculated [103]. Detailed mining methods of these social contextual impact factor values are out of the scope of this thesis.

3.3 A Contextual Trust-Oriented Social Network Structure

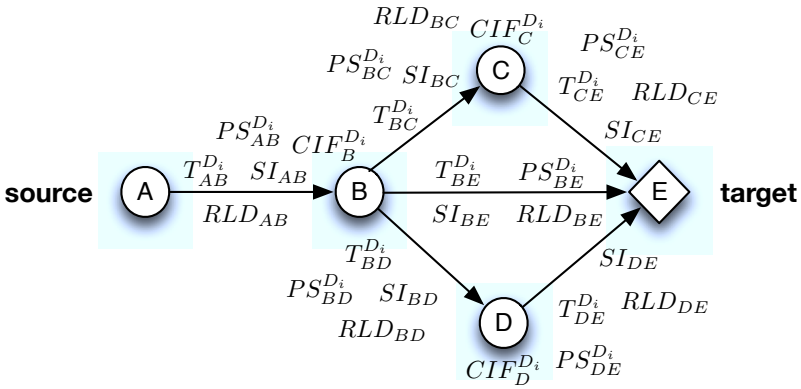

Figure 3.1: A contextual trust-oriented social network

Based on the social contextual impact factors identified above, a new structure for complex contextual trust-oriented social networks is shown in Fig. 3.1, where for each participant (e.g., B), the community impact factor (e.g., $CIF_B^{D_i}$) is added, and for each pair of participants who have direct interactions (e.g., A and B), the social intimacy degree (e.g, SI_{AB}), trust (e.g., $T_{AB}^{D_i}$), preference similarity (e.g., $PS_{AB}^{D_i}$) and residential location distance (e.g., RLD_{AB}) are added.

3.4 Conclusion

In this chapter, we have introduced the social contextual information proposed to be included in social networks. In addition, based on these social contexts, the corresponding social contextual impact factors are proposed which have significant influences on trust evaluation. Furthermore, we have proposed a complex contextual trust-oriented social network structure, which contains the social contextual impact factors, including social intimacy degree, community impact factor, preference similarity and residential location distance. This structure can better reflect the social network structure in the real-world scenarios, and thus it is proposed to be the basis to develop reasonable trust management and evaluation models in OSNs.

Chapter 4

Trust Network Extraction in Large-Scale Trust-Oriented Social Networks

4.1 The Trust Network Extraction Problem

As introduced in *Chapter 1*, an Online Social Network (OSN) can be represented as a graph, where each node represents a participant and each link between two nodes corresponds to a real-world or online interaction. In the social network depicted in Fig. 1.1, suppose A is looking for a badminton coach and E is a badminton coach. In such a situation, as indicated in the theory of Social Psychology [23, 91] and Computer Science [48, 78], A can evaluate the trustworthiness of E based on the trust network from A to E by using trust transitivity and trust propagation methods [48, 78]. Therefore, in OSNs, such a trust network is critical and the basis for the trust evaluation for two nonadjacent participants, as it contains some important intermediate participants, the trust relations between those participants and social context. All of them have important influences on the trust evaluation between two unknown participants in OSNs. Extracting the trust network between two unknown participants becomes a fundamental and essential step before performing trust propagation [48, 83, 85].

As introduced in *Chapter 2.3.2*, extracting a sub network from a cyclic network has been proved to be an NP-Complete problem [6]. Therefore, in real applications,

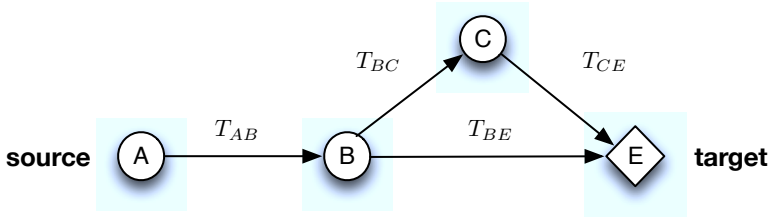

Figure 4.1: A sub trust network (TN-part-1)

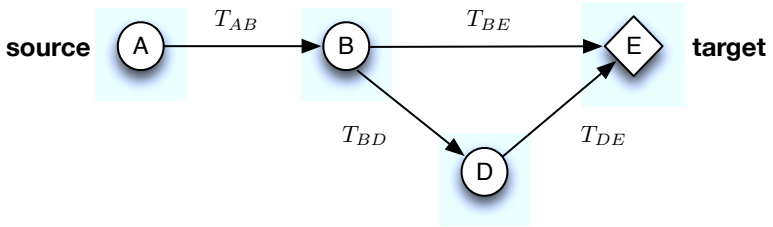

Figure 4.2: A sub trust network (TN-part-2)

only part of the trust network can be extracted for trust evaluation, which consequently affects the reliability of trust evaluation in response to a trust query given in a certain context.

For example, in Fig. 1.1 shown in Fig. 1, suppose C has no knowledge of badminton, and personally, C does not know E well. T_{CE} was given by C as C bought a used car from E. But D is good at playing badminton and D is familiar with E, T_{DE} was given by D as D usually plays badminton with E. Fig. 4.1 and Fig. 4.2 depict two possible sub trust networks (denoted as TN-part-1, and TN-part-2) extracted from the whole trust network in Fig. 1.1.

In the trust query of *looking for a badminton coach* given by A, based on the trust theory in Social Psychology [23, 91], D's recommendation of E may be considered more credible than C. From the two figures, we can see that TN-part-1 does not contain the important intermediate node D and the important trust relation T_{DE}. Therefore, based on the trust theory in Social Psychology [23, 91], the trust evaluation result delivered based on TN-part-1 is less reasonable than that delivered based on

TN-part-2. Namely, it is not reasonable to adopt the trust value given by a participant, who has no knowledge about badminton, based on the sale of a used car to evaluate the trustworthiness of a person to be a badminton coach. Therefore, given a trust query under a certain context (e.g., looking for a badminton coach), a trust network has *higher quality* (e.g., TN-part-2) if the network contains more important intermediate participants (e.g., D), their trust relations (e.g., T_{BD} and T_{DE} in Fig. 4.2) and the corresponding contextual information (e.g., D is familiar with E and D has expertise in the domain of playing badminton in Fig. 4.2), and excludes those less important (irrelevant) nodes (e.g., C) and links (e.g., T_{CE}). Such a trust network with *higher quality* can deliver more reasonable trust evaluation results. Therefore, it is necessary and important to address the challenging NP-Complete trust network extraction problem to provide a high quality trust network as the foundation for the trust evaluation between two unknown participants.

In the literature, to the best of our knowledge, there are no proposed methods for the NP-Complete social trust network extraction problem. But some resource discovery methods, which are developed for P2P networks, can be used for trust network extraction in OSNs, as it has similar properties as the trust network extraction problem. Such as 1) Time To Live Breadth First Search (TTL-BFS) method [19, 37], 2) Random Walk Search (RWS) method [45, 46], and 3) High Degree Search (HDS) method [2]. But these methods do not consider the social contexts, including social relationships, social positions, residential locations and preferences of participants. As indicated in Social Psychology [16, 75, 135], all the above social contextual information has significant influence on both social interactions and trust evaluation. In addition, a source participant may specify some constraints on social contexts to reflect his/her trust evaluation criteria in trust network extraction. However, it is not supported by the above existing resource discovery methods in P2P networks.

This chapter first proposes a new concept, called QoTN (Quality of Trust Network) and proposes a social context-aware trust network extraction method with QoTN constraints. Moreover, to address the NP-Complete trust network extraction problem, this

chapter then proposes an approximation algorithm, called SCAN, based on the Monte Carlo method, and two heuristic algorithms, called H-SCAN and H-SCAN-K based on K-Best-First Search (KBFS) method [34], bidirectional search (i.e., search from both the source and the target nodes simultaneously) [57] and our proposed optimization and heuristic search strategies. The experimental results illustrate that on average our methods can extract trust networks with higher quality by consuming less execution time than the existing methods.

4.2 Social Context-Aware Trust Network Extraction Models

In this section, we first discuss the influence of social contextual impact factors on social interactions and social connections, and then propose a new concept called Quality of Trust Network (QoTN) and a trust network utility calculation method, all of which are the key components of our social context based trust network extraction model.

4.2.1 The Influence of Social Context on Social Interactions and Social Connections

In Computer Science, based on the statistics of 1000 publications from 18 countries on ISI Web of Knowledge (apps.webofknowledge.com) [129], in 54% of the papers, the first author and the last author had the same address. In addition, based on the statistics on Flickr (flickr.com) - an online photo sharing social network [103], any two participants in photo sharing usually have similar preferences. These examples illustrate that the social context of two participants (e.g., residential location and preference) have influence on their social interactions in different domains (e.g., in research and in photo sharing), and thus can affect social connections between participants. This feature also has been validated by Social Psychology theory [16, 44, 88, 136]. Next, we give an example to illustrate the influences of social context on social connections.

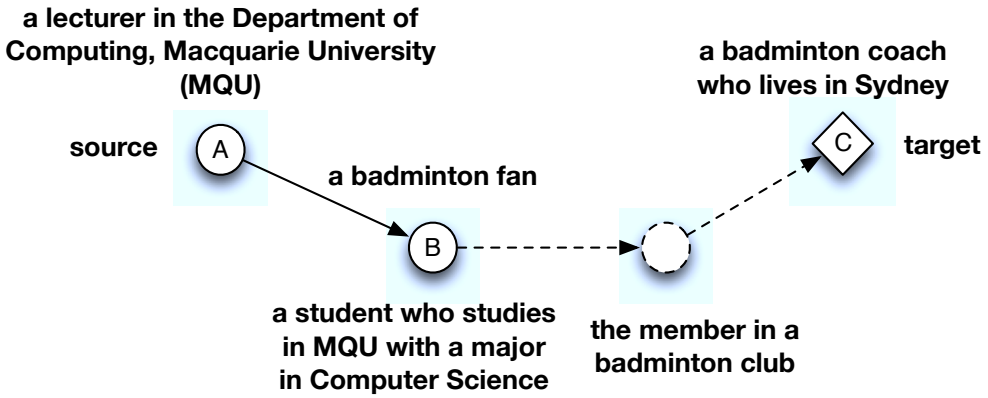

Figure 4.3: The influence of social context on social connections

As depicted in Fig 4.3, A is a lecturer in the Department of Computer Science of Macquarie University in Sydney, Australia, and he/she has social interactions with B as B is a student of A. C is a badminton coach and lives in the same city with both A and B. B does not have direct interactions with C. In the OSN, suppose A is looking for a badminton coach, (i.e., A is the source and C is the target), we could see that B and C have a high probability to have a social connection as both of them like playing badminton and thus they may be connected via some other members in a badminton club.

From this example, we can see that the social context similarity between two unknown participants can affect the social interactions and thus affect the probability of their social connections. Based on this property, we can estimate a social connection based on the social context similarity between participants. Next we propose a social context similarity method by Eq. (4.1).

Let $Smi^{D_i}_{v_m,v_t}$ denote the social context similarity between v_m and v_t in domain i (v_m is nonadjacent with v_t), which can be calculated by the following Eq. (4.1). This method considers the influence of different social contexts including preference similarity, the similarity of community impact factors and residential location distance in different social networks.

$$Smi^{D_i}_{v_m,v_t} = \omega_1 * PS^{D_i}_{v_m,v_t} + \omega_2 * |CIF^{D_i}_{v_m} - CIF^{D_i}_{v_t}| + \omega_3 * RLD_{v_m,v_t} \quad (4.1)$$

where ω_1, ω_2 and ω_3 are the weights of social impact factors and $\omega_1 + \omega_2 + \omega_3 = 1$. These weights can be different in different social networks. For example, in a social community, if the preference similarity between participants has more impact on social connection, ω_1 should be given a relatively high value.

The larger the social context similarity between two participants, the more likely for them to having social connections. In our model, the social context similarity is considered in the node selection of trust network extraction, where the larger the similarity between a node and the target which are nonadjacent, the more likely for the node to be selected (e.g., B has a higher likelihood to be selected as it has similar social context with target C in Fig. 4.3.)

4.2.2 Quality of Trust Network (QoTN)

In addition to the influence of social contexts, our model also considers different trust evaluation criteria in trust network extraction. We first present a new concept, *Quality of Trust Network*, as below.

Definition 2: *Quality of Trust Network* (QoTN) is the ability of a contextual trust network to guarantee a certain level of trustworthiness in trust evaluation, taking $T,\ SI,\ CIF,\ PS,\ RLD$ as attributes.

In our model, a source participant can specify multiple constraints of QoTN attributes (i.e., T, SI, CIF, PS and RLD) for intermediate nodes and their links in a trust network, as the requirements of trust network extraction in different domains. Let $QoTN^{(\eta)}_{v_s,v_t}$ ($\eta \in \{T,\ SI,\ CIF,\ PS,\ RLD\}$) denote the QoTN constraint of η in the trust network from v_s to v_t (throughout this thesis, v_s denotes the source and v_t denotes the target in a social network). For example, to *hire employees*, v_s, a hiring manager, can specify the QoTN constraints as $\{QoTN^{T}_{v_s,v_t} > 0.3,\ QoTN^{SI}_{v_s,v_t} > 0.3,\ QoTN^{PS}_{v_s,v_t} > 0.3,\ QoTN^{RLD}_{v_s,v_t} > 0.3,\ QoTN^{CIF}_{v_s,v_t} > 0.8\}$, if he/she believes the

community impact factor of each of the intermediate participants is more important in the domain of *recruitment*.

4.2.3 Trust Network Utility

In our model, we define the utility (denoted as $\mathcal{U}$) as the measurement of the trustworthiness of an extracted trust network. In a given domain, the utility function takes the QoTN attributes T, SI, CIF, PS and RLD as the arguments in Eq. (4.2)

$$\mathcal{U}(v_s, v_t) = \sum_{j=1}^{N} T_j + \sum_{j=1}^{N} SI_j + \sum_{j=1}^{M} CIF_j + \sum_{j=1}^{N} PS_j + \sum_{j=1}^{N} RLD_j \tag{4.2}$$

where M is the number of intermediate nodes and N is the number of corresponding links in the trust network,

The goal of trust network extraction is to extract the optimal trust network from the source v_s to the target v_t that satisfies multiple QoTN constraints and yields the highest utility, which is an NP-Complete problem as it covers finding the longest simple path (a simple path is an acyclic path) in a graph which has been proved to be NP-Complete [6].

4.3 The Proposed SCAN Algorithm for Trust Network Extraction

4.3.0.1 Social Context-Aware Social Interaction Probability

In the real world, the *normal distribution* has been widely used since the 18th century to model the relative frequency of physical and social phenomena [60], for example, the IQ, income and reading skills of people in a general population, the box-office performance of feature films, the output of journal articles by scientists, and the number of violent acts committed by male teenagers [1]. The probability density function of the normal distribution is as Eq.(4.3) (see the function image in Fig. 4.4).

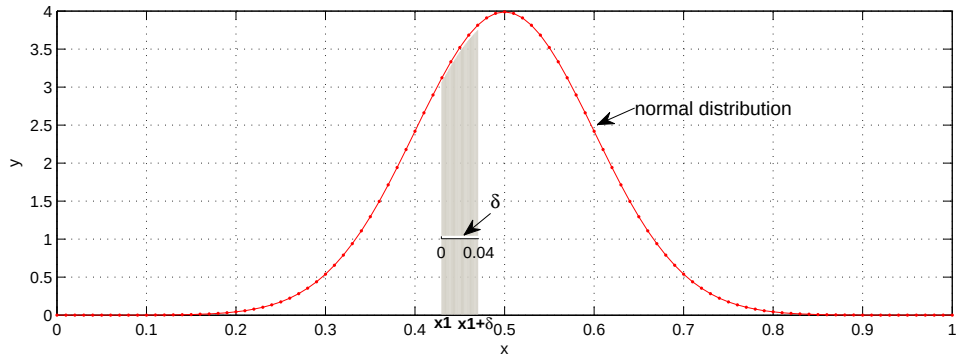

Figure 4.4: Normal distribution

$$y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.3}$$

where parameters μ and σ are the mean and standard deviation respectively, controlling the curve of the function image.

In our model, we assume the probability distribution of a social interaction between any two participants with the social contextual impact factors also follows the normal distribution. Let $P(A \rightarrow B|X)$ denote the probability of A and B have social interactions, where X is the value of each item included in Eq (4.1) (i.e., $PS_{AB}^{D_i}$, RLD_{AB}) and $|CIF_A^{D_i} - CIF_B^{D_i}|$).

Then, based on mathematical theory of the integration [11], $P(A \rightarrow B|X)$ can be calculated by Eq. (4.4). In this equation, δ is the length of each small interval (the horizontal axis between 0 and 1 is divided into several small intervals [11]). If X is in one of the intervals (e.g., in the interval $[x_1, x_1 + \delta]$ in Fig. 4.4), $P(A \rightarrow B|X)$ is the integration of Eq. (4.3) with a lower limit X and an upper limit $X + \delta$ (in the case shown in Fig. 4.4, where $X = x_1$). Namely, $P(A \rightarrow B|X)$ is equal to the corresponding area of the trapezoid with curved edges in an interval $[X, X + \delta]$ (e.g., the shadowed area in Fig. 4.4). In addition, the parameters μ and σ in Eq. (4.4) can be computed by applying social statistics methods and mathematical theories in a social network [11, 12]. But this problem is out of the scope of this thesis. Finally, based on probability theory [11], the aggregated social interaction probability between A and B

(denoted as $AP(A \rightarrow B)$) can be calculated by Eq. (4.5).

$$P(A \rightarrow B|X) = \int_{X}^{X+\delta} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-\mu)^2}{2\sigma^2}} d(X) \tag{4.4}$$

$$AP(A \rightarrow B) = \prod P(A \rightarrow B|X) \tag{4.5}$$

In our model, the aggregated social interaction probability will be considered in the node selection of trust network discovery, where the larger the probability of a node to have a social interaction with the target, the more likely for the node to be selected.

To solve the NP-Complete trust network discovery problem with QoTN constraints, we propose a Social Context-Aware trust Network discovery (SCAN) algorithm, by adopting the Monte Carlo method and two optimization strategies.

4.3.1 Monte Carlo Method

Monte Carlo method Monte Carlo method [43] is a computational algorithm that relies on repeated random sampling to solve a problem. The specific areas of application of the Monte Carlo method include computational physics, physical chemistry, global illumination computations, finance and business, and computational mathematics (e.g. numerical integration and numerical optimization) [43, 104]. It is also one of the techniques with good efficiency for solving NP-complete problems [43, 104]. In the literature, based on the Monte Carlo method, a number of algorithms have been proposed for solving NP-Complete problems [43, 73, 104].

4.3.2 Algorithm Description of SCAN

Based on the Monte Carlo method, we propose an approximation algorithm for Social Context-Aware trust Network (SCAN) selection problem. In SCAN, initially, the source participant v_s is regarded as the current expansion node (the node as the start point of the next step search), and SCAN searches all the neighboring nodes of v_s

(denoted as $v_s.neighboring\_node$) to investigate whether the current node and its corresponding links satisfy the QoTN constraints. If all QoTN constraints can be satisfied, the neighboring node is called *a feasible node* (denoted as v_f). Given that the larger the outdegree of a node, the more likely the node is to have a connection with others [2], SCAN calculates the *selection probability* of all the feasible nodes (denoted as $SCP(v_f \rightarrow v_t)$) based on $AP(v_f \rightarrow v_t)$ and the outdegree of v_f (denoted as $deg^+(v_f)$) by Eq. (4.6).

$$SCP(v_f \rightarrow v_t) = AP(v_f \rightarrow v_t) \cdot \frac{deg^+(v_f)}{MAX(deg^+)} \tag{4.6}$$

where $MAX(deg^+)$ is the maximal value of the outdegree of the nodes in a social network.

After that, based on their selection probabilities, SCAN selects one of the feasible nodes as the next expansion node (denoted as v_{exp}), where the higher the selection probability of a node, the more likely for the node to be selected. During the process, a cycle in a path is avoided by the strategy in [109], as it leads to inefficiency and ineffectiveness of the network discovery [91]. Finally, SCAN repeats the above search process at each v_{exp} until it finds v_t or reaches the threshold of search hops (denoted as λ_h, on average $\lambda_h \leq 7$ due to the *small-world* phenomenon of social networks. During the search process, in addition to the basic Monte Carlo method, we adopt the following two optimization strategies to improve the efficiency of our algorithm.

Optimization Strategy 1: Avoiding Repeated Feasibility Investigations in Simulations. In each search step, the Monte Carlo method investigates the feasibility of all the neighboring nodes of the current v_{exp} (e.g., investigating v_x, v_y and v_z in Fig. 4.5). In multiple simulations of the Monte Carlo method, a node may be selected as a v_{exp} more than once. In such a situation, the feasibility investigation needs to be performed repeatedly, leading to low efficiency. To address this issue, in SCAN, if a neighboring node is infeasible, (e.g., v_x in Fig. 4.5), the corresponding link from the current v_{exp} to the neighboring node (e.g., $v_m \rightarrow v_x$) will be removed. Then, upon reaching

the same v_{exp} (e.g., v_m) in the subsequent simulations, SCAN does not investigate its neighboring nodes repeatedly as all of them are feasible.

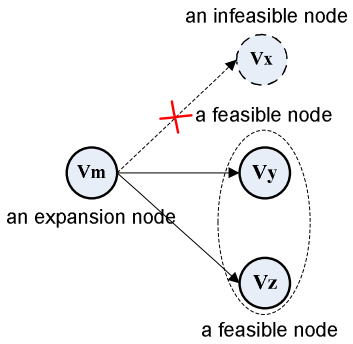

Figure 4.5: Unsatisfied nodes

Optimization Strategy 2: Avoiding Repeated Probability Calculations in Simulations. In multiple simulations of the Monte Carlo method, if the same v_{exp} is selected more than once (e.g., v_m in Fig. 4.5 is selected more than once), its feasible neighboring nodes' selection probabilities will have to be calculated repeatedly, leading to low efficiency. To address this issue, at each feasible node, SCAN records its selection probability (denoted as $v.selection$), thereby avoiding repeated calculations in the subsequent simulations.

Given a group of QoTN constraints, and a pair of v_s and v_t in a complex contextual social network, the process of SCAN includes the following steps.

4.3.3 The Process of SCAN

Initialization: At each node v_i, set $v_i.probability\_status = 0$, which indicates the social interaction probabilities of v_i's neighboring nodes (denoted as $v_i.neighboring\_nodes$) have not been calculated. In addition, all the nodes are marked as unvisited ($v_i.visit = 0$), and set $v_{exp} = v_s$.

Step 1: Based on $v_{exp}.probability\_status$, SCAN performs the following search strategies.

If $v_{exp}.probability\_status \neq 1$, SCAN investigates the feasibility of v_j, $v_j \in$

$\{v_{exp}.neighboring\_nodes\}$) as follows.

(a) **Case 1:** If $v_j = v_t$, then SCAN terminates the current search, and starts a new simulation from *Initialization*.

(b) **Case 2:** If $v_j \neq v_t$ and v_j is a feasible node, then SCAN calculates $SCP(v_j \rightarrow v_t)$, and sets $v_j.selection = SCP(v_j \rightarrow v_t)$ and newline $v_{exp}.probability\_status = 1$.

(c) **Case 3:** If $v_j \neq v_t$ and v_j is an infeasible node, based on the number of the infeasible neighboring nodes of v_{exp} (denoted as $v_{exp}.infeasible\_number$), SCAN performs the following search strategies.

(c-1-1): If $v_{exp}.infeasible\_number = v_{exp}.neighboring\_nodes$ and $v_{exp} = v_s$, then SCAN terminates, failing to return a trust network that satisfies QoTN constraints.

(c-1-2): If $v_{exp}.infeasible\_number = v_{exp}.neighboring\_nodes$ and $v_{exp} \neq v_s$, then SCAN terminates the current search and starts a new simulation from *initialization*.

(c-1-3): If $v_{exp}.infeasible\_number \neq v_{exp}.neighboring\_nodes$, go to *Step 2*.

Step 2. If $v_{sel}.visit = 0$, calculate the probability of v_j to be a v_{exp} by the following Eq.(4.7).

$$p(v_j) = \frac{v_j.selection}{\sum v_k.selection} \; v_k \in \{v_{exp}.neighboring\_node\} \tag{4.7}$$

Step 3. Select one of the feasible neighboring nodes (denoted as v_{sel}) based on their probabilities obtained by Eq. (4.7). Then based on *Optimization Strategy 1*, set $v_{sel}.visit = 1$, to avoid cycles in the subsequent search steps of the current simulation.

Step 4. Set $v_{exp} = v_{sel}$, and continue the search from *Step 1* until the number of searching hops reaches λ_h (on average $\lambda_h \leq 7$[101]).

The time complexity of SCAN is $O(sld)$, where s is the number of simulations; l is the average length of the social trust paths from v_s to v_t; d is the maximal outdegree

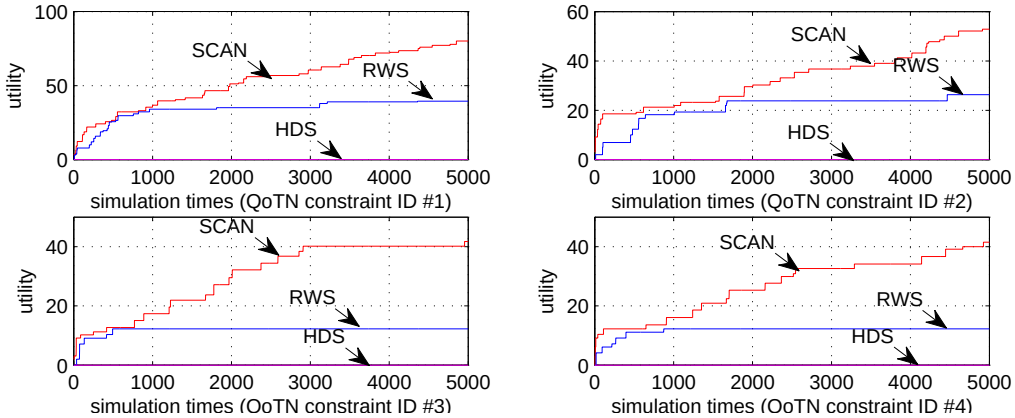

Figure 4.6: The utilities of extracted trust networks with 4 hops

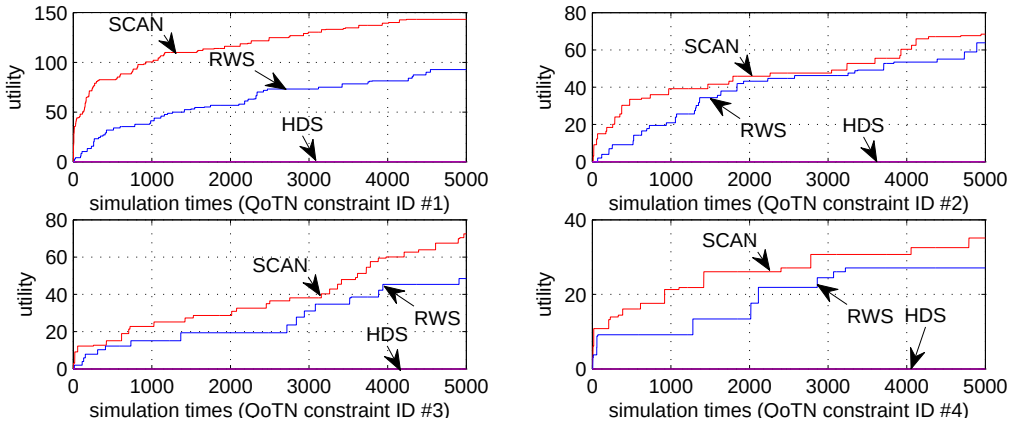

Figure 4.7: The utilities of extracted trust networks with 6 hops

of the nodes in social networks. In social networks, on average, $l < 7$ [101]. Thus the time complexity of SCAN is $O(sd)$, which is better than FBS (Flooding Based Search) with the time complexity of $O(d^{TTL})$ (TTL is Time To Live, introduced in *Section 2.3*), and the same as those of both RWS (Random Walk Search) and HDS (High Degree Search).

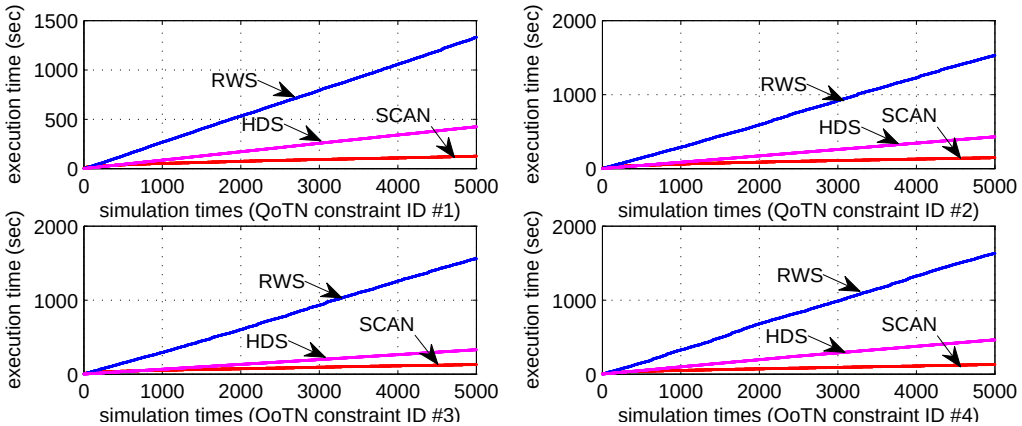

Figure 4.8: The execution time (4 hops)

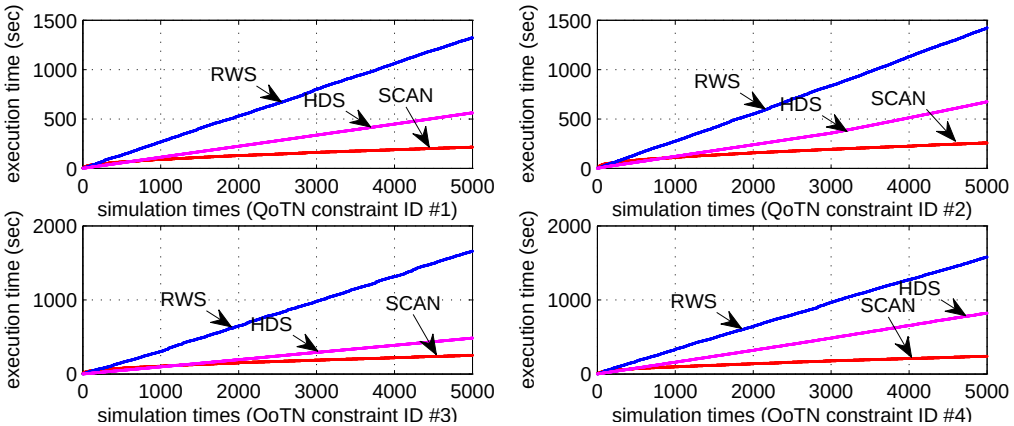

Figure 4.9: The execution time (6 hops)

4.4 Experiments on SCAN

4.4.1 Experimental Setup

Firstly, in order to evaluate the performance of our proposed algorithm SCAN on trust network discovery, we need a dataset that contains social network structures. The *Enron* email dataset (https://www.cs.cmu.edu/ enron/) 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 [79, 82, 96, 117]. To validate our pro-

posed algorithm, we select the *Enron* email dataset with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) as the dataset for our experiments. Secondly, we randomly select a source and a target from the dataset, and compare our SCAN with other methods in all the three categories, i.e., FBS, RWS and HDS (see *Chapter 2.2.3*). Thirdly, we set four groups of QoTN constraints as listed in Table 4.1 and set the social interaction probability to approximately follow the normal distribution with $\mu = 0.5$ and $\sigma = 0.1$. Finally, since the detailed mining method of social contextual impact factor values is out of the scope of this thesis, their values are generated by using the function $normrnd(\mu, \sigma)$ in *Matlab*, which generates random numbers in the range of $[0, 1]$, following the normal distribution with μ and σ.

Each of SCAN, FBS, RWS and HDS is implemented using Matlab R2008a running on an Lenovo 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 relational database. The results are plotted in Fig. 4.6 to Fig. 4.9, where the execution time and the utilities of the extracted trust network for each of the algorithms are averaged based on 10 independent runs. In each run, we perform up to 5000 simulations.

Table 4.1: The settings of QoTN constraints

ID	$QoTN(T)$	$QoTN(SI)$	$QoTN(PS)$	$QoTN(RLD)$	$QoTN(\rho)$
1	0.10	0.10	0.10	0.10	0.10
2	0.15	0.15	0.15	0.15	0.15
3	0.20	0.20	0.20	0.20	0.20
4	0.25	0.25	0.25	0.25	0.25

4.4.2 Results and Analysis

During the execution of FBS with TTL=2, MATLAB runs out of memory. This could be caused by the large outdegrees of the nodes in the two search hops. Therefore, we compare the extracted trust networks' utilities delivered by SCAN, RWS and HDS and their execution time.

Fig. 4.6 to Fig. 4.7 plot the extracted trust networks' utilities with different QoTN

Table 4.2: The comparison of the utility

Simulations	Difference of path utility				
	4 hops	5 hops	6 hops	7 hops	total
1000	19.24%more	45.02%more	112.37%more	114.15%more	72.69%more
2000	62.40%more	61.68%more	63.29%more	118.71%more	76.52%more
3000	103.84%more	88.18%more	41.08%more	104.62%more	84.43%more
4000	114.80%more	101.68%more	40.13%more	84.42%more	85.25%more
5000	139.55%more	92.51%more	37.55%more	64.78%more	71.12%more

Table 4.3: The comparison of execution time (5000 simulations)

Algorithms	The sum of the average execution time (sec)				
	4 hops	5 hops	6 hops	7 hops	total
SCAN	532.5590	772.2890	910.7940	1.1280e+003	3.3436e+003
RWS	6.0538e+003	5.8048e+003	5.9819e+003	6.3017e+003	2.4142e+004
HDS	1.6466e+003	2.0008e+003	2.3238e+003	2.5434e+003	8.5146e+003
SCAN/RWS	0.0880	0.1330	0.1523	0.1790	0.1385
SCAN/HDS	0.3234	0.3836	0.3919	0.4435	0.3927

constraints with 4 and 6 search hops (the figures for 5 and 7 hops are similar to those of 4 and 6). From them we could see that firstly, with the same simulation times, our proposed SCAN can deliver much higher network utilities than all the other methods in all cases. In addition, since HDS cannot find v_t after 5000 times search from v_s in all cases, the extracted trust network's utility delivered by HDS is always *zero*. Thus, we compare the average utilities based on different QoTN constraints delivered by SCAN and RWS in Table 4.2. From them we could see that, on average, SCAN can deliver extra 72.69%, 76.52%, 84.43%, 85.25% and 71.12% more utilities respectively than RWS with 1000, 2000, 3000, 4000 and 5000 simulations. This is because SCAN takes into account the influence of social context on social interactions, where the larger the probability that a node has a social interaction with v_t, the more likely for the node to be selected. This method increases the probability of finding a trust path from v_s to v_t at each search run. In addition, SCAN considers the QoTN constrains, and can avoid searching infeasible nodes, improving the effectiveness of each search.

Fig. 4.8 to Fig. 4.9 plot the execution time of SCAN, RWS and HDS with different QoTN constraints with 4 and 6 search hops (the figures for 5 and 7 hops are similar

to those of 4 and 6). From the results, we could see that the execution time of SCAN is less than that of RWS in all cases. In addition, when the simulation times are less than 1000, the execution time of SCAN is similar to HDS. But with the increase of simulation times, HDS consumes much more execution time than SCAN. The average execution time of HDS, RWS and SCAN in each of 4 to 7 search hops is listed in Table 4.3. Based on the statistics of all executions, on average, the execution time of SCAN is only 13.85% and 39.27% of those of RWS and HDS respectively. This is because SCAN avoids repeated feasibility investigation (*by Optimization Strategy 1*) and repeated selection probability calculation (*by Optimization Strategy 2*).

Summary: Based on the above experimental results and analysis, we conclude that our proposed SCAN outperforms all the existing methods significantly in both *execution time* and *the quality of the extracted trust networks*. Therefore, SCAN is an efficient and effective algorithm for the trust network discovery with QoTN constrains in complex contextual social networks.

4.5 The Proposed H-SCAN for Trust Network Extraction

4.5.1 K-Best-First Search (KBFS)

K-Best-First Search (KBFS) algorithm [34] is based on the Best-First Search method, which expands up to the best K nodes in the $OpenSet$ (i.e., a set to store all candidates) in each cycle of node expansion. KBFS is one of the best heuristic algorithms with good efficiency for solving NP-complete problems, such as the Number Partitioning problem and n-Puzzle problem [6, 34]. The K best nodes selection method can also be used in node selection in trust network extraction. However, KBFS is not directly designed for the trust network extraction problem. During search, for example, KBFS may (1) access a node with $deg^+ = 0$, and (2) repeatedly investigate the feasibility of an expansion node's neighboring nodes.

4.5.2 Algorithm Description of H-SCAN

Based on KBFS, we propose a Heuristic algorithm for Social Context-Aware trust Network (H-SCAN) selection problem. In H-SCAN, initially, the source participant v_s is regarded as the current expansion node, and H-SCAN searches all the neighboring nodes of v_s (denoted as $v_s.neighbors$) to investigate whether the current node and its corresponding links satisfy the QoTN constraints. If all QoTN constraints can be satisfied, the neighboring node is called *a feasible node* (denoted as v_f). Because the larger the outdegree of a node, the more likely for the node to have a connection with others [2], H-SCAN calculates the *selection probability* of each the feasible nodes v_f (i.e., $SCP(v_f \rightarrow v_t)$ in Eq. 4.6).

After that, H-SCAN selects up to K feasible neighboring nodes which have the K maximum selection probabilities, as the next expansion nodes (i.e., v_{exp}). Finally, H-SCAN repeats the above search process at each v_{exp} until it reaches the threshold of search hops (i.e., λ_h, on average $\lambda_h \leq 7$ due to the *small-world* phenomenon of social networks. During the search process, we adopt the following three optimization strategies to improve the efficiency of H-SCAN.

Optimization Strategy 1: Avoid Investigating the Intermediate Nodes with $deg^- > 0$ and $deg^+ = 0$. According to the *power-law* characteristic of social networks, most of the nodes in a social network have a small outdegree [103]. Therefore, there can be many nodes with $deg^- > 0$ and $deg^+ = 0$. During KBFS search, a node (denoted as v_x, $v_x \neq v_t$) with $deg^+(v_x) = 0$ and $deg^-(v_x) > 0$ (e.g., v_x in Fig. 4.10) may be investigated as a candidate of expansion nodes. As there is no social trust path linking v_x and v_t, the investigation of such a node may lead to low efficiency. To avoid this problem, H-SCAN does not investigate the feasibility of v_x and does not select such a node as an expansion node in the subsequent search. This strategy improves the efficiency and effectiveness of the trust network discovery.

Optimization Strategy 2: Avoid Repeatedly Accessing the Neighboring Nodes of An Expansion Node. During KBFS search, an expansion node v_{exp} may be se-

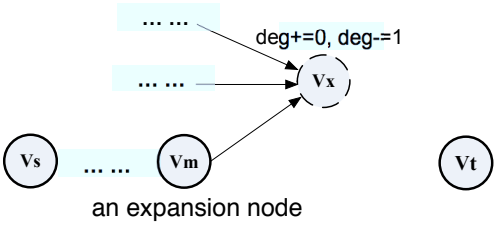

Figure 4.10: A case of accessing the node with $deg^+ = 0$

lected more than once in different search hops (e.g., $v_{exp} = v_c$ in Fig. 4.11). In such a situation, the feasibility of v_{exp}'s neighboring nodes will be repeatedly investigated (e.g., v_d in Fig. 4.11), leading to low efficiency. To address this issue, upon reaching the same v_{exp} (e.g., v_b) in different search hops, H-SCAN does not investigate its neighboring nodes repeatedly as all of them have been visited in previous search, thus saving execution time.

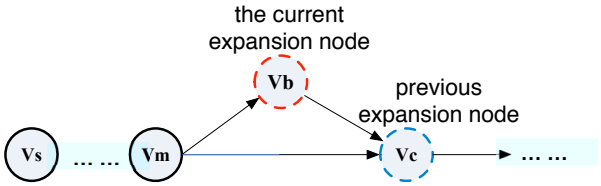

Figure 4.11: Repeated selecting the same expansion node in different search steps

4.5.3 The Process of H-SCAN

Given a group of QoTN constraints, and a pair of v_s and v_t in a complex contextual social network, the process of H-SCAN includes the following steps. The pseudo-code of H-SCAN is given in Algorithm 4.5.

Initialization: At each node v_k, set $v_k.expansion = 0$, which indicates v_k's neighboring nodes has not been selected as an expansion node. In addition, set two sets to record up to K current expansion nodes (i.e., $ClosedSet$ [116]) and all the candidates for the search of the next hop (i.e., $OpenSet$ [116]), respectively. Furthermore, put v_s

Algorithm 1: H-SCAN

```
   Data: MT(v_s, v_t), v_s, v_t, QoTN^η_{v_s,v_t}, K, λ_h
   Result: TN(v_s, v_t)
1  begin
2      Set v_k.expansion = 0, v_k.preNumber = 0, hops = 1, ClosedSet = v_s, OpenSet = ∅;
3      while hops ≤ λ_h do
4          for each v_i ∈ ClosedSet do
5              v_i.expansion = 1;
6              for each v_j ∈ adj[v_x] do
7                  if QoTN^η_{v_i,v_j} can be satisfied and v_j.expansion = 1 then
8                      v_j.preNumber = v_j.preNumber + 1;
9                      m = v_j.preNumber;
10                     v_j.preNode(m) = v_i;
                   end
11                 if QoTN^η_{v_i,v_j} can be satisfied and v_j.expansion = 0 then
12                     if v_j = v_t then
13                         v_j.preNumber = v_j.preNumber + 1;
14                         m = v_j.preNumber;
15                         v_j.preNode(m) = v_i;
                       end
16                     else if deg^+(v_j) > 0 and v_j ≠ v_t then
17                         v_j.preNumber = v_j.preNumber + 1;
18                         m = v_j.preNumber;
19                         v_j.preNode(m) = v_i;
20                         Put v_j into OpenSet;
                       end
                   end
               end
           end
21         ClosedSet = up to the K Maximal SCP(v_j → v_t) , v_j ∈ OpenSet;
22         hops = hops + 1;
       end
23     if v_t.preNumber > 0 then
24         Put v_t into preset;
25         while preSet ≠ ∅ do
26             for each v_m ∈ preSet do
27                 Delete v_m from preSet;
28                 for i = 1 to v_m.preNumber do
29                     Add v_m, v_m.preNode(i) and v_m.preNode(i) → v_m into TN(v_s, v_t);
30                     if v_m.preNode(i) ≠ v_s then
31                         Put v_m.preNode(i) into preSet;
                       end
                   end
               end
           end
       end
   end
```

into $ClosedSet$ and set the number of current search hop as one (denoted as $hops = 1$) *(lines 1-2 in Algorithm 1).*

Step 1: If there exists any $v_i.expansion = 0,\ (v_i \in ClosedSet)$ and $hops \leq \lambda_h$, get v_i from $ClosedSet$, and mark $v_i.expansion = 1$. Otherwise, go to *Step 3 (lines 3-5 in Algorithm 1).*

Step 2: Investigates v_j, $(v_j \in v_i.neighbors)$. If v_j is a feasible node, based on the

value of $v_j.expansion$, H-SCAN performs the following search strategies.

Situation 1: If $v_j.expansion = 1$ and the corresponding QoTN constraints can be satisfied, then the number of the preceding nodes of v_j increases by 1 (denoted as $v_j.preNumber = v_j.preNumber + 1$). In addition, let $m = v_j.preNumber$. At v_j, store the m^{th} preceding node of v_j (denoted as $v_j.preNode(m) = v_i$) *(lines 6-10 in Algorithm 1)*.

Situation 2: If $v_j.expansion = 0$ and the corresponding QoTN constraints can be satisfied, based on the status of v_j, H-SCAN performs the following search strategies.

Case 1: If $v_j = v_t$, then the number of the preceding nodes of v_j increases by 1 (i.e., $v_j.preNumber = v_j.preNumber + 1$). In addition, let $m = v_j.preNumber$, at v_j, store the m^{th} preceding node of v_j (denoted as $v_j.preNode(m) = v_i$) *(lines 11-15 in Algorithm 1)* .

Case 2: If $v_j \neq v_t$ and $deg^+(v_j) > 0$, then the number of the preceding nodes of v_j is increased 1 (i.e., $v_j.preNumber = v_j.preNumber + 1$). In addition, let $m = v_j.preNumber$, at v_j, store the m^{th} preceding node of v_j (i.e., $v_j.preNode(m) = v_i$). Finally, put v_j into $OpenSet$ *(lines 16-20 in Algorithm 1)*.

Step 3: Set $hops = hops + 1$. If $hops <= \lambda_h$ and $OpenSet \neq \emptyset$, select up to K candidate (denoted as $v_{cand}(K)$) which have the K maximal $SCP(v_{cand}(K) \rightarrow v_t)$ from $OpenSet$ and put them into $ClosedSet$ *(lines 21-22 in Algorithm 1)*.

Step 4: If $v_t.preNumber > 0$, construct the trust network from v_s to v_t (denoted as $TN(v_s, v_t)$) by searching the preceding nodes from v_t layer by layer until reaching v_s. Otherwise, it fails to return a trust network that satisfies QoTN constraints *(lines 23-31 in Algorithm 1)*.

H-SCAN contains two parts, i.e., part 1: network search (*Step 1* to *Step 3*) and part 2: trust network construction (*Step 4*). At each search hop, H-SCAN selects up

to K expansion nodes; therefore, the time complexity of H-SCAN in network search (i.e., the first part) is $O(Km\lambda_h)$, where K is the number expansion nodes selected at each search hop; m is the maximal outdegree of the nodes in a social network, and λ_h is the maximal search hops. In addition, in the worst case, there are $K(\lambda - 1)$ intermediate nodes and each intermediate node has K preceding nodes. Then the time complexity of H-SCAN in trust network construction (i.e., the second part) is $O(K^2\lambda_h)$. In social networks, on average, $\lambda_h < 7$ [101], and $K \leq m$. Thus the time complexity of H-SCAN is $O(Km)$, which is better than TTL-BFS (Time To Live based Breadth First Search) with the time complexity of $O(m^{TTL})$, and the same as both RWS (Random Walk Search) and HDS (High Degree Search). Since H-SCAN considers the social contextual impact factors and adopts our proposed optimization strategies, it can deliver higher utility and consume less execution time than TTL-BFS, HDS and RWS.

4.6 The Proposed H-SCAN-K for Trust Network Extraction

4.6.1 Drawbacks of H-SCAN

H-SCAN considers social contexts and the constraints specified by a source, and thus it can greatly improve the effectiveness in trust network extraction. However, H-SCAN still has some drawbacks that can lead to losing important nodes and links in trust network extraction.

At each search step, if there are more than K neighboring nodes, H-SCAN selects fixed K neighbors (e.g., v_1 to v_k in Case 1 and Case 2 in Fig. 4.12) with top K selection probabilities. This strategy (1) may neglect some *marginal nodes* that also have high likelihood to connect to the target (denoted as v_{mg}^{+}) (e.g., the selection probability of v_k is 0.81 and that of v_{k+1} is 0.8 in *Case 1* in Fig. 4.12, then $v_{mg}^{+} = v_{k+1}$), and (2) may select some marginal nodes that have low likelihood to connect to the target (denoted

as v_{mg}^{-}) (e.g., the selection probability of v_{k-1} is 0.8 and that of v_k is 0.3 in *Case 2* in Fig. 4.12, then $v_{mg}^{-} = v_k$).

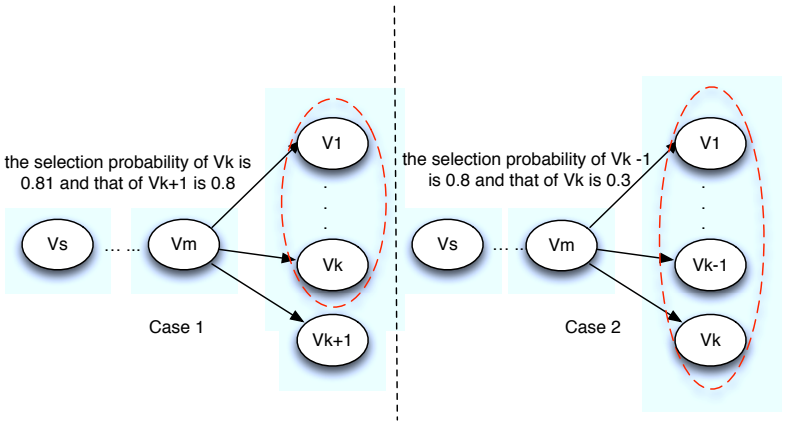

Figure 4.12: Drawbacks in H-SCAN

To address the above drawbacks included in KBFS and H-SCAN, we propose four optimization strategies and two heuristic search strategies in the following subsections. Then, based on these strategies, we propose a Heuristic Social Context-Aware trust Network extraction algorithm (H-SCAN-K), where constraints and the social context similarity between participants are considered.

4.6.2 Algorithm Description of H-SCAN-K

In H-SCAN-K, initially, the source participant v_s is regarded as the current expansion node, and H-SCAN-K searches all the neighboring nodes of v_s (denoted as $NE(v_s)$) to investigate whether the current node and its corresponding links satisfy the QoTN constraints. If all QoTN constraints can be satisfied, the neighboring node is called *a feasible node* (denoted as v_f). The larger the outdegree of a node, the more likely for the node to have a social connection with others [2]. Thus, H-SCAN-K calculates the *selection probability* of each of the feasible nodes v_f in a specified domain (denoted as $SCP_{v_f,v_t}^{D_i}$) based on $Smi_{v_f,v_t}^{D_i}$ and the outdegree of v_f (i.e., $deg^{+}(v_f)$) by Eq. (4.8).

$$SCP^{D_i}_{v_f,v_t} = Smi^{D_i}_{v_f,v_t} \cdot \frac{deg^+(v_f)}{MAX(deg^+)} \tag{4.8}$$

where $MAX(deg^+)$ is the maximal outdegree of all nodes in a social network.

After that, based on $SCP^{D_i}_{v_f,v_t}$, in addition to the two optimization strategies proposed in H-SCAN, H-SCAN-K selects the next expansion nodes (i.e., v_{exp}) based on two *Optimization Strategies* and two *Heuristic Search Strategies* discussed below. Finally, H-SCAN-K repeats the above search process at each v_{exp} until it reaches the threshold of search hops (i.e., λ_h, on average $\lambda_h \leq 7$ conforming to the small-world phenomenon). These strategies can improve the effectiveness and efficiency of trust network extraction. This has been validated in our experiments (see experiments in Section 4.7).

Optimization Strategy 1: Avoid Selecting Marginal Nodes v^-_{mg}. In H-SCAN, if the number of the neighboring nodes of v_{exp} is greater than K, the fixed K nodes with top K selection probabilities will be selected as the expansion nodes for the subsequent search (e.g., $K = 3$ in Fig. 4.13). However, as we discussed in *Section 4.5*, there may be some *marginal nodes* (i.e., v^-_{mg}) among the selected K nodes (suppose the number of margin nodes v^-_{mg} is $n \in [1, K-1]$) whose selection probabilities are less than that of the $(K-n)^{th}$ node (denoted as $v_{(K-n)^{th}}$) minus a small value (denoted as gap). i.e., $SCP^{D_i}_{v_{(K-n)^{th}},v_t} - SCP^{D_i}_{v^-_{mg},v_t} > gap$. That is, although these nodes have been selected as part of the K expansion nodes, they have much lower likelihood to connect to the target than other expansion nodes (e.g., v_3 in Fig. 4.13). Extracting the trust network via these nodes can lead to low effectiveness. To avoid this problem, H-SCAN-K investigates K' ($0 < K' < K$) nodes. If there exist K'' ($K'' \leq K'$) v^-_{mg} nodes in all the neighbors of an expansion node (e.g., $K' = K'' = 1$ in Fig. 4.13), then only $K - K''$ nodes will be selected as the expansion nodes in the subsequent search (e.g., only v_1 and v_2 are in the $ClosedSet$ (a set stores all the expansion nodes) of H-SCAN-K in Fig. 4.13).

Optimization Strategy 2: Avoid Neglecting Marginal Nodes v^+_{mg}. In H-SCAN,

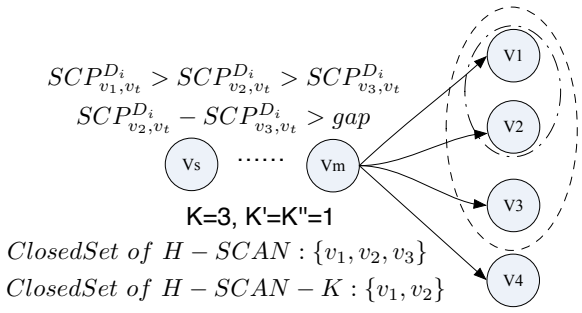

Figure 4.13: v^-_{mg} nodes

there may be some *marginal nodes* (i.e., v^+_{mg}) whose selection probabilities are greater than that of the K^{th} selected expansion node (denoted as $v_{K^{th}}$) minus gap (e.g., v_8 in Fig. 4.14 is a marginal node v^+_{mg}). i.e. $SCP^{D_i}_{v_K,v_t} - SCP^{D_i}_{v^+_{mg},v_t} < gap$. Although these nodes are not included in the K selected expansion nodes, they still have a high likelihood to connect to the target (e.g., v_8 in Fig. 4.14). However, these nodes are neglected by H-SCAN, leading to low effectiveness in trust network extraction. To avoid this problem, in H-SCAN-K, suppose there are n^* $(n^* > K)$ candidates in the search of the current layer (e.g., $n^* = 4$ in Fig. 4.14), in addition to the K best nodes, H-SCAN-K investigates the selection probabilities of another K^* nodes ($K + K^* \leq n*$). If there exist K^{**} $(K^{**} \leq K^*)$ v^+_{mg} nodes in the search of a layer (e.g., $K^* = K^{**} = 1$ in Fig. 4.14), then $K + K^{**}$ nodes will be selected as the expansion nodes in the subsequent search of the next layer (e.g., all v_5, v_6, v_7 and v_8 are in the $ClosedSet$ of H-SCAN-K in Fig. 4.14).

In addition to the above optimization strategies, we propose two heuristic search strategies and adopt them into a bidirectional search from v_s and v_t respectively to extract the trust networks.

Heuristic Strategy 1 (Forward Search From v_s To v_t): The *Forward Search* procedure extracts the trust network by searching the nodes that have social connection with v_s and have high likelihood to connect to v_t. In forward search, at each search step, H-SCAN-K computes the social context similarity between each neighboring

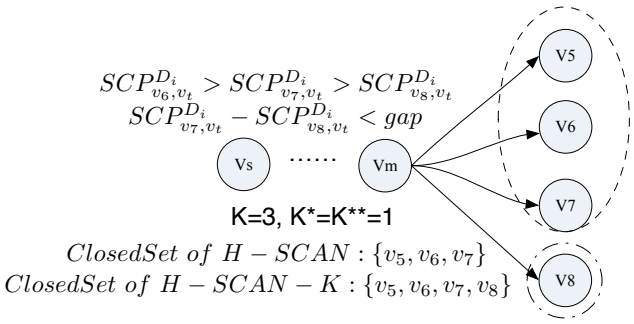

Figure 4.14: v^+_{mg} nodes

node and v_t (no direct link with v_t) based on Eq. (4.1), and calculates the corresponding selection probability based on Eq. (4.8). Then H-SCAN-K selects up to $K + K^{**}$ (or $K - K^{''}$) expansion nodes based on *Optimization Strategies 1 to 4*. During this process, if a neighboring node has been selected by the following *Heuristic Strategy 2* (i.e., the node has social connection with v_t), the node will be regarded as an expansion node for subsequent search and the link from the current expansion node to the selected neighboring node is added into the extracted trust network. H-SCAN-K will then continue the above search process until the number of search hops reaches six.

Heuristic Strategy 2 (Backward Search From v_t To v_s): The *Backward Search* procedure extracts the trust network by searching the nodes which have social connections with v_t and have a high likelihood to connect to v_s. At each search step, H-SCAN-K computes the social context similarity between v_s and those nodes that have social connections with v_t based on Eq. (4.1), and calculates the corresponding selection probability based on Eq. (4.8). Then H-SCAN-K selects up to $K + K^{**}$ (or $K - K^{''}$) expansion nodes based on the *Optimization Strategies 1 to 4*. During this process, if a neighboring node has been selected by the above *Heuristic Strategy 1* (i.e., the node has social connection with v_s), the node will be regarded as an expansion node for subsequent search and the link from the pre-visited neighboring node to the current expansion node will be added into the extracted trust network. H-SCAN-K will then continue the above search until the number of the search hops reaches six.

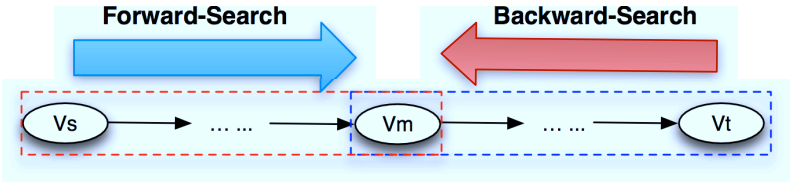

Figure 4.15: The property of bidirectional search

Our bidirectional search strategy is based on the social context similarity between an intermediate node and v_t, and the node and v_s (e.g., v_m in Fig. 4.15), which tends to link those intermediate nodes which connect to both v_s and v_t (e.g., v_m), which can improve the efficiency and effectiveness of trust network extraction (see experiments in Section 4.7)

4.6.3 The Process of H-SCAN-K

Given a group of QoTN constraints, and a pair of v_s and v_t in a large-scale and complex contextual trust-oriented social network, the process of H-SCAN-K includes the following steps. The pseudo-code of H-SCAN-K is given in *Algorithm 2* to *Algorithm 4* (the notations used in the algorithm are explained in the *Appendix*).

Algorithm 2: H-SCAN-K

Data: $v_s, v_t, QoTN^{\eta}_{v_s,v_t}, K, K^*, K', gap$
Result: $TN(v_s, v_t)$
begin
  $v.fvisit = 0, v.bvisit = 0, hops = 0\ \lambda_h = 6$;
  $f - ClosedSet = v_s, b - ClosedSet = v_t$;
  while $hos <= \lambda_h$ **do**
    Forward-Search;
    Backward-Search;
    $hops = hops + 1$;
  end
  establish the trust network based on $f - TN$ and $b - TN$;
end

Initialization: At each node v_k, set $v_k.fvisit = 0$ and $v_k.bvisit = 0$, which indicates that v_k has not been selected as an expansion node in *Forward Search* and *Backward Search* respectively. In addition, set two sets to record the current expansion

Algorithm 3: Forward-Search

```
1  begin
       Set f − OpenSet = ∅;
2      for each v_i ∈ f − ClosedSet do
3          for each v_j ∈ NE(v_i) do
4              if QoTN^η_{v_i,v_j} can be satisfied then
5                  if v_j.fvisit == 1 || v_j == v_t then
6                      add v_j and v_i → v_j into f − TN;
                   end
7                  else if v_j.bvisit = 1 then
8                      add v_j and v_i → v_j into f − TN;
9                      put v_j into f − ClosedSet;
10                     v_j.fvisit = 1;
11                     else if v_j.bvisit = 0 then
12                         add v_j and v_i → v_j into f − TN;
13                         put v_j into f − OpenSet;
14                         v_j.fvisit = 1;
                       end
                   end
               end
           end
       end
15     if f − ClosedSet ≠ ∅ then
16         select K + K** (or K − K'') expansion nodes from f − ClosedSet;
       end
17     else
18         stop Forward-Search;
       end
19     return f − TN;
   end
```

nodes in forward search and backward search respectively (i.e., $f - ClosedSet$ and $b - ClosedSet$) to record all the candidates of expansion nodes for subsequent search (i.e., $f - OpenSet$ and $b - OpenSet$). Furthermore, set the maximal search hop λ_h as 6, put v_s into $f - ClosedSet$ and v_t into $b - ClosedSet$, then set the number of the current search hop as one *(lines 1-3 in Algorithm 2)*.

Step 1 (Forward-Search): If there exists any $v_i.fivisit = 0$ ($v_i \in f - ClosedSet$), get v_i from $f - ClosedSet$; otherwise, terminate *Forward-Search* and return the extracted trust network (denoted as $f - TN$) *(lines 1-2 in Algorithm 3)*.

Step 2: Investigate v_j, ($v_j \in NE(v_i)$). If v_j is a feasible node, based on the status of v_j, H-SCAN-K performs the following search strategies *(lines 3-5 in Algorithm 3)*

Situation 1: If $v_j.fvisit = 1$ or $v_j = v_t$, add v_j and $v_i \rightarrow v_j$, into $f - TN$ *(lines 6-7 in Algorithm 3)*.

Situation 2: If $v_j.fvisit = 0$ and $v_j.bvisit = 1$, add v_j and $v_i \rightarrow v_j$, into

Algorithm 4: Backward-Search

```
1  begin
       Set b − OpenSet = ∅;
2      for each v_b ∈ b − ClosedSet do
3          for each v_a ∈ PreNE(v_b) do
4              if QoTN^η_{v_a,v_b} can be satisfied then
5                  if v_a.bvisit == 1 || v_a == v_s then
6                      add v_a and v_a → v_b into b − TN;
                   end
7                  else if v_a.fvisit = 1 then
8                      add v_a and v_a → v_b into b − TN;
9                      put v_a into b − ClosedSet;
10                     v_a.bvisit = 1;
11                     else if v_a.fvisit = 0 then
12                         add v_a and v_a → v_b into b − TN;
13                         put v_a into b − OpenSet;
14                         v_a.bvisit = 1;
                       end
                   end
               end
           end
       end
15     if b − ClosedSet ≠ ∅ then
16         Select K + K** (or K − K′′) expansion nodes from b − ClosedSet;
       end
17     else
18         Stop Backward-Search;
       end
19     return b − TN;
   end
```

$f - TN$, and put v_j into $f - ClostedSet$. Then set $v_j.fvisit = 1$ *(lines 8-10 in Algorithm 3)*.

Situation 3: If $v_j.fvisit = 0$ and $v_j.bvisit = 0$, add v_j and $v_i \rightarrow v_j$, into $f - TN$, and put v_j into $f - OpenSet$. Then set $v_j.fvisit = 1$ *(lines 11-14 in Algorithm 3)*.

Step 3 : Select $K + K^{**}$ (or $K - K''$) expansion nodes from $f - OpenSet$; put them into $f - ClosedSet$ respectively; and return $f - TN$. *(lines 15-19 in Algorithm 4)*.

Step 4 (Backward-Search): If there exists any $v_b.bvisit = 0$ ($v_a \in b{-}ClosedSet$), get v_b from $b - ClosedSet$; otherwise, terminate *Backward-Search* and return the extracted trust network (denoted as $b - TN$) *(lines 1-2 in Algorithm 4)*.

Step 5: Investigate v_a, which has direct link to v_b (i.e., $v_a \rightarrow v_b$, denoted as $v_a \in PreNE(v_b)$). If v_a is a feasible node, based on the status of v_a, H-SCAN-K performs

the following search strategies *(lines 3-5 in Algorithm 4)*.

Situation 1: If $v_a.bvisit = 1$ or $v_a = v_s$, add v_a and $v_a \rightarrow v_b$, into $b-TN$ *(lines 6-7 in Algorithm 4)*.

Situation 2: If $v_a.fvisit = 0$ and $v_a.fvisit = 1$, add v_a and $v_a \rightarrow v_b$, into $b - TN$, and put v_a into $f - ClostedSet$. Then set $v_a.bvisit = 1$ *(lines 8-10 in Algorithm 4)*.

Situation 3: If $v_a.fvisit = 0$ and $v_a.bvisit = 0$, add v_a and $v_a \rightarrow v_a$, into $b - TN$, and put v_a into $b - OpenSet$. Then set $v_a.bvisit = 1$ *(lines 11-14 in Algorithm 4)*.

Step 6 : Select $K + K^{**}$ (or $K - K''$) expansion nodes from $b - OpenSet$; put them into $b - ClosedSet$ respectively; return $b - TN$. *(lines 15-19 in Algorithm 4)*.

Step 7: Set $hops = hops + 1$. If both *Forward-Search* and*Backward-Search* terminate, H-SCAN-K establish the trust network based on $f - TN$ and $b - TN$ by searching the preceding nodes from v_t and succeeded nodes from v_s layer by layer respectively. *(lines 7-8 in Algorithm 2)*.

H-SCAN-K performs a bidirectional search strategy, i.e., *Forwards Search* and *Backward Search*. Each search procedure can be divided into two parts, i.e., *trust network search* (*Step 1* to *Step 6*) and *trust network construction* (*Step 7*). Let $m = MAX(deg^+)$. In the worst case, at each search hop, H-SCAN-K selects up to $K+K^{**}$ expansion nodes; therefore, the time complexity of trust network search (i.e., the first part) is $O((K + K^{**})m\lambda_h)$, where K is the number expansion nodes selected at each search hop; K^{**} is the number of v^+_{mg}; λ_h is the maximal search hops. In addition, in the worst case, there are $(K + K^{**})(\lambda - 1)$ intermediate nodes and each intermediate node has $K + K^{**}$ preceding nodes. Then the time complexity of H-SCAN-K in trust network construction (i.e., the second part) is $O((K+K^{**})\lambda_h)$. In social networks, on average, $\lambda_h < 7$ [101], and $(K + K^{**}) \leq m$. Thus the time complexity of H-SCAN-K is $O(m^2)$, which is better than TTL-BFS (Time To Live based Breadth First Search)

with the exponential time complexity of $O(m^{TTL})$, and it is the same as RWS (Random Walk Search), HDS (High Degree Search) and our previous H-SCAN. Since H-SCAN-K adopts our proposed optimization strategies and novel heuristic search strategies in a bidirectional search, it can deliver results with higher utility and consume less execution time than each of TTL-BFS, HDS, RWS and H-SCAN. The experimental results illustrate the significant performance difference.

4.7 Experiments on H-SCAN and H-SCAN-K

The objective of the experiments is to compare the performance difference between our proposed H-SCAN-K and the existing methods on two real datasets of social networks.

4.7.1 Datasets

We select two real datasets of social networks to conduct experiments. They are *Enron* email dataset (cs.cmu.edu/enron/) and *Epinions* dataset (trustlet.org), whose social network structures are formed based on different applications.

4.7.1.1 Enron Email Dataset

The social network based on the *Enron* email dataset is formed by sending and receiving emails, and it has been proved to possess the small-world and power-law characteristics of social networks. This dataset has in fact been widely used in the studies of social networks [79, 82, 96, 117]. Thus, we select the *Enron* email dataset with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) for our experiments.

4.7.1.2 Epinions Dataset

Epinions (epinions.com) is an online website that provides reviews of products, where participants could specify the trust relations between each other based on the quality of

Table 4.4: The settings of QoTN constraints

ID	QoTN(T)	QoTN(SI)	QoTN(PS)	QoTN(RLD)	QoTN(CIF)
1	0.2	0.2	0.1	0.1	0.1
2	0.1	0.1	0.2	0.2	0.1
3	0.1	0.1	0.1	0.2	0.2

Table 4.5: Algorithms compared in the experiments

Algorithm ID	Algorithm Name
1	Time To Live-Breadth First Search (TTL-BFS) [19]
2	High Degree Search (HDS) [2]
3	Random Walk Search (RWS) [45]
4	H-SCAN [85]
5	H-SCAN-K+HS1
6	H-SCAN-K+HS12

product reviews. The *Epinions* dataset contains a trust-oriented social network, where the trust relations specified by a truster to a trustee form each link. The *Epinions* dataset has also been proved to possess the properties of social networks [20], and has been widely used in the studies of trust in online social networks [24, 87]. Thus, we select the *Epinions* dataset available at TrustLet (trustlet.org) with 88,180 nodes (participants) and 71,7667 links (formed by trust relations specified by participants) for our experiments.

4.7.2 Experimental Setup

We randomly select 5 pairs of source nodes and target nodes from each of the two social network datasets for extracting the trust network between them. In addition, in order to have more detailed investigations on the performance of H-SCAN-K, we introduce two versions of it. The first one, denoted as H-SCAN-K+HS1, adopts our optimization strategies and *Heuristic Search Strategy 1* only (i.e., the forward search from v_s). The second one, denoted as H-SCAN-K+HS2, adopts optimization strategies and *Heuristic Search Strategies 1 and 2* (bidirectional search). Then we compare their performance with other methods including TTL-BFS, RWS, HDS and our proposed

H-SCAN. Table 4.5 lists these algorithms.

In addition, considering the small-world characteristic, we set the maximal search hops of all the algorithms to 6. Moreover, we set three groups of QoTN constraints as listed in Table 4.4. Finally, the impact factor values are generated by using the function $rand()$ in *Matlab*, which can simulate different cases in real social networks. As we introduced in *Chapter 2.3.2*, these values can be mined by using data mining methods but this is not within the scope of this thesis.

TTL-BFS, RWS, HDS, H-SCAN, H-SCAN-K+HS1 and H-SCAN-K+HS2 are implemented using Matlab R2008a running on a desktop with an Intel Core i5 CPU (2.80GHZ), 4GB RAM, Windows 7 Professional operating system and MySql 5.1.35 relational database. The results are plotted in Fig. 4.16 to Fig. 4.23, where the execution time for each of the algorithms and the utilities of the trust network extracted by RWS (as its search is based on random walks) are averaged based on 3 independent runs.

4.7.3 Results and Analysis

We use the ratio of utility to execution time (i.e., utility/execution time, termed as *performance ratio*) to illustrate the efficiency and effectiveness of trust network extraction. The larger the ratio, the better the performance of an algorithm in trust network extraction. Next, we analyse the experimental results in detail.

Result and Analysis #1: Fig. 4.16 and Fig. 4.17 plot the performance ratios of TTL-BFS method on *Enron* email dataset and *Epinions* dataset respectively. From these figures, we can see that only in a few cases (3 out of 25 cases in *Enron* email dataset, and 2 out of 25 cases in *Epinions* dataset), TTL-BFS can extract the trust networks (e.g., S1 in Fig. 4.16 and Fig. 4.17). In most of the cases (i.e., 90% of all cases), TTL-BFS cannot extract any trust networks (e.g., S2 in Fig. 4.16 and Fig. 4.17). In addition, in all the cases in which trust networks can be extracted, only in one of them (networkID=4 in *Enron* email dataset), TTL-BFS can extract the trust network

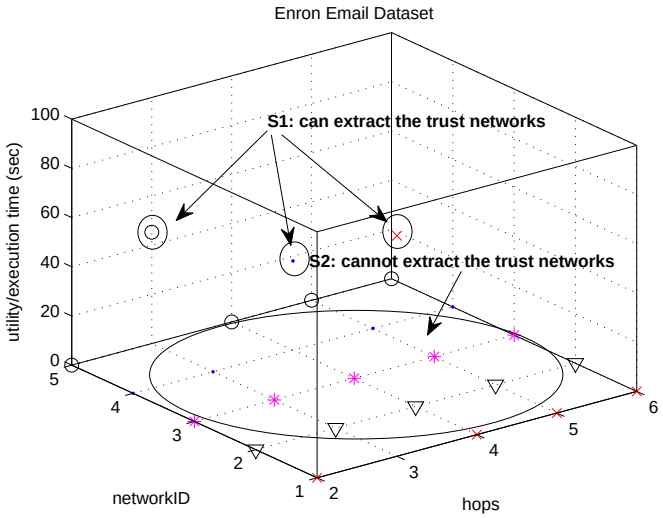

Figure 4.16: The average performance ratio delivered by TTL-BFS on Enron email dataset

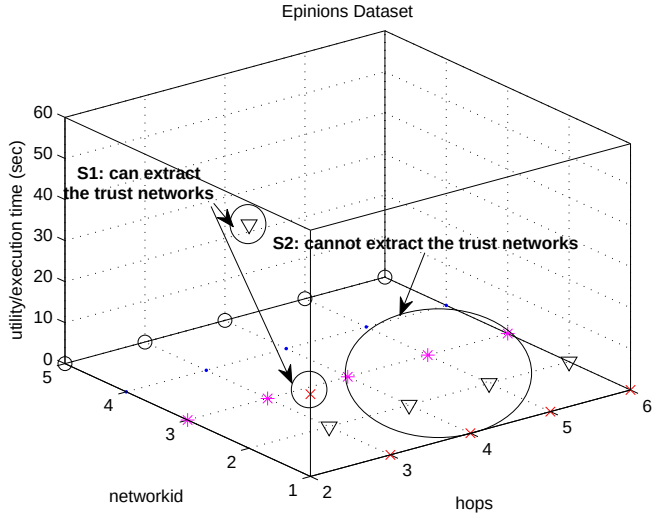

Figure 4.17: The average performance ratio delivered by TTL-BFS on Epinions dataset

by searching 4 hops from v_s. In the rest of the cases in which trust networks can be extracted, TTL-BFS can only extract the trust network by searching 2 to 3 hops from v_s. If the number of search hops is greater than 4, TTL-BFS cannot extract any trust network in all cases (even when the execution time is greater than those of the other algorithms).

This is because the time complexity of TTL-BFS is exponential in TTL (i.e., $O(m^{TTL})$). Namely, the execution time of TTL-BFS will increase exponentially with the increase of the search hops (i.e., the TTL number). Therefore, it cannot extract the social network when the maximal path length is more than 3. This can lead to losing many important nodes and links in trust network extraction, and thus, TTL-BFS is inapplicable to trust network extraction in large-scale trust-oriented social networks. So, in the following experiments, we only compare the performance ratios of HDS, RWS, H-SCAN H-SCAN-K+HS1 and H-SCAN-K+HS2, and their execution time.

Result and Analysis #2: Fig.4.18 and Fig. 4.19 plot the performance rations of the above 5 algorithms on *Enron* email dataset and *Epinions* dataset respectively. From these figures, we can see that in all cases of both datasets, the utilities of the solutions delivered by HDS are always equal to *zero* (even when HDS has more execution time than RWS, H-SCAN, H-SCAN-K+HS1 and H-SCAN-K+HS2). This is because HDS searches nodes based the descending order of their outdegrees only, without considering the likelihood of any social connection between a node and the target. This can lead to low effectiveness.

Result and Analysis #3: In addition to HDS, from Fig. 4.18 and Fig. 4.19, we can see that both our proposed H-SCAN-K+HS1 and H-SCAN-K+HS2 have larger performance ratios than HDS, RWS and H-SCAN. Table 4.6 lists their performance ratios, where we can see that on average the performance ratio of H-SCAN-K+HS1 is *3.55 times more* than that of H-SCAN, and *59.95 times more* than that of RWS. The performance ratio of H-SCAN-K+HS2 is *3.57 times more* than that of H-SCAN and *60.7 times more* than that of RWS. Namely, with the same execution time, our H-SCAN-K can extract the trust networks with much better utilities than RWS and

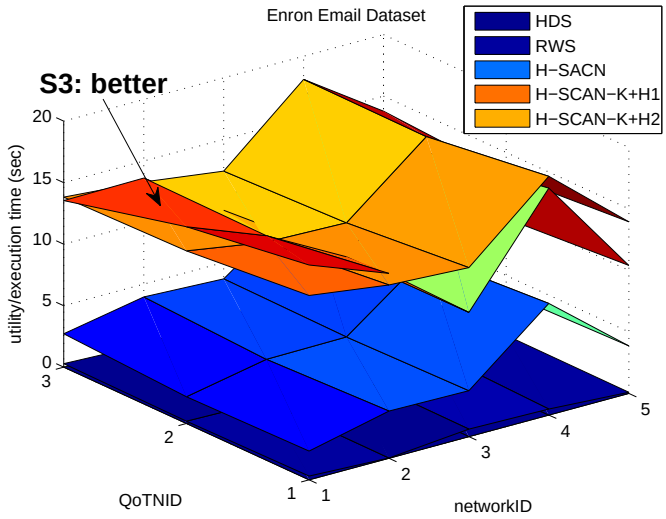

Figure 4.18: The comparison of average performance ratio on Enron email dataset

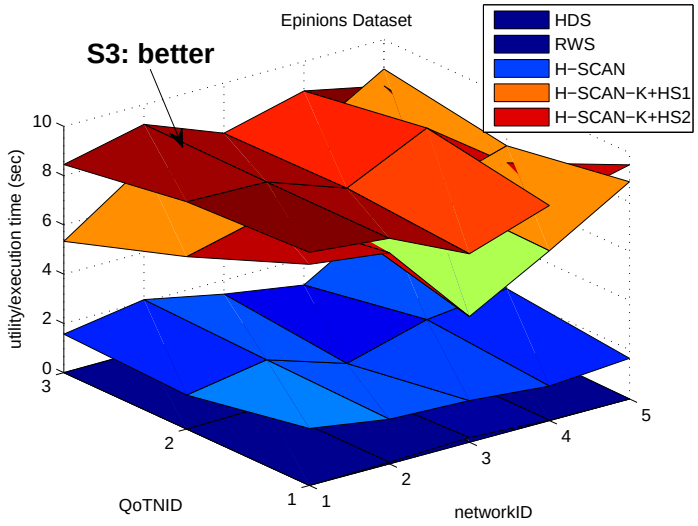

Figure 4.19: The comparison of average performance ratio on Epinions dataset

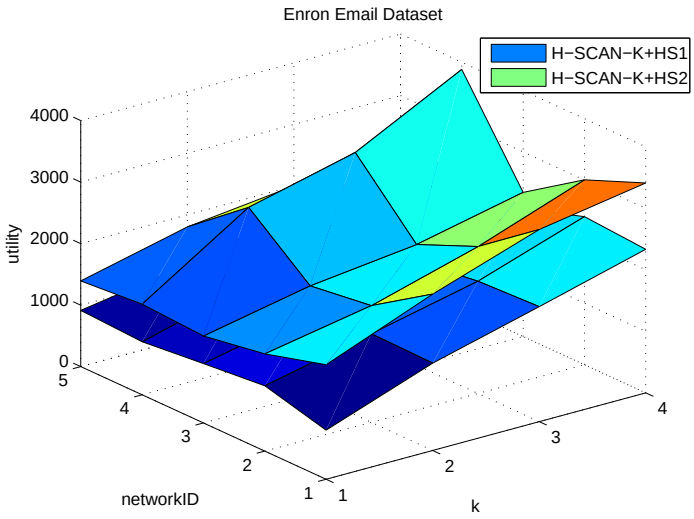

Figure 4.20: The average utilities delivered by H-SCAN-K+HS1 and H-SCAN-K+HS2 on Enron email dataset

H-SCAN.

This is because H-SCAN-K takes into account the influence of social contexts on social interactions, where the larger the likelihood for a node to have social connections with v_t and v_s, the more likely for the node to be selected in search. This method increases the probability of finding a good quality trust path from v_s to v_t. In addition, H-SCAN-K adopts the proposed optimization strategies, and thus it can (1) avoid accessing nodes with $deg^+ = 0$ and accessing the neighboring nodes of the same expansion node repeatedly (*by Strategies 1 and 2 in H-SCAN*), (2) neglect the marginal nodes (i.e., v_{mg}^-) which have a low likelihood to connect to v_t in all candidates (*by Strategy 1 in H-SCAN-K*), and (3) consider those marginal nodes (i.e., v_{mg}^+) with a high likelihood to connect to v_t in all candidates (*by Strategy 2 in H-SCAN-K*).

To sum up, H-SCAN-K greatly outperforms RWS and H-SCAN in the efficiency and the quality of the extracted trust networks. Next, we compare the performance of H-SCAN-K+HS1 and H-SCAN-K+HS2.

Result and Analysis #4: Fig. 4.20 to Fig. 4.23 plot the average utility and the average execution time for each of H-SCAN-K+HS1 and H-SCAN-K+HS2 with different K values in *Enron* email dataset and *Epinions* dataset. From these figures, we

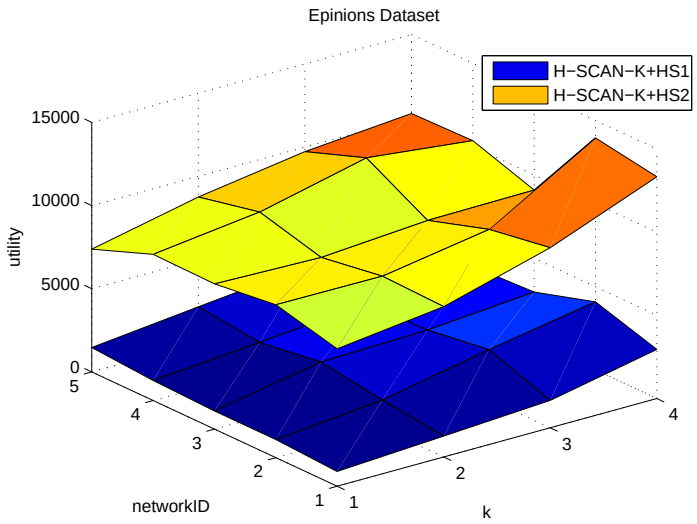

Figure 4.21: The average utilities delivered by H-SCAN-K+HS1 and H-SCAN-K+HS2 on Epinions dataset

can see that the utilities and the execution time of H-SCAN-K+HS2 in all cases are always greater than those of H-SCAN-K+HS1. This is because in addition to the forward search from v_s in H-SCAN-K+HS1, H-SCAN-K+HS2 performs the backward search from v_t to search the nodes that have social connections with v_t and high likelihood to connect to v_s. This search can deliver better utilities but consumes more execution time. Next, we compare their performance ratios.

From Fig. 4.18 and Fig. 4.19, we can see that in 19 out of 30 extracted trust networks (6 from *Enron* email dataset, 13 from *Epinions* dataset), H-SCAN-K+HS2 can deliver much larger performance ratios than H-SCAN-K+HS1 (e.g. S3 in Fig. 4.18 and Fig. 4.19). From Table 4.6, we can see that on average the performance ratio of H-SCAN-K+HS2 is *1.25% more* than that of H-SCAN-K+HS1. That is, although H-SCAN-K+HS2 consumes more execution time, it can extract trust networks with much better utilities than H-SCAN-K+HS1, and thus H-SCAN-K+HS2 is more efficient and effective than H-SCAN-K+HS1 in trust network extraction.

This is because in addition to the forward search, H-SCAN-K+HS2 considers the nodes that have a social connection with v_t and have a high likelihood to connect to v_s

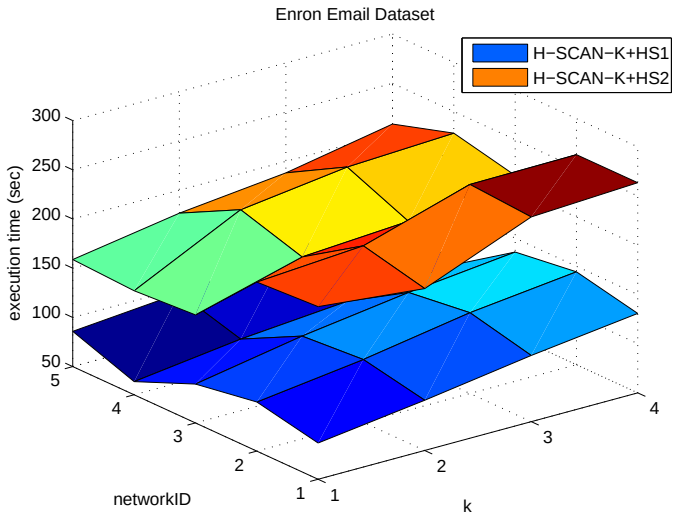

Figure 4.22: The average execution time of H-SCAN-K+HS1 and H-SCAN-K+HS2 on Enron email dataset

in backward search. This bidirectional search tends to search the intermediate nodes that have a high likelihood to connect to both v_s and v_t. If these nodes can connect to v_s during the backward search, H-SCAN-K+HS2 can deliver better utilities than H-SCAN-K+HS1. The experiments have shown that in most cases, H-SCAN-K+HS2 outperforms H-SCAN-K+H1S, and on average, it can extract more important nodes and links with higher efficiency than H-SCAN-K+HS1 (i.e., 1.25% more performance rations) in trust network extraction.

The two versions of H-SCAN-K have different characteristics. H-SCAN-K+HS1 will use less execution time but will extract a trust network with less utility than H-SCAN-K+HS2. In contrast, H-SCAN-K+HS2 is to guarantee the utility of the extracted trust network but will consume more execution time than H-SCAN-K+HS1.

4.7.4 Summary

Based on the above experimental results and analysis, we conclude that TTL-BFS and HDS are not suitable for trust network extraction in large-scale social networks due to low effectiveness and efficiency of their search strategies. Although RWS and H-

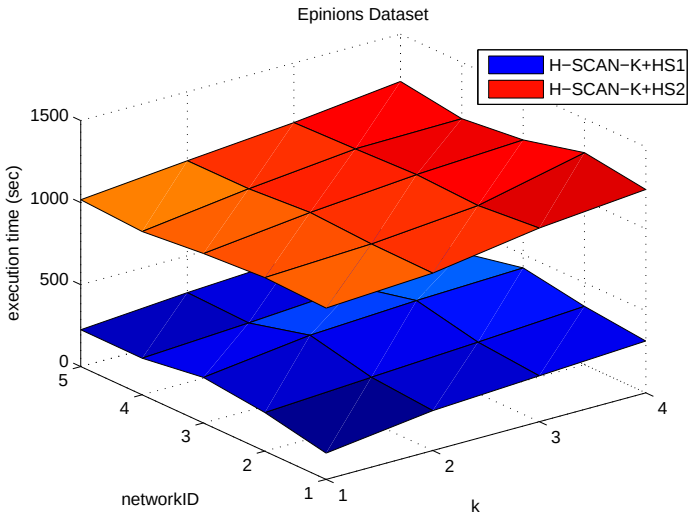

Figure 4.23: The average execution time of H-SCAN-K+HS1 and H-SCAN-K+HS2 on Epinions dataset

SCAN can be used to extract trust networks, our proposed H-SCAN-K greatly outperforms them in the *efficiency* and the *quality of the extracted trust networks*. Therefore, H-SCAN-K is an efficient and effective algorithm for trust network extraction with QoTN constraints in large-scale trust-oriented social networks. H-SCAN-K can extract high quality trust networks which provide the basis to deliver a reasonable trust evaluation result between two unknown participants.

Table 4.6: Comparison of performance ratio

Algorithm ID	The average performance ratio										
	NetworkID (Enron email dataset)					NetworkID (Epinions dataset)					Average
	1	2	3	4	5	6	7	8	9	10	
HDS	0	0	0	0	0	0	0	0	0	0	0
RWS	0.2779	0.0342	0.6539	0.5792	0.1131	0	0.0375	0.0196	0.0023	0.1067	0.1824
H-SCAN	2.4111	3.7528	3.6394	8.6094	3.7507	1.7544	1.9464	1.3436	1.3776	2.2297	3.0815
H-SCAN-K+HS1	14.2814	13.4567	**12.9856**	**18.5261**	**13.6309**	7.1036	7.9726	5.9070	6.9332	**8.5431**	10.934
H-SCAN-K+HS2	**15.6723**	**14.0537**	9.7914	17.4620	10.3775	**9.0494**	**9.1718**	**7.8810**	**9.0424**	8.2024	**11.0704**

4.8 Conclusion

In this Chapter, we have proposed a general concept QoTN (Quality of Trust Network), and a novel social context-aware trust network extraction model in large-scale trust-oriented social networks. In addition, we have proposed a new Social Context-Aware trust Network discovery algorithm (SCAN) by adopting the Monte Carlo method and our proposed optimization strategies. Furthermore, we have proposed a new Heuristic Social Context-Aware trust Network discovery algorithm, called H-SCAN, by adopting K-Best-First Search (KBFS) method and our proposed optimization strategies. Finally, we have proposed a new Heuristic Social Context-Aware trust Network extraction algorithm, called H-SCAN-K, by adopting K-Best-First Search (KBFS) method, bidirectional search (i.e., search from both the source and the target nodes simultaneously) and our proposed novel heuristic strategies. The experimental results demonstrate the superior performance of the proposed algorithms in both the quality of the extracted trust networks and the execution time.

Our proposed highly efficient and effective trust network extraction methods provide a good foundation for performing trust inference methods to deliver reasonable trust evaluation results, which can be used in the social network based recommendation systems to help find the most trustworthy recommenders.

Chapter 5

Finding the Optimal Social Trust Path

In Chapter 4 we have introduced the trust network extraction methods in large-scale trust-oriented social networks. After extracting the trust network, we can evaluate the trustworthiness of the target by using trust propagation methods to propagate the trust via the social trust paths in the trust network. However, in large-scale trust-oriented social networks, there could be tens of thousands of social trust paths between a source participant and a target one [70]. Evaluating the trustworthiness of the target participant based on all these social trust paths is very time consuming, and thus cannot be applied into real applications [6]. Alternatively, we can search the *optimal* path yielding the most trustworthy trust propagation result from multiple paths. We call this the optimal social trust path selection problem that is known to be a challenging research problem [78].

In the literature, Lin *et al*. [77] proposed 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. This method neglects *trust information* between participants. In another work [53], the path with the maximal propagated trust value is selected as the most trustworthy social trust path. In addition, Wang *et al*. [125] proposed a social trust path selection method where a source participant can specify a threshold for the aggregated trust value for a social trust path in a trust network. If the aggregated trust value of a social trust path is greater than the specified threshold, the trust path is kept for trust evaluation.

Moreover, existing trust path selection methods do not consider the social contexts,

including the social relationship between participants and the social position of participants. These social contexts have significant influence on trust propagation [3, 102].

In addition, 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 may have different social trust path selection criteria, and thus, should be able to set certain constraints on the above social context in trust path selection. However, such a capability is not supported by existing methods [53, 77].

In this chapter, we first propose a novel concept, *Quality of Trust* (QoT), and models the multiple QoT constrained optimal social trust path selection problem as a Multi-Constrained Optimal Path (MCOP) selection problem, which is proved to be NP-Complete in [67]. Then, we propose an approximation algorithm, called MONTE_K based on the Monte Carlo method, and two heuristic algorithms, called H_OSTP and MFPB-HOSTP based on the Dijkstra's shortest path algorithm and our novel search strategies. The experimental results illustrate that the proposed methods outperform the existing methods in both the quality of the identified social trust path and the execution time.

5.1 Quality of Trust (QoT) and QoT Attributes Aggregation

5.1.1 Quality of Trust (QoT)

In Service-Oriented Computing (SOC), QoS (Quality of Service) consists of a set of attributes, used to illustrate the ability of services to guarantee a certain level of performance [40]. Similar to QoS, we propose a new concept, *Quality of Trust*.

Definition 3: *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 (SI), and community impact factor (CIF), as attributes.

In service invocations, users 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, to satisfy different trust evaluation criteria, a source participant can specify multiple end-to-end constraints for QoT attributes (i.e., T, SI and CIF) as the requirements of trust propagation in a social trust path of different domains.

Let $Q^{\mu}_{v_s,v_t}$ ($\mu \in \{T,\ SI,\ CIF\}$) denote the end-to-end constraint of QoT attribute μ for the paths between v_s and v_t in a certain domain (throughout this thesis, v_s denotes the source participant and v_t denotes the target participant in a social network). For example, as shown in Fig. 1.1, to *hire employees*, A, a retailer manager specifies the end-to-end QoT constraints for the social trust paths from A to M as $Q_{A,M} = \{Q^{T}_{A,M} > 0.3,\ Q^{SI}_{A,M} > 0.3,\ Q^{CIF}_{A,M} > 0.8\}$, if he/she believes the social position of participants is more important in the domain of *employment*. But when looking for new customers for *selling products*, A could specify QoT constraints as $Q_{A,M} = \{Q^{T}_{A,M} > 0.8,\ Q^{SI}_{A,M} > 0.3,\ Q^{CIF}_{A,M} > 0.3\}$, if he/she believes the social relationships between participants are more important in the domain of *product sale*.

5.1.2 QoT Attribute Aggregation

To specify end-to-end QoT constraints, we propose the QoT attribute aggregation methods as follows.

5.1.2.1 Trust Aggregation

The trust values between a source participant and the target participant in a social path can be aggregated based on trust transitivity property (i.e., if A trusts B and B trusts C, then A trusts C to some extent) [48]. Since trust is discounted with the increase of transitivity hops [23], in our model, we adopt the strategy proposed in [73, 124], where 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 in a certain domain is calculated as in Eq. (5.1). This strategy has been widely used in the literature as a trust aggregation method [10, 83, 124].

$$T_{p(a_1,...,a_n)} = \prod_{(a_i,a_{i+1})\in p(a_1,...,a_n)} T_{a_i\ a_{i+1}} \tag{5.1}$$

This aggregated trust value will be combined with the social intimacy degree and the community impact factor in the following context to select the optimal social trust path.

5.1.2.2 Social Intimacy Degree Aggregation

Firstly, social intimacy between participants decays with the increasing number of hops between them in a social trust path [72, 102]. In addition, in the real world, the intimacy degree decays fast when its value approaches 1 (the slope approaches the minimum). In contrast, the intimacy degree decays slowly when its value approaches zero (the slope approaches the maximum) [18, 56]. Namely, the decay speed of the social intimacy degree is non-linear in social networks. The aggregated SI value in path $p(a_1, ..., a_n)$ can be calculated by Eq. (5.2) whose function image is a *hyperbolic curve*, fitting the characteristic of social intimacy attenuation [102].

$$SI_{p(a_1,...,a_n)} = \prod_{(a_i,a_{i+1})\in p(a_1,...,a_n)} SI_{a_i\ a_{i+1}} \tag{5.2}$$

5.1.2.3 Community Impact Factor Aggregation

As illustrated in social psychology [99], in the same society, the community impact factor of a participant *does not decay* with the increase of transitivity hops in a certain domain. Thus, the aggregated CIF value of $p_{(a_1,...a_n)}$ in a certain domain can be calculated by Eq. (5.3).

$$CIF_{p(a_1,...,a_n)} = \frac{\sum_{i=2}^{n-1} CIF_{a_i}}{n-2} \tag{5.3}$$

5.1.3 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, SI and CIF as the arguments in Eq. (5.4). The non-linearity of the trust and intimacy decay between two non-adjacent individuals have been considered in the computation of the utility function that is used in path selection.

$$\mathcal{F}_{p(a_1,\ldots,a_n)} = \omega_T * T_{p(a_1,\ldots,a_n)} + \omega_{SI} * SI_{p(a_1,\ldots,a_n)} + \omega_{CIF} * CIF_{p(a_1,\ldots,a_n)} \tag{5.4}$$

where ω_T, ω_{SI} and ω_{CIF} are the weights of T, SI and CIF in domain i respectively; $0 < \omega_T, \omega_{SI}, \omega_{CIF} < 1$ and $\omega_T + \omega_{SI} + \omega_{CIF} = 1$. A source participant can specify different weights for different QoT attributes in path selection. For example, if a source participant believes the social position of participants is more important in the domain of employment, he/she can specify a relative high value for ω_{CIF}. In contrast, if he/she regards the social relationship is more important, he/she can specify a relatively high value for ω_{SI}.

A *feasible path (solution)* is the social trust path that can satisfy multiple end-to-end QoT constraints. The goal of optimal social trust path selection is to select the *optimal path (solution)* that yields the best utility with the weights specified by the source participant from all *feasible paths (solutions)*.

5.2 The Proposed MONTE_K for Optimal Social Trust Path Selection

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 that is NP-Complete [67]. In this section, we first analyse some existing approximation algorithms for the MCOP selection problem and then propose an efficient

approximation algorithm, MONTE_K, based on the Monte Carlo method [43] and our optimization strategies

5.2.1 Existing Approximation Algorithms

In the literature, several approximation algorithms have been proposed for the MCOP selection problem.

Korkmaz *et al*. [67] propose a heuristic algorithm H_MCOP for the multiple-constrained optimal path selection in service invocation. In this algorithm, both multi-constrained values and QoS attributes values are aggregated based on Eq. (5.5).

$$g_\lambda(p) \triangleq (\frac{q_1(p)}{Q^1_{v_s,v_t}})^\lambda + (\frac{q_2(p)}{Q^2_{v_s,v_t}})^\lambda + ... + (\frac{q_m(p)}{Q^m_{v_s,v_t}})^\lambda \tag{5.5}$$

where $\lambda \geq 1$; $q_i(p)$ is the aggregated value of the i^{th} QoS attribute of path p (e.g., the total cost of the services in a path formed by service invocation); $Q^i_{v_s,v_t}$ is the i^{th} QoS constraint value of the selected path between v_s and v_t (e.g., $Q^{cost}_{v_s,v_t} \leq \$100$).

H_MCOP first adopts the Dijkstra's shortest path algorithm [31] to find the path with the minimum g_λ from v_t to v_s, which intends to investigate whether there exists a feasible solution satisfying all end-to-end QoS constraints in a sub-network. In this process, at each intermediate node v_k, the aggregated value of each QoS attribute for the identified path from v_k to v_t is computed and recorded. If there exists at least one feasible solution, then these aggregated values are used in another search from v_s to v_t, which intends to identify a feasible path from v_s to v_t with the minimal cost of services.

H_MCOP was one of the most promising algorithms for the MCOP selection problem as it outperformed prior existing algorithms in both algorithm efficiency and solution quality [67].

Consequently, in the field of Service-Oriented Computing (SOC), Yu *et al*. [134] propose an approximation algorithm, MCSP_K to solve the quality-driven service selection problem that is also the MCOP selection problem. This method keeps only K

paths from a source node to each intermediate node, aiming to reduce the search space and execution time. Their K-path selection is based on Eq. (5.6).

$$\xi(p) \triangleq max\{(\frac{q_1(p)}{Q^1_{v_s,v_t}}), (\frac{q_2(p)}{Q^2_{v_s,v_t}}), ..., (\frac{q_m(p)}{Q^m_{v_s,v_t}})\} \tag{5.6}$$

From Eq. (5.6), if any QoS attribute value does not satisfy the corresponding QoS constraint in path p, then $\xi(p) > 1$. In their search strategies, the paths with up to K minimum ξ values are kept at each intermediate node. This method never prunes any feasible path if it exists. In their service candidate graph, all services are categorised into different service sets based on their functionalities. 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 will become exponential. Therefore, the algorithm does not fit large-scale social networks.

Some other algorithms [137, 138] adopt integer linear programming to solve the service selection problem with multi-QoS constraints. But in [134] they have been proved to have low efficiency in finding a near-optimal solution in large-scale networks.

5.2.2 Algorithm Description of MONTE_K

In MONTE_K, we adopt the following two optimization strategies.

Optimization Strategy 1: *K-path selection*. Let v_s denote a source participant and v_t denote the target one. According to Eq. (5.6), the lower the ξ value of a path, the higher the probability for that path to be a feasible solution. Thus, given a partially identified social trust path from v_s to v_x ($v_x \neq v_t$), we calculate the ξ values of the paths from v_s to each neighboring node of v_x and record up to K neighboring nodes that yield up to K minimum ξ values as candidates for selection.

As this strategy selects no more than K neighbors at each step in social trust path

selection, it can reduce the search space and deliver higher efficiency than MCBA.

Optimization Strategy 2: *Optimization at dominating nodes.* If the indegree of v_y ($v_y \neq v_s$) is greater than 1 in the social network, then node v_y is regarded as a *dominating node*. To obtain a near-optimal solution, MONTE_K performs multiple simulations. In the first simulation, if a social trust path from v_s to v_y (denoted as path p_1) is selected, we store the utility $\mathcal{F}$, ξ value and the aggregated value of each QoT attribute of p_1 at v_y. In all subsequent simulations, if a different social path from v_s to v_y (denoted as path p_y, where $y > 1$) is selected, the optimization is performed in the following situations.

Situation 1: If $v_y = v_t$ and $\mathcal{F}(p_y) < \mathcal{F}(p_1)$, it indicates p_y is worse than p_1. Thus we replace the values of p_y (i.e., T, SI, CIF, $\mathcal{F}$ and ξ) with the one stored at v_y .

Situation 2: If $v_y = v_t$ and $\mathcal{F}(p_y) > \mathcal{F}(p_1)$, it indicates p_y is better than p_1. Thus, we store the values of p_y at v_y.

Situation 3: If $v_y \neq v_t$, $\mathcal{F}(p_y) < \mathcal{F}(p_1)$ and $\xi(p_y) > \xi(p_1)$, it indicates p_y is worse than p_1. Thus we replace the values of p_y with the one stored at v_y .

Situation 4: If $v_y \neq v_t$, $\mathcal{F}(p_y) > \mathcal{F}(p_1)$ and $\xi(p_y) < \xi(p_1)$, it indicates p_y is better than p_1. Thus, we store the values of p_y at v_y.

Following Strategy 2, the dominating node v_y records T, SI, CIF, $\mathcal{F}$ and ξ values of the locally optimal social trust path from v_s to v_y. The optimization at v_y can guarantee that the delivered solution from v_s to v_y is locally optimal.

5.2.3 The Process of MONTE_K

Initialization: Mark the status of all nodes in the network as unvisited. Add v_s into set $temp\_P$ that stores the solution (i.e., identified social trust path). Let $Min\_K(v_u)$ be a set that stores up to K neighboring nodes of node v_u (*lines 1 to 3 in Algorithm 5*).

Step 1: Get an unvisited node v_u from $temp\_P$ and mark v_u as visited. Select up to K neighboring nodes of v_u based on *strategy 1* and put these nodes into $Min\_K(v_u)$ (*lines 4 to 10 in Algorithm 5*).

Algorithm 5: MONTE_K

Data: $MT(v_s, v_t)$, QoT constraints, v_s, v_t, K-path number
Result: $\mathcal{F}$, P_{st}
/*v_s, v_t: a source and the target participants; $\mathcal{F}$: utility; P_{st}: identified social trust path; $temp\_P$: partly identified social trust path; $deg^-(v)$: the indegree of v; $\mathcal{P}(v)$: the probability of v to be selected; $adj[v_u]$: neighboring nodes of v_u; $Min\_K(v_u)$: the set stores up to K neighboring nodes of v_u; $\mathcal{F}_{old}(p_v)$, $\xi_{old}(p_v)$, $T_{old}(p_v)$, $SI_{old}(p_v)$, $CIF_{old}(p_v)$, $temp_{old}\_P(p_v)$: values stored at dominating node v; */
begin
1 **Mark** the status of all node as **unvisited**, $P_{st} = \emptyset$, $temp\_P \longleftarrow v_s$;
2 **while** *(unvisited node exists in $temp\_P$)* **do**
3 **Get** unvisited node v_u from $temp\_P$
4 **Mark** v_u as **visited**;
5 **for** *each* $v_i \in adj[v_u]$ **do**
6 **Calculate** $T(p_{v_i})$, $SI(p_{v_i})$, $CIF(p_{v_i})$ and $\xi(p_{v_i})$
end
7 **if** *size($adj[v_u]$)*$> K$ **then**
8 **Put** v_i with K minimum $\xi(p_{v_i})$ into $Min\_K(v_u)$;
end
9 **else**
10 **Put** $v_i \in adj[v_u]$ into $Min\_K(v_u)$;
end
11 **for** *each* $v_i \in Min\_K(v_u)$ **do**
12 $\mathcal{F}(p_{v_i}) = \omega_T * T(p_{v_i}) + \omega_{SI} * SI(p_{v_i}) + \omega_{CIF} * CIF(p_{v_i})$;
13 $\mathcal{P}(v_i) = \mathcal{F}(p_{v_i}) \,/\, \sum_{i=1}^{size(Min\_K(v_u))} \mathcal{F}(p_{v_i})$;
end
14 **Generate** a random number $rand \in [0, 1]$;
15 **Select** the m^{th} node v_j such that $rand \leq \sum_{i=1}^{m} \mathcal{P}(v_i)$;
16 **if** $\xi(p_{v_j}) > 1$ **then**
17 **Break**;
end
18 **else**
19 **if** $deg^-(v_j) > 1$ **then**
20 **Optimization at Dominating Nodes** ($\mathcal{F}(p_{v_j}), \xi(p_{v_j}), T(p_{v_j}), SI(p_{v_j}), CIF(p_{v_j}), temp\_P$);
end
21 **if** $v_j = v_t$ **then**
22 $\mathcal{F} = \mathcal{F}(p_{v_j})$, $P_{st} = temp\_P$;
23 **Return** $\mathcal{F}$ and P_{st};
end
24 **else**
25 $T(p_{v_j}) = T_{new}(p_{v_j})$, $SI(p_{v_j}) = SI_{new}(p_{v_j})$;
26 $CIF(p_{v_j}) = CIF_{new}(p_{v_j})$, $temp\_P = temp\_P_{new}$;
27 **Put** v_j into $temp\_P$;
end
end
end
end

Algorithm 6: Optimization at Dominating Nodes

Data: $\mathcal{F}(p_{v_j}), \xi(p_{v_j}), v, temp\_P, T(p_{v_j}), SI(p_{v_j}), CIF(p_{v_j})$
Result: $\mathcal{F}_{new}(p_{v_j}), \xi_{new}(p_{v_j}), temp\_P_{new}, T_{new}(p_{v_j}), SI_{new}(p_{v_j}), CIF_{new}(p_{v_j})$
begin
1 **Get** $(node(v_j).\mathcal{F}, \xi, T, SI, CIF, temp\_P)$ that stored at v_j;
2 **Put** these values into $(\mathcal{F}_{old}(p_{v_j}), \xi_{old}(p_{v_j}), T_{old}(p_{v_j}), SI_{old}(p_{v_j}), CIF_{old}(p_{v_j}), temp\_P_{old})$;
3 **if** $v_j \neq v_t$ **then**
4 **if** $\mathcal{F}(p_{v_j}) < \mathcal{F}_{old}(p_{v_j})$ *and* $\xi(p_{v_j}) > \xi_{old}(p_{v_j})$ **then**
5 **Put** $\{\mathcal{F}_{old}(p_{v_j}), \xi_{old}(p_{v_j}), T_{old}(p_{v_j}), SI_{old}(p_{v_j}), CIF_{old}(p_{v_j}), temp\_P_{old}\}$ **into** $\{\mathcal{F}_{new}(p_{v_j}), \xi_{new}(p_{v_j}), T_{new}(p_{v_j}), SI_{new}(p_{v_j}), CIF_{new}(p_{v_j}), temp\_P_{new}\}$
end
6 **else**
7 **if** $\mathcal{F}(p_{v_j}) > \mathcal{F}_{old}(p_{v_j})$ *and* $\xi(v) < \xi_{old}(p_{v_j})$ **then**
8 **Put** $\{\mathcal{F}(p_{v_j}), \xi(p_{v_j}), T(p_{v_j}), SI(p_{v_j}), CIF(p_{v_j}), temp\_P\}$ **into** $\{\mathcal{F}_{new}(v), \xi_{new}(v), T_{new}(v), SI_{new}(v), CIF_{new}(v), temp\_P_{new}\}$;
9 **Update**$(node(v_j).\mathcal{F}, \xi, T, SI, CIF, temp\_P)$ with these values;
end
end
end
10 **else**
11 **if** $\mathcal{F}(p_{v_j}) < \mathcal{F}_{old}(p_{v_j})$ **then**
12 **Put** $\{\mathcal{F}_{old}(p_{v_j}), \xi_{old}(p_{v_j}), T_{old}(p_{v_j}), SI_{old}(p_{v_j}), CIF_{old}(p_{v_j}), temp\_P_{old}\}$ **into** $\{\mathcal{F}_{new}(p_{v_j}), \xi_{new}(p_{v_j}), T_{new}(p_{v_j}), SI_{new}(p_{v_j}), CIF_{new}(p_{v_j}), temp\_P_{new}\}$
end
13 **else**
14 **Put** $\{\mathcal{F}(p_{v_j}), \xi(p_{v_j}), T(p_{v_j}), SI(p_{v_j}), CIF(p_{v_j}), temp\_P\}$ **into** $\{\mathcal{F}_{new}(p_{v_j}), \xi_{new}(p_{v_j}), T_{new}(p_{v_j}), SI_{new}(p_{v_j}), CIF_{new}(p_{v_j}), temp\_P_{new}\}$;
15 **Update** $(node(v_j).\mathcal{F}, \xi, T, SI, CIF, temp\_P)$ with these values;
end
end
16 **Return**$(\mathcal{F}_{new}(p_{v_j}), \xi_{new}(p_{v_j}), T_{new}(p_{v_j}), SI_{new}(p_{v_j}), CIF_{new}(p_{v_j}), temp\_P_{new})$
end

Step 2: For each $v_i \in Min\_K(v_u)$, calculate the probability of v_i for selection, based on the utility of the social trust path from v_s to v_i via v_u (denoted as path p_{v_i}). The probability of v_i to be selected is $\mathcal{P}(p_{v_i}) = \frac{\mathcal{F}(p_{v_i})}{\sum_{i=1}^{size(Min\_K(v_u))} \mathcal{F}(p_{v_i})}$ (*lines 11 to 13 in Algorithm 5*).

Step 3: Select v_j from set $Min\_K(v_u)$ based on a random number $rand \in [0, 1]$ and $\{\mathcal{P}(p_{v_i})\}$. If $\xi(p_{v_j}) \leq 1$ and the indegree of v_j is greater than 1, then v_j is a dominating node and thus performs the optimization at v_j based on *Strategy 2*. If $\xi(p_{v_j}) > 1$, it indicates that no feasible solution has been delivered in this simulation (*lines 14 to 27 in Algorithm 5 and lines 1 to 15 in Algorithm 6*).

Step 4: If $v_j \neq v_t$, add v_j into $temp\_P$ and go to *Step 1*. If $v_j = v_t$, return $temp\_P$ and $\mathcal{F}(temp\_P)$ (*line 16 in Algorithm 6*).

According to the *power-law* characteristic [103], only a few nodes have a large

outdegree in social networks (e.g., in *Enron* email corpus[1], 94.7% nodes have an outdegree less than 15). Therefore, in MONTE_K, each node can keep a small search space without pruning a large number of neighboring nodes (i.e., candidates) of a node in K-path selection, which results in high efficiency and a higher probability of finding the optimal solution. The time complexity of MONTE_K is $O(slmK)$, where s is the number of simulations; l is the average length of the shortest social trust paths from a source participant to the target one in social networks; m is the maximal outdegree of nodes in social networks and K is the argument specified for K-path selection. In social networks, usually $l < 7$ according to the *small-world* characteristic [103]. Thus the time complexity of MONTE_K is $O(smK)$. By both addressing the characteristics of social networks and adding optimization strategies in the algorithm design, MONTE_K can deliver better solutions with less execution time than existing methods.

5.3 Experiments on MONTE_K

5.3.1 Experiment Settings

The *Enron* email dataset has been proved to possess the small-world and power-law characteristics of social networks and has been widely used in the studies of social networks [49, 96, 117]. In addition, the social intimacy degree between participants and the community impact factor of participants can be calculated through mining the subjects and contents of emails [96]. Therefore, in contrast to other real social network datasets (e.g., Epinions[1] and FilmTrust), the *Enron* email dataset fits our proposed complex social network structure very well. Thus, to verify our proposed algorithm, we select the *Enron* email corpus with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) as the dataset, and conduct experiments on it.

As the complexity of MCSP_K [134] is exponential in finding an optimal social

[1] http://epinions.com/

trust path in social networks, it is ignored in our experiments. Instead, we compare MONTE_K with H_MCOP [67] and MCBA [73] in both execution time and the utilities of identified social paths. Since this thesis does not focus on detailed data mining techniques, in our experiments, the T, SI and CIF values are randomly generated. The end-to-end QoT constraints are set as $Q_{v_s,v_t} = \{Q^T_{v_s,v_t} \geq 0.05,\ Q^{SI}_{v_s,v_t} \geq 0.001,\ Q^{CIF}_{v_s,v_t} \geq 0.3\}$ and the weights of attributes in utility function are set as $\omega_T = 0.25,\ \omega_{SI} = 0.25$ and $\omega_{CIF} = 0.5$.

All three algorithms 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.

5.3.2 Results and Analysis

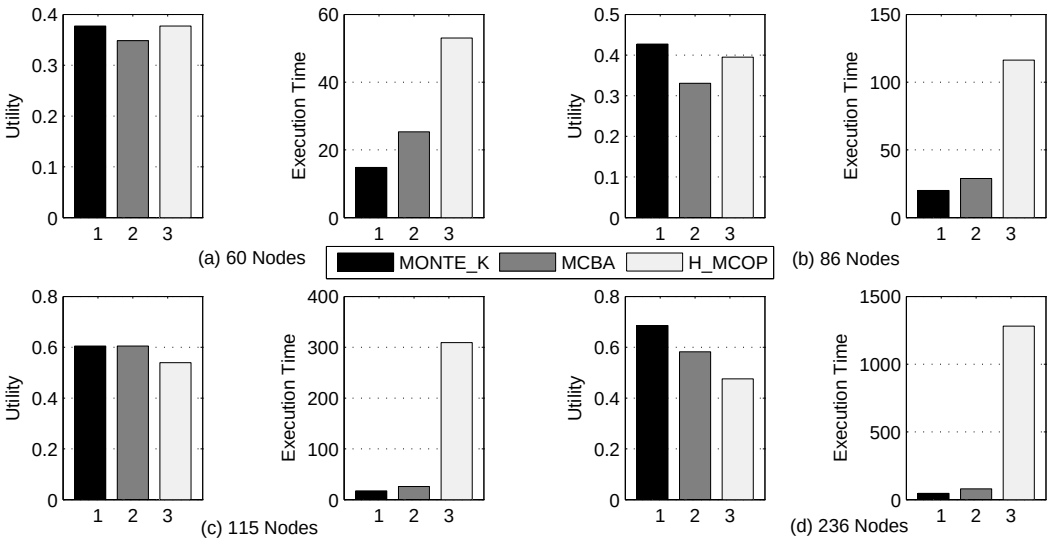

Figure 5.1: Maximal length of paths is 4 hops

In this experiment, in order to evaluate the performance of our proposed approximation algorithm in the sub-networks of different scales and structures, we first randomly select 16 pairs of source and target nodes from *Enron* email dataset. We then extract the corresponding 16 sub-networks between them by using the exhaustive search method. Among them, the maximal length of a social trust path varies from 4 to 7

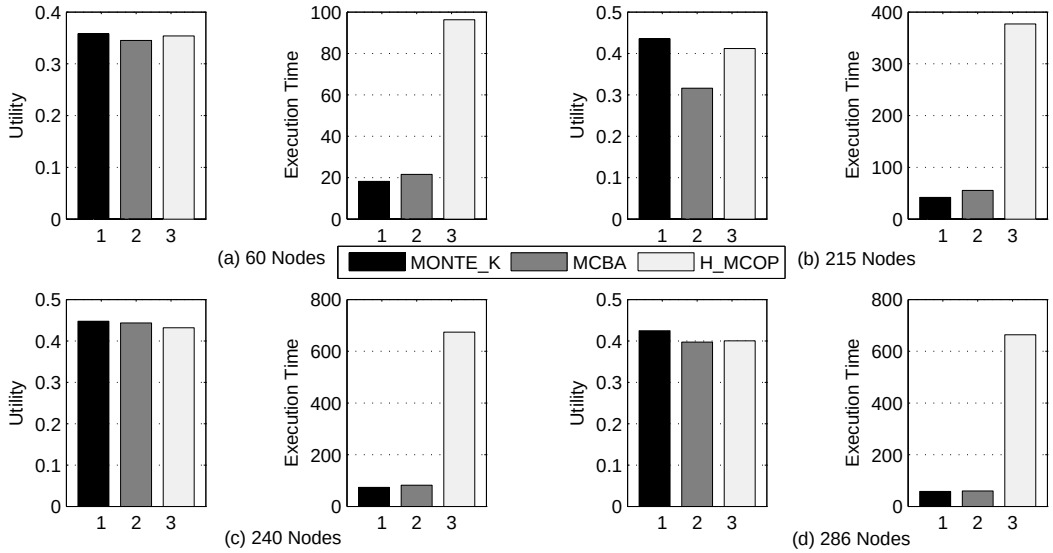

Figure 5.2: Maximal length of paths is 5 hops

hops following the *small-world* characteristic. The properties of these sub-networks are listed in Table 5.1.

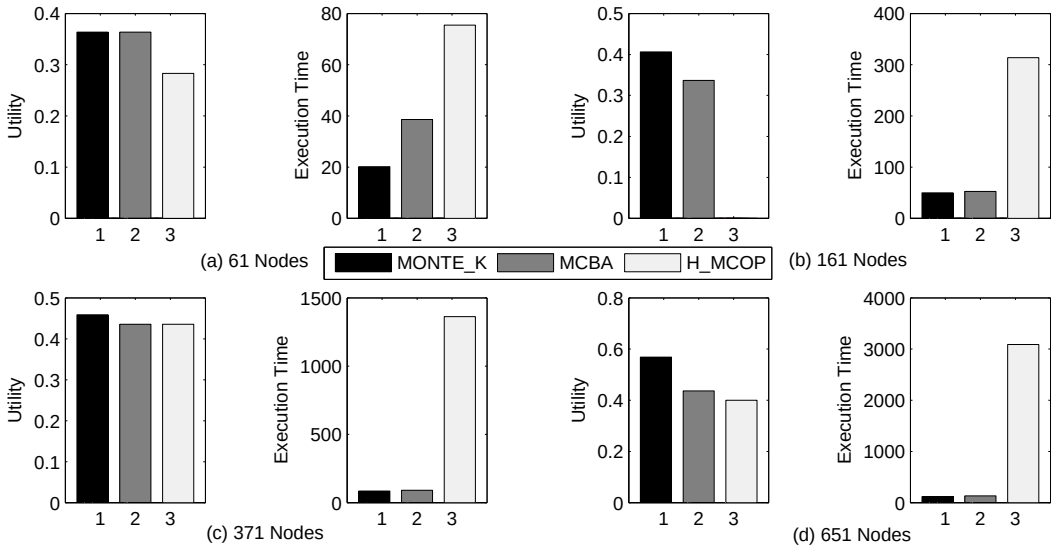

Figure 5.3: Maximal length of paths is 6 hops

The number of simulations of MONTE_K and MCBA in each sub-network is also listed in Table 5.1. The average outdegree in all these sub-networks is 3.77 and 95.5% nodes have an outdegree less than 15. Hence we set $K = 15$ in the K-path selec-

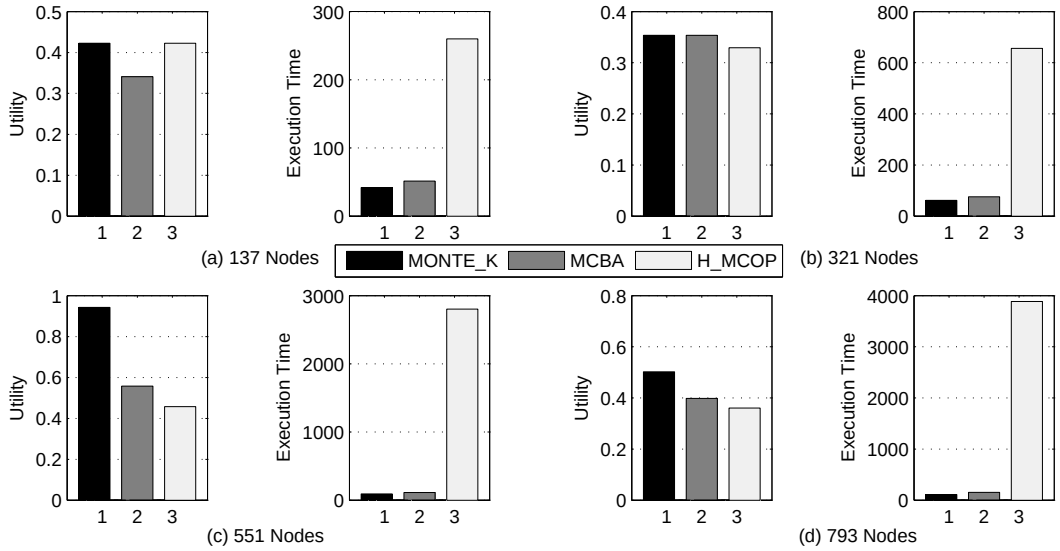

Figure 5.4: Maximal length of paths is 7 hops

tion, without pruning a large number of neighboring nodes of most nodes following the *power-law* characteristic [103]. Secondly, we perform 500 repeated experiments for MONTE_K and MCBA in each sub-network and record the utilities of the identified social trust paths in each experiment. The maximal utilities of the social trust paths identified in all 500 experiments by MONTE_K and MCBA are selected for the comparison with that yielded by H_MCOP. The average execution time of each of MONTE_K and MCBA in each sub-network is recorded based on 500 repeated experiments. The execution time of H_MCOP is averaged based on 5 independent executions. The results are plotted in Fig. 5.1 to Fig. 5.4.

Utility: We can see that in any of 16 cases, MONTE_K does not yield any utility worse than that of H_MCOP while in most sub-networks, the utilities of social trust paths identified by MONTE_K are better than those of H_MCOP (*see Fig. 5.1(a, c, d), Fig. 5.2(a) to (d), Fig. 5.3 (a) to (d) and Fig. 5.4 (b) to (d)*). The sum of utilities computed by MONTE_K is 12.23% more than that of H_MCOP in 4 hops sub-networks, 4.27% more in 5 hops, 60.62% more in 6 hops and 41.51% more in 7 hops. This is because when a trust path with the maximal utility is a feasible solution, H_MCOP can identify it as the optimal solution. However, when the identified trust

Table 5.1: Properties of different social networks

ID	Max Hops	Number of Nodes	Number of Links	Max Outdegree	Max Indegree	Simulation Times
1	4	60	113	16	15	100
2	4	86	192	20	32	100
3	4	115	257	41	82	150
4	4	236	1321	74	91	200
5	5	60	107	9	18	100
6	5	215	528	34	48	200
7	5	240	655	32	49	250
8	5	286	749	31	84	300
9	6	61	124	17	32	100
10	6	161	355	43	46	200
11	6	371	1623	56	48	350
12	6	651	2475	173	151	450
13	7	137	373	48	18	200
14	7	321	860	39	38	350
15	7	551	3265	122	91	400
16	7	793	3411	83	89	500

path is not a feasible solution, H_MCOP can hardly find a near-optimal solution and some times yields an infeasible one even when a feasible solution exists (*see Fig. 5.3(b) where the utility computed by H_MCOP is 0*).

Regarding the utility of identified paths, MONTE_K also outperforms MCBA in most cases and is no worse than MCBA in all cases. The sum of utilities computed by MONTE_K is 17.25% more than that of MCBA in 4 hops sub-networks, 10.89% more in 5 hops, 14.30% more in 6 hops and 34.60% more in 7 hops. This is because *Strategy 2* in MONTE_K guarantees that the solutions identified by later simulations will be no worse than the current one.

Execution Time: From Fig. 5.1 to Fig. 5.4, we can observe that the execution time of MONTE_K is significantly less than that of H_MCOP in all sub-networks. The total execution time of MONTE_K is only 5.92% of that of H_MCOP in 4 hops sub-networks, 10.58% in 5 hops, 5.63% in 6 hops and 4.05% in 7 hops. In particular, in the most complex sub-network with 793 nodes, 3411 links and 7 hops (*see the last row of Table 5.1*), the execution time of MONTE_K is only 2.88% of that of H_MCOP

(*see Fig. 5.4 (d)*). From the above results, we can see that MONTE_K is much more efficient than H_MCOP for identifying the optimal social trust path, especially in larger scale sub-networks. (*see Fig. 5.1(d), Fig. 5.2(c, d), Fig. 5.3(c, d) and Fig. 5.4(b) to (d)*).

In addition, the execution time of MONTE_K is also less than MCBA. The total execution time of MONTE_K is 91.48% of that of MCBA in 4 hops sub-networks, 87.72% in 5 hops, 86.94% in 6 hops and 78.25% in 7 hops. This is because in MONTE_K, when any QoT constraint of a trust path from the source participant to an intermediate node v_k can not be satisfied, MONTE_K starts a new simulation, rather than searching the social trust path from v to the target. Thus, the execution time of MONTE_K is less than MCBA in all sub-networks. And the greater the number of hops, the less the execution time of MONTE_K than MCBA.

Through the above experiments conducted in the sub-networks with different scales and structures, we can see that our proposed approximation algorithm, MONTE_K, addresses the characteristics of online social networks well and thus can deliver better near-optimal solutions with less execution time than existing approximation algorithms.

5.4 The Proposed H_OSTP for Optimal Social Trust Path Selection

In this section, we propose an efficient heuristic algorithm, H_OSTP, for the QoT constrained optimal social trust path selection. 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, SI and CIF) of the identified path from v_t to each intermediate node v_k. 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.

5.4.1 Algorithm Description of H_OSTP

In social trust path selection, if a path satisfies multiple QoT constraints, it means that each aggregated QoT attribute (i.e., T, SI or CIF) of that path should be larger than the corresponding QoT constraint. Therefore, we propose an objective function in Eq. (5.7) to investigate whether the aggregated QoT attributes of a path can satisfy the QoT constraints in a certain domain. From Eq. (5.7), 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 $\delta(p) \leq 1$.

$$\delta(p) \triangleq max\{(\frac{1-T_p}{1-Q^T_{v_s,v_t}}),(\frac{1-SI_p}{1-Q^{SI}_{v_s,v_t}}),(\frac{1-CIF_p}{1-Q^{CIF}_{v_s,v_t}})\} \tag{5.7}$$

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 [31]. 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 $p^{b(\delta)}_{v_k \to v_t}$) is identified and $T_{p^{b(\delta)}_{v_k \to v_t}}$ $SI_{p^{b(\delta)}_{v_k \to v_t}}$ and $CIF_{p^{b(\delta)}_{v_k \to v_t}}$ 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 $\exists \varphi \in \{T, SI, CIF\}$ that $\varphi_{p_s} < Q^{\varphi}_{v_s,v_t}$. 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. □

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 $v_n \in \{neighboring\ nodes\ of\ v_s\}$ 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_n (denoted as path $p^{f(u)}_{v_s \to v_n}$). Let $p^{b(\delta)}_{v_n \to v_t}$ denote the path from v_n to v_t identified in the *Backward_Search* procedure, then a *foreseen path* from v_s to v_t via v_n (denoted as $fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t} = p^{f(u)}_{v_s \to v_n} + p^{b(\delta)}_{v_n \to v_t}$) can be identified. Let h denote the number of hops of path $fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t} = p^{f(u)}_{v_s \to v_n} + p^{b(\delta)}_{v_n \to v_t}$. The aggregated QoS attribute values of $fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}$ can be calculated as $T_{fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}} = T_{p^{f(u)}_{v_s \to v_n}} * T_{p^{b(\delta)}_{v_n \to v_t}}$, $SI_{fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}} = (SI_{p^{f(u)}_{v_s \to v_n}} * SI_{p^{b(\delta)}_{v_n \to v_t}})/h^{\alpha}$ ($\alpha \geq 1$ is the argument for controlling the attenuation speed of SI) and $CIF_{fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}} = (CIF_{p^{f(u)}_{v_s \to v_n}} + CIF_{p^{b(\delta)}_{v_n \to v_t}})/(h-1)$. According whether $fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}$ is feasible, H_OSTP adopts the following searching strategies.

Situation 1: If each aggregated QoT attribute of $fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}$ satisfies the corresponding end-to-end QoT constraint, then H_OSTP chooses the next node from v_n 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 $fp^{f(u)+b(\delta)}_{v_s \to v_n \to v_t}$ does not satisfy the corresponding end-to-end QoT constraint, then H_OSTP does not search the path from v_n and the link $v_s \to v_n$ 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 $v_s \to v_n$.

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 iden-

Algorithm 7: H_OSTP

Data: $MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t
Result: p_t, $\mathcal{F}(p_t)$
begin
1 $p_s = \emptyset$, $p_t = \emptyset$;
2 **Backward_Search** ($MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t);
3 **if** $\delta(p_s) > 1$ **then**
4 **Return** no feasible solution;
end
5 **else**
6 **Forward_Search** ($MT(v_s, v_t)$, $AQ^{\mu}(p^{b(\delta)}_{v_k \rightarrow v_t})$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t);
7 **Return** p_t and $\mathcal{F}(p_t)$;
end
end

tified 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 $v_s, v'_1, ..., v'_n, v_t$. 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 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. □

If there exists only one feasible solution in the sub-network, it can be identified by both the *Backward_Search* procedure and the *Forward_Seach* 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_Seach* procedure is near-optimal or optimal, which is better than the one identified by the *Backward_Search* procedure.

Algorithm 8: Backward_Search

Data: $MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t
Result: p_s
begin
1 **Set** $v_x.\delta = \infty$ $(v_x \neq v_t)$, $v_t.\delta = 0$, $S_x = \emptyset$;
2 **Add** v_t into S_x;
3 **while** $S_x \neq \emptyset$ **do**
4 $v_a.\delta = min(v^*_a.\delta)$ $(v^*_a \in S_x)$;
5 **for** *each* $v_b \in adj[v_a]$ **do**
6 h is the number of hops of the path from v_t to v_b;
7 $\delta(p^{b(\delta)}_{v_b \to v_t}) = max[(1 - AQ^T(p^{b(\delta)}_{v_b \to v_t}) * MT(v_a, v_b).T/(1 - Q^T_{v_s,v_t}), (1 - AQ^{SI}(p^{b(\delta)}_{v_j \to v_t}) * MT(v_a, v_b).SI/h^{\alpha})/(1 - Q^{SI}_{v_s,v_t}), (1 - (AQ^{CIF}(p^{b(\delta)}_{v_j \to v_t}) + MT(v_a, v_b).CIF))/(h - 1)/(1 - Q^{CIF}_{v_s,v_t})]$;
8 **if** $v_b \notin S_x$ **then**
9 Put v_b into S_x;
10 $pre_x(v_b) = v_a$;
end
11 **else if** $\delta(p^{b(\delta)}_{v_b \to v_t}) < AQv_b.\delta$ **then**
12 $v_b.\delta = \delta(p^{b(\delta)}_{v_b \to v_t})$;
13 update $AQ^{\mu}(p^{b(\delta)}_{v_b \to v_t})$;
14 Put v_b into S_x;
15 $pre_x(v_b) = v_a$;
end
end
16 **Remove** v_a from S_x;
end
17 $p_s \leftarrow pre_x(v_s)$ to $pre_x(v_t)$;
18 **Return** p_s
end

5.4.2 The Process of H_OSTP

Step 1: Start the *Backward_Search* procedure. Add v_t into S_x. Select the node v_a from S_x, where the δ value of the path from v_t to v_a (i.e., $p^{b(\delta)}_{v_a \to v_t}$) is the minimum of all δ of the paths from v_t to v^*_a $(v^*_a \in S_x)$ (*lines 1-2 in Algorithm 7 and lines 1 to 4 in Algorithm 8*).

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 $p^{b(\delta)}_{v_b \to v_t}$). 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 $T_{p^{b(\delta)}_{v_b \to v_t}}$, $SI_{p^{b(\delta)}_{v_b \to v_t}}$ and $CIF_{p^{b(\delta)}_{v_b \to v_t}}$ at v_b. Add v_b into S_x and set $pre_x(v_b) = v_a$ (*lines 5 to 15 in Algorithm 8*).

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 3 to 4 in Algorithm 7 and lines 16 to*

Algorithm 9: Forward_Search

Data: $MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t
Result: p_t, $\mathcal{F}(p_t)$
begin
1 Set $\mathcal{F}' = 1/\mathcal{F}$, $v_y.\mathcal{F}' = \infty$ ($v_y \neq v_s$), $v_s.\mathcal{F}' = 0$, $S_y = \emptyset$;
2 Add v_s into S_y;
3 **while** $S_y \neq \emptyset$ **do**
4 $v_i.\mathcal{F}' = min(v_i^*.\mathcal{F}')$ ($v_i^* \in S_y$);
5 **for** *each* $v_j \in adj[v_i]$ **do**
6 h' is the number of the hops of the foreseen path from v_s to v_t via v_j;
7 $temp_T = AQ^T(p^{f(u)}_{v_s \to v_i}) * MT(v_i, v_j).T * AQ^T(p^{b(\delta)}_{v_j \to v_t})$
8 $temp_{SI} = AQ^{SI}(p^{f(u)}_{v_s \to v_i}) * MT(v_i, v_j).SI * AQ^{SI}(p^{b(\delta)}_{v_j \to v_t})$
9 $temp_{CIF} = AQ^{CIF}(p^{f(u)}_{v_s \to v_i}) + MT(v_i, v_j).CIF + AQ^{CIF}(p^{b(\delta)}_{v_j \to v_t})$;
10 **if** $temp_T \geq Q^T_{v_s,v_t}$ *and* $temp_{SI}/h'^{\alpha} \geq Q^{SI}_{v_s,v_t}$ *and* $temp_{CIF}/(h'-1) \geq Q^{CIF}_{v_s,v_t}$ **then**
11 **if** $v_j \notin S_y$ **then**
12 Put v_j into S_y;
13 $pre_y(v_j) = v_i$;
end
14 **else if** $\mathcal{F}'(p^{f(u)}_{v_s \to v_j}) < v_j.\mathcal{F}'$ **then**
15 $v_j.\mathcal{F}' = \mathcal{F}'(p^{f(u)}_{v_s \to v_j})$;
16 update $AQ^{\mu}(p^{f(u)}_{v_s \to v_j})$;
17 Put v_j into S_y;
18 $pre_y(v_j) = v_i$;
end
end
end
19 **Remove** v_i from S_y;
end
20 $p_t \leftarrow Pre_y(v_t)$ to $Pre_y(v_s)$;
21 **Return** p_t and $\mathcal{F}(p_t)$);
end

18 in Algorithm 8).

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 5 to 6 in Algorithm 7 and lines 1 to 4 in Algorithm 9*).

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 $p^{f(u)}_{v_s \to v_j}$). If the current $1/\mathcal{F}(p^{f(u)}_{v_s \to v_j})$ is less than the value recorded at node v_j, then calculate each aggregated QoT attribute value $T_{p^{f(u)}_{v_s \to v_j}}$, $SI_{p^{f(u)}_{v_s \to v_j}}$ and $CIF_{p^{f(u)}_{v_s \to v_j}}$. If each aggregated QoT value can satisfy the corresponding QoT constraint, then replace the stored $1/\mathcal{F}(p^{f(u)}_{v_s \to v_j})$ with the current $1/\mathcal{F}(p^{f(u)}_{v_s \to v_j})$ at v_j and set $pre_y(v_j) = v_i$. Otherwise, set $MT(v_i, v_j).T = 0$, $MT(v_i, v_j).SI = 0$ and

$MT(v_i, v_j).CIF = 0$ (*lines 5 to 18 in Algorithm 9*).

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 $pre_y(v_t)$ (*line 7 in Algorithm 7 and lines 19 to 21 in Algorithm 9*).

H_OSTP consumes twice the execution time of Dijkstra's shortest path algorithm. The time complexity of H_OSTP 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_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 6.4.2*).

5.5 Experiments on H_OSTP

5.5.1 Experiment Settings

As introduced in *Section 4.4.1*, Enron email dataset fit the proposed contextual trust-oriented social network well. Thus, we also select the *Enron* email corpus with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) as the dataset for our experiments.

As we analysed in Section 5.2.1, H_MCOP is the most promising algorithm for the MCOP selection. Based on it, several approximation algorithms [74, 134] 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 [67] in both execution time and the utilities of identified social trust paths (*see section 6.4.2*). In our experiments, the T, SI and CIF 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 $Q_{v_s,v_t} = \{Q^T_{v_s,v_t} \geq 0.05,\ Q^{SI}_{v_s,v_t} \geq$

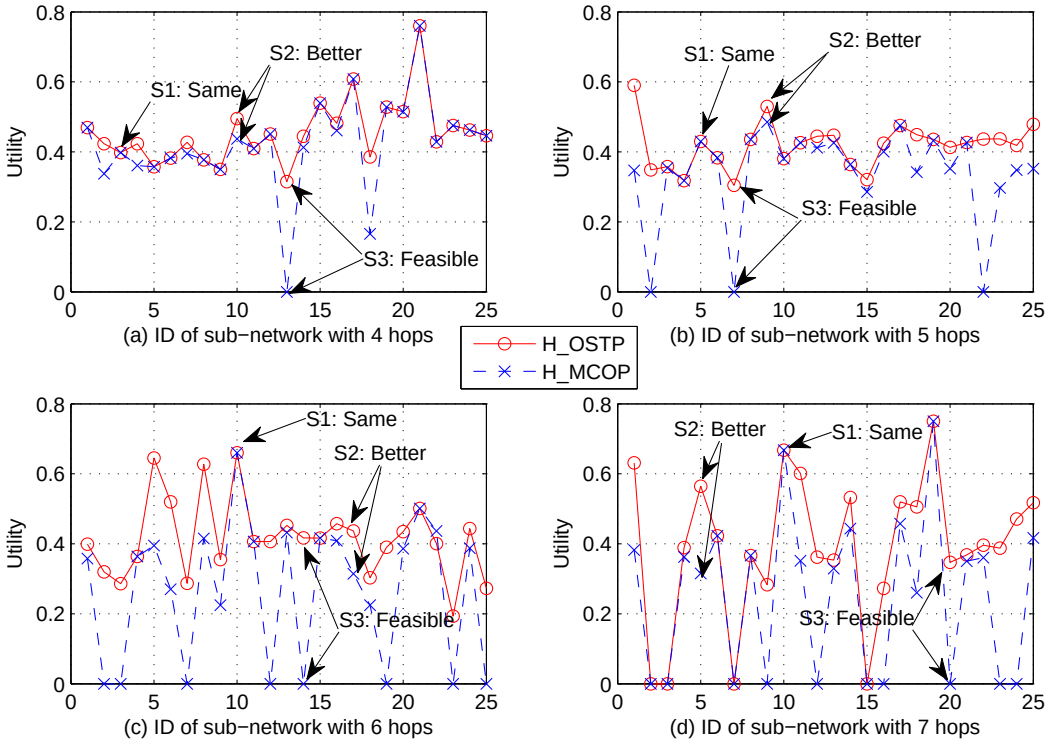

Figure 5.5: The comparison in path utilities of sub-networks

$0.001,\ Q^{CIF}_{v_s,v_t} \geq 0.3\}$ and the weights of attributes in the utility function specified by the source participant are set as $\omega_T = 0.25,\ \omega_{SI} = 0.25$ and $\omega_{CIF} = 0.5$.

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.

5.5.2 Results and Analysis

Table 5.2: 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

Table 5.3: The comparison of utility

Algorithms	The sum of utility			
	4 hops	5 hops	6 hops	7 hops
H_OSTP	11.3515	10.4770	10.3937	9.7074
H_MCOP	10.5265	8.4712	6.6006	6.2363
difference	10.78% more	12.37% more	15.75% more	15.57% more

Table 5.4: The comparison of execution time

Algorithms	The sum of execution time (sec)			
	4 hops	5 hops	6 hops	7 hops
H_OSTP	133.9208	449.6327	1.1924e+003	2.2585e+003
H_MCOP	222.9832	875.9788	2.2262e+003	4.4913e+003
difference	39.94% less	48.67% less	46.44% less	49.71% less

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[1]. 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 5.2 list 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_OSTP and H_MCOP. The results are plotted in Fig. 5.5 and 5.6 where the execution time of each of H_OSTP and H_MCOP is averaged based on the 5 independent runs.

Results (Utility). From Fig. 5.5, 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. 5.5 (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. 5.5 (a) to (d)). The sum of utilities computed by H_OSTP and H_MCOP in the sub-

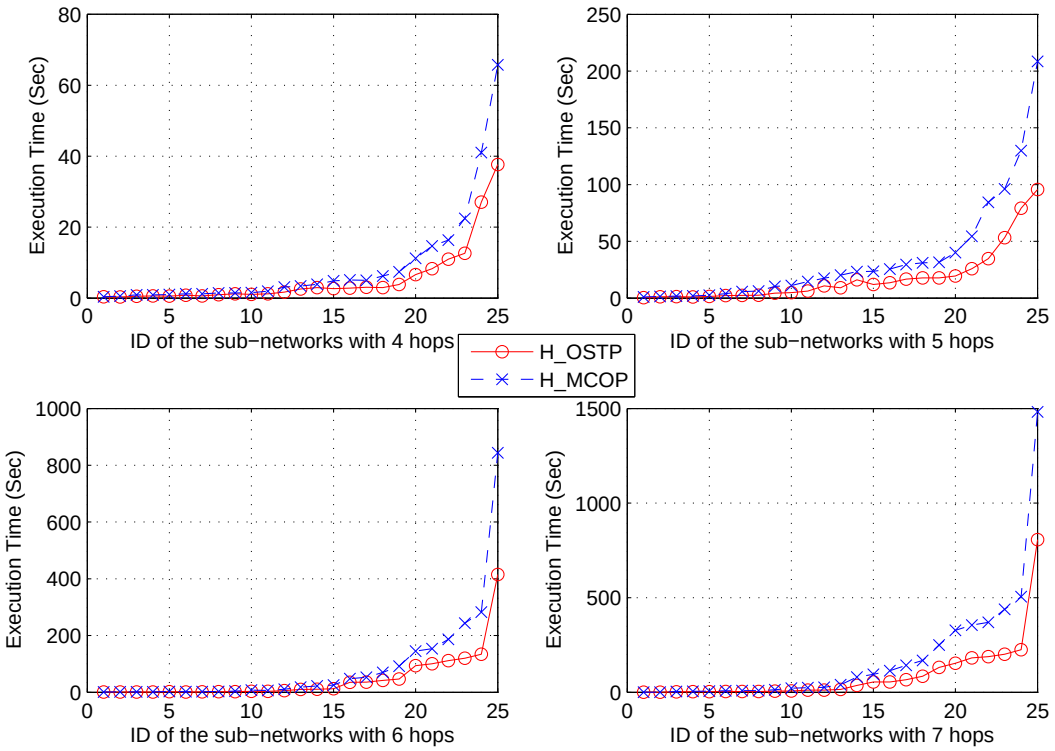

Figure 5.6: The comparison in Execution time

networks with each group of hops is listed in Table 5.3. From the Table, 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 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. 5.5 (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. 5.5 (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. 5.6, 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 5.4. From the table, 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 $\lambda \rightarrow \infty$ [74]. 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.

5.6 The Proposed MFPB-HOSTP for Optimal Social Trust Path Selection

5.6.1 The Advantage and Disadvantage of H_OSTP

Advantage: H_OSTP could detect whether there exist a feasible solution in a sub-network, as it adopts a new objective function $\delta(p)$ which is better than that of H_MCOP.

If there exists at least one feasible solution, H_OSTP does not deliver any solution which is worse in quality than that of H_MCOP, and could possibly deliver better solutions than H_MCOP. In addition, when a foreseen path is infeasible (i.e., at least one aggregated QoT attribute value of the path does not satisfy the corresponding QoT constraint), the corresponding link between nodes is deleted, which reduces the search space and makes H_OSTP more efficient than H_MCOP [78].

Disadvantage: Although H_OSTP significantly outperforms existing approximation algorithms in both the efficiency and the quality of identified social trust paths, it still has a disadvantage called the *imbalance problem of QoT attributes*, which may cause a failed feasibility estimation of a foreseen path in the forward search procedure from v_s to v_t, and deliver a solution with a low utility that is not near optimal. We analyse the disadvantage of H_OSTP below in detail.

If a *feasible solution* (i.e., a path where the aggregated value of each QoT attribute satisfies the corresponding QoT constraint) exists in the sub-network between v_s and v_t, H_OSTP performs the *Forward_Search* procedure, where H_OSTP investigates the feasibility of the foreseen path $fp_{v_s \to v_k \to v_t}^{f(u)+b(\delta)}$ to estimate whether a feasible solution can be delivered by following $p_{v_s \to v_k}^{f(u)}$. But this strategy may give a failed feasibility estimation. Namely, even if $fp_{v_s \to v_k \to v_t}^{f(u)+b(\delta)}$ is infeasible, there may still exist a feasible solution identified by following $p_{v_s \to v_k}^{f(u)}$ in the sub-network.

We use the following example to illustrate the imbalance problem of QoT attributes in H_OSTP. Fig. 5.7 depicts a social network between v_s and v_t, which contains five intermediate nodes v_1 to v_5, and the aggregated QoT attribute values computed by the *Backward_Search* procedure at each of these nodes is listed in Table 5.5. Suppose that v_s specifies the QoT constraints as $Q_{v_s,v_t}^{T} > 0.3, Q_{v_s,v_t}^{SI} > 0.3$ and $Q_{v_s,v_t}^{CIF} > 0.2$. Based on the search strategy introduced in Section 5.4, at v_4, H_OSTP concatenates the social trust path $p_{v_s \to v_4}^{f(u)}$ with $p_{v_4 \to v_t}^{b(\delta)}$ to form a foreseen path $fp_{v_s \to v_4 \to v_t}^{f(u)+b(\delta)}$ with the aggregated QoT attributes values as $T = 0.2$, $SI = 0.48$ and $CIF = 0.5$, which is infeasible (note: the aggregated $T = 0.2$ does not satisfy the corresponding constraint $Q_{v_s,v_t}^{T} > 0.3$). In such a situation, H_OSTP deletes the link $v_2 \to v_4$ in $p_{v_s \to v_4}^{f(u)}$ and

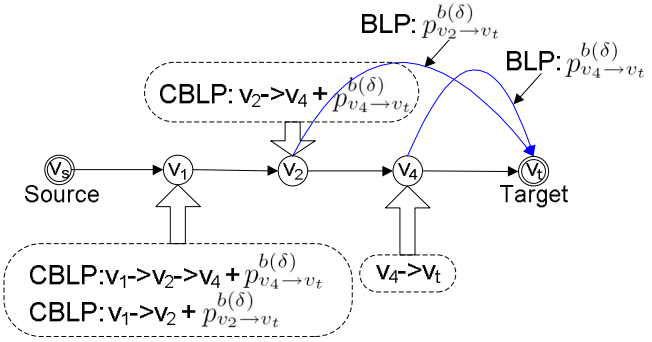

Figure 5.7: Limitation of H_OSTP

Table 5.5: Social trust paths and the aggregated QoT attributes values

Path	Nodes and Links	T	SI	CIF
$p^{f(u)}_{v_s \to v_4}$	$v_s \to v_1 \to v_2 \to v_4$	0.4	0.8	0.5
$p^{b(\delta)}_{v_4 \to v_t}$	$v_4 \to v_3 \to v_5 \to v_t$	0.5	0.6	0.5
$p^{b(T)}_{v_4 \to v_t}$	$v_4 \to v_t$	0.8	0.45	0.5
path $v_2 \to v_t$	$v_2 \to v_t$	0.75	0.4	0.4

selects another path $v_s \to v_1 \to v_2 \to v_t$ as the near-optimal social trust path between v_s and v_t. Suppose the QoT attributes have the same weights in the utility function, then the utility of this path is 0.35.

However, as shown in Fig. 5.7, the aggregated values of QoT attributes of another path $v_4 \to v_t$ (denoted as $p^{b(T)}_{v_4 \to v_t}$) are $T = 0.8$, $SI = 0.45$ and $CIF = 0.5$. If we concatenate $p^{f(u)}_{v_s \to v_4}$ and $p^{b(T)}_{v_4 \to v_t}$ together, a new foreseen path $fp^{f(u)+b(T)}_{v_s \to v_4 \to v_t}$ is formed that is feasible. In such a situation, the path $v_s \to v_1 \to v_2 \to v_4 \to v_t$ with a utility of 0.39 is selected as the solution, which has a better quality than the one identified by H_OSTP (i.e., the utility=0.35).

From the above example, we can see that the foreseen path formed by concatenating path $p^{f(u)}_{v_s \to v_k}$ with path $p^{b(\delta)}_{v_k \to v_t}$ may not accurately estimate whether there exists a feasible a solution identified by following $p^{f(u)}_{v_s \to v_k}$ in the forward search procedure. This is because during searching $p^{b(\delta)}_{v_k \to v_t}$, one of the aggregated values of the QoT attributes may be already close to the corresponding QoT constraints (e.g., $T = 0.5$ of $p^{b(\delta)}_{v_4 \to v_t}$ in Fig. 5.7). In such a situation, if the aggregated values of that QoT attribute is also close

to the corresponding QoT constraint in $p^{f(u)}_{v_s \to v_k}$ (e.g., $T = 0.4$ of $p^{f(u)}_{v_4 \to v_t}$ in Fig. 5.7), the foreseen path at v_k is usually infeasible. This is the typical imbalance problem of QoT attributes (e.g., the imbalance problem of T at v_4 in Fig 5.7), which may lead to a failed feasibility estimation of a foreseen path. In such a situation, H_OSTP cannot identify a social trust path with a high utility that is near-optimal.

5.6.2 Algorithm Description of MFPB-HOSTP

We first introduce some definitions below that are used to describe our algorithm.

Definition 4: (Backward Local Path (BLP)): In a sub-network from v_s to v_t, a Backward Local Path (BLP) is the path from v_t to an intermediate node v_k, identified by the backward search from v_t to v_s.

Based on *Definition 5*, path $p^{b(\delta)}_{v_k \to v_t}$ identified by the backward search procedure is a BLP.

Definition 5: (Forward Local Path (FLP)): In a sub-network from v_s to v_t, a Forward Local Path (FLP) is the path from v_s to an intermediate node v_k, identified by the forward search from v_s to v_t.

Based on *Definition 6*, path $p^{f(u)}_{v_s \to v_t}$ identified by the forward search procedure is an FLP. A foreseen path can be formed at the same intermediate node v_k by concatenating an FLP that ends at node v_k and a BLP that starts from node v_k.

Definition 6: (Composite Backward Local Path (CBLP)): in a sub-network between v_s and v_t, a Composite Backward Local Path (CBLP) is the path which is composed of the BLP with the minimal δ and the links of BLP with the maximal aggregated value for one of the QoT attributes.

Based on the above definitions, we propose a novel Multiple Foreseen Path-Based Heuristic algorithm for Optimal Social Trust Path selection (MFPB-HOSTP) in complex social networks that inherits the advantages of H_OSTP (i.e., the objective function) and aims to overcome its disadvantage (i.e., the imbalance problem of QoT attributes). Our MFPB-HOSTP also bidirectionally searches a sub-network (i.e., by

employing both a backward search and a forward search procedure) by adopting the Dijkstra's shortest path algorithm [31]. But our algorithm employs different search strategies with H_OSTP.

In the backward search procedure from v_t to v_s, at each intermediate node v_k, in addition to BLP $p^{b(\delta)}_{v_k \to v_t}$, MFPB-HOSTP first identifies the BLPs with the maximal aggregated T, SI and CIF values respectively (denoted as $p^{b(\mu)}_{v_k \to v_t},\ \mu \in \{T, SI, CIF\}$). When facing with the imbalance problem of QoT attribute μ at v_k (e.g., T at v_4 in Fig. 5.7), the identified BLPs $p^{b(\mu)}_{v_k \to v_t}$ are concatenated with the identified FLP, forming other foreseen paths (e.g., $fp^{f(u)+b(T)}_{v_s \to v_4 \to v_t}$ in Fig. 5.7), helping avoid a failed feasibility estimation of a foreseen path and having a chance to deliver a better solution than H_OSTP (e.g., the path $v_s \rightarrow v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_t$ in Fig. 5.7). However, greedily maximizing the aggregated value of the QoT attribute may cause a new imbalance problem of QoT attributes (see a detailed analysis in *Step 2* in the following section of *Algorithm Description*). Therefore, MFPB-HOSTP then identifies some CBLPs the number of which depends on the number of intermediate nodes of $p^{b(\mu)}_{v_k \to v_t}$ ($\mu \in \{T, SI, CIF\}$). When facing with the new imbalance problem of QoT attributes at v_k, these CBLPs are used to be concatenated with the FLP to balance QoT attributes in the newly formed foreseen paths, which could increase the probability of delivering a solution with high utility that is near-optimal (see a detailed analysis in *Step 2* in the following section of *Algorithm Description*).

The backward search procedure could illustrate whether there exists a feasible solution in a sub-network (it is proved in *Theorem 1* in the following section of *Algorithm Description*). If there exists at least one feasible solution, MFPB-HOSTP performs a forward search procedure from v_s to v_t. This procedure intends to identify the path with the maximal utility by using the Dijkstra's shortest path algorithm [31]. When facing with the imbalance problem of QoT attributes at v_k, MFPB-HOSTP concatenates the FLP (i.e., $p^{f(u)}_{v_s \to v_k}$) with BLPs and CBLPs, forming multiple foreseen paths, instead of one foreseen path only in H_OSTP. This strategy could effectively help address the imbalance problem of QoT attributes in path selection, and thus helping avoid

a failed feasibility estimation of a foreseen path in the social path selection.

5.6.3 The Process of MFPB-HOSTP

In this section, we give a more detailed description of our proposed MFPB-HOSTP algorithm.

Backward_Search: In the *Backward_Search* procedure, MFPB-HOSTP searches the sub-network from v_t to v_s to investigate whether there exists a feasible solution in the sub-network. In this process, at each intermediate node v_k, several BLPs and CBLPs from v_t to v_k are identified. The identification of these paths can be divided into the following 4 steps.

Algorithm 10: MFPB-HOSTP

Data: $MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$
Result: $p^{forward}_{v_s \to v_t}$, $\mathcal{F}(p^{forward}_{v_s \to v_t})$
begin
  $p^{forward}_{v_s \to v_t} = \emptyset$, $p^{backward}_{v_s \to v_t} = \emptyset$;
  Backward_Search($MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$);
  if $\delta(p^{backward}_{v_s \to v_t}) > 1$ **then**
    Return no feasible solution;
  end
  else
    Forward_Search($MT(v_s, v_t)$, $AQ^{\mu}(p^{b(\delta)}_{v_k \to v_t})$, $AQ^{\mu}(p^{b(\mu)}_{v_k \to v_t})$, $AQ^{\mu}(p^{CBLP(\mu)}_{v_k \to v_t})$,
    $\mu \in \{T, SI, CIF\}$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$);
    Return $p^{forward}_{v_s \to v_t}$ and $\mathcal{F}(p^{forward}_{v_s \to v_t})$;
  end
end

Step 1 (identify the BLP with the minimal δ): In social trust path selection, if a path satisfies multiple QoT constraints, the aggregated value of each QoT attribute (i.e., T, SI or CIF) of that path should be larger than the corresponding QoT constraint. From Eq. (5.7), we can see that if any aggregated QoT attribute value of a social trust path does not satisfy the corresponding QoT constraint, then $\delta(p) > 1$. Otherwise $\delta(p) \leq 1$.

To investigate whether there exists a feasible solution in a sub-network, in this step, MFPB-HOSTP identifies the path from v_t to v_s with the minimal δ (i.e., $p^{b(\delta)}_{v_s \to v_t}$) based on the Dijkstra's shortest path algorithm [31]. In the searching process, at each in-

Algorithm 11: Backward_Search ()

Data: $MT(v_s, v_t)$, $Q^{T}_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$
Result: $\delta(p^{backward}_{v_s \to v_t})$, $AQ^{\mu}(p^{b(\delta)}_{v_k \to v_t})$, $AQ^{\mu}(p^{b(\mu)}_{v_k \to v_t})$, $AQ^{\mu}(p^{CBLP(\mu)}_{v_k \to v_t})$, $(\mu \in \{T, SI, CIF\})$
begin
  Set $v_x.d = \infty$ $(v_x \neq v_t)$, $v_t.d = 0$, $S_x = \emptyset$, $p^{b(\delta)}_{v_t \to v_t} = v_t$;
  Add v_t into S_x;
  while $S_x \neq \emptyset$ **do**
    $v_a.d = min(v^*_a.d)$ $(v^*_a \in S_x)$;
    for *each* $v_b \in adj[v_a]$ **do**
      if $v_b \notin S_x$ **then**
        Put v_b into S_x;
        $p^{b(\delta)}_{v_b \to v_t} = v_b \to v_a + p^{b(\delta)}_{v_a \to v_t}$;
      end
      else if $\delta(v_b \to v_a + p^{b(\delta)}_{v_a \to v_t}) < v_b.d$ **then**
        Update $v_b.d$ and $AQ^{\mu}(p^{b(\delta)}_{v_b \to v_t})$, $(\mu \in \{T, SI, CIF\})$;
        $p^{b(\delta)}_{v_b \to v_t} = v_b \to v_a + p^{b(\delta)}_{v_a \to v_t}$;
      end
    end
    Remove v_a from S_x;
  end
  $p^{backward}_{v_s \to v_t} = p^{b(\delta)}_{v_s \to v_t}$;
  if $\delta(p^{backward}_{v_s \to v_t}) \leq 1$ **then**
    Computing Max_$T(MT(v_s, v_t), Q^{T}_{v_s,v_t}, Q^{SI}_{v_s,v_t}, Q^{CIF}_{v_s,v_t})$;
    Computing Max_$SI(MT(v_s, v_t), Q^{T}_{v_s,v_t}, Q^{SI}_{v_s,v_t}, Q^{CIF}_{v_s,v_t})$;
    Computing Max_$CIF(MT(v_s, v_t), Q^{T}_{v_s,v_t}, Q^{SI}_{v_s,v_t}, Q^{CIF}_{v_s,v_t})$;
  end
end

termediate node v_k, BLP $p^{b(\delta)}_{v_k \to v_t}$ is identified and the aggregated QoT attribute values of these paths (i.e., $T_{p^{b(\delta)}_{v_k \to v_t}}$ $SI_{p^{b(\delta)}_{v_k \to v_t}}$ and $CIF_{p^{b(\delta)}_{v_k \to v_t}}$) are computed and recorded. According to the *Theorem 1* in H_OSTP, the *Backward_Search* procedure can investigate whether there exists a feasible solution in the sub-network.

The *Backward_Search* procedure of MFPB-HOSTP can always identify the path with the minimal δ. If $\delta_{min} > 1$, it indicates there is no feasible solution in the sub-network, then the algorithm terminates. If $\delta_{min} \leq 1$, it indicates there exists at least one feasible solution and the identified path is a feasible solution. In such a case, the algorithm will perform the following steps to deliver a near-optimal solution.

Step 2 (identify the BLP with the maximal aggregated T value and the corresponding CBLPs): In this step, at each intermediate node v_k, MFPB-HOSTP first identifies the BLP with the maximal aggregated T value (i.e., $p^{b(T)}_{v_k \to v_t}$), and then identifies several corresponding CBLPs which are composed of part of $p^{b(T)}_{v_k \to v_t}$ and a BLP with the minimal δ from v_t to each intermediate node in $p^{b(T)}_{v_k \to v_t}$.

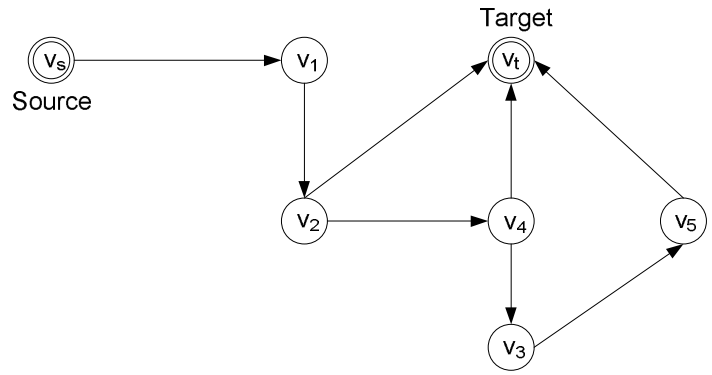

Figure 5.8: Multiple CBLPs in backward search procedure

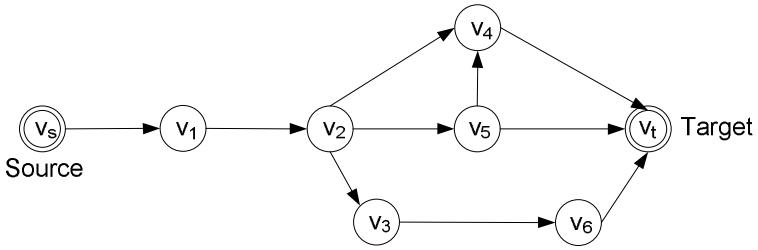

Figure 5.9: The CBLP in path selection

(a): identify the BLPs with the maximal T. MFPB-HOSTP first identifies the path from v_t to v_s with the maximal aggregated T value (i.e., $p^{b(T)}_{v_s \to v_t}$) based on the Dijkstra's shortest path algorithm [31]. In the searching process, at each intermediate node v_k, BLP $p^{b(T)}_{v_k \to v_t}$ (e.g., BLP $v_4 \to v_t$ in Fig. 5.7) and the aggregated QoT attributes' values of $p^{b(T)}_{v_k \to v_t}$ are computed and recorded. When facing with the imbalance problem of T at v_k, BLP $p^{b(T)}_{v_k \to v_t}$ is concatenated with the FLP $p^{f(u)}_{v_k \to v_t}$, forming a new foreseen path $fp^{f(u)+b(T)}_{v_s \to v_k \to v_t}$ (e.g., the foreseen path $v_1 \to v_2 \to v_4 \to v_t$ in Fig. 5.7). This foreseen path could be used as a reference to estimate whether there exists a feasible solution identified by following $p^{f(u)}_{v_s \to v_k}$. This strategy could help avoid a failed feasibility estimation of a foreseen path caused by the imbalance problem of T at v_k.

(b): identify the CBLPs based on the BLPs with the maximal T. Greedily maximizing the aggregated T value without considering other QoT attributes values in $p^{b(T)}_{v_k \to v_t}$ may lead to the new imbalance problem of QoT attributes (i.e., SI and CIF). Therefore, in addition to $p^{b(T)}_{v_k \to v_t}$, suppose there are M intermediate nodes (denoted as

Table 5.6: BLPs, CBLPs, and the aggregated QoT attributes values

Path	Nodes and Links	T	SI	CIF
$p^{f(u)}_{v_s \to v_2}$	$v_s \to v_1 \to v_2$	0.3	0.8	0.5
$p^{b(\delta)}_{v_2 \to v_t}$	$v_2 \to v_4 \to v_t$	0.25	0.5	0.4
$p^{b(T)}_{v_2 \to v_t}$	$v_2 \to v_5 \to v_4 \to v_t$	0.7	0.1	0.3
$p^{CBLP^1(T)}_{v_2 \to v_t}$	$v_2 \to v_5 \to v_t$	0.5	0.2	0.3
path $v_3 \to v_t$	$v_3 \to v_6 \to v_t$	0.4	0.2	0.3

$v_l,\ l \in [1, M]$) in path $p^{b(T)}_{v_k \to v_t}$, MFPB-HOSTP then identifies M *Composite Backward Local Paths* at v_k (denoted as $p^{CBLP^M(T)}_{v_k \to v_t}$ which are composed of $p^{b(T)}_{v_k \to v_l}\ l \in [1, M]$ and $p^{b(\delta)}_{v_l \to v_t},\ l \in [1, M]$. For example, as shown in Fig. 5.8, since there is no intermediate node between v_4 and v_t in BLP $p^{b(T)}_{v_4 \to v_t}$ (i.e., M=0), MFPB-HOSTP only identifies one BLP $p^{b(T)}_{v_4 \to v_t} = v_4 \to v_t$. Since there exists an intermediate node v_4 between v_2 and v_t in BLP $p^{b(T)}_{v_2 \to v_t}$ (i.e., $M = 1$), in addition to $p^{b(T)}_{v_2 \to v_t}$, MFPB-HOSTP identifies one CBLP $p^{CBLP^1(T)}_{v_2 \to v_t} = (v_2 \to v_4) + p^{b(\delta)}_{v_4 \to v_t}$. Similarly, at v_1 there exist two intermediate nodes between v_1 and v_t in BLP $p^{b(T)}_{v_1 \to v_t}$ (i.e., $M = 2$), MFPB-HOSTP identifies two CBLPs. They are CBLP $p^{CBLP^1(T)}_{v_1 \to v_t} = (v_1 \to v_2 \to v_4) + p^{b(\delta)}_{v_4 \to v_t}$ and CBLP $p^{CBLP^2(T)}_{v_1 \to v_t} = (v_1 \to v_2) + p^{b(\delta)}_{v_2 \to v_t}$. When facing with the new imbalance caused by the BLP with the maximal T, the M CBLPs at v_k are concatenated with the FLP $p^{f(u)}_{v_s \to v_k}$. This strategy could help avoid a failed feasibility estimation of a foreseen path caused by the new imbalance problem of other two QoT attributes (i.e., SI and CIF) at v_k. Next we use an example to illustrate the effectiveness of CBLPs in solving the new imbalance problem of QoT attributes.

Fig. 5.9 depicts a sub-network between v_s and v_t. Table 5.6 lists the FLP at v_2, the BLP at v_2, the corresponding CBLP at v_2, and the aggregated values of QoT attributes of these paths. Suppose that the QoT constraints specified by source participant v_s are $Q^T_{v_s,v_t} = 0.12$, $Q^{SI}_{v_s,v_t} = 0.15$ and $Q^{CIF}_{v_s,v_t} = 0.3$. We could see that the foreseen path $fp^{f(u)+b(\delta)}_{v_s \to v_2 \to v_t}$ is infeasible due to the imbalance problem of T at v_2 ($T = 0.075 < Q^T_{v_s,v_t} = 0.12$). Then MFPB-HOSTP concatenates the FLP with BLP $p^{b(T)}_{v_2 \to v_t}$ to form another foreseen path $fp^{f(u)+b(T)}_{v_s \to v_2 \to v_t}$.

Algorithm 12: Computing Max_T ()

Data: $MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$
Result: $AQ^{\mu}(p^{b(T)}_{v_k \to v_t})$ and $AQ^{\mu}(p^{CBLP(T)}_{v_k \to v_t})$, $(\mu \in \{T, SI, CIF\})$
begin
  Set $v_x.d = \infty$ $(v_x \neq v_t)$, $v_t.d = 0$, $S_x = \emptyset$, $p^{b(T)}_{v_t \to v_t} = v_t$, $p^{CBLP(T)}_{v_t \to v_t} = v_t$;
  Add v_t into S_x;
  while $S_x \neq \emptyset$ **do**
    $v_a.d = min(v^*_a.d)$ $(v^*_a \in S_x)$;
    for *each* $v_b \in adj[v_a]$ **do**
      $obj = 1/AQ^T(p^{b(\delta^T)}_{v_a \to v_t} + v_a \to v_b)$;
      if $v_b \notin S_x$ **then**
        Put v_b into S_x;
        $p^{b(T)}_{v_b \to v_t} = v_b \to v_a + p^{b(T)}_{v_a \to v_t}$;
      end
      else if $obj < v_b.d$ **then**
        Update $AQ^T(p^{b(T)}_{v_b \to v_t})$;
        $v_b.d = obj$;
        $p^{b(T)}_{v_b \to v_t} = v_b \to v_a + p^{b(T)}_{v_a \to v_t}$;
      end
      for $i = 1$ *to* M **do**
        $p^{CBLP^i(T)}_{v_b \to v_t} = p^{CBLP^i(T)}_{v_a \to v_t}$;
        $AQ^{\mu}(p^{CBLP^i(T)}_{v_b \to v_t}) = AQ^{\mu}(p^{CBLP^i(T)}_{v_a \to v_t})$;
      end
      $p^{CBLP^{M+1}(T)}_{v_b \to v_t} = v_b \to v_a + p^{b(\delta)}_{v_a \to v_t}$;
    end
    Remove v_a from S_x;
  end
end

However, we could see there arises a new imbalance problem of SI, where the aggregated SI value of $fp^{f(u)+b(T)}_{v_s \to v_2 \to v_t}$ does not satisfy the corresponding QoT constraint ($r = 0.08 < Q^{SI}_{v_s,v_t} = 0.15$) and thus the foreseen path is infeasible. In such a situation, suppose $p^{b(\delta)}_{v_5 \to v_t} = v_5 \to v_t$, at v_2, MFPB-HOSTP identifies the CBLP $p^{CBLP^1(T)}_{v_2 \to v_t} = v_2 \to v_5 \to v_t$ and concatenates it with the FLP to balance the aggregated SI value. In such a situation, the foreseen path $fp^{f(u)+CBLP^1(T)}_{v_s \to v_2 \to v_t}$ is feasible. Assume the QoT attributes have the same weight in the utility function, with the assistance of CBLP $p^{CBLP^1(T)}_{v_2 \to v_t}$, MFPB-HOSTP could select the path $v_s \to v_1 \to v_2 \to v_5 \to v_t$ with the utility of 0.117 as the solution. Otherwise, the path $v_s \to v_1 \to v_3 \to v_6 \to v_t$ with the utility of 0.107 will be selected, which is worse than the one (i.e., utility is 0.117) identified with the assistance of CBLPs.

From this example, we could see that when facing with the new imbalance problem of QoT attributes caused by greedily maximizing the aggregated QoT attributes val-

Algorithm 13: Computing Max_SI ()

Data: $MT(v_s, v_t)$, $Q^T_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$
Result: $AQ^{\mu}(p^{b(SI)}_{v_k \to v_t})$ and $AQ^{\mu}(p^{CBLP(SI)}_{v_k \to v_t})$, $(\mu \in \{T, SI, CIF\})$
begin
  Set $v_x.d = \infty$ $(v_x \neq v_t)$, $v_t.d = 0$, $S_x = \emptyset$, $p^{b(r)}_{v_t \to v_t} = v_t$, $p^{CBLP(r)}_{v_t \to v_t} = v_t$;
  Add v_t into S_x;
  while $S_x \neq \emptyset$ **do**
    $v_a.d = min(v^*_a.d)$ $(v^*_a \in S_x)$;
    for *each* $v_b \in adj[v_a]$ **do**
      $obj = 1/AQ^{SI}(p^{b(\delta^{SI})}_{v_a \to v_t} + v_a \to v_b)$;
      if $v_b \notin S_x$ **then**
        Put v_b into S_x;
        $p^{b(SI)}_{v_b \to v_t} = v_b \to v_a + p^{b(SI)}_{v_a \to v_t}$;
      end
      else if $obj < v_b.d$ **then**
        Update $AQ^r(p^{b(r)}_{v_b \to v_t})$;
        $v_b.d = obj$;
        $p^{b(SI)}_{v_b \to v_t} = v_b \to v_a + p^{b(SI)}_{v_a \to v_t}$;
      end
      for $i = 1$ *to* M **do**
        $p^{CBLP^i(SI)}_{v_b \to v_t} = p^{CBLP^i(SI)}_{v_a \to v_t}$;
        $AQ^{\mu}(p^{CBLP^i(SI)}_{v_b \to v_t}) = AQ^{\mu}(p^{CBLP^i(SI)}_{v_a \to v_t})$;
      end
      $p^{CBLP^{M+1}(SI)}_{v_b \to v_t} = v_b \to v_a + p^{b(\delta)}_{v_a \to v_t}$;
    end
    Remove v_a from S_x;
  end
end

ues in BLPs, CBLPs could help avoid a failed feasibility estimation caused by a new imbalance problem of QoT attributes. Thus with the assistance of CBLPs, MFPB-HOSTP could deliver a better solution in some cases. In the process of identifying these BLPs and CBLPs, if there exist two overlapping paths (i.e., they have the same aggregated QoT attributes values), MFPB-HOSTP keeps only one of them for further search, saving execution time.

Step 3 (identify the BLP with the maximal aggregated SI value and the corresponding CBLPs):

(a): identify the BLPs with the maximal SI. Similar to *Step 2*, in order to avoid the imbalance problem of SI, in this step, at each intermediate node v_k, MFPB-HOSTP first identifies the BLP with the maximal aggregated SI value (denoted as $p^{b(SI)}_{v_k \to v_t}$) based on the Dijkstra's shortest path algorithm [31]. In this search process, at v_k, the aggregated values of QoT attributes of $p^{b(SI)}_{v_k \to v_t}$ are computed and recorded.

Algorithm 14: Computing Max_CIF ()

Data: $MT(v_s, v_t)$, $Q^{T}_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$
Result: $AQ^{\mu}(p^{b(CIF)}_{v_k \to v_t})$ and $AQ^{\mu}(p^{CBLP(CIF)}_{v_k \to v_t})$, $(\mu \in \{T, SI, CIF\})$
begin
  Set $v_x.d = \infty$ $(v_x \neq v_t)$, $v_t.d = 0$, $S_x = \emptyset$, $p^{b(CIF)}_{v_t \to v_t} = v_t$, $p^{CBLP(CIF)}_{v_t \to v_t} = v_t$;
  Add v_t into S_x;
  while $S_x \neq \emptyset$ **do**
    $v_a.d = min(v^*_a.d)$ $(v^*_a \in S_x)$;
    for *each* $v_b \in adj[v_a]$ **do**
      $obj = 1/AQ^{CIF}(p^{b(\delta^r)}_{v_a \to v_t} + v_a \to v_b)$;
      if $v_b \notin S_x$ **then**
        Put v_b into S_x;
        $p^{b(CIF)}_{v_b \to v_t} = v_b \to v_a + p^{b(CIF)}_{v_a \to v_t}$;
      end
      else if $obj < v_b.d$ **then**
        Update $AQ^{CIF}(p^{b(CIF)}_{v_b \to v_t})$;
        $v_b.d = obj$;
        $p^{b(CIF)}_{v_b \to v_t} = v_b \to v_a + p^{b(CIF)}_{v_a \to v_t}$;
      end
      for $i = 1$ *to* M **do**
        $p^{CBLP^i(CIF)}_{v_b \to v_t} = p^{CBLP^i(CIF)}_{v_a \to v_t}$;
        $AQ^{\mu}(p^{CBLP^i(CIF)}_{v_b \to v_t}) = AQ^{\mu}(p^{CBLP^i(CIF)}_{v_a \to v_t})$;
      end
      $p^{CBLP^{M+1}(CIF)}_{v_b \to v_t} = v_b \to v_a + p^{b(\delta)}_{v_a \to v_t}$;
    end
    Remove v_a from S_x;
  end
end

Algorithm 15: Path_Selection ()

Data: $MT(v_s, v_t)$, S_y, v_a, v_b
Result: $p^{f(u)}_{v_s \to v_b}$, $AQ^{\mu}(p^{f(u)}_{v_s \to v_b})$, $\mu \in \{T, SI, CIF\}$
begin
  if $v_b \notin S_y$ **then**
    Put v_b into S_y and $p^{f(u)}_{v_s \to v_b} = p^{f(u)}_{v_s \to v_a} + v_a \to v_b$;
  end
  else if $1/\mathcal{F}(p^{f(u)}_{v_s \to v_a} + v_a \to v_b) < v_b.d$ **then**
    Update $AQ^{\mu}(p^{f(u)}_{v_s \to v_b})$;
    $p^{f(u)}_{v_s \to v_b} = p^{f(u)}_{v_s \to v_a} + v_a \to v_b$;
  end
end

When facing with the imbalance problem of SI at v_k, BLP $p^{b(SI)}_{v_k \to v_t}$ is concatenated with the FLP $p^{f(u)}_{v_s \to v_k}$, forming a new foreseen path $fp^{f(u)+b(SI)}_{v_s \to v_k \to v_t}$. This foreseen path is used as a reference to estimate whether there exists a feasible solution identified by following $p^{f(u)}_{v_s \to v_k}$. This strategy could avoid a failed feasibility estimation of a foreseen path caused by the imbalance problem of SI at v_k.

(b): identify the CBLPs based on the BLPs with the maximal SI. To avoid the

Algorithm 16: Forward_Search ()

Data: $MT(v_s, v_t)$, $AQ^{\mu}(p_{v_k \to v_t}^{b(\delta)})$, $AQ^{\mu}(p_{v_k \to v_t}^{b(\mu)})$, $AQ^{\mu}(p_{v_k \to v_t}^{CBLP(\mu)})$, $\mu \in \{T, SI, CIF\}$, Q_{v_s,v_t}^{T}, Q_{v_s,v_t}^{SI}, Q_{v_s,v_t}^{CIF}
Result: $p_{v_s \to v_t}^{forward}$, $\mathcal{F}(p_{v_s \to v_t}^{forward})$
begin
  Set $v_y.d = \infty$ $(v_y \neq v_s)$, $v_s.d = 0$, $S_y^1 = S_y^2 = \emptyset$,$p_{v_s \to v_s}^{f(u)} = v_s$;
  Add v_s into S_y^1 and S_y^2;
  while $S_y^1 \neq \emptyset$ *and* $S_y^2 \neq \emptyset$ **do**
    $v_a^1.d = min(v_a^*.d)$ $(v_a^* \in S_y^1)$;
    $v_a^2.d = min(v_a^{2*}.d)$ $(v_a^{2*} \in S_y^2)$;
    if $v_a^1 = v_a^2$ *and* $v_a^1.d^1 = v_a^2.d^2$ **then**
      for *each* $v_b \in adj[v_a^1]$ **do**
        if $fp_{v_s \to v_b \to v_t}^{f(u)+b(\delta)}$ *is feasible* **then**
          Path_Selection($MT(v_s, v_t)$, S_y^1, v_a^1, v_b);
        end
        else if $fp_{v_s \to v_b \to v_t}^{f(u)+b(\delta)}$ *is infeasible* **then**
          if *one of* $\{fp_{v_s \to v_j \to v_t}^{f(u)+b(\mu)}\}$ *and* $\{fp_{v_s \to v_j \to v_t}^{f(u)+CBLP^M(\mu)}\}$ *is feasible* **then**
            Path_Selection($MT(v_s, v_t)$, S_y^2, v_a^1, v_b);
          end
        end
      end
    end
    else
      for *each* $v_b \in adj[v_a^1]$ **do**
        if $fp_{v_s \to v_b \to v_t}^{f(u)+b(\delta)}$ *is feasible* **then**
          Path_Selection($MT(v_s, v_t)$, S_y^1, v_a^1, v_b);
        end
      end
      for *each* $v_b \in adj[v_a^2]$ **do**
        if *one of* $\{fp_{v_s \to v_j \to v_t}^{f(u)+b(\mu)}, fp_{v_s \to v_j \to v_t}^{f(u)+CBLP^M(\mu)}\}$ *is feasible* **then**
          Path_Selection($MT(v_s, v_t)$, S_y^2, v_a^2, v_b);
        end
      end
    end
    Remove v_a^1 from S_y^1 and v_a^2 from S_y^2;
  end
  Return $p_{v_s \to v_t}^{forward}$ =max_utility($p_{v_s \to v_a^1 \to v_t}^{f(u)}, p_{v_s \to v_a^2 \to v_t}^{f(u)}$) and $\mathcal{F}(p_{v_s \to v_t}^{forward})$;
end

new imbalance problem of QoT attributes caused by greedily maximizing SI value, MFPB-HOSTP then identifies M CBLPs at each intermediate node v_k, which are composed of $p_{v_k \to v_l}^{b(SI)},\ l \in [1, M]$ and $p_{v_l \to v_t}^{b(\delta)},\ l \in [1, M]$. When facing with the new imbalance problem of QoT attributes caused by maximizing SI value, the identified M CBLPs at v_k are concatenated with the FLP $p_{v_s \to v_k}^{f(u)}$, to estimate whether there exists a feasible solution identified by following the FLP. This could help avoid a failed feasibility estimation of a foreseen path caused by the new imbalance problem of the other two QoT attributes (i.e., T and CIF) at v_k.

Step 4 (identify the BLP with the maximal aggregated CIF value and the corresponding CBLPs):

(a): identify the BLPs with the maximal CIF. To avoid the imbalance problem of CIF, in this step, at each intermediate node v_k, MFPB-HOSTP first identifies the BLP with the maximal aggregated CIF value (denoted as $p_{v_k \to v_t}^{b(CIF)}$) based on the Dijkstra's shortest path algorithm [31]. In this search process, at each v_k, the aggregated QoT attributes values of $p_{v_k \to v_t}^{b(CIF)}$ are computed and recorded. When facing with the imbalance problem of CIF at v_k, BLP $p_{v_k \to v_t}^{b(CIF)}$ is concatenated with the FLP $p_{v_k \to v_t}^{f(u)}$, forming a new foreseen path $fp_{v_s \to v_k \to v_t}^{f(u)+b(CIF)}$. This strategy could help avoid a failed feasibility estimation of a foreseen path caused by the imbalance problem of CIF at v_k.

(b): identify the CBLPs based on the BLPs with the maximal CIF. To avoid the new imbalance problems of QoT attributes caused by greedily maximizing CIF value, MFPB-HOSTP then identifies M CBLPs at each intermediate node v_k, which are composed of $p_{v_k \to v_l}^{b(CIF)},\ l \in [1, M]$ and $p_{v_l \to v_t}^{b(\delta)},\ l \in [1, M]$. When facing with the new imbalance problem of QoT attributes caused by the BLP with the maximal CIF at v_k, the M CBLPs at v_k are concatenated with the FLP $p_{v_s \to v_k}^{f(u)}$, to estimate the feasibility of searching by following the FLP. This could avoid a failed feasibility estimation of a foreseen path caused by the new imbalance problem of the other two QoT attributes (i.e., T and SI) at v_k.

In summary, the *Backward_Search* procedure can illustrate whether there exists a feasible solution in a sub-network. In addition, if a feasible solution exists, compared with the *Backward_Search* procedure of H_OSTP, MFPB-HOSTP identifies the BLP with the maximal aggregated value of each of the QoT attributes. Furthermore, to solve a new imbalance problem of QoT attributes caused by greedily maximizing the aggregated values of QoT attributes, MFPB-HOSTP also identifies several CBLPs, which are composed of part of the BLP with the minimal δ and part of the BLP with the maximal aggregated value of each of the QoT attributes. When facing with an imbalance problem of QoT attributes, the identified BLPs and CBLPs will be used in

the following *Forward_Search* procedure aiming to avoid a failed feasibility estimation of a foreseen path in H_OSTP and deliver a near-optimal solution. Next we discuss the search strategies adopted in the following *Forward_Search* procedure of MFPB-HOSTP.

Forward_Search: In the forward search from v_s to v_t, MFPB-HOSTP uses the BLPs and CBLPs identified by the above *Backward_Search* procedure to investigate whether there exists another path $p_{v_s \to v_t}^{forward}$, which is better in quality than the above path $p_{v_s \to v_t}^{backward} = p_{v_s \to v_t}^{b(\delta)}$ returned in the *Backward_Search* procedure (i.e., whether $\mathcal{F}(p_{v_s \to v_t}^{forward}) > \mathcal{F}(p_{v_s \to v_t}^{backward})$).

In this procedure, MFPB-HOSTP searches the path with the maximal $\mathcal{F}$ value from v_s to v_t. Assume node $v_m \in \{neighboring\ nodes\ of\ v_s\}$ is selected based on the Dijkstra's shortest path algorithm (i.e., FLP $p_{v_s \to v_m}^{f(u)}$ is identified). Then, MFPB-HOSTP concatenates the FLP with BLP $p_{v_m \to v_t}^{b(\delta)}$ to form a foreseen path $fp_{v_s \to v_m \to v_t}^{f(u)+b(\delta)}$. If the foreseen path is feasible, MFPB-HOSTP then chooses the next node from v_m with the maximal $\mathcal{F}$ value. Otherwise, MFPB-HOSTP concatenates the FLP with the BLPs with the minimal T, SI and CIF respectively to form three foreseen paths $\{fp_{v_s \to v_m \to v_t}^{f(u)+BLP(\mu)}\ (\mu \in \{T, SI, CIF\})\}$. According to the feasibility of these foreseen paths, MFPB-HOSTP adopts the following search strategies.

Situation 1: If one of $\{fp_{v_s \to v_m \to v_t}^{f(u)+b(\mu)}\ (\mu \in \{T, SI, CIF\})\}$ is feasible, MFPB-HOSTP adopts the following two strategies to identify two social trust paths and selects the feasible social trust path with the higher utility value as the final solution.

1. **Strategy 1:** MFPB-HOSTP identifies one path by choosing the next node from v_m with the maximal $\mathcal{F}$ value.

2. **Strategy 2:** MFPB-HOSTP identifies another path by searching another neighboring node of v_s with the maximal $\mathcal{F}$, which is the same as the search strategy adopted in H_OSTP.

Situation 2: If all $\{fp_{v_s \to v_m \to v_t}^{f(u)+b(\mu)}\ \mu \in \{T, SI, CIF\}\}$ are infeasible, then at v_m, MFPB-HOSTP concatenates the FLP with the CBLPs to form the foreseen paths

(i.e., $\{fp_{v_s \to v_m \to v_t}^{f(u)+CBLP^M(\mu)}$). According to the feasibility of these foreseen paths, MFPB-HOSTP adopts the following search strategies.

1. **Sub-situation 2.1:** If one of $\{fp_{v_s \to v_m \to v_t}^{f(u)+CBLP^M(\mu)}$ is feasible, MFPB-HOSTP identifies two social trust paths based on *Strategies 1 and 2* in the above *Situation 1*, and selects the feasible social trust path with the higher utility as the final solution.

2. **Sub-situation 2.2:** If all of $\{fp_{v_s \to v_m \to v_t}^{f(u)+CBLP^M(\mu)}$ are infeasible, MFPB-HOSTP does not search the path from v_m. Instead, MFPB-HOSTP performs the *Forward_Search* procedure to search the path from v_s in the sub-network without taking link $v_s \to v_m$ into consideration.

Based on the *Theorem 2* in H_OSTP, we can see that the social trust path $p_{v_s \to v_t}^{forward}$ identified by the *Forward_Search* procedure of MFPB-HOSTP can not be worse than the feasible social trust path $p_{v_s \to v_t}^{backward}$ identified by the *Backward_Search* procedure. Namely, $\mathcal{F}(p_{v_s \to v_t}^{forward}) \geq \mathcal{F}(p_{v_s \to v_t}^{backward})$. In addition, if there exists only one feasible solution in the sub-network, it can be identified by both the *Backward_Search* procedure and the *Forward_Seach* procedure, and it is the optimal solution. Otherwise, if there exist more than one feasible solutions in the sub-network, then the solution identified by the *Forward_Seach* procedure is near-optimal or optimal, which is better than the one identified by the *Backward_Search* procedure.

5.6.4 Summary

Based on the above discussion, during the *Backward_Search* procedure, MFPB-HOSTP could illustrate whether there exists a feasible solution in a sub-network (it is proved by *Theorem 1* in H_OSTP). If a feasible solution exists, MFPB-HOSTP then identifies several BLPs and CBLPs at each intermediate node rather than only one BLP in H_OSTP. During the *Forward_Search* procedure, MFPB-HOSTP delivers a near-optimal solution which is no worse than the one returned by the *Backward_Search*

procedure (it is proved by *Theorem 2* in H_OSTP). In this search process, the identified BLPs and CBLPs are used to concatenate with the FLP, forming multiple foreseen paths rather than one foreseen path only in H_OSTP. These foreseen paths could help avoid a failed feasibility estimation of a foreseen path caused by the imbalance problem of QoT attributes.

In the *Backward_Search* procedure, in order to identify 4 BLPs for the minimal δ and the maximal value of each QoT attribute (i.e, T, SI and CIF), MFPB-HOSTP adopts the Dijkstra's shortest path algorithm 4 times with the time complexity of $O(4*(NlogN+E))$ [31] (N is the number of nodes and E is the number of links). In addition, in the worst case, the time complexity of identifying the CBLPs for three QoT attributes by MFPB-HOSTP is $O(3*(KN))$, where K is the maximal path length in a sub-network. So, the time complexity of the Backward_Search procedure is $O(4*(NlogN+E)+3*KN)$.

In the *Forward_Search* procedure, in the worst case, MFPB-HOSTP adopts the Dijkstra's shortest path algorithm twice with the time complexity of $O(2*(NlogN+E))$ [31]. In addition, in the worst case, the time complexity of evaluating the feasibility of foreseen paths is $O(KE)$. So, the time complexity of MFPB-HOSTP is $O(NlogN+KE)$.

In social networks, following the *small-world*[2] characteristic, it is usually the case that $K \leq 7$ [101]. Therefore, the time complexity of MFPB-HOSTP is $O(NlogN+E)$, which is the same as that of H_OSTP. But our proposed heuristic algorithm has better search strategies than H_OSTP. Thus MFPB-HOSTP delivers a solution no worse than that of H_OSTP, and as our experiments confirm, MFPB-HOSTP can deliver better solutions than H_OSTP in some cases.

[2]The average path length between any two nodes is about 6 hops in a social network.

Table 5.7: The setting of QoT constraints

Constraint ID	$Q^{T}_{v_s,v_t}$	$Q^{SI}_{v_s,v_t}$	$Q^{CIF}_{v_s,v_t}$
1	0.01	0.01	0.01
2	0.05	0.05	0.05
3	0.1	0.1	0.1
4	0.15	0.15	0.15
5	0.2	0.2	0.2
6	0.25	0.25	0.25
7	0.3	0.3	0.3
8	0.35	0.35	0.35
9	0.4	0.4	0.4
10	0.2	0.05	0.05
11	0.05	0.2	0.05
12	0.05	0.05	0.2
13	0.25	0.05	0.05
14	0.05	0.25	0.05
15	0.05	0.05	0.25
16	0.3	0.05	0.05
17	0.05	0.3	0.05
18	0.05	0.05	0.3
19	0.35	0.05	0.05
20	0.05	0.35	0.05
21	0.05	0.05	0.35
22	0.4	0.05	0.05
23	0.05	0.4	0.05
24	0.05	0.05	0.4

5.7 Experiments on MFPB-HOSTP

5.7.1 Experiment Settings

We select the *Enron* email dataset with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) for our experiments. As we analysed in Chapter 5.4, our proposed H_OSTP outperforms prior algorithms in both efficiency and the quality of identified social trust path. Therefore, in order to study the performance of our proposed algorithm, we compare MFPB-HOSTP with H_OSTP in both execution time and the utilities of the identified social trust paths. In our experiments, since the detailed mining method of QoT attribute values (i.e., T, SI and CIF) is out

Table 5.8: The setting of the weight of QoT attributes

Weight ID	w_T	w_{SI}	w_{CIF}
1	0.5	0.25	0.25
2	0.25	0.5	0.25
3	0.25	0.25	0.5

Table 5.9: 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	20	393	1543
5	1	49	90	20	680	2670
6	1	48	74	20	1300	6396
7	1	40	64	20	964	4955

of the scope of this thesis, and they could have different values in different applications, the QoT attribute values are randomly generated by using *rand()* in Matlab.

Source participants may specify different QoT constraints for the social trust path selection in different domains. In order to investigate the performance of MFPB-HOSTP with different QoT constraints values, 24 sets of QoT constraints are specified and listed in Table 5.7, which cover some possible settings of QoT constraints. In some cases (i.e., constraint IDs 1 to 9), the values of QoT constraints are the same, and in the rest of the cases (i.e., constraint IDs 10 to 24), the constraint of one QoT attribute (i.e., T, SI or CIF) is larger than the values of the other two QoT attributes. In addition, in order to investigate the performance of MFPB-HOSTP in path selection with different weights of the QoT attributes in the utility function, three sets of weights are specified and listed in Table 5.8, where T, SI and CIF are given a lager weight than other two QoT attributes respectively.

In order to study the performance of our proposed heuristic algorithm in the sub-networks of different scales and structures, we first randomly select 80 pairs of source and target participants from the *Enron* email dataset[1]. We then extract the corresponding 80 sub-networks between them by using the exhaustive search method. Among them, the maximal length of a social trust path varies from 4 to 7 hops following the

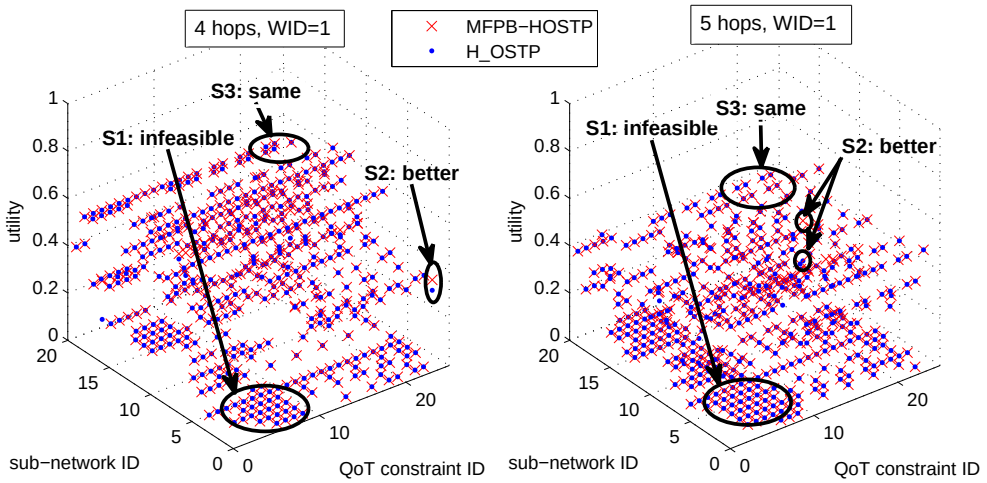

Figure 5.10: The path utilities of sub-networks with 4 and 5 hops based on WID=1

small-world characteristic. These sub-networks are grouped by the number of hops. In each group they are ordered by the number of nodes in them. Table 5.9 lists the properties of the simplest and the most complex sub-networks in each group of hops. The simplest sub-network has 33 nodes and 56 links (4 hops), while the most complex sub-network has 1300 nodes and 6396 links (6 hops). With each sub-network, we run MFPB-HOSTP and H_OSTP 3 times independently to calculate the average execution time.

Both MFPB-HOSTP and H_OSTP 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.

5.7.2 Results and Analysis

Results and analysis of path utility. Fig. 5.10 to Fig. 5.15 plot the path utilities of the identified social trust paths in the sub-networks categorised in groups of hops. From these figures, we can observe that if there are no feasible solutions in a sub-network, both of MFPB-HOSTP and H_OSTP can investigate the infeasibility (e.g., case S1 in Fig. 5.10 to Fig. 5.15). This is because both of them perform a backward search

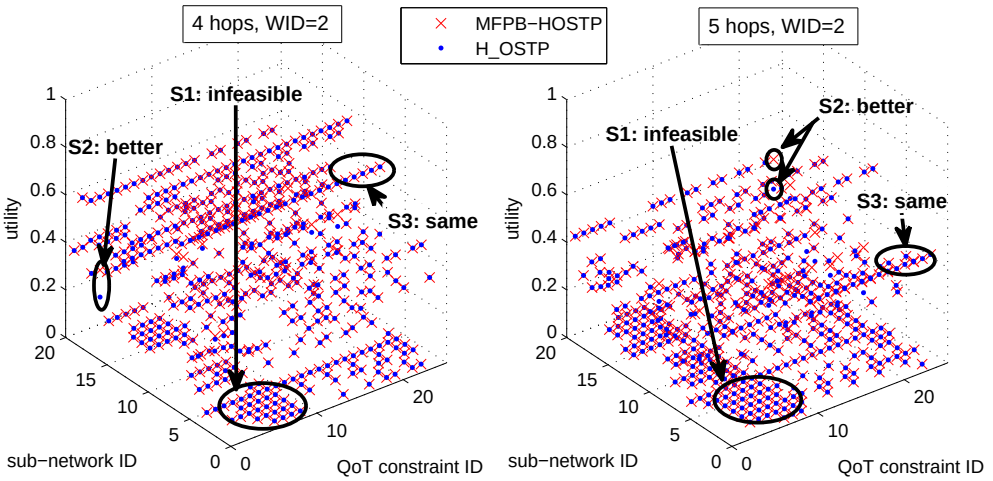

Figure 5.11: The path utilities of sub-networks with 4 and 5 hops based on WID=2

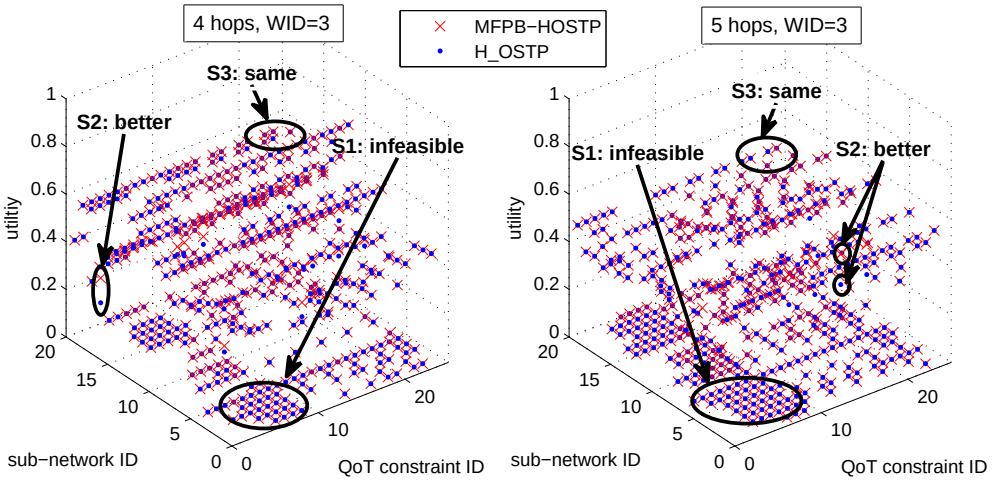

Figure 5.12: The path utilities of sub-networks with 4 and 5 hops based on WID=3

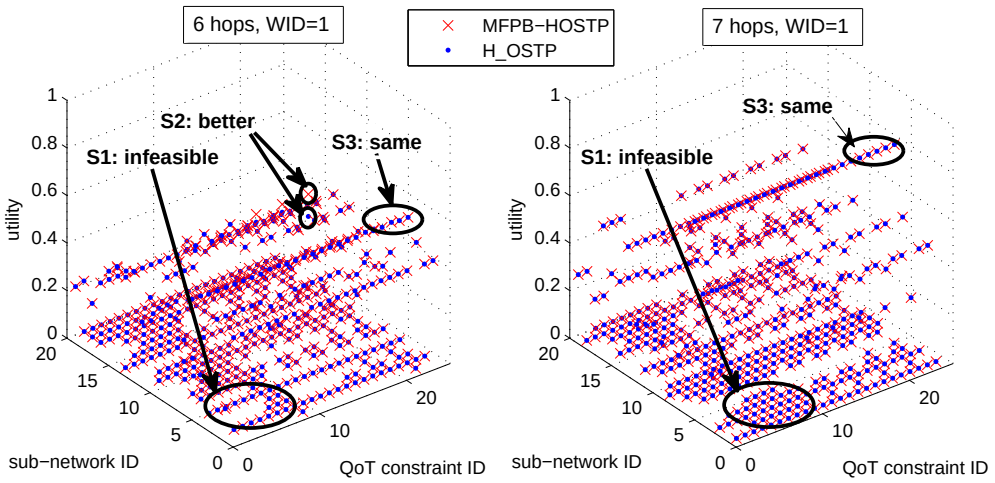

Figure 5.13: The path utilities of sub-networks with 6 and 7 hops based on WID=1

from v_t to v_s to identify the social trust path with the minimal δ. It has been proved in *Theorem 1* that this procedure can always investigate whether there exists a feasible solution in a sub-network.

From Fig. 5.10 to Fig. 5.15, we can see that in all cases of the 80 sub-networks, our MFPB-HOSTP does not yield any feasible social trust path with a utility worse than that of H_OSTP (e.g., cases S2 and S3 in Fig. 5.10 to Fig. 5.15). This is because in the *Forward_Search* procedure, if there is no imbalance problem of QoT attributes, MFPB-HOSTP identifies the same social trust path with H_OSTP. When facing with an imbalance problem of QoT attributes, MFPB-HOSTP identifies two social trust paths, out of which one path is identified by using the same search strategy adopted in H_OSTP (see *Strategy 2* of *Situation 1* in *Section 5.6.3*), and selects the feasible path with the higher utility as the solution. Therefore, MFPB-HOSTP does not yield any solution worse than that of H_OSTP in any cases.

According to our experimental results, in 27 out of 75 sub-networks with feasible solutions (i.e., 36% of total sub-networks with feasible solutions), MFPB-HOSTP can deliver better social trust paths than H_OSTP (e.g., case S2 in Fig. 5.10 to Fig. 5.15). The sums of utilities computed by MFPB-HOSTP and H_OSTP in these sub-

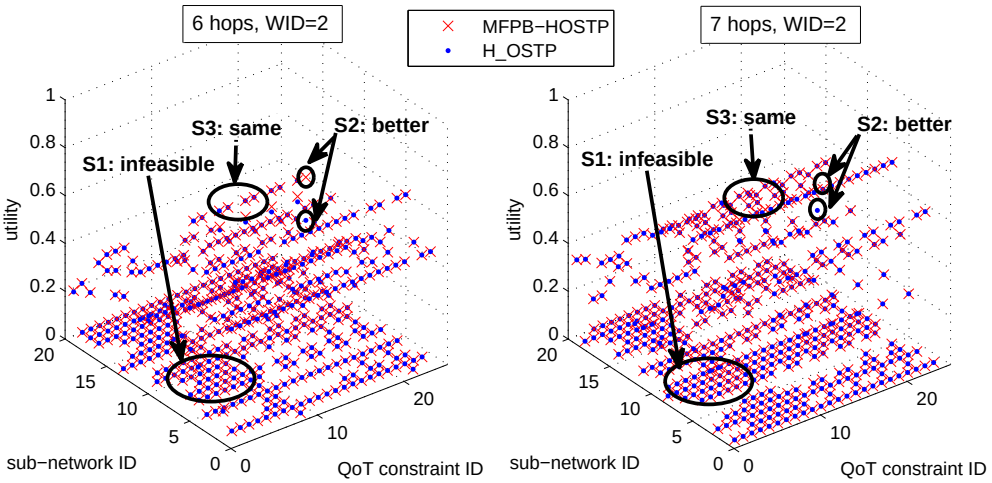

Figure 5.14: The path utilities of sub-networks with 6 and 7 hops based on WID=2

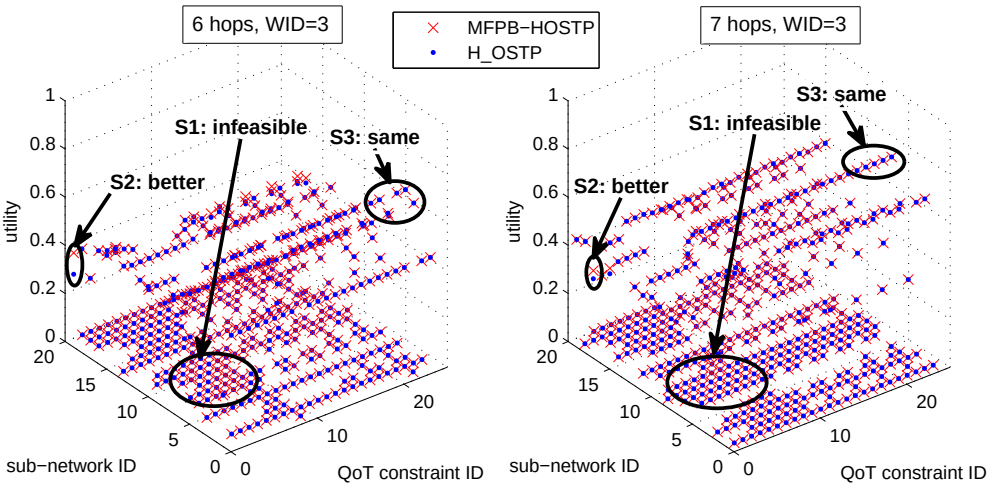

Figure 5.15: The path utilities of sub-networks with 6 and 7 hops based on WID=3

networks with each group of hops are listed in Table 5.10, where we can see that the sum of utilities of our proposed MFPB-HOSTP algorithm is 15.94% more than that of H_OSTP in 4 hops sub-networks, 46.51% more in 5 hops, 12.63% more in 6 hops and 17.79% more in 7 hops. This is because when facing with an imbalance problem of QoT attributes at an intermediate node v_k, in addition to $p^{b(\delta)}_{v_k \to v_t}$, more BLPs are concatenated with the FLP identified by the forward search procedure, forming multiple foreseen paths and helping avoid a failed feasibility estimation. Thus MFPB-HOSTP can deliver a better solution than H_OSTP in some cases.

Table 5.10: The comparison of path utility

Algorithms	The sum of path utility (sec)				
	4 hops	5 hops	6 hops	7 hops	total
MFPB-HOSTP	11.7634	11.2517	6.3161	2.1140	31.4452
H_OSTP	10.1459	7.6797	5.6076	1.7947	25.2279
difference	15.94%more	46.51%more	12.63%more	17.79%more	24.64%more

Results and analysis of the execution time.

Fig. 5.16 to Fig. 5.17 plot the average execution time of the social trust path selection with three different weights of QoT attributes. From these figures we can see that in most cases (i.e., 3082/5760=53.5% of total cases), MFPB-HOSTP has the same execution time as that of H_OSTP (e.g., case S4 in Fig. 5.16 to Fig. 5.17). This is because if no feasible solution exists in the sub-network, based on *Theorem 1* in H_OSTP, both of MFPB-HOSTP and H_OSTP can identify this and stop the search process, resulting in the same execution time. In addition, in the rest of the cases, MFPB-HOSTP consumes more execution time than H_OSTP (e.g., case S5 in Fig. 5.16 to Fig. 5.17). This is because if a feasible solution exists in a sub-network, at each intermediate node v_k, in addition to $p^{b(\delta)}_{v_s \to v_k}$, MFPB-HOSTP identifies multiple BLPs (i.e., the BLPs with the maximal aggregated value of each of QoT attribute and M CBLPs for each QoT attribute) in the *Backward_Search* procedure, rather than one BLP only in H_OSTP (see Section 5.6.3). Moreover, when facing with the imbalance problem of QoT attributes at v_k, MFPB-HOSTP needs to identify two social trust paths. The total execution time of each of MFPB-HOSTP and H_OSTP in sub-networks with each group of hops is listed in Table 5.11, where we conclude that the difference of the execution

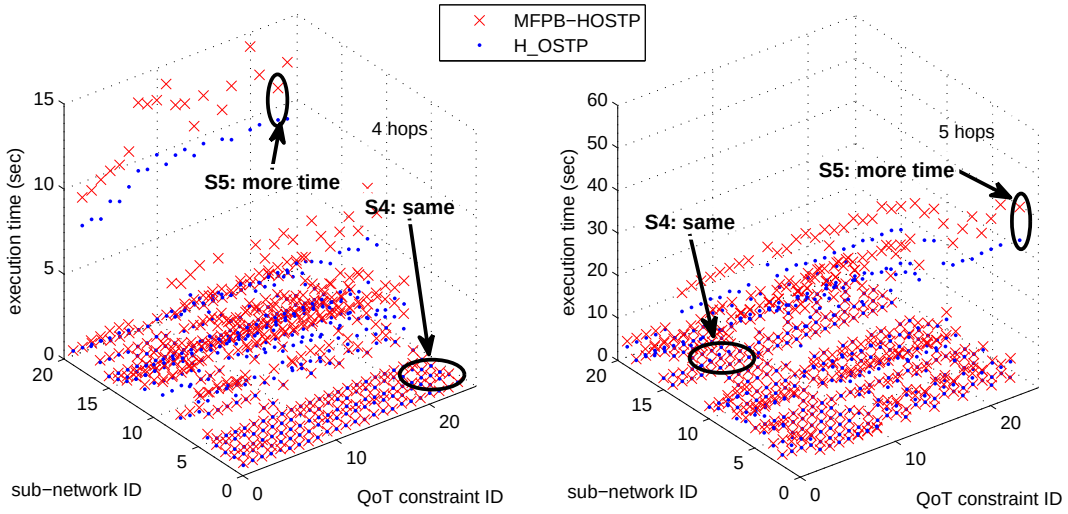

Figure 5.16: The execution time of sub-networks with 4 and 5 hops

time between MFPB-HOSTP and H_OSTP is similar in sub-networks with each group of hops. On average, the execution time of MFPB-HOSTP is 1.288 times of that of H_OSTP.

Through the above experiments conducted on the sub-networks with different scales and structures, we can see that on average MFPB-HOSTP consumes 1.288 times of the execution time of H_OSTP while delivering better solutions in sub-networks. Since MFPB-HOSTP has the same polynomial time complexity (i.e, $O(NlogN + E)$) as H_OSTP, MFPB-HOSTP is superior to H_OSTP when applied to large-scale social networks.

Table 5.11: The comparison of execution time

Algorithms	The sum of execution time (sec)				
	4 hops	5 hops	6 hops	7 hops	total
MFPB-HOSTP	7.6478e+003	2.3537e+004	2.5621e+004	4.2355e+004	9.9161e+004
H_OSTP	5.7831e+003	1.8529e+004	1.9903e+004	3.2776e+004	7.6991e+004
difference	1.3224:1	1.2703:1	1.2873:1	1.2922:1	1.2880:1

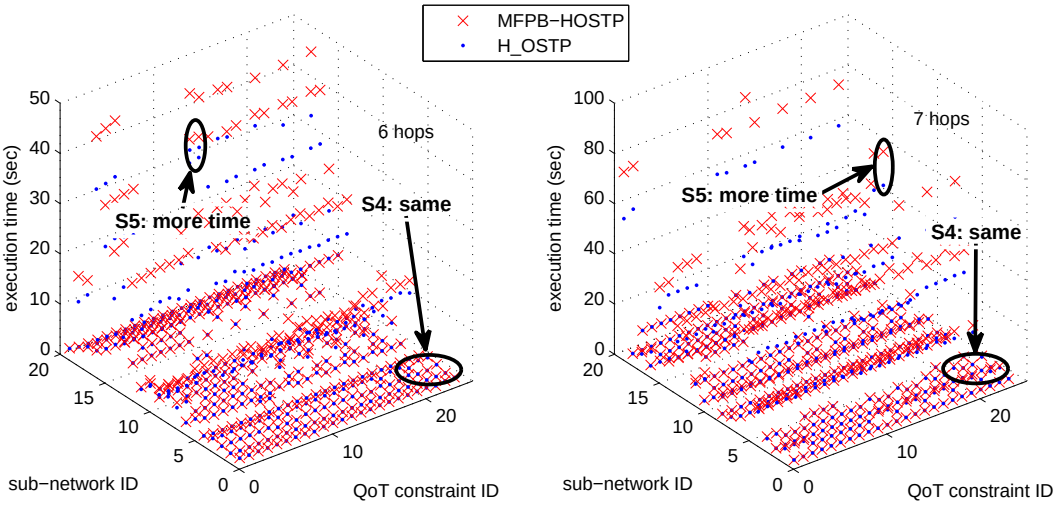

Figure 5.17: The execution time of sub-networks with 6 and 7 hops

5.8 Conclusion

In this Chapter, we have proposed a general concept QoT (Quality of Trust), and modelled the QoT constrained optimal social trust path selection as the classical Multiple Constrained Optimal Path (MCOP) problem, which is NP-Complete [67]. For solving the NP-Complete optimal social trust path selection problem, we have proposed an approximation algorithm, called MONTE_K, by adopting the Monte Carlo method and our proposed optimization strategies. In addition, we have proposed a heuristic algorithm, called H_OSTP based on the Dijkstra's shortest path algorithm and our proposed optimization strategies. Furthermore, to address the drawbacks included in H_OSTP (i.e., the imbalance problem of QoT attributes), we have proposed a heuristic algorithm, called MFPB-HOSTP, an efficient heuristic algorithm, where multiple foreseen paths are formed, helping avoid a failed feasibility estimation of a foreseen path caused by the imbalance problem of QoT attributes. The results of experiments conducted on real social network datasets demonstrate that the proposed methods outperform the existing methods in optimal social trust path selection with good efficiency.

The trust path selection methods proposed in this chapter can select the most trust-

worthy social trust path to efficiently and effectively deliver reasonable propagated trust values between two unknown participants, which is significant and essential in the trust management of OSNs. In social network based real applications, the proposed methods can for instance help a buyer to find the most trustworthy seller who sells the products preferred by the buyer, or help an employer to find the most trustworthy potential employees.

Chapter 6

Finding K Optimal Social Trust Paths

In *Chapter 5*, we have introduced the proposed optimal social trust path algorithms. But these studies focus on selecting only one social trust path between a source participant and a target participant. As illustrated in cognitive science [68], people are willing to believe what they have been told most often and by the possibility of the greatest number of different sources. Therefore, in order to obtain a more reasonable trust evaluation result of a target participant, a source participant may refer to multiple social trust paths from the source participant to the target one. This requires identifying K ($K \geq 2$) optimal social trust paths, yielding the K most trustworthy trust propagation results based on the constraints specified by the source participant. Since the selection of any one of the K optimal social trust paths based on multiple constrains is the classical MCOP selection problem, which has been proved to be NP-Complete [67], the Multiple Constrained K Optimal Social Trust Paths (MCOP-K) selection is also an NP-Complete problem. But the existing algorithms [33, 94, 100] for K paths selection attempt to find the K shortest paths without any end-to-end constrains, and this is not an NP-Complete problem. Thus, they cannot be used for the MCOP-K selection problem.

In this chapter, in order to solve the NP-Complete MCOP-K selection problem in complex trust-oriented social networks, we propose a new efficient Heuristic algorithm for the K Optimal Social Trust Path selection based on the Dijkstra's shortest path algorithm [31] and our optimization strategies, called H-OSTP-K. In addition, we have conducted extensive experiments on a real online social network dataset, the *Enron*

email dataset. Experimental results demonstrate that H-OSTP-K outperforms existing methods in the quality of identified social trust paths and the efficiency.

6.1 K Optimal Social Trust Paths Selection

In this section, we first analyse some existing algorithms for K shortest paths selection and then propose an efficient heuristic algorithm H-OSTP-K for the NP-Complete MCOP-K selection in complex social networks.

6.2 Existing Algorithms

K shortest paths selection has been used in many applications, such as power transmission route selection, automatic translation between natural languages, and biological sequence alignment [33]. In the literature, several algorithms have been proposed to solve the K shortest paths selection problem, including (1) algorithms to find K *general shortest path* (paths allowing loops), and (2) algorithms to find K *simple shortest paths* (paths without loops) [33]. As a social trust path may contain loops [48], we introduce some existing algorithms for finding K general shortest paths as follows.

The algorithms for finding K general shortest paths can be classified into two categories. They are (1) K general paths selection based on the Dijkstra's shortest algorithm [31], and (2) K general paths selection based on A^* algorithm.

6.2.1 Category 1

In *Category 1*, Fox [39] proposed a K paths selection algorithm, where each intermediate node v_k, $(v_k \neq v_s)$ records up to K minimal path lengths from v_s to v_k. At each step, up to K nodes are selected from a priority queue as the expansion nodes based on the maximal path length record at the nodes. If a node is selected, the algorithm counts the number of times it has been visited. If all the nodes have been visited K times, the K shortest paths from v_s to each node of the sub-network are selected. Miaou

[100] proposed a similar algorithm by using a binary heap to store the priority queue, which improves the efficiency of K path selection. The time complexity of this type of algorithm is $O(m + Knlogn)$. Throughout this chapter, K $(K \geq 2)$ stands for the number of selected paths, m for the number of links, and n for the number of nodes.

6.2.2 Category 2

In *Category 2*, Yen proposed a classic K general shortest paths selection algorithm based on the A^* algorithm [133]. This algorithm first computes the shortest path from v_s to v_t. Then it regards each node of the newly discovered shortest path as a *deviation node*. For each deviation node, this algorithm executes a single-source shortest path algorithm from the deviation node to v_t, forming a candidate *deviation path*. The next shortest path is chosen from all the candidates deviation paths with the minimal path length. This process continues until K different shortest paths are finally determined. In addition, Martins [94] improved the runtime performance of Yen's algorithm by ordering the deviation node based on deviation paths' length. Furthermore, Eppsten [33] proposed a well-known K general shortest paths selection algorithm. This algorithm builds a shortest path tree rooted at the target node first, and then selects certain links outside the shortest path tree, forming the paths to be discovered. The time complexity of Eppsten's algorithm reaches $O(m + nlogn + K)$, which is also the lowest bound of the K general paths selection problem.

The above algorithms address the K general shortest path selection problem well. However, they are all deterministic and thus cannot be used to solve the NP-Complete MCOP-K selection problem [6].

6.3 The Proposed H-OSTP-K for K Optimal Social Trust Paths Selection

In this section, we propose a novel heuristic algorithm H-OSTP-K, for the K optimal social trust path selection with end-to-end QoT constraints in complex social networks. In H-OSTP-K, we first adopt the *Backward_K-Search* procedure from v_t to v_s to (1) investigate whether there exists a feasible solution in the sub-network, (2) indicate the number of feasible solutions when this number is less than K ($K \geq 2$), and (3) record the aggregated QoT attributes (i.e., T, SI and CIF) of the identified K paths from v_t to each intermediate node v_k. If there exists at least one feasible solution, we then adopt the *Forward_K-Search* procedure to search the network from v_s to v_t to deliver up to K near-optimal solutions with the K best utilities (the utility function has been defined in *Section 5.1.3*).

In MOCP-K selection, if a path satisfies multiple QoT constraints, it means that each aggregated QoT attribute of that path should be larger than the corresponding QoT constraint. Based on this observation, we adopt the objective function proposed in *Section 5.4.1* (Eq. (5.7)) to investigate whether the aggregated QoT attributes of a path can satisfy the QoT constraints, where $\delta(p) \leq 1$, *if and only if* each aggregated QoT attribute of a social trust path satisfies the corresponding QoT constraint. Otherwise $\delta(p) > 1$.

6.3.1 Algorithm Description of H-OSTP-K

Backward_K-Search: Assume there exist at least K social trust paths in the sub-network. In the backward search from v_t to v_s, H-OSTP-K identifies K social trust paths from v_t to v_s (denoted as $p^{B_1}_{v_s \to v_t}$ to $p^{B_K}_{v_s \to v_t}$) with the K minimal δ based on the Dijkstra's shortest path algorithm [31]. In the searching process, at v_k, the aggregated QoT attributes of K paths from v_t to v_k with the K minimal δ are recorded. According to the *Theorem 1* proposed in *Section 5.4.1*, the *Backward_K-Search* procedure can

investigate whether there exists a feasible solution in the sub-network. In addition, according to *Theorem 3* given below, this procedure can also indicate the number of feasible solutions when there exist less than K feasible solutions in the sub-network (*see Algorithm 18*).

Theorem 3: In the *Backward_K-Search* procedure, the process of identifying K paths with the K minimal δ can indicate the number of feasible solutions when there exist less than K feasible solutions in a sub-network.

Proof: Let $p^{B_1}_{v_s \to v_t}, ..., p^{B_K}_{v_s \to v_t}$ be the K paths identified by the *Backward_Search* procedure from v_t to v_s with the K minimal δ value, and $\mathcal{S}$ is the number of feasible solutions in the subnetwork between v_s and v_t. In the identified K paths from v_s to v_t, if there exists G $(0 < G < K)$ paths (denoted as $p^{B_1}_{v_s \to v_t}, ..., p^{B_G}_{v_s \to v_t}$), where $\delta(p^{B_1}_{v_s \to v_t}) \leq 1, ..., \delta(p^{B_G}_{v_s \to v_t}) \leq 1$, then based the theorems in proposed in *Section 5.4.1*, there exist at least G feasible solutions in the sub-network between v_s and v_t (i.e., $\mathcal{S} \geq G$). In addition, the *Backward_Search* procedure can always identify K paths with K minimal δ value [100]. Therefore, there exist no more than G feasible solutions in the sub-network between v_s to v_t (i.e., $\mathcal{S} \leq G$). Then $\mathcal{S} = G$. □

Without loss of generality, we assume there are at least K social trust paths in the sub-network, though not all of them are feasible solutions. The *Backward_K-Search* can always identify K paths with the K minimal δ. In all the identified K paths, 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. In addition, if there exist G $(0 < G < K)$ paths, where the δ values of these paths are no more than one, it means there are G feasible solutions in the sub-network.

Forward_K-Search: Assume there exist at least K $(K \geq 2)$ feasible solutions in the sub-network. In the *Forward_K-Search* procedure, the aggregated QoT attribute values recorded at each v_k is adopted to identify whether there exist further K paths $p^{F_1}_{v_s \to v_t}, ..., p^{F_K}_{v_s \to v_t}$, each of which is better than the corresponding path of $p^{B_1}_{v_s \to v_t}, ..., p^{B_K}_{v_s \to v_t}$ (i.e.,$\mathcal{F}(p^{F_1}_{v_s \to v_t}) > \mathcal{F}(p^{B_1}_{v_s \to v_t}), ..., \mathcal{F}(p^{F_K}_{v_s \to v_t}) > \mathcal{F}(p^{B_K}_{v_s \to v_t})$ (*see Algorithm 19*).

Algorithm 17: H-OSTP-K

Data: $M, Q^T_{v_s,v_t}, Q^{SI}_{v_s,v_t}, Q^{CIF}_{v_s,v_t}, v_s, v_t, K$
/ M is an adjacency matrix that represents the sub-network between v_s and v_t */*
Result: $\mathcal{F}(p^{F_1}_{v_s \to v_t}) ... \mathcal{F}(p^{F_G}_{v_s \to v_t})$

```
begin
    p_s = ∅, p_t = ∅;
    Backward_K-Search (M, Q^T_{v_s,v_t}, Q^{SI}_{v_s,v_t}, Q^{CIF}_{v_s,v_t}, v_s, v_t, K);
    if Min δ(p^{B_1}_{v_s→v_t}) ... δ(p^{B_K}_{v_s→v_t}) > 1 then
        return no feasible solution;
    end
    else
        return G, BAQoT(v).T, BAQoT(v).SI, BAQoT(v).CIF;
        /* G is the number of feasible solution identified by the Backward_K-Search procedure, and BAQoT records
        the aggregated QoT attributes in the backward search. */
        Forward_K-Search (M, BAQoT(v).T, BAQoT(v).SI, BAQoT(v).CIF, Q^T_{v_s,v_t}, Q^{SI}_{v_s,v_t}, Q^{CIF}_{v_s,v_t},
        v_s, v_t, G);
        Return F(p^{F_1}_{v_s→v_t}), ..., F(p^{F_G}_{v_s→v_t});
    end
end
```

In this procedure, H-OSTP-K first searches the path with the K 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 in the i^{th} path ($i \in [1, K]$). H-OSTP-K calculates the aggregated QoT attribute values of the path from v_s to v_m (denoted as path $p^{F_i}_{v_s \to v_m}$). Then K *foreseen paths* from v_s to v_t via v_m (denoted as $fp^{F_i+B_\sigma}_{v_s \to v_m \to v_t} = p^{F_i}_{v_s \to v_m} + p^{B_\sigma}_{v_m \to v_t}$ ($\sigma \in [1, K]$)) are formed. Depending on whether $fp^{F_i+B_\sigma}_{v_s \to v_m \to v_t}$ is feasible, H-OSTP-K adopts the following searching strategies.

Situation 1: If each aggregated QoT attribute of one of the foreseen paths from v_s to v_t via v_m, (i.e., $fp^{F_i+B_\sigma}_{v_s \to v_m \to v_t}$ ($\sigma \in [1, K]$) satisfies the corresponding end-to-end QoT constraint, then v_m is put into the priority queue for the next search step.

Situation 2: If all the foreseen paths $fp^{F_i+B_\sigma}_{v_s \to v_m \to v_t}$ ($\sigma \in [1, K]$) are infeasible, v_m is not put into the priority queue. Subsequently, H-OSTP-K performs the *Forward_K-Search* procedure to search the path from v_s in the sub-network without taking the link $v_s \to v_m$ into consideration.

Theorem 4: If v_t is selected from the priority queue, then a social trust path from v_s to v_t is identified (denoted as p_t). If any of the K optimal social trust paths has not been identified, p_t is one of the K optimal social trust paths.

Algorithm 18: Backward_K-Search

Data: M, $Q^{T}_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t, K
Result: $BAQoT(v).T$, $BAQoT(v).SI$, $BAQoT(v).CIF$

begin
  set $v_x.\delta = \infty$ ($v_x \neq v_t$), $v_t.\delta = 0$, $S_x = \emptyset$, $v_x.bvisit = 0$, $G = 0$;
  add v_t into S_x;
  while $S_x \neq \emptyset$ **do**
    $S_{Top^K} = K\ min(v^*_a.\delta)$ ($v^*_a \in S_x$);
    /* *S_x is the priority queue in the backward search, and S_{Top^K} is a set that contains the K minimal δ values.* */;
    for *each* $v_a \in S_{Top^K}$ **do**
      if $v_a == v_s$ *and* $v_a.\delta \leq 1$ **then**
        $G = G + 1$;
      end
      for *each* $v_b \in adj[v_a]$ **do**
        /* *$adj[v_a]$ are all neighboring nodes of v_a* */
        $p_b = v_b$ to v_t via v_a;
        if $v_b \notin S_x$ **then**
          put v_b into S_x;
        end
        else if $\delta(p_b) < max(v_b.\delta)$ **then**
          update $BAQoT(v_b).T$, $BAQoT(v_b).SI$, $BAQoT(v_b).CIF$;
          put v_b into S_x;
        end
      end
      $v_a.bvisit = v_a.bvisit + 1$;
      /* *the visited times of v_a plus one* */
      if $v_a.bstatus == K$ **then**
        remove v_a from S_x;
      end
    end
  end
  Return G, $BAQoT(v).T$, $BAQoT(v).SI$, $BAQoT(v).CIF$;
end

Proof: Let $p^{F*}_{v_s \to v_t}$ denote the path from v_s to v_t that is selected from the priority queue at the J^{th} step. Let $p^{F_1}_{v_s \to v_t}, ..., p^{F_K}_{v_s \to v_t}$ denote the K optimal social trust paths from v_s to v_t identified by the *Forward_K-Search* procedure. If $p^{F*}_{v_s \to v_t} \notin \{p^{F_1}_{v_s \to v_t}, ..., p^{F_K}_{v_s \to v_t}\}$, then $\mathcal{F}(p^{F*}_{v_s \to v_t})$ is less than any of $\{\mathcal{F}(p^{F_1}_{v_s \to v_t}), ..., \mathcal{F}(p^{F_K}_{v_s \to v_t})\}$. At the J^{th} step, in addition to v_t, there are $K - 1$ nodes selected from the priority queue. Thus, at least one node in paths $\{p^{F_1}_{v_s \to v_t}, ..., p^{F_K}_{v_s \to v_t}\}$ is not selected at the J^{th} step. Then $\mathcal{F}(p^{F*}_{v_s \to v_t})$ is greater than one of $\{\mathcal{F}(p^{F_1}_{v_s \to v_t}), ..., \mathcal{F}(p^{F_K}_{v_s \to v_t})\}$, which contradicts that $\mathcal{F}(p^{F*}_{v_s \to v_t})$ is less than any of $\{\mathcal{F}(p^{F_1}_{v_s \to v_t}), ..., \mathcal{F}(p^{F_K}_{v_s \to v_t})\}$. Therefore, *Theorem 4* is correct. □

From *Theorem 3* and *Theorem 4*, H-OSTP-K can identify the number of solutions when there is less than K solutions, and the first K identified paths by H-OSTP-K are the top K paths. Based on these properties, we propose two optimization strategies to improve the efficiency of the *Forward K-Search* procedure.

Algorithm 19: Forward_K-Search

Data: M, $BAQoT(v).T$, $BAQoT(v).SI$, $BAQoT(v).CIF$, $Q^{T}_{v_s,v_t}$, $Q^{SI}_{v_s,v_t}$, $Q^{CIF}_{v_s,v_t}$, v_s, v_t, G
Result: $\mathcal{F}(p^{F_1}_{v_s \to v_t}), ..., \mathcal{F}(p^{F_G}_{v_s \to v_t})$

```
1  begin
2      set F' = 1/F, v_y.F' = ∞ (v_y ≠ v_s), v_s.F' = 0, S_y = ∅, v_y.fvisit = 0;
3      add v_s into S_y;
4      J = G;
       /* J is the number of unidentified paths from v_s to v_t. */
5      while S_y ≠ ∅ do
6          S_{Top^J}(F') = K min(v_i^*.F') (v_i^* ∈ S_y);
           /* S_y is the priority queue in the forward search, and S_{Top^J} is a set that contains the J minimal F' values.
           */
7          for each v_i ∈ S_{Top^J}(F') do
8              if v_i == v_t and v_a.δ ≤ 1 then
9                  J = J − 1;
                   /* Only J − 1 paths need to be identified in the following search. */
               end
10             for each v_j ∈ adj[v_i] do
                   /* adj[v_i] are all neighboring nodes of v_i */
11                 p_j = v_s to v_j via v_i;
12                 if ∃fp^{F_i+B_j}_{v_s→v_j→v_t} (i, j ∈ [1, G]) is feasible then
13                     if v_j ∉ S_y then
14                         put v_j into S_y;
                       end
15                     else if F'(p_j) < Max(v_j.F') then
16                         update FAQoT(v_b).T, FAQoT(v_b).SI, FAQoT(v_b).CIF;
                           /* FAQoT records the aggregated QoT attributes in the forward search. */
17                         put v_j into S_y;
                       end
                   end
               end
18             v_i.fvisit = v_i.fvisit + 1;
               /* the visited times of v_i plus one */
19             if v_i.fvisit == K* then
20                 remove v_i from S_x;
               end
           end
       end
21     Return F(p^{F_1}_{v_s→v_t}), ..., F(p^{F_G}_{v_s→v_t});
   end
```

Optimization Strategy 1: The *Forward_K-Search* procedure is to identify up to K optimal social trust paths which are feasible. if there exist G $(0 < G < K)$ feasible solutions identified by the *Backward_K-Search* procedure based on *Theorem 1* in a sub-network, the *Forward_K-Search* procedure does not need to search K paths but G paths from v_s to v_t.

Optimization Strategy 2: If v_t has been selected J $(1 \leq J < K)$ times from the priority queue, in the following process, H-OSTP-K only needs to search $K - J$ optimal social trust paths from v_s to v_t.

Then, if there exist l $(1 \leq l \leq K)$ feasible solutions, the *Forward_K-Search* procedure can identify them all, and they are the l optimal social trust paths. Otherwise, this procedure can identify K feasible solutions which are not worse than those identified by the *Backward_K-Search* procedure. Namely, *Theorem 1* and *Theorem 2* can guarantee the effectiveness of our algorithm.

Since H-OSTP-K adopts the Dijkstra's shortest path algorithm based K general social trust paths selection method twice, it has the same time complexity of $O(m + Knlogn)$ as that of the algorithms in *Category 1*.

6.4 Experiments on H-OSTP-K

6.4.1 Experiment Settings

To validate our proposed algorithm, we select the *Enron* email dataset with 87,474 nodes (participants) and 30,0511 links (formed by sending and receiving emails) as the dataset for our experiments. Firstly, in order to study the performance of our proposed heuristic algorithm in sub-networks of different scales and structures, we first randomly select 100 pairs of source and target participants from the *Enron* email dataset. We then extract the corresponding 100 sub-networks between them by using the exhaustive search 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 in them. 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).

Secondly, as we have analysed in *Section 6.2*, existing K general shortest paths selection algorithms are all deterministic algorithms, and cannot be used for solving the NP-Complete MCOP-K problem [6]. Therefore, to study the performance of our heuristic H-OSTP-K, we first compare the maximal utility of the identified K social

Table 6.1: The setting of QoT constraints

Constraint ID	$Q^{T}_{v_s,v_t}$	$Q^{SI}_{v_s,v_t}$	$Q^{CIF}_{v_s,v_t}$
#1	0.05	0.05	0.05
#2	0.1	0.05	0.05
#3	0.05	0.1	0.05
#4	0.05	0.05	0.1

trust paths with that of our previously proposed H_OSTP, which so far outperforms the other existing algorithms for the NP-Complete Multiple Constrained Optimal social trust Path (MCOP) selection problem in complex contextual trust-oriented social networks. In addition, since existing methods are not suitable for the NP-Complete MCOP-K selection problem, in order to study the efficiency of our proposed optimization strategies, we compare the execution time of H-OSTP-K with that of the modified version of H-OSTP-K without our proposed optimization strategies (denoted as H-WOP-K) (*see Section 6.4.2*).

Finally, to investigate the performance of H-OSTP-K in social trust path selection with different QoT constraints, four groups of QoT constraints are set and listed in Table 6.1. In addition, the three QoT attributes are given the same weights in the utility function. Furthermore, since the detailed mining method of QoT attributes values are out of scope of this thesis, these QoT attributes values are randomly generated by using $rand()$ in *Matlab*.

Each of H-OSTP-K, H-WOP-K and H_OSTP is 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. The results are plotted in Fig. 6.1 to Fig. 6.6, where the execution time of each of H-OSTP-K and H-WOP-K is averaged based on 3 independent executions.

6.4.2 Results and Analysis

Comparison of Path Utility: Fig. 6.1 plots the path utilities of the identified social trust path by H_OSTP and the maximal path utility of the identified $K(K = 2)$ optimal

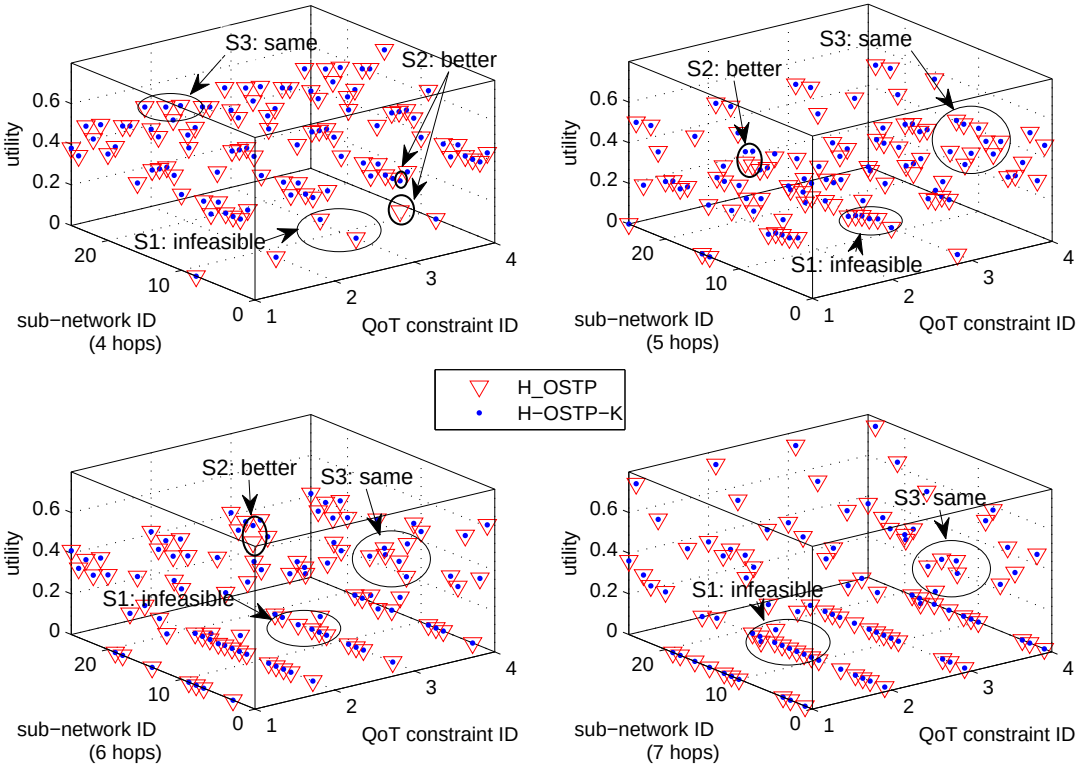

Figure 6.1: The path utilities of sub-networks with each group of hops

social trust paths by H-OSTP-K in sub-networks in 4 groups. From these figures, we can observe that in some sub-networks (i.e., 32 out of 100 sub-networks), if there is no feasible solution, both H-OSTP-K and H_OSTP can investigate the infeasibility (e.g., S1 in Fig. 6.1), and thus the path utilities in these sub-networks are zero. This is because that H_OSTP also computes δ_{min} in the backward search from v_t to v_s based on the Dijkstra's shortest path algorithm. Therefore, based on the theorems proposed in *Chapter 5.4*, both of them can always investigate whether there is a feasible solution existing in a sub-network.

In addition, from Fig. 6.1, we can see that in some cases (i.e., 49 out of 100 sub-networks), H-OSTP-K can deliver the same path utilities with those of H_OSTP (e.g., S3 in Fig. 6.1). This is because that firstly, in a sub-network, if the path with the

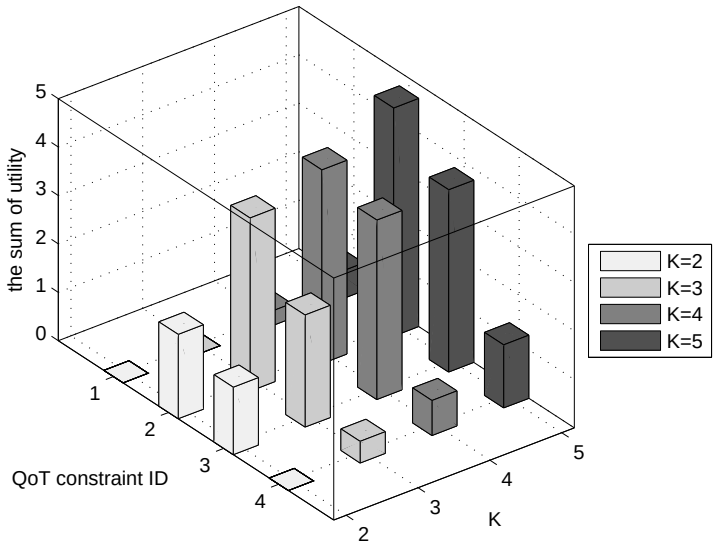

Figure 6.2: The sum of path utilities with different K values

maximal path utility in the K paths identified by H-OSTP-K and the path identified by H_OSTP are selected based on the same foreseen path formed at each of the intermediate nodes of these paths, according to the searching strategies in H_OSTP proposed in *Chapter 5.4*, the two paths are the same. Secondly in a sub-network, if there exists only one feasible solution, both H-OSTP-K and H_OSTP can identify it, and thus they deliver the same path utility.

Furthermore, from Fig. 6.1, we can also see that H-OSTP-K can deliver better social trust paths than H_OSTP (e.g., S2 in Fig. 6.1) in some cases (i.e., 19 out of 100 sub-networks). In addition, as depicted in Fig. 6.2, given the same constraint ID, the larger the K value, the greater the sum of the utility of these sub-networks. Table 6.2 lists the sum of utilities computed by H-OSTP-K and H_OSTP in these sub-networks, where we can see that the sum of utilities computed by our proposed heuristic algorithm is 49.66% higher than that of H_OSTP in 4 hops sub-networks, 20.24% higher in 5 hops, 13.39% higher in 6 hops. On average, the path utility computed by H-OSTP-K is 20.29% higher than that of H_OSTP. This is because that in H-OSTP-K, it is to form K foreseen paths rather than only one foreseen path in H_OSTP, and thus H-OSTP-K

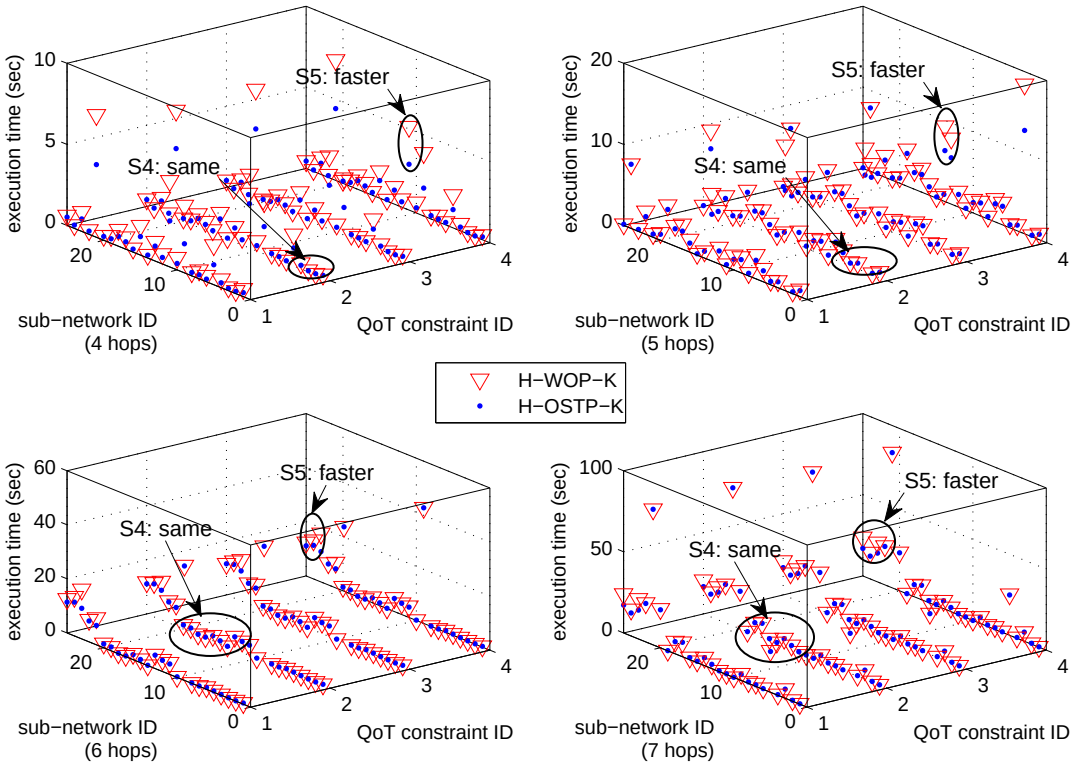

Figure 6.3: The execution time of $K = 2$

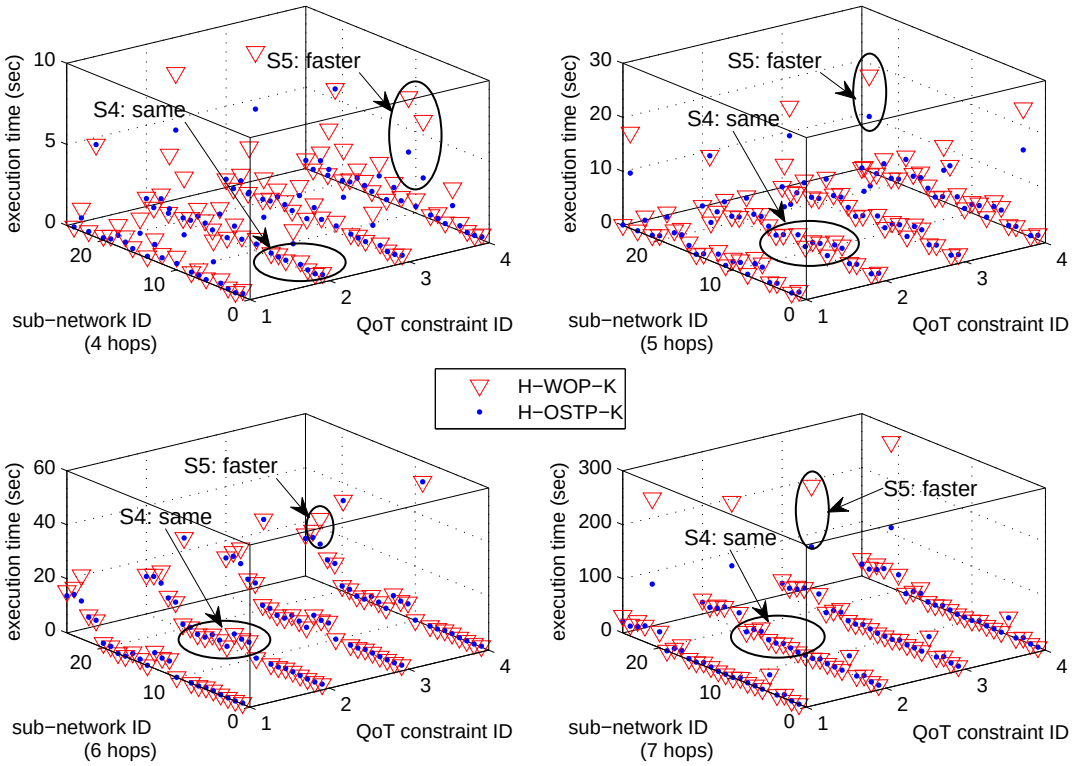

Figure 6.4: The execution time of $K = 3$

Table 6.2: The comparison of path utility

Algorithms	The sum of utility				
	4 hops	5 hops	6 hops	7 hops	total
H-OSTP-K	2.5461	17.9369	8.0839	0	28.5669
H_OSTP	1.7012	14.9174	7.1295	0	23.7481
difference	49.66% higher	20.24% higher	13.39% higher	0	20.29% higher

have more chances to deliver a better social trust path.

Comparison of Execution Time:

Since H-WOP-K has the same functionality as H-OSTP-K, they both deliver the same path utilities of K paths in a sub-network. Therefore, we only compare the difference in their execution time, and the experiment results are plotted in Fig. 6.3 to Fig. 6.6.

From Fig. 6.3 to Fig. 6.6, we can observe that the execution time of H-OSTP-K is the same as that of H-WOP-K in some sub-networks (e.g., S4 in Fig. 6.3 to Fig. 6.6). This is because if there is no feasible solution in a sub-network, H-OSTP-K only performs the *Backward_K-Search* procedure which has the same search strategy with H-WOP-K. Therefore, they have the same execution time.

In addition, from these figures, we can also observe that the execution time of H-OSTP-K is less than that of H-WOP-K in other sub-networks (e.g., S5 in Fig. 6.3 to Fig. 6.6). The total execution time of each of H-OSTP-K and H-WOP-K in each group of hops is listed in Table 6.3, where we can see that the total execution time of our proposed heuristic algorithm is only 41.86% of that of H-WOP-K in 4 hops sub-networks, 70.60% in 5 hops, 89.51% in 6 hops and 51.03% in 7 hops. On average, H-OSTP-K is 37.22% faster than H-WOP-K. From the above results, we can see that H-OSTP-K is much more efficient than H-WOP-K. The reasons are twofold. Firstly based on *Theorem 3*, if there exist G $(0 < G < K)$ feasible solutions, then H-OSTP-K searches only G optimal social trust paths from v_s to v_t, significantly saving execution time (see details in *Optimization Strategy 1*). Secondly, based on *Theorem 4*, assuming there exist K $(K \geq 2)$ feasible solutions and v_t has been selected J $(0 < J <$

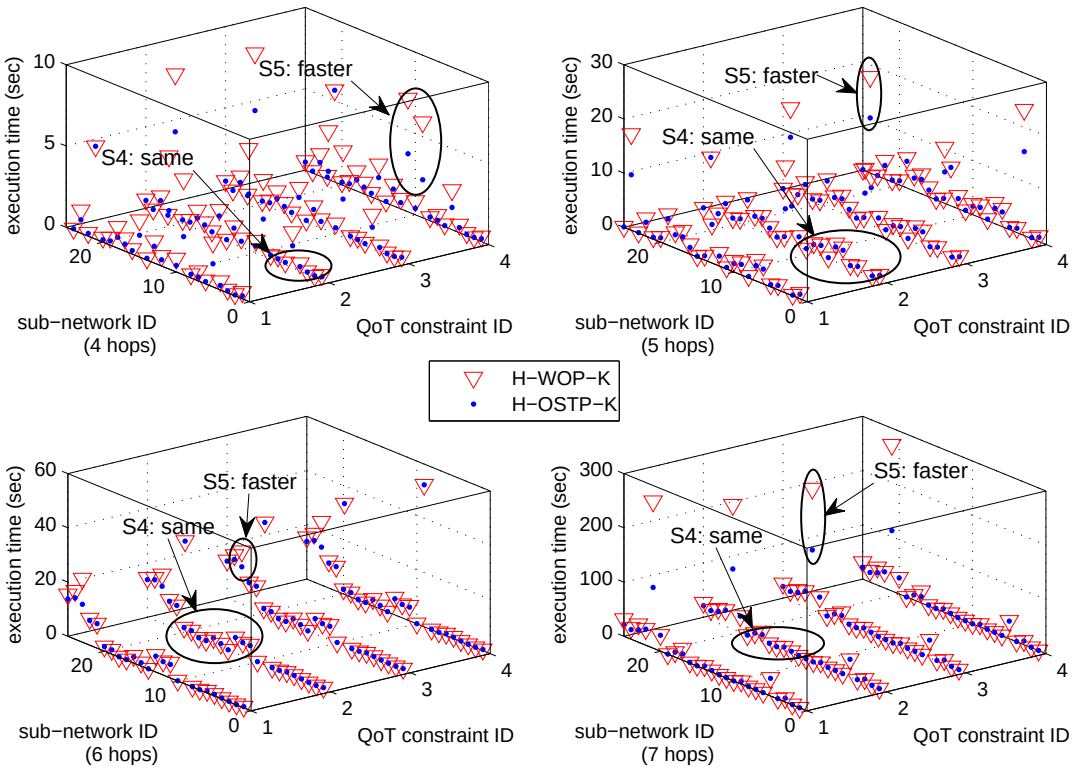

Figure 6.5: The execution time of $K = 4$

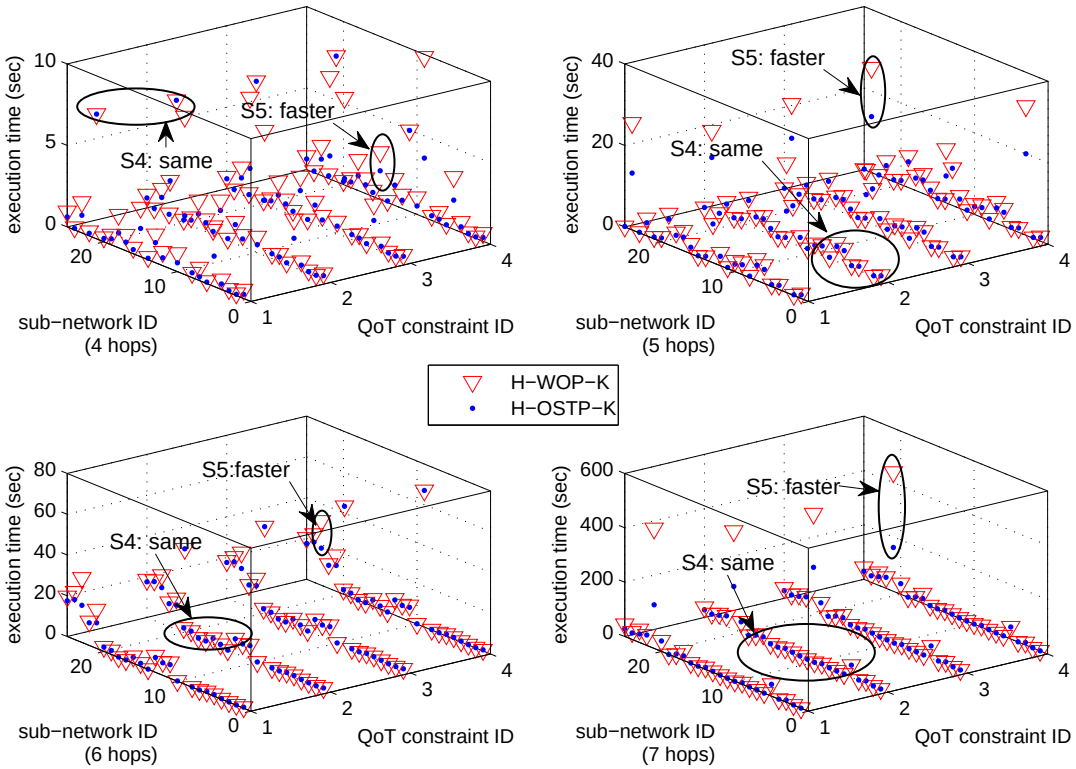

Figure 6.6: The execution time of $K = 5$

Table 6.3: The comparison of execution time

Algorithms	The sum of execution time (sec)				
	4 hops	5 hops	6 hops	7 hops	total
H-OSTP-K	1.4152e+003	4.1990e+003	9.4535e+003	2.1763e+004	3.6831e+004
H-WOP-K	2.2380e+003	5.4336e+003	1.0445e+004	3.2421e+004	5.0538e+004
difference	58.14% less	29.40% less	10.49% less	48.97% less	37.22% less

K) times from the priority queue, then in the following searching steps, H-OSTP-K searches only $K - J$ optimal social trust paths from v_s to v_t, and thus saves execution time (see details in *Optimization Strategy 2*).

Through the above experiments conducted in sub-networks with different scales and structures, we can see that H-OSTP-K is an effective and efficient algorithm for MCOP-K selection in complex trust-oriented social networks.

6.5 Conclusion

In this Chapter, we have analysed the existing K optimal paths selection algorithms for the top K shortest path selection problem. Then, in order to solve the NP-Complete multiple QoT constrained K optimal social trust paths selection problem. we have proposed an efficient heuristic algorithm H-OSTP-K based on the Dijkstra's shortest path algorithm and several novel search strategies. The results of experiments conducted on real datasets of social networks demonstrate that H-OSTP-K significantly outperforms the existing methods in the quality of identified social trust paths and the efficiency.

The K optimal social trust path selection method proposed in this chapter can identify more than one trustworthy trust paths, which helps deliver more reasonable trust evaluation results.

Chapter 7

Trust Transitivity in Complex Contextual Trust-Oriented Social Networks

In social networks, if there is a social trust path linking two nonadjacent participants, the source participant can evaluate the trustworthiness of the target one along an existing path based on the *trust transitivity* property (i.e., if A trusts B and B trusts C, then A trusts C to some extent) under certain semantic constraints [63]. In each of the selected social trust paths by the algorithms in *Chapters 5 and 6*, the computation of the value of trust for the target participant requires an understanding of how trust is transitive along the trust path, which is a critical and challenging problem in OSNs [48, 51]. In the literature, several trust transitivity models have been proposed [48, 50, 51, 113, 124], but they have the following drawbacks.

Firstly, as illustrated in Social Psychology [3, 76, 102], the *social relationships* between participants (e.g., the one between an employer and an employee), the *social positions* of participants (e.g., a supervisor as a referee in his postgraduate's job application) and the *preference similarity* between participants (e.g., whether both of them like to play badminton) have significant influence on trust transitivity. However, to the best of our knowledge, these impact factors are not fully considered by existing trust transitivity models. Secondly, a source participant may have different criteria in evaluating the trustworthiness of the target participant [91], impacting on trust transi-

tivity results. However, the specification of evaluation criteria is not supported by any existing method. Finally, trust transitivity formalized in the existing models does not follow the nature of trust decay illustrated in social psychology, namely, trust decays slowly in a certain number of early hops (specified by a source participant) from a source participant, and then decays fast until the trust value approaches the minimum [44, 61].

In this Chapter, we first propose a novel concept, Quality of Trust Transitivity (QoTT) to illustrate the ability of a social trust path to guarantee a certain level of quality in trust transitivity. Then, based on the properties of trust illustrated in social psychology, a new Multiple QoTT Constrained Trust Transitivity (MQCTT) model is introduced. Experimental results demonstrate that the proposed trust transitivity model follows both the principles in social psychology and the properties of trust, and thus it computes more reasonable trust values than existing methods.

7.1 Trust Properties and the Quality of Trust Transitivity

In this section, we first analyze trust properties and then propose a novel concept Quality of Trust Transitivity (QoTT).

7.1.1 The properties of Trust Transitivity

As illustrated in social psychology, trust has the following properties:

Property 1: Subjective. As illustrated in social psychology [54, 91], trust is a subjective phenomenon that is defined by the psychological experiences of the individual who bestows it, reflecting subjective attitudes that affect participants' thinking based on subjective evaluation criteria that can vary in different domains.

Property 2: Transitive. Trust can be transitive from one to another with a discount [23]. In addition, trust transitivity needs certain constraints [23, 63]. Namely, if A

trusts B in the domain of *teaching C++*, and B trusts C in the domain of *repairing a car*, then the trust cannot be transitive from A to C via B in the domain of *teaching C++*. However, if A also trusts B in *repairing a car* (in the same domain that B trusts C), then trust can be transitive from A to C in this domain.

Property 3: Decay. In trust transitivity, trust decays with the increase of transitivity hops along a social trust path [23]. In addition, the general decay is non-linear [61, 91] and can be divided into three phases. ***Phase 1: (Slow Decay Phase)*** In this phase, trust decays slowly in transitivity along a social trust path from a source participant within a certain number of hops (e.g., from 1 to 3 hops in Fig. 7.1). This is because the source participant may consider the familiarity with the trustee to extend no more than a certain number of transitivity hops. ***Phase 2: (Fast Decay Phase)*** With the increase of transitivity hops, the trust decay speed increases in trust transitivity until the trust value approaches the minimum (e.g., from 4 to 6 hops in Fig. 7.1). This is because that in this phase, the trustee is becoming stranger to the source participant than the case in *Phase 1*. ***Phase 3: (Slow Decay Phase)*** When the trust value between the source participant and the trustee is approaching the minimum, the trust decay speed changes from fast to slow (e.g., from the 6^{th} hop in Fig. 7.1). This is because in this phase, the trustee has become a stranger to the source participant.

Let λ_1 denote the number of hops of trust transitivity in Phase 1 (e.g., $\lambda_1 = 3$ in Fig. 7.1) and λ_2 denote the number of the hops where trust approaches to zero in Phase 3 (e.g., $\lambda_2 = 8$ in Fig. 7.1). Their values can be specified by participants based on their own trust evaluation criteria in a certain domain [61, 91]. Then even the trust transitivity follows the general trust decay trend along a social trust path, based on different λ_1 and λ_2 values specified by the source participants, they can obtain different trust transitivity results of the target along the social trust path.

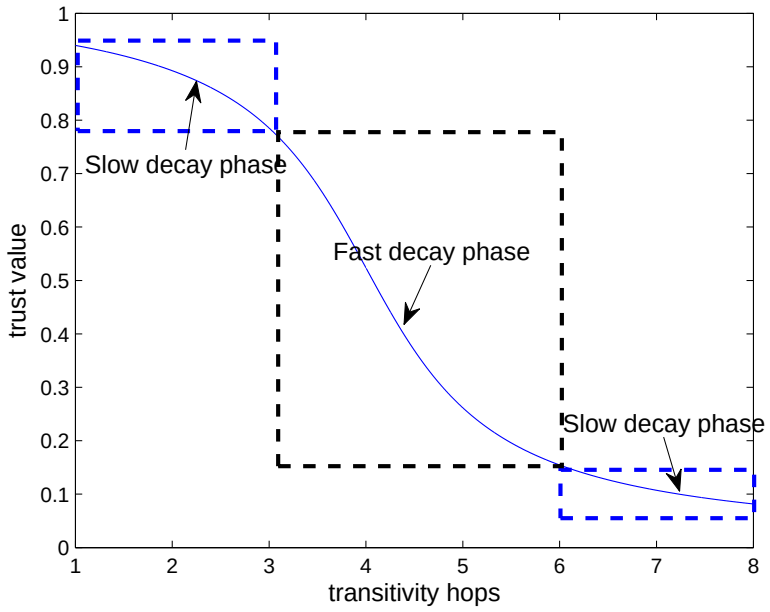

Figure 7.1: General trust decay with the increase of transitivity hops

7.1.2 Quality of Trust Transitivity (QoTT)

In Service-Oriented Computing (SOC), QoS embodies a set of attributes to illustrate the ability of services to guarantee a certain level of performance [40]. Similar to the QoS, we propose a novel concept, *Quality of Trust Transitivity*, which in general incorporates any attribute that impacts on trust transitivity.

Definition 4. *Quality of Trust Transitivity* (QoTT) is the ability of a social trust path to guarantee a certain level of quality of trust transitivity, taking trust (T), social intimacy degree (SI), community impact factor (CIF) and preference similarity (PS) as attributes.

7.1.3 QoTT Constraints

Based on *Property 1* of trust, in our model, a source participant can specify multiple end-to-end QoTT constraints for QoTT attributes as the requirements of the Quality of trust transitivity along a social trust path. Let $QoTT^{\mu'}$ denote the QoTT constraints

for the aggregated QoTT attribute μ' ($\mu' \in \{T,\ SI,\ CIF,\ PS\}$) in a social trust path. Then the aggregated trust value, social intimacy degree and community impact factor value of a social trust path can be computed by using the corresponding aggregation methods in Eqs (5.1) to (5.3) proposed in *Chapter 5.1.2*. In the following, we introduce a method for the aggregation preference similarity value between participants in a trust path.

As illustrated in social psychology [76], if two participants have the same preference to an object, they have a high preference similarity which *does not decay* with the increase of the number of transitivity hops. Thus, the aggregated PS value of path $p_{(a_1,...a_n)}$ in a certain domain can be calculated by Eq. (7.1).

$$PS_{p(a_1,...,a_n)} = \frac{\sum_{k=2}^{n-1} PS_{a_k}}{n-2} \tag{7.1}$$

Then based on our model, a reliable trust transitivity result can be computed along a social trust path, *if and only if* each aggregated QoTT attribute value of the social trust path satisfies the corresponding end-to-end QoTT constraint.

Since the QoTT constraints, λ_1 and λ_2 in trust transitivity are subjectively specified by source participant in trust transitivity, these parameters are called *subjective impact parameters*.

7.2 Multiple QoTT Constrained Trust Transitivity Model

In this section, we propose a novel Multiple QoTT Constrained Trust Transitivity (MQCTT) model, where both subjective impact parameters and objective impact factors are considered.

7.2.1 The Process of MQCTT

In a social trust path $p_{(a_1,...,a_n)}$, with the λ_1 and λ_2 specified by the source participant a_1, we take a_{j+1} (where there are j hops between a_1 and a_{j+1} ($j \leq n-1$)) as an example to

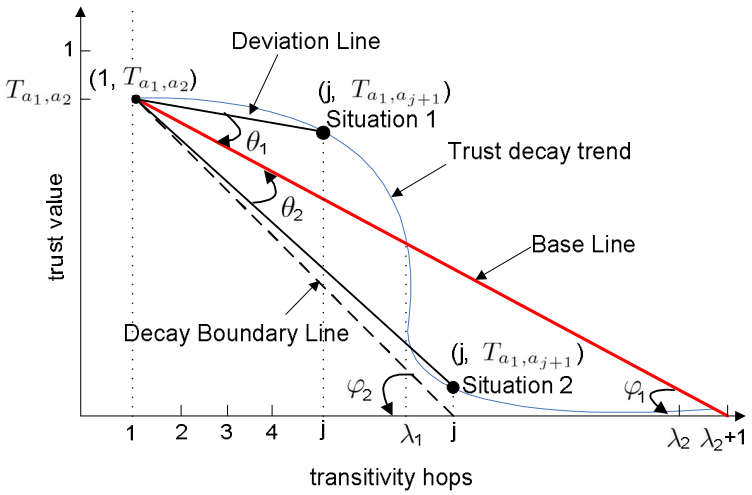

Figure 7.2: Trust transitivity model

introduce the calculation of the trust transitivity result $T_{a_1,a_{j+1}}$ by our MQCTT model.

Step 1 (average trust decay speed): Based on *Property 3* of trust, trust decays to zero when the number of transitivity hops is greater than λ_2 ($\lambda_2 > 1$). As depicted in Fig. 7.2, we draw a *Base Line* that starts from coordinate $(1,\ T_{a_1,a_2})$, which corresponds to the first hop of trust transitivity with the initial trust value T_{a_1,a_2} and ends at $(\lambda_2 + 1,\ 0)$, where the number of trust transitivity hops is greater than λ_2, leading to the trust value of zero. This line and its slope can illustrate the *average trust decay speed* along $p_{(a_1,\ldots,a_n)}$ in trust transitivity.

Step 2 (intersection angle θ): After identifying the average trust decay speed, based on *Property 3* of trust, if $j \leq \lambda_1$, the trust decay speed should be slower than the average trust decay speed. Therefore, $(j,\ T_{a_1,a_{j+1}})$ should be above the *Base Line* (i.e., *Situation 1* in Fig. 7.2). If $\lambda_1 < j \leq \lambda_2$ $(j,\ T_{a_1,a_{j+1}})$ should be under the *Base Line* (i.e., *Situation 2* in Fig. 7.2). A *Deviation Line* that starts from $(1,\ T_{a_1,a_2})$ and ends at $(j,\ T_{a_1,a_{j+1}})$ can be drawn, where an *intersection angle* θ is formed (i.e., $\theta_1 < 0$ in *Situation 1* or $\theta_2 > 0$ in *Situation 2*). Since the $T_{a_1,a_{j+1}}$ is determined by θ, λ_1 and λ_2, in the following steps, we will introduce how to compute the value of θ, and further compute $T_{a_1,a_{j+1}}$ along path $p_{(a_1,\ldots,a_n)}$.

Step 3 (the scope of θ): Before computing the value of θ, we first determine the

scope of θ. Since trust decays in transitivity with the hops via the social trust path from a source participant [23], the minimal value of θ is equal to the *interaction angle* from the *Base Line* to the horizontal axis (i.e., φ_1 in Fig. 7.2), which can be calculated by Eq. (7.2). In addition, based on *Property 3* of trust, if and only if $j > \lambda_2$, $T_{a_1,a_{j+1}}$ decays to zero. We draw a *Decay Boundary Line* that starts from $(1,\ T_{a_1,a_2})$ and ends at $(j,\ 0)$ $j > \lambda_1$ to indicate the trust decay boundary. Then the maximal value of θ is equal to the interaction angle from *Decay Boundary Line* to the horizontal axis (i.e., φ_2 in Fig. 7.2) minus φ_1, i.e., $\varphi_2 - \varphi_1$, where φ_2 can be calculated by Eq. (7.3). Then $\theta \in (\varphi_1, \varphi_2 - \varphi_1)$

$$\varphi_1 = arctan(\frac{T_{a_1,a_2}}{\lambda_2}), \quad \lambda_2 > 1 \text{ and } \varphi_1 \in (0, \frac{\pi}{2}) \tag{7.2}$$

$$\varphi_2 = arctan(\frac{T_{a_1,a_2}}{j-1}), \quad 1 < j \leq \lambda_2 \text{ and } \varphi_2 \in (0, \frac{\pi}{2}) \tag{7.3}$$

Step 4 (logistic function): As illustrated in *Property 3* of trust, the general trust decay follows the curve plotted in Fig. 7.1. Therefore, the increase of θ is non-linear and follows the curve depicted in Fig. 7.3. In mathematics, the *logistic function* is known to be the most accurate one to model phenomenons possessing non-linear increases with such a trend, and has been widely used in the real-world, e.g., modeling the *non-linear population growth* in ecology, the *non-linear growth of tumors* in medicine and the *nonlinearity of clamp signals* in neural networks [64]. Therefore, to compute an accurate θ value and further obtain a more reasonable trust transitivity result, we use the *logistic function* as in Eq. (7.4) to model the increase of θ. The function curve is plotted in Fig. 7.3.

$$\theta = \begin{cases} [\frac{2*\varphi_1}{1+e^{(\xi-j)}}] - \varphi_1 & \text{for } 1 < j \leq \lambda_1 \\ \\ [\frac{2*(\varphi_2-\varphi_1)}{1+e^{(\xi-j)}}] - (\varphi_2 - \varphi_1) & \text{for } \lambda_1 < j \leq \lambda_2 \end{cases} \tag{7.4}$$

where ξ is the argument controlling the function curve.

Step 5 (computing θ value): After modeling the increase of θ by using Eq. (7.4), it is necessary to calculate the arguments of Eq. (7.4), and further compute θ value. From Fig. 7.3, we can see that ξ is the argument controlling the number of transitivity hops when $\theta = 0$. Then based on *Property 2* of trust, if $0 < j \leq \lambda_1$, then $\xi > \lambda_1$, which ensures $\theta < 0$ (i.e., *Situation 1* in Fig. 7.2). Otherwise if $\lambda_1 < j \leq \lambda_2$, then $\xi < \lambda_1$, which ensures $\theta > 0$ (i.e., *Situation 2* in Fig. 7.2). Then ξ can be calculated by Eq. (7.5) and Eq. (7.6).

$$\begin{aligned} \tau = SI_{p(a_1,\ldots,a_{j+1})} + CIF_{p(a_1,\ldots,a_{j+1})} \\ + PS_{p(a_1,\ldots,a_{j+1})} + T_{p(a_1,\ldots,a_{j+1})} \end{aligned} \tag{7.5}$$

$$\xi = \begin{cases} \lambda_1 + \frac{\tau}{1-\tau} & \text{for } 1 < j \leq \lambda_1 \\ \\ \lambda_1 - \frac{1-\tau}{\tau} & \text{for } \lambda_1 < j \leq \lambda_2 \end{cases} \tag{7.6}$$

Note that Eq. (7.5) and Eq. (7.6) have the following characteristics:

Characteristic 1: if $1 < j \leq \lambda_1$ and $\tau \to 0$, then $\xi \to \lambda_1^+$ and thus $\theta \to 0$. In such a situation, the *Deviation Line* tends to coincide with the *Base Line*. Namely, the trust decay speed approaches the average trust decay speed when all QoTT attribute values approach zero.

Characteristic 2: If $1 < j \leq \lambda_1$ and $\tau \to 1$, then $\xi \to \infty$ and thus $\theta \to \varphi$. In this situation, the *Deviation Line* tends to be parallel with the horizontal axis. Namely, the trust decay speed approaches zero, when all the QoTT attribute values approach one.

Similarly, we can obtain the same characteristics above when $\lambda_1 < j \leq \lambda_2$, following the principles in social psychology and the properties of trust.

Step 6 (computing $T_{a_1,a_{j+1}}$ based on θ): After computing θ based on Eq. (7.4) and the slope of *Base Line* (denoted as k_1) based on Eq. (7.7) respectively,

$$k_1 = \frac{T_{a_1,a_2}}{\lambda_2 + 1}, \tag{7.7}$$

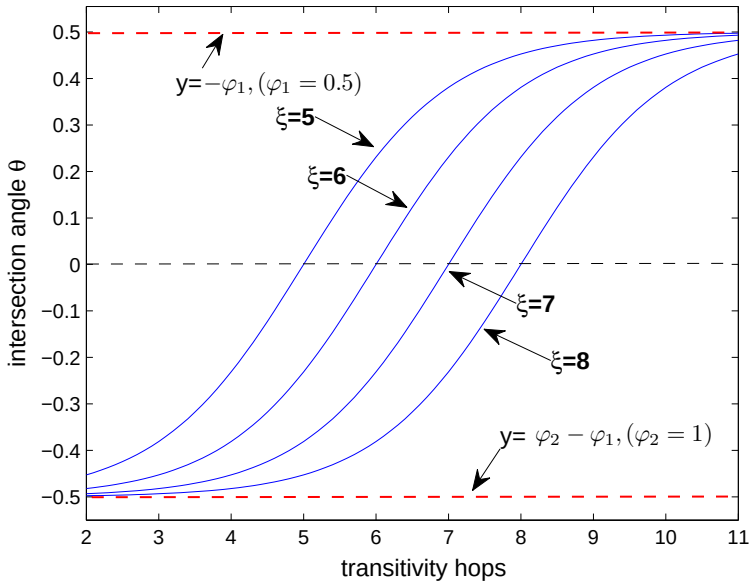

Figure 7.3: Increase of intersection angle θ

the slope of *Deviation Line* (denoted as k_2) can be calculated with Eq. (7.8).

$$tan(\theta) = (\frac{k_1 - k_2}{1 + k_1 k_2}), \quad \theta \in (-\varphi_1, \varphi_2 - \varphi_1) \tag{7.8}$$

After obtaining k_2, $T_{a_1,a_{j+1}}$ can be calculated by Eq. (7.9).

$$T_{a_1,a_{j+1}} = T_{a_1,a_2} + k_2 \cdot j, \quad 1 < j \leq \lambda_2 \text{ and } k_2 < 0 \tag{7.9}$$

$T_{a_1,a_{j+1}}$ is computed based on both objective impact factors (i.e., T, SI, CIF and PS), and subjective impact parameters (i.e., QoTT constraints, λ_1 and λ_2), thus it is different from $T_{p_{(a_1,\ldots,a_{j+1})}}$ which is only one of the above factors impacting on $T_{a_1,a_{j+1}}$ value.

Table 7.1: Extracted sub-networks

ID	Max hops	Number of nodes	Number of links
1	4	61	155
2	4	104	237
3	5	158	389
4	5	215	619
5	6	228	667
6	6	445	1418
7	7	551	3265
8	7	750	3301

Table 7.2: Selected trust transitivity models

Model Number	Category	Strategy	Authors
model 1	first	multiplication	Walter *et. al* [124]
model 2	second	average	Golbeck *et. al* [48]
model 3	third	confidence-based	Guha *et. al* [51]

7.3 Experiments on MQCTT

7.3.1 Experiment Settings

Firstly, in order to evaluate the performance of our proposed MQTTC model, we conduct experiments on sub-networks of different scales and structures, extracted from the *Enron* email dataset[1] which contains 87,474 nodes and 30,0511 links. This dataset has been widely used in the studies of social networks [96, 117]. We randomly select 8 pairs of source and target participants, and then extract the corresponding 8 sub-networks between them by using an exhaustive search method. Among these sub-networks, the maximal length of a social trust path varies from 4 to 7 hops, following the *small-world* characteristic [50][2]. These sub-networks are listed in Table 7.1.

Secondly, to compare MQCTT with existing trust transitivity models, we select one model from each of the categories of the existing trust transitivity methods introduced in the Chapter of *Literature Review* (Chapter 2) and list them in Table 7.2). In addition, we select three domains in our experiments, including (1) *product sales*, (2) *hiring*

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

[2]The average path length between any two nodes is about 6.6 hops in a social network

Table 7.3: Subjective impact parameters of three domains

Domain (NO.)	$QoTT^{T}$	$QoTT^{SI}$	$QoTT^{CIF}$	$QoTT^{PS}$	λ_1	λ_2
product sales (1)	0.1	0.05	0.05	0.05	3	4
hiring employees (2)	0.05	0.05	0.1	0.05	4	5
making friends (3)	0.05	0.05	0.05	0.1	5	6

employees and (3) *making friends*. The values of the subjective impact parameters specified by a source participant are listed in Table 7.3. Furthermore, the values of SI, CIF and PS can be mined in social networks by using data mining techniques. But this is out of the scope of this thesis. Without loss of generality, the values QoTT attributes are randomly generated by using *rand()* in *Matlab*.

Finally, as all trust transitivity models including MQCTT are used to compute the trust value along a social trust path, we compare the most reliable trust transitivity results of all models obtained from the optimal social trust path in a sub-network. The optimal social trust path without QoTT constraints is selected by using the existing optimal algorithm in [53], and the path with QoTT constraints in MQCTT is selected by using the optimal algorithm in [83].

All four trust transitivity models 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.

7.3.2 Results and Analysis

7.3.2.1 Scenario 1: trust transitivity based on different subjective impact parameters

To investigate the performance of the MQCTT model with different subjective impact parameters, we set the same T, SI, CIF and PS values in the three domains.

From the experimental results plotted in Fig. 7.4, we can see that each of the existing trust transitivity models yields the same trust values in the three domains (e.g. S1 in Fig. 7.4). However, based on *Property 1* of trust, a source participant may have

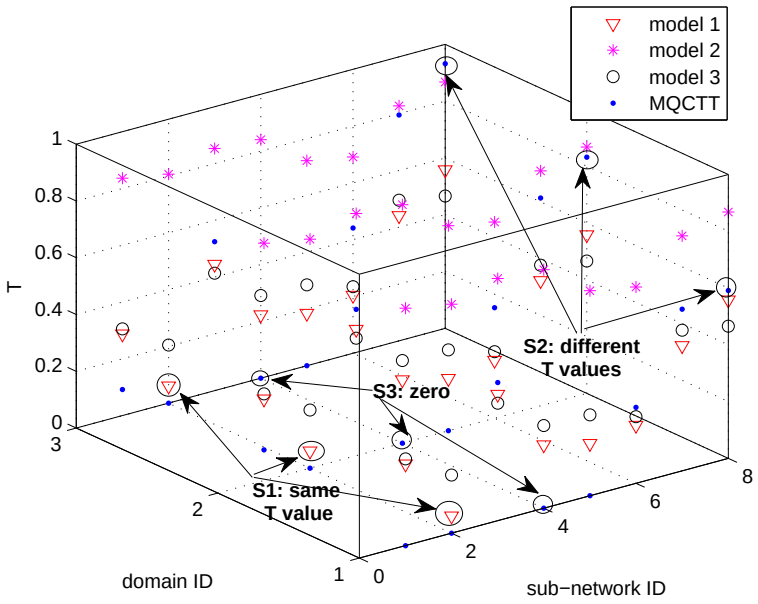

Figure 7.4: Trust values computed based on different subjective impact parameters

different evaluation criteria in the trust transitivity of different domains, leading to different trust values along the same social trust path. Thus, *existing trust transitivity models neglect this property*.

In contrast, our MQCTT model considers different values of subjective impact parameters specified by the source participant. Therefore, the trust values computed by our MQCTT model are different in the three domains based on the source participant's different trust evaluation criteria (e.g., S2 in Fig. 7.4), following *Property 1* of trust. In addition, if no social trust path can satisfy the QoTT constraints in the sub-network, or the number of transitivity hops is greater than λ_2, the source participant will not establish a trust relation with the target participant. Then the trust values of the target participant are equal to zero (e.g., $T = 0$ in S3 in Fig. 7.4). This follows *Properties 2 and 3* of trust. However, existing methods neglect these properties.

7.3.2.2 Scenario 2: trust transitivity with different social relationships

To investigate the performance of all models in trust transitivity with different social relationships, SI value is decreased to $SI' = SI/1.5$, and the rest of the QoTT at-

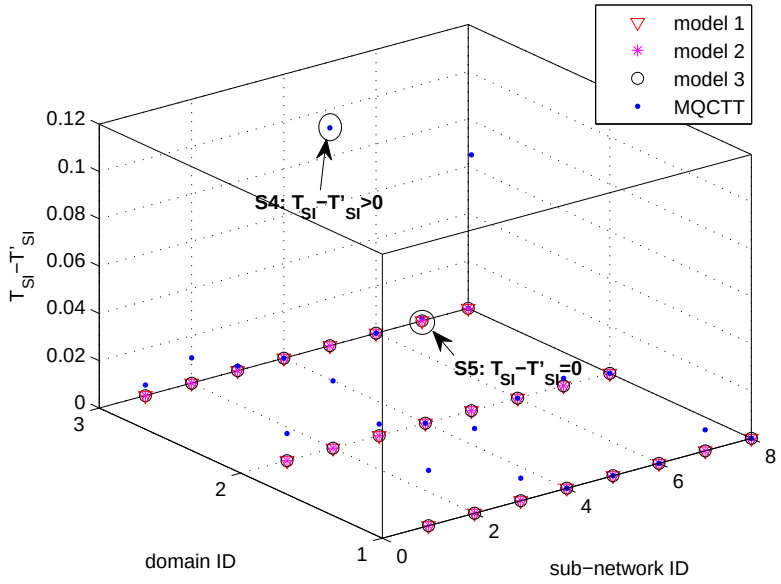

Figure 7.5: The results of $T_{SI} - T'_{SI}$

tributes have the same values with those in scenario 1.

Fig. 7.5 plots the trust transitivity results computed based on SI (denoted as T_{SI}) minus those computed based on SI' (denoted as T'_{SI}), i.e., $T_{SI} - T'_{SI}$. We can see that in some cases, $T_{SI} - T'_{SI} > 0$ in the MQCTT model (e.g., S4 in Fig. 7.5). Namely, the trust value computed by our MQCTT model decreases with the decrease of r value when the social trust path satisfies QoTT constraints, which follows *Principle 1*. In contrast, the trust values computed by each of the three existing trust transitivity models are the same, *neglecting the influence of social relationships*.

In addition, in MQCTT, if the aggregated SI and SI' values in a path do not satisfy the corresponding QoTT constrains, $T_{SI} = T'_{SI} = 0$ (e.g., S5 in Fig. 7.5). This follows *Property 3* of trust.

7.3.2.3 Scenario 3: trust transitivity with different social positions

To investigate the performance of these models in trust transitivity with different social positions, CIF is decreased to $CIF' = CIF/1.5$. The rest of the QoTT attributes

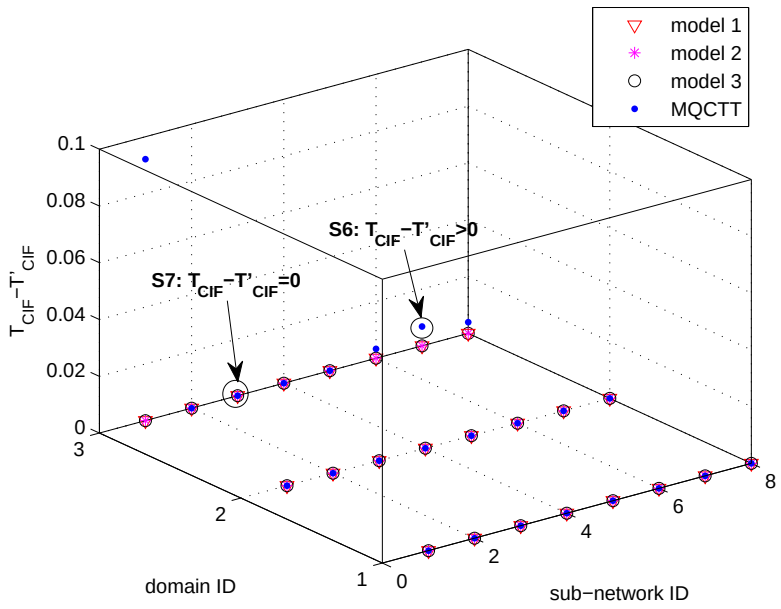

Figure 7.6: The results of $T_{CIF} - T'_{CIF}$

have the same values with those in scenario 1.

Fig. 7.6 plots the trust transitivity results computed based on CIF (termed as T_{CIF}) minus those computed based on CIF' (termed as T'_{CIF}), i.e., $T_{CIF} - T'_{CIF}$. We can see that in some cases in *domain 3*, $T_{CIF} - T'_{CIF} > 0$ in our MQCTT model (e.g., S6 in Fig. 7.6). Namely, the trust value decreases with the decrease of CIF value when the social trust path satisfies the QoTT constraints, which follows *Principle 1*. In contrast, the trust values computed by each of three existing trust transitivity models are the same in each domain, *neglecting the influence of social positions*.

In addition, in MQCTT, if the aggregated CIF and CIF' value in a path do not satisfy the corresponding QoTT constrains, $T_{CIF} = T'_{CIF} = 0$ (e.g., S7 in Fig. 7.6). This follows *Property 3* of trust.

7.3.2.4 Scenario 4: trust transitivity based on different preference similarity

To investigate the performance of these models in trust transitivity with different preference similarity, PS is decreased to $PS' = PS/1.5$. The rest of the QoTT attributes

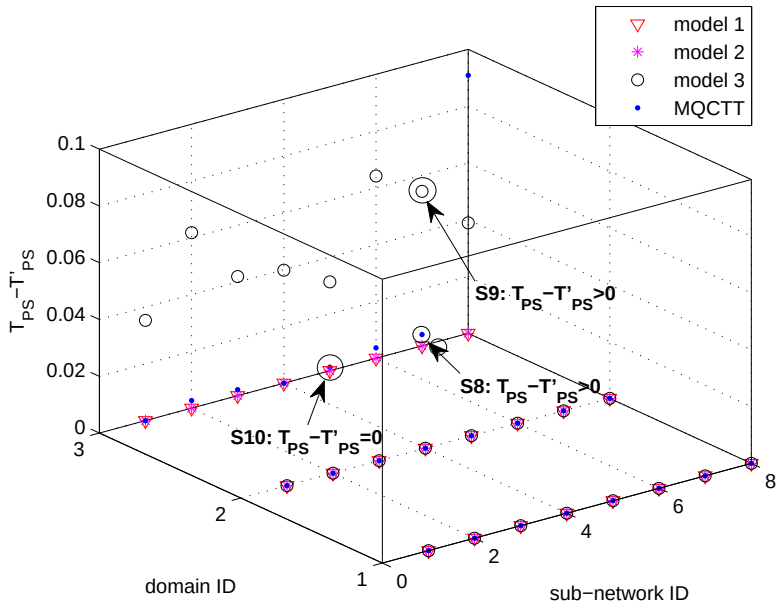

Figure 7.7: The results of $T_{PS} - T'_{PS}$

have the same values with those in scenario 1.

Fig. 7.7 plots the trust transitivity results computed based on PS (termed as T_{PS}) minus those computed based on CIF' (termed as T'_{PS}), i.e., $T_{PS} - T'_{PS}$. We can see that in some cases, $T_{PS} - T'_{PS} > 0$ in our MQCTT model (e.g., S8 in Fig. 7.7). Namely, the trust value computed by our proposed MQCTT model decreases with the decrease of S value when the social trust path satisfies the QoTT constraints, which follows *Principle 1*. In contrast, only the work in [51] follows this principle (e.g., S9 in Fig. 7.7), while other two models neglect the influence of the preference similarity between participants.

In addition, in MQCTT, if the aggregated PS and PS' values in a path do not satisfy the corresponding QoTT constrains, $T_{PS} = T'_{PS} = 0$ (e.g., S10 in Fig. 7.7). This follows *Property 3* of trust. However, the existing methods, including the model in [51] do not follow this trust property.

Summary: Based on the above experimental results and our analysis in the four scenarios, we can see that our proposed MQCTT model not only follows the principles in social psychology, but also follows the trust properties. Therefore, MQCTT can

compute a more reasonable trust value of the target participant than existing models.

7.4 Conclusion

In this chapter, we have proposed a general concept of Quality of Trust Transitivity (QoTT) and proposed a novel Multiple QoTT Constrained Trust Transitive (MQCTT) model in complex contextual trust-oriented social networks. Furthermore, we have conducted experiments on the datasets of real social networks. Experimental results have demonstrated that our MQCTT model follows principles in social psychology and properties of trust, and thus it computes more accurate trust transitivity results than existing methods.

The trust transitivity model proposed in this chapter can help compute reasonable propagated trust values along the selected social trust paths, which can support a participant to make a correct decision during the service selections or collaborations with other unknown participants in OSNs.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In recent years, many people have joined in Online Social Networks (OSNs). In the social networking platforms, the participants conduct a variety of activities, like seeking employees and movie recommendations. In these activities, trust is one of the most important indications for participants' decision making. However in OSNs, most participants do not have direct interactions previously (e.g., there is no previous interaction between an employer and an employee, and there is no previous interaction between a movie recommender and a movie recommendee). Therefore, evaluating trust between two unknown participants becomes significant and necessary.

In this thesis, in order to provide an efficient and effective trust evaluation method to deliver a reasonable trust value, three major contributions have been made. The contributions are summarised below.

1. The first contribution of the work presented in this thesis is trust network extraction in contextual trust-oriented social networks. This is a fundamental step to perform any trust evaluation methods.

   (a) The current social network structures do not consider the social contextual information, including social relationships, social positions, preferences and residential locations. These social contextual information have significant influence on trust management. In our model, several social contextual impact factors have been proposed. In addition, we have proposed a

contextual trust-oriented social network structure which contains the above factors, reflecting the social networks in real world scenarios better.

(b) In our trust network extraction method, a general concept Quality of Trust Networks (QoTN) has been proposed, which contains the social contextual impact factors, including the social intimacy degree, the community impact factor, the preference similarity and the residential location distance as attributes. The value of QoTN can illustrate the ability of the extracted trust network to deliver a trustworthy trust evaluation result.

(c) As discussed before, a source participant may have different trust evaluation criteria in trust evaluation, and social context can impact on the social interactions between two participants. To address these issues, we have proposed a social context-aware trust network extraction model with QoTN constraints. In the literature, there are no algorithms for the NP-Complete QoTN constrained trust network extraction problem. We have proposed several approximation algorithms and heuristic algorithms. Experimental results have demonstrated the superior performance of the proposed methods.

2. The second contribution of the work presented in this thesis is trust path selection in contextual trust-oriented social networks. As evaluating trust via all the social trust paths in a large-scale extracted trust network is computationally infeasible, selecting those trust paths which can deliver most trustworthy trust evaluation results is necessary and significant.

(a) In our trust path selection method, a general concept Quality of Trust (QoT) has been proposed, which contains the social contextual impact factors, including the social intimacy degree and the community impact factor, as they have significant influence on trust path selection. The value of QoT can illustrate the ability of a social trust path(s) to guarantee a certain level of trust in trust evaluation.

(b) In our model, a source participant can specify multiple end-to-end QoT constraints to reflect their trust evaluation criteria. Then the trust path selection problem is modeled as the classical Multiple Constrained Optimal Path (MCOP) selection problem, which is NP-Complete.

(c) The proposed algorithms for trust path selection in the literature do not consider the social context including social relationship and community impact factor. We have proposed several approximation algorithms and heuristic algorithms for optimal social trust path selection, and a heuristic algorithm for K optimal social trust paths selection by considering the social contexts and adopting our novel search strategies. Experimental results have demonstrated the proposed algorithms outperform the existing methods in both the quality of the identified trust path(s) and the efficiency.

3. The third contribution of the work presented in this thesis is a novel model of trust transitivity in contextual trust-oriented social networks. After identifying the trustworthy social trust path(s), in order to compute the reasonable trust value of the target, understanding how trust is propagated along the trust path is a critical and challenging problem.

(a) In our trust path selection method, a general concept Quality of Trust Transitivity (QoTT) has been proposed, which contains the social contextual impact factors, including social intimacy degree, community impact factor and preference similarity, as they have significant influence on trust transitivity. The value of QoTT can illustrate the ability of a social trust path to guarantee a certain level of quality of trust transitivity.

(b) Then based on the properties of trust illustrated in social psychology, we have proposed a new Multiple QoTT Constrained Trust Transitivity (MQCTT) model. Experimental results demonstrate that the proposed trust transitivity model follows both the principles in social psychology and the proper-

ties of trust, and thus it computes more reasonable trust values than existing methods.

8.2 Future Work

In relation to foundational studies, trust situation between two participants is dynamic. Therefore, in order to compute a more reasonable trust evaluation result, in addition to the current proposed trust management models, we plan to study an efficient and effective method to compute the updated trust situation based on monitoring and analysing the conversations between participants in OSNs in real-time. Then, based on the trust information, a mining method will be proposed to find the most trustworthy participant in a specific area, which can help individuals find high quality recommendations and help companies find marketing targets in the area.

In relation to real applications, our work provides several key techniques to many applications with social networks as the backbone. Based on them, a social network based trust-oriented recommendation system can be developed, which maintains a social network with complex social contextual information. In such a system, our trust management methods can help, for example, help a buyer to find the most trustworthy seller who sells the product preferred by the buyer. Similarly, a new generation of social network based recruitment system and a new generation of social network based CRM system can be developed, where the proposed trust management methods can help an employer to find the most trustworthy potential employees, or help a retailer to find loyal customers.

Chapter 9

Notations Used in This Thesis

Table 9.1: Notations Used in Chapter 3

Notation	Representation	First occurrence
$CIF_A^{D_i}$	the Community Impact Factor of A in domain i	Section 3.2.3
$deg^-(A)$	the indegree of A	Section 3.1.2.2
$deg^+(A)$	the outdegree of A	Section 3.1.2.3
$NE(v)$	the neighboring nodes of v	Section 3.1
$PF_A^{D_i}$	A's preference in domain i	Section 3.1.1.2
$PS_{A,B}^{D_i}$	the preference similarity between A and B in domain i	Section 3.2.4
$RL(A)$	the residential location where A lives in	Section 3.1.1.3
$RLD_{A,B}$	the residential location between A and B	Section 3.2.5
SC	the social context in a social network	Definition 1
SI_{AB}	the Social Intimacy Degree between A and B	Section 3.2.2
$SP_A^{D_i}$	A's social position in domain i	Section 3.1.1.1
$SR_{A,B}^{TY_j}$	the j^{th} type of social relationship between A and B	Section 3.1.2.1
$T_{AB}^{D_i}$	the trust value that A assigns to B in domain i	Section 3.2.1

Table 9.2: Notations Used in Chapter 4

Notation	Representation	First occurrence
$b-ClosedSet$	a set to store the expansion nodes in the Backward-Search	Section 4.6.3
$b-OpenSet$	a set to store the candidates of expansion nodes in the Backward-Search	Section 4.6.3
$f-ClosedSet$	store the expansion nodes	Section 4.6.3
$f-OpenSet$	store the candidates of expansion nodes	Section 4.6.3
K'	the number of the candidates of v_{mg}^{-} in the search of a layer	Section 4.6.2
K''	the number of v_{mg}^{-} in the search of a layer	Section 4.6.2
K^{*}	the number of the candidates of v_{mg}^{+} in the search of a layer	Section 4.6.2
K^{**}	the number of v_{mg}^{+} in the search of a layer	Section 4.6.2
d	the maximal outdegree of the nodes in a social network	Section 4.5.3
M	the number of the intermediate nodes in a trust network	Section 4.2.2
$MT(v_s, v_t)$	the complex trust-oriented social network between v_s and v_t	Section 4.5
N	the number of the corresponding links in a trust network	Section 4.2.2
$OpenSet$	store the candidates of expansion nodes	Section 4.5
$PreNE(v_b)$	preceding neighboring nodes of v_b (the nodes have direct links to v_b)	Section 4.6.3
$QoTN$	Quality of Trust Network	Section 4.2.2
$SCP(v_f \rightarrow v_t)$	the normal distribution based selection probability between v_f and v_t	Section 4.3.2
$SCP_{v_f,v_t}^{D_i}$	the context similarity based selection probability between v_f and v_t in domain i	Section 4.6.2
$Smi_{v_f,v_t}^{D_i}$	the social context similarity between v_f and v_t in domain i	Section 4.6.2
$\mathcal{U}$	the utility of an extracted trust network	Section 4.2.3
$v.bvisit$	the status indicates whether a node is selected in Backward-Search	Section 4.6.3
v_{exp}	an expansion node	Section 4.3.2
v_f	a feasible node	Section 4.3.2
$v.fvisit$	the status indicates whether a node is selected in Forward-Search	Section 4.6.3

Table 9.3: Notations Used in Chapter 4 (continued)

Notation	Representation	First occurrence
v_m	an intermediate node	Section 4.2.1
v_{mg}^{+}	a marginal node which has a low probability to connect with the target	Section 4.6.1
v_{mg}^{-}	a marginal node which has a low probability to connect with the target	Section 4.6.1
v_s and v_t	the source and target respectively	Section 4.2.2
λ_h	the threshold of search hops	Section 4.3.2

Table 9.4: Notations Used in Chapter 5

Notation	Representation	First occurrence
$AQ^{\mu}(p)$	the aggregated value of QoT attribute ($\mu \in \{T,\ SI,\ CIF\}$) of path p	Section 5.6.2
BLP	backward local path	Definition 4
CBLP	composite backward local path	Definition 6
$CIF_{p_{(a_1,\ldots,a_n)}}$	the aggregated community impact factor of path $p_{(a_1,\ldots,a_n)}$	Section 5.1.2.3
FLP	forward local path	Definition 5
$fp_{v_s\rightarrow v_n\rightarrow v_t}^{f(u)+b(\delta)}$	the foreseen path from v_s to v_t via v_n	Section 5.4.1
$\mathcal{F}_{p_{(a_1,\ldots,a_n)}}$	the utility of path $p_{(a_1,\ldots,a_n)}$	Section 5.1.3
$p_{v_s\rightarrow v_t}^{backward}$	the social trust path identified by the *Backward_Search* procedure	Section 5.6.2
$p_{v_k\rightarrow v_t}^{b(CIF)}$	the BLP from v_k to v_t with the maximal aggregated CIF value	Section 5.6.2
$p_{v_k\rightarrow v_t}^{b(SI)}$	the BLP from v_k to v_t with the maximal aggregated SI value	Section 5.6.2
$p_{v_k\rightarrow v_t}^{b(T)}$	the BLP from v_k to v_t with the maximal aggregated T value	Section 5.6.2
$p_{v_n\rightarrow v_t}^{b(\delta)}$	the path from v_n to v_t with the minimal δ value	Section 5.4.1
$p_{v_k\rightarrow v_t}^{CBLP^M(CIF)}$	the CBLP from v_k to v_t with part of $p_{v_k\rightarrow v_t}^{b(CIF)}$, M is the number of the intermediate nodes of $p_{v_k\rightarrow v_t}^{b(CIF)}$	Section 5.6.2

Table 9.5: Notations Used in Chapter 5 (continued)

Notation	Representation	First occurrence
$p_{v_k \to v_t}^{CBLP^M(SI)}$	the CBLP from v_k to v_t with part of $p_{v_k \to v_t}^{b(SI)}$, M is the number of the intermediate nodes of $p_{v_k \to v_t}^{b(SI)}$	Section 5.6.2
$p_{v_k \to v_t}^{CBLP^M(T)}$	the CBLP from v_k to v_t with part of $p_{v_k \to v_t}^{b(T)}$, M is the number of the intermediate nodes of $p_{v_k \to v_t}^{b(T)}$	Section 5.6.2
$p_{v_s \to v_t}^{forward}$	the social trust path identified by the *Forward_Search* procedure	Section 5.6.2
$p_{v_s \to v_n}^{f(u)}$	the path from v_s to v_n with the maximal utility	Section 5.4.1
QoT	Quality of Trust	Definition 3
Q_{v_s,v_t}^{μ}	the end-to-end QoT constraint of QoT attribute μ $\mu \in \{T,\ SI,\ CIF\}$)	Section 5.1.1
$SI_{p_{(a_1,\ldots,a_n)}}$	the aggregated social intimacy degree of path $p_{(a_1,\ldots,a_n)}$	Section 5.1.2.2
$T_{p_{(a_1,\ldots,a_n)}}$	the aggregated trust value of path $p_{(a_1,\ldots,a_n)}$	Section 5.1.2.1
v_k	an intermediate node in a sub-network	Section 5.2.1
v_n	an neighboring node of v_s	Section 5.4.1
v_u	an unvisited node	Section 5.2.3
$g_\lambda(p)$	the objective function defined in H_MOCP	Section 5.2.1
$\xi(p)$	the objective function defined in MCSP_K	Section 5.2.1
$\delta(p)$	the objective function defined in H_OSTP	Section 5.4.1
μ	QoT attributes	Section 5.1.1

Table 9.6: Notations Used in Chapter 6

Notation	Representation	First occurrence
$fp_{v_s \to v_m \to v_t}^{F_i+B_K}$	K *foreseen paths* from v_s to v_t via v_m	Section 6.3.1
$p_{v_s \to v_t}^{B_K}$	K social trust paths from v_t to v_s with the K minimal δ	Section 6.3.1
$p_{v_s \to v_t}^{F_K}$	K social trust paths from v_s to v_t with the K maximal utility	Section 6.3.1

Table 9.7: Notations Used in Chapter 7

Notation	Representation	First occurrence
k_1	the slope of Base Line	Section 7.2
k_2	the slope of Deviation Line	Section 7.2
$PS_{p(a_1,...,a_n)}$	the aggregated PS value of path $p_{(a_1,...a_n)}$ in a certain domain	Section 7.1.3
QoTT	Quality of Trust Transitivity	Section 7.1.2
$T_{a_1,a_{j+1}}$	trust transitivity result between a_1 and a_{j+1}	Section 7.2.1
λ_1	the number of hops of trust transitivity in Phase 1	Section 7.1
λ_2	the number of the hops where trust $p_{(a_1,...a_n)}$ approaches to zero in Phase 3	Section 7.1
θ	intersection angle	Section 7.1
μ'	QoTT attributes	Section 7.1.3

Bibliography

[1] M. Abell, J. Braselton, and J. Rafter. *Statistics with Maple*. Elsevier Science and Technology, 2012.

[2] L. A. Adamic, R. M. Lukose, and B. A. Huberman. Local search in unstructured networks. In *Handbook of Graphs and Networks*. Wiley, 2005.

[3] P. S. Adler. Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, 12(12):215–234, 2001.

[4] S. S. Andeleeb. An experimental investigation of satisfaction and commitment in marketing channels: The role of trust and dependence. *Journal of Retailing*, 72(1):77–93, 1996.

[5] R. Ashri, S. Ramchurn, J. Sabater, M. Luck, and N. Jennings. Trust evaluation through relationship analysis. In *AAMAS'05*, pages 1005–1011.

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

[7] E. Barnett and M. Casper. A definition of social environment. *American Journal of Public Health*, 91(3), 2001.

[8] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In *IJCAI*, pages 2677–2682, 2007.

[9] E. Berscheid and H. T. Reis. Attraction and close relationships. In *The Handbook of Social Psychology*.

[10] J. Bi, J. Wu, and W. Zhang. A trust and reputation based anti-spim method. In *INFOCOM'08*, pages 2485–2493, 2008.

[11] M. Bittinger. *Basic Mathematics*. Addison Wesley, 2002.

[12] H. M. Blalock. *Social Statistics*. New York: McGraw-Hill, 1979.

[13] P. Blau. A theory of social integration. *American Journal of Sociology*, 65(6):545–556, 1960.

[14] D. Boyd and N. Ellison. Social network sites: Definition, history and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2007.

[15] J. L. Braddach and R. G. Eccles. Price, authority, and trust: From ideal types to plural forms. *Annual Review of Sociology*, 15:97–118, 1989.

[16] D. J. Brass. *A Socal Network Prespective On Industral/organizational psychology*. Industrial/Organizational Handbook, 2009.

[17] J. Brockner, P. A. Siegel, J. P. Tyler, and C. Martin. When trust matters: The moderating effect of outcome favorability. *Administrative Science Quarterly*, 43:558–583, 1997.

[18] R. S. Burt. Decay functions. *Social Networks*, 22(4):1–28, 2000.

[19] N. Chang and M. Liu. Revisiting the ttl-based controlled flooding search: Optimality and randomization. In *MobiCom'04*, pages 85–99.

[20] P. Chia and G. Pitsilis. Exploring the use of explicit trust link for filtering recommenders: A study on epinions.com. *Journal of Information Processing*, 19:332–344, 2011.

[21] Y.-S. Cho, G. V. Steeg, and A. Galstyan. Co-evolution of selection and influence in social networks. In *AAAI*, 2011.

[22] S. Chow and R. Holden. Toward and understanding of loyalty: The moderating role of trust. *Journal of Managerial*, 43:558–583, 1997.

[23] B. Christianson and W. S. Harbison. Why isn't trust transitivie? In *International Workshop on Security Protocols*, pages 171–176, 1996.

[24] F. Chua and E.-P. Lim. Trust network inference for online rating data using generative models. In *KDD'10*, pages 889–898.

[25] K. Cook. *Trust in Society*. New York: Russell Sage Foundation, 2001.

[26] P. Cui, F. Wang, S. Yang, and L. Sun. Item-level social influence prediction with probabilistic hybrid factor matrix factorization. In *AAAI*, 2011.

[27] M. Dalton. *Men Who Manage*. New York: Wiley, 1959.

[28] R. J. Deluga. The relation between trust in the supervisor and subordinate organizational citizenship behavior. *Military Psychology*, 7(1):1–16, 1995.

[29] M. Deutsch. Scooperation and trust. some theoretical notes. In *Nebraska Symposium on Motivation*. Nebraska University Press, 1962.

[30] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(3):97–166, 2001.

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

[32] M. Dodgeson. Learning, trust, and technological collaboration. *Human Relations*, 46(1):77–95, 1993.

[33] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1999.

[34] A. Felner, S. Kraus, and R. E. Korf. Kbfs: K-best-first search. *Annals of Mathematics and Artificial Intelligence*, 39:19–39, 2003.

[35] T. Ferdinand. *Community and Society*. Michigan State University Press, 1887.

[36] T. Ferdinand. *The Division of Labor in Society*. New York: Free Press, 1893.

[37] I. Filali and F. Huet. Dynamic ttl-based search in unstructured peer-to-peer networks. In *CCGrid'10*, pages 438–447.

[38] S. Fiske. *Social Beings: Core Motives in Social Psychology*. John Wiley and Sons Press, 2009.

[39] B. FoX. K-th shortest paths and applications to the probabilistic networks. *Joint National Meeting of ORSA/TIMS*, 23, 1975.

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

[41] F. Fukuyama. *Trust: The Social Virtues and The Creation of Prosperity*. New York: Free Press, 1996.

[42] F. Fukuyama. *The Moral Foundations of Trust*. Cambridge University Press, 2002.

[43] J. Gentle, W. Hardle, and Y. Mori. *Handbook of Computational Statistics*. Springer, 2004.

[44] J. Gimpel, K. Karnes, J. Mctague, and S. Pearson-Merkowitz. Distance-decay in the political geography of friends-and-neighbors voting. *Political Geography*, 27:231–252, 2008.

[45] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *IEEE INFORCOM'04*, pages 120–130.

[46] C. Gkantsidis, M. Mihall, and A. Saberi. Random walks in peer-to-peer networks: algorithm and evaluation. *Performance Evaluation*, 63(3):241–263, 2006.

[47] J. Golbeck and J. Hendler. Accuracy of metrics for inferring trust and reputation. In *Proceedings of 14th International Conference on Knowledge Engineering and Knowledge Management*, 2004.

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

[49] J. Goldstein, A. Kwasinksi, 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.

[50] E. Gray, J. Seigneur, Y. Chen, and C. Jensen. Trust propagation in small world. In *iTrust'03*, pages 239–254, 2003.

[51] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW'04*, pages 403–412, 2004.

[52] S. Guo, M. Wang, and J. Leskovec. The role of social networks in online shopping information passing, price of trust, and consumer choice. In *EC'11*, pages 130–137, 2011.

[53] C. Hang, Y. Wang, and M. Singh. Operators for propagating trust and their evaluation in social networks. In *AAMAS'09*, pages 1025–1032, 2009.

[54] R. Hardin. *Trust and Trustworthiness*. Russell Sage Foundation, 2002.

[55] P. W. Holland and S. Leinhardt. Transitivity in structural models of small groups. *Comparative Group Studies*, 2:107–124, 1971.

[56] Y. Hu, Y. Fan, and Z. Di. Orientation in social networks. *Physics and Society*, 1(1), 2009.

[57] P. Ira. *Bi-directional Search*. Edinburgh University Press, 1971.

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

[59] S. Jones. *TRUST-EC: Requirements for Trust and Confidence in E-commerce*. CEC, 1999.

[60] M. Jorge and X. Yi. *Return to RiskMetrics: The Evolution of a Standard*. RiskMetrics, 2001.

[61] A. Jøsang, E. Gary, and M. Kinateder. Analysing topologies of transitive trust. In *FAST'03*, 2003.

[62] A. Jøsang, E. Gary, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[63] A. Jøsang and S. Pope. Semantic constraints for trust transitivity. In *APCCM'05*, pages 59–68, 2005.

[64] S. E. Kingsland. *Modeling Nature*. University of Chicago Press, 1995.

[65] M. A. Konovsky and S. D. Pugh. Citizenship behavior and social exchange. *Academy of Management Journal*, 37(3):656–669, 1994.

[66] K. Konrad, G. Fuchs, and J. Bathel. Trust and electronic commerce ł more than a technical problem. In *The 18th Symposium on Reliable Distributed Systems*, 1999.

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

[68] R. F. Korte. Biases in decision making and implications for human resource development. *Advances in Developing Human Resources*, 5(4):440–457, 2003.

[69] R. F. Korte. Biases in decision making and implications for human resource development. *Advances in Developing Human Resources*, 5(4):440–457, 2003.

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

[71] 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, 2007.

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

[73] L. Li, Y. Wang, and E. Lim. Trust-oriented composite services selection and discovery. In *ICSOC'09*, pages 50–67, 2009.

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

[75] S. Lichtenstein and P. Slovic. *The construction of preference*. Cambridge University Press, 2006.

[76] S. Lichtenstein and P. Slovic. *The Construction of Preference*. Cambridge University Press, 2006.

[77] 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, 2009.

[78] 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.

[79] 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.

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

[81] G. Liu, Y. Wang, and M. A. Orgun. Social context-aware trust network discovery in complex contextual social networks. In *AAMAS'12*.

[82] G. Liu, Y. Wang, and M. A. Orgun. Trust transitivity in complex social networks. In *AAAI'11*, pages 1222–1229.

[83] G. Liu, Y. Wang, and M. A. Orgun. Optimal social trust path selection in complex social networks. In *AAAI'10*, pages 1397–1398, 2010.

[84] G. Liu, Y. Wang, M. A. Orgun, and E.-P. Lim. Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks. *IEEE Transactions on Services Computing (TSC)*, 2011.

[85] G. Liu, Y. Wang, M. A. Orgun, and H. Liu. Discovering trust networks for the selection of trustworthy service providers in complex contextual social networks. In *ICWS'12*, pages 384–391.

[86] G. Liu, Y. Wang, and D. Wong. Multiple qot constrained social trust path selection in complex social networks. In *TrustCom'12*.

[87] D. Lo, D. Surian, K. Zhang, and E.-P. Lim. Mining direct antagonistic communities in explicit trust networks. In *CIKM'11*, pages 1013–1018.

[88] N. Luhmann. *Trust and Power*. Wiley, 1979.

[89] H. Ma, I. King, and M. R. Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–19, 2011.

[90] H. Ma, T. Zhou, M. Lyu, and I. King. Improving recommender systems by incorporating social contextual informaiton. *ACM Transactions on Information Systems*, 29(2), 2011.

[91] R. Mansell and B. Collins. *Trust and crime in information societies*. Edward Elgar Publishing, 2005.

[92] I. Markova, A. Gillespie, and J. Valsiner. *Trust and Distrust: Sociocultural Perspectives*. Information Age Publishing, 2008.

[93] S. Marsh. *Formalising trust as a computational concept*. PhD Thesis, 1994.

[94] E. Martins, M. Pascoal, and J. Santos. Deviation algorithms for ranking shortest paths. *International Journal of Foundations of Computer Science*, 10(3):247–261, 1999.

[95] D. J. McAllister. Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations. *Academy of Management Journal*, 38:24–59, 1995.

[96] 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.

[97] I. H. McKnight and N. L. Chervany. *The meanings of trust*. Technical Report, 1996.

[98] M. Mcpherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[99] K. Merton. The role set problems in sociological theory. *British Journal of Sociology*, 8(2):110–113, 1957.

[100] S. Miaou and S. Chin. Computing k-shortest path for nuclear spent fuel highway transportation. *European Journal of Operational Research*, 53:64–80, 1975.

[101] S. Milgram. The small world problem. *Psychology Today*, 2(60), 1967.

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

[103] 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, 2007.

[104] D. Morton and E. Popova. Monte-carlo simulation for stochastic optimization. *Encyclopedia of Optimization*, pages 2337–2345, 2009.

[105] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2), 2003.

[106] K. O'Hara and W. Hutton. *Trust:From Socrates to Spin*. Icon Books Ltd, 2004.

[107] C. D. Parks, R. F. Henager, and S. D. Scamahorn. Trust and reactions to messages of intent in social dilemmas. *Journal of Conflict Resolution*, 40(1):134–151, 1996.

[108] T. Parsons. *The Structure of Social Action: A Study in Social Theory with Special Reference to a Group of European Writers*. NY: The Free Press, 1937.

[109] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[110] I. Pool and M. Kochen. Contacts and influence. *Social Networks*, 1:1–48, 1978.

[111] D. Povey. Trust management. In *http://security.dstc.edu.au/presentations/trust/*, 1999.

[112] C. L. Prell. Community networking and social capital: Early investigation. *Journal of Computer Mediated Communication*, 8(3), 2003.

[113] D. Quercia, S. Hailes, and L. Capra. Lightweight distributed trut propogation. In *ICDM'07*, pages 282–291, 2007.

[114] G. A. Rich. The sales manager as a role model: Effects on trust, job satisfaction, and performance of salespeople. *Journal of the Academy of Marketing Science*, 25(4):319–328, 1997.

[115] P. S. Ring and A. Van. Developmental processes of cooperative interorganizational relationships. *Academy of Management Review*, 19:90–118, 1994.

[116] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[117] F. L. S. Yoo, Y. Yang and I. Moon. Mining social networks for personalized email prioritization. In *KDD'09*, pages 967–976, 2009.

[118] J. Scott and P. J. Carrington. *The Sage Handbook of Social Network Analysis*. Sage Publications, 2011.

[119] J. P. Scott. *Social Network Analysis: A Handbook*. Sage Publications, 2000.

[120] A. B. Seligman. *The Problem of Trust*. Princeton University Press, 2000.

[121] J. E. Swan, M. R. Bowers, and L. D. Richardson. Customer trust in the salesperson: An integrative review and meta-analysis of the empirical literature. *Journal of Business Research*, 44(2):93–107, 1999.

[122] P. Sztompka. *Trust: A Sociological Theory*. Cambridge University Press, 1999.

[123] 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.

[124] F. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *AAMAS Journal*, 16(1):57–74, February 2008.

[125] G. Wang and J. Wu. Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Generation Computer Systems*, 17:529–538, 2011.

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

[127] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[128] A. C. Wicks, S. L. Berman, and T. M. Jones. The structure of optimal trust: Moral and strategic implications. *Academy of Management Review*, 24(1):99–116, 1999.

[129] J. Wren, K. Kozak, K. Johnson, S. Deakyne, L. Schilling, and R. Dellavalle. A survey of perceived contributions to papers based on byline position and number of authors. *EMBO Report*, 8(11):988–991, 2007.

[130] S. Yang, J. Zhang, and I. Chen. Web 2.0 services for identifying communities of practice. In *SCC*, pages 130–137, 2007.

[131] Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li. Social context summarization. In *SIGIR'11*, pages 255–264.

[132] I. Yaniv and E. Kleinberger. Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes*, 83(2):260–281, 2000.

[133] J. Yen. Shortest path network problems. *Mathematical Systems in Economics*, 1975.

[134] 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.

[135] R. Zajonc. Interpersonal attraction and attitude similarity. *Journal of Abnormal and Social Psychology*, 62(3):713–715, 1961.

[136] R. Zajonc. Mere exposure: A gateway to the subliminal. *Current Directions in Psychological Science*, 10(6):224–228, 2011.

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

[138] 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.