Searching For Expertise in Social Networks: A Simulation of Potential Strategies

Jun Zhang
School of Information
University of Michigan
Ann Arbor, MI, 48109
junzh@umich.edu

Mark S. Ackerman
EECS and School of Information
University of Michigan
Ann Arbor, MI, 48109
ackerm@umich.edu

ABSTRACT
People search for people with suitable expertise all of the time in their social networks – to answer questions or provide help. Recently, efforts have been made to augment this searching. However, relatively little is known about the social characteristics of various algorithms that might be useful. In this paper, we examine three families of searching strategies that we believe may be useful in expertise location. We do so through a simulation, based on the Enron email data set. (We would be unable to suitably experiment in a real organization, thus our need for a simulation.) Our emphasis is not on graph theoretical concerns, but on the social characteristics involved. The goal is to understand the tradeoffs involved in the design of social network based searching engines.

Categories and Subject Descriptors
H.5.3 [**Group and Organization Interfaces**]: Computer-supported cooperative work, Evaluation/methodology; H.3.3 [**Information Search and Retrieval**]: Search process

General Terms
Human Factors, Experimentation, Algorithms

Keywords
Computer-Supported Cooperative Work, CSCW, expertise location, expertise finding, expertise sharing, social computing, social networks, information seeking, organizational simulations

1. INTRODUCTION
Imagine you have a question that is blocking your work. For example, you might need help understanding a warning message from a critical application, and you're unable to locate a document explaining the message. Or as another example, you might need to understand how to work around a specific rule for ordering equipment.

In both of these situations, someone knows the answer to the question. Finding that person, however, can be difficult. Ideally, we would like to find a person who knows the correct answer to that specific problem. Additionally, we would like to ask only the appropriate person and to find a person who has enough free time to answer the question.

In reality, of course, answering questions is not so easy. People are busy, they may lack the requisite expertise to answer the question, or they may lack the social graces to answer well. As a first step, you may not know whom to ask.

Systems that help find others with appropriate expertise are called expertise finders or expertise location engines. These have been explored in a series of studies, including Streeter et al. [23] and McDonald and Ackerman [18] as well as the studies in Ackerman et al. [1]. Newer systems, that use the social network of an organization to help find people, have also been explored, most notably in Yenta [11], ReferralWeb [15], ER [18], and MARS [30]. These systems attempt to leverage the social network within an organization to help find the appropriate others, thus reducing the need for specialized data. This is a critical requirement for expertise finders, as requiring specialized data for expertise location makes adoption difficult at best.

Because each of these newer social network based expertise finders uses social network data (which may be derived in a number of ways), we can now move away from research emphasis on the systems and towards an examination of the algorithms used to search the social networks.

This paper surveys three algorithms in the open literature; it also adds several additional algorithms. These new algorithms, as will be seen, have interesting social characteristics. The main contribution of the paper, accordingly, is to examine those algorithms using a simulation testbed in order to evaluate them and understand their relative tradeoffs. We believe this work is critical if progress is to be made on finding methods and mechanisms for expertise location.

The paper proceeds as follows: First, we survey the related research. Second, we introduce our simulation experiments, including the data set we used, the algorithms we evaluated, and the performance measures we used. Third, we describe our analyses and findings. At last, we discuss the design implications and future work.

2. SEARCHING IN SOCIAL NETWORKS
In this section, we first review the rich literature about searching for people in social networks. Then we examine the computational approaches used for finding people in social networks, when the person is known in advance and when he is not. The more interesting case for us is the latter, since this is the expertise location problem.

GROUP'05, November 6–9, 2005, Sanibel Island, Florida, USA.

2.1 Small World

The classic study on searching in social networks is the "small world" experiment. In late 1960s, Milgram and Travers found that subjects could successfully send a small packet (with a name, the city, and the profession of the recipient on it) from Nebraska to people in Boston [19][24]. The subjects did so, even though they had only local knowledge of their acquaintances, by passing the packet to an acquaintance that they believed to be closest to the target. Travers and Milgram found the average length of acquaintance chain is roughly six. The result of this experiment indicated that the social network is searchable and that the paths linking people are short, the so-called "six degrees of separation".

A key question in such an experiment is how people select the next person to whom to forward the packet or message. Potentially each subject has hundreds of acquaintances, but picks one, which ultimately leads to a short chain between the sender and the target. Later similar experiments found that geographic proximity and similarity of profession to the target person were the most frequently used criteria by subjects [16][6][9].

Recently, mathematical models have been proposed to explain why these simple heuristics are good at forming short paths [17][26]. These models assume that the social network usually has a structure, in which individuals are grouped together by occupation, location, interest, and so on. As well, these groups are grouped together into bigger groups and so forth. The difference in people's group identities defines their social distance. By choosing individuals who have the shortest social distance to the target at each step, people can gradually reach the target in a short path with only local information about their own immediate acquaintances.

2.2 Searching For Expertise in Social Networks

These studies on the small world problem have led to two lines of computationally-based approaches that concern searching for people within social networks. The first is an automatization of the small world approach, where the target is known by name or unique identifier [3][28]. The second is locating a person with some specific expertise or knowledge. We consider the latter.

In an expertise location or expert finder problem, a suitable person or set of people is not known in advance. One must be found by matching people against a list of attributes.

A number of expertise location systems have been developed. For example, Who Knows [23] found people with appropriate expertise by doing latent semantic indexing of project reports, Yenta [11] found people by searching email archives in a distributed manner, and Expertise Recommender [18] used locally meaningful data to recommend sets of potential answerers for queries. Other work is surveyed in Ackerman, Wulf and Pipek[1].

Yu and Singh's referral system [30] is, as far as we know, the only paper that explicitly argues for a specific expertise-finding algorithm. In their experiment, they use the similarity between a query vector and a neighbor's expertise vector, plus some consideration of its historical referring performance, as the criteria for picking the next agent in a referral graph. The simulation results using a scientific co-authorship network suggest that this strategy can help people find experts in such a network.

Yu and Singh's algorithm is a useful first step, but their approach has limitations. There are several issues. First, the query vector and expertise vector in their experiment are manually coded; and each is a combination of preset topics from a taxonomy. This approach is not practical in real world scenarios: Questions are usually extremely detailed, and people cannot be categorized as one specific type of expert. Second, they had only a rudimentary consideration of the impact of the *social* network structure on the searching process. Finally, and most importantly for this paper, they did not compare the performance of their algorithm with other possible algorithms; thus, the relative benefits and tradeoffs of their algorithm is unknown. Nonetheless, Yu and Singh's algorithm is an important candidate for examination.

In addition to Yu and Singh's algorithm, several other algorithms can be adapted to the expertise location problem. Adamic et al's best connected search (BCS) algorithm [3][4], which makes use of the skewed degree distribution of many networks[1], can also be used to find experts. By passing the query to highly connected nodes first, BCS can spread a query quickly in the network. However, Adamic et al. also found that the BCS algorithm is not always efficient in all networks. Nonetheless, Adamic et al's algorithm may be valuable in many cases, and we will also include it in our investigation. Breadth First Search (BFS) [21], which broadcasts a query to every person in a social network, has the strength of finding the closest expert available. But it can have a high cost both computationally and socially in that many people can be bothered.

Thus there are three lines of potential algorithms in the open literature that need be examined. To our knowledge, these algorithms have never been evaluated together nor their tradeoffs and social characteristics examined. These social characteristics include standard attributes of social networks:

- Connections among people are not uniformly distributed. Unlike a theoretically constructed graph, the connections among people in a social network are highly meaningful and vary greatly [25].
- The connections between two individuals can have different strengths. There is a strength of association between individuals. This strength of association varies and is not always symmetrical. Usually, in social networks, the strength of association is divided roughly into strong and weak ties [12].
- People in a social network vary in their expertise, status, availability, and sociability. Unlike theoretically constructed graphs and computational agents, a person weighs his use of his social network by considering these additional characteristics.

These social characteristics could lead to sizeable differences in the way information is transferred, affecting the performance of the searching algorithms. For instance, weak ties have been found to be important in helping people get new information [12] and adopt innovations [7]. In Dodd et al.'s small world experiment, successful searches were also found to be conducted primarily through intermediate to weak strength ties.

[1] In such a network, many nodes just have one or several links and a few nodes have many links.

These social characteristics, in addition to computational efficiency, will guide our outcome measures. The following section introduces the outcome measures, but only after introducing the experimental testbed and the examined algorithms.

3. SIMULATION

In this section, we firstly discuss the simulation as our experimental apparatus. Second, we describe the data set we used and its limitations. Third, we introduce our new algorithms along with those previously proposed algorithms. Fourth, we describe the simulation process and the data we collected. Finally, we describe the evaluation criteria we used to compare these algorithms.

3.1 Simulation as Experiment

It may seem odd, at first glance, that we would wish to examine the social considerations and tradeoffs of these expertise locating (EL) approaches using simulations instead of field or laboratory experiments. However, simulations appear to be a much more fruitful experimental apparatus or testbed for examining these issues. This is unusual for CSCW investigations, and some explanation is required.

Constructing well-controlled laboratory experiments of the size required to effectively test these algorithms would at best be extremely difficult. On the other hand, while it might be possible to construct Internet-based experiments of a suitable size, these would be uncontrolled. Alternatively, Internet subjects would be required to run special applications (e.g. email monitoring software or email indexing software); this is extremely unlikely. Finally, a real organization (of a sufficiently large size) would provide us with enough users and use. However, we have been unable to convince any large corporation to either provide us with all of the company's email or to introduce experiments into their ongoing communication systems. It is unlikely that we will.

Accordingly, we examined simulations as a potential experimental apparatus for our investigations. We felt that the major problem with using simulations was the threat to the validity of our results.

Simulations are often too artificial. Overly rational agents, a small set of experimental categories and of agent behaviors (so as to be tractable), and severe limitations on methods of choice can lead to problematic social findings because of the restrictions. This is of course not necessarily true - one can look to the insightful simulations of Hutchens or Axelrod [14][5]. While socially limited, their restricted operationalizations have led to insights about cultural production and coordination, respectively.

In the following work, we have tried to avoid artificiality in two ways. First, we constructed our simulations using a data set from a real organization rather than using artificially or theoretically constructed data. The Enron email data set will be discussed below.

We also tried to operationalize our outcomes in a manner that was not overly restrictive. Of course, any operationalization in an experimental situation must be restricted, if nothing else to be concrete. Our operationalizations, which will also be discussed below, have implied limitations and restrictions. We have tried to account for these limitations both in our discussion and by doing detailed sensitivity analyses on our results, explicitly to look for the effect of such restrictions. We discuss these sensitivity analyses, and where we were unable to do one, below.

3.2 Simulation Data

The simulation data set is the well known Enron email dataset [8]. After cleaning the data, it contains data from 147 employees, mostly senior management of Enron. There are a total of 517,431 messages in the data set.

To construct a social network, a sub sample of 32766 messages that were exchanged among these 147 employees was used to construct a directed graph. As shown in figure 1, the network is a relative dense internal social network with 147 nodes. The density of the network is 0.096, and the average shortest path is 2.498.

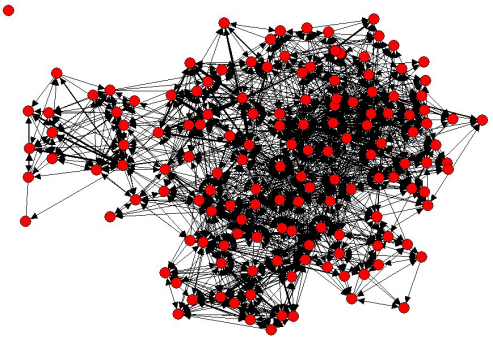


Figure 1: Enron Email Network

Figure 2 displays the cumulative out-degree[2] distribution of the network. It is highly skewed with some nodes having high degree in the tail.

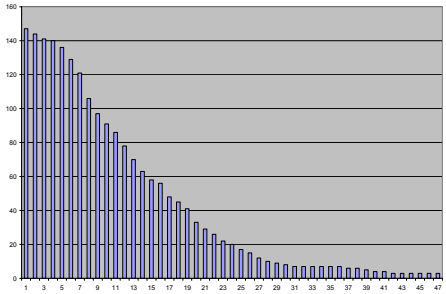


Figure 2: Cumulative Out-Degree Distribution of Enron email network

Second, using a standard document indexing method [29], we indexed all the messages that a person sent or received for each of the other 146 users. The indexed result for a single user is a keyword vector, in which a keyword is weighted by its term frequency inverse document (message) frequency. These indexes are used as information profiles for the users.

By doing this, we get a testbed with a network structure and information profiles derived from a real organization. There are, however, two possible limitations with this data set.

First, the simulation network is not the complete email social network of the Enron organization. It consists of the management

[2] Here we simply use the number of other users to whom a user had sent emails as his/her out-degree.

level subset of that network. Compared to the complete organization network (which we cannot obtain), this network is likely to be more dense with a smaller average shortest path. This is because general employees usually have a lower probability than managers of knowing people in other groups. Furthermore, managers may have different information profiles (or expertise) than other employees, such as engineers or clerical personnel. To examine whether the data set has different properties than the full social network, we constructed two new networks by removing edges that were weaker than a threshold and by removing high degree nodes. This gave us social networks with different network characteristics. We ran the same simulations on these two additional networks as a sensitivity analysis; we discuss the impact on the findings below.

Second, using this data set to determine expertise may be problematic. A keyword in one's email folder does not necessarily mean that one has expertise concerning that keyword. This is a limitation of our operationalization: we are assuming a perfect match between the information profiles and expertise. Determining an ontology of expertise and determining where it is located in an organization is a significant, ongoing research problem [2], and we believe this operationalization is a good surrogate. Furthermore, for transactive knowledge, communication is likely to be an indicator. Accordingly the information profiles are not only an indicator of this aspect of organizational expertise, they serve as an approximation of the location of other types of organizational expertise.

We believe that despite these limitations, the Enron data set gives us a realistic testbed, reducing the amount of artificiality in the simulation. We will discuss where its limitations affect the results below.

3.3 Searching Strategies Evaluated

We evaluated total 8 searching strategies from three families in this simulation; include 3 found in the public literatures and 5 proposed based on related theories. They are shown in Table 1.

Family	Name	Heuristic	From
General computational	BFS	Breadth first Search	Classic AI
	RWS	Random walk	Adamic et al.
Network structure based	BCS	Best connected	Adamic et al.
	WTS	Weak tie	* Granovetter, Burt
	STS	Strong tie	* Granovetter
	CSS	Cosine similarity	* Wasserman et al.
	HDS	Hamming distance	* Hamming
Similarity based	ISS	Information scent	Extract from Yu and Singh

Table 1. Evaluated Algorithms (* indicates algorithms that we proposed based on related works)

All these strategies are based on the information that a user can gather or derive locally from their email communications with peers. There may be additional strategies, such as using people's position in physical space or in an organizational hierarchy [3], but information available in the Enron data set limits us to the ones examined in this paper. In any case, the strategies examined here are, we believe, the most important ones to examine first.

Details of these algorithms are:

Breadth First Search (BFS) broadcasts a query to all of one's neighbors instead of picking a neighbor according to a heuristic. It can find the target closest to the source but with extremely high bandwidth costs (as in p2p file sharing networks).

Random Walk Search (RWS) randomly chooses one of the current query holder's neighbors to whom to spread the query. RWS will be our baseline to determine whether other heuristics are "better" in spreading a query.

Best Connected Search (BCS) is the algorithm Adamic et al. used. The only difference from their algorithm is that we construct our network as a directed network. Social relations are not symmetrical, and it may affect people's information seeking behavior in a social network. In any case the Enron data set has both outgoing and incoming messages. We use out-degree of connectivity instead of both in-degree and out-degree in evaluating one's neighbors.

Weak Tie Search (WTS) is proposed based on Granovetter's weak tie concept as discussed above. There are different ways of measuring relationship strength [22], here we simply assume that a peer who receives the fewest messages from a user is a weak tie and will be chosen as the next one to whom to forward the query.

Strong Tie Search (STS) is proposed as a comparison with WTS. It picks the neighbor who has received the most messages from the current user. It may be a reasonable strategy in practice because there is usually lower social cost when one asks for help through strong ties.

Hamming Distance Search (HDS) and *Cosine Similarity Search (CSS*) strategies are two structural dissimilarity strategies based on definitions of structure equivalence in social network studies [25]. HDS picks the neighbor who has the most uncommon friends from the current user. The definition of Hamming distance [13] favors the nodes with high out-degree. HDS could be viewed as an improved version of BCS. CCS decreases the high degree impact by dividing the Hamming distance by the total number of out-degree relations (friends) a neighbor has.

Information Scent Search (ISS) is extracted from Yu and Singh's algorithm [30], leaving aside the sociability learning part. ISS picks the next person who has the highest match score (which we call information scent) between the query and his profile. Our implementation of the algorithm is slightly different from Yu and Singh, since we needed to adapt their algorithm to the Enron data set. (Remember that Yu and Singh used only 19 categories or keywords.) We use the automatic generated keywords profile instead.

3.4 Process and Data Collection

We manually generated 147 questions by picking keywords from one or several messages from the sent folder of each of 147 email users. Each question has three to five keywords that do not include common words, such as "hi" "the", "ok", and "enron". Thus, we assumed each question has at least one expert available in the network.

During each round of the simulation, a question and an asker are selected at random. Each searching strategy is executed simultaneously at each round.

The match between a query and a person is calculated in two steps:

1) Message level matching: This is a standard information retrieval matching based on the TF/IDF measure. In general, if a message has the exactly the same combination of keywords as the query, it has the highest score; if it has only several keywords out of many, it has a low score.

2) Personal level aggregation: A person might have multiple messages with different matching scores that related to the query. This raises some issues. For instance, how could we compare a person who only has one document that has a very good match score with another person who has hundreds of related messages none of which is a good match? We chose to weight the documents by their ranking in one person's results in the personal level aggregating step.

   So, a person's match to the query is measured as $\sum_{i=1}^{n} MessageScore(i)/(i+1)$. We use the top 20 messages.

The criterion of a satisfied match is calculated by multiplying the best match score available, which is pre-calculated using a global search, with a satisfaction factor S (S=0.8 here).

The general query propagating process is as follows:

1) A user receives a query message (or the asker has a query).

2) The simulation engine searches all of the user's directed neighbors' information profiles. If there is a match above a desired threshold, it returns that person to the asker and stops the search. If there is not, the BFS strategy will broadcast the query to all of the -neighbors; other strategies will pick a neighbor according to their definitions. The visited node's ID is appended to the query message so a node would not be visited twice. Except for BFS, the asker starts a new searching path if the previous path reaches a dead end[4].

3) The query will be continually propagated in the network until no node is not visited (BFS) or no path is left (other strategies).

Note that in step 2, we assume that each user has knowledge of his direct neighbors' knowledge or has access to their profiles. It corresponds to transactive memory [27]. It is also the assumption used by Adamic et al in their small world experiments.

The data we collected during each round of the simulation include: asker's information scent on the query, steps (people used) to complete a query, number of paths tried (how many times a query needed to be restarted), number of people used, and the expertise score of the target. Since not all queries are successful because some nodes are not reachable from some other nodes, we record the number of search failures as well. After all rounds (N=30,000) are finished, we summarize overall how many times a user has been queried in each strategy.

We then calculate the out-degree and in-degree of each user. We used these to analyze their influence on the performance of algorithms.

3.5 Evaluation Criteria

Compared to searching a file in peer-to-peer file networks or searching for a person in a small world experiment, searching for expertise in social networks is a far more complex process. It involves many more social interactions. Speed and computational resource are not the only concerns; psychological and social costs are very important. After a social network based expertise system is adopted into an organization, the searching activities will be embedded into people's daily lives. So, an evaluation should not only consider the computational performance per query, but also needs to consider the social consequences of the strategies.

Based on these considerations and related work, besides analyzing the result from a computational efficiency perspectives, we compare the social cost of the evaluated algorithms using three measures:

- Number of people used per query (how many people were bothered).
- Depth of query chain (i.e., how deep the query went).
- Total labor distribution in all queries.

The number of people used per query is the measure Adamic et al. used in their simulation [3]. It counts how many nodes (people) processed the query during a search. It is a measure of social cost per query as well as the speed of the algorithm. When searching for information in social networks, we usually want to bother as few people as possible. If each used person took one unit of time to process a query and the query is propagated sequentially, we want the search process to be fast and bother the fewest number of people possible.

The depth of query chain measure, in many cases, is equal to the number of people used per query. It becomes different when there is more than one path used for a query. The depth of query chain counts only the number of people involved in the final successful path. In real life, less distance also means a high probability of getting response from an expert.

Labor distribution measures the overall social cost in an organization related to people's expertise seeking activities. Different from the people used per query, it counts how frequently a person is used by each searching strategy (during an entire simulation).

[4] It could reach a dead end or the Time-to-live (TTL) of the query message expires. The TTL is set to infinite in this simulation.

4. DATA ANALYSIS

In this section, first, we describe the general computation results. Second, we introduce the general findings related to the social cost measures. Third, we briefly analyze the impact of social characteristics on these algorithms. Finally, we discuss the sensitivity of the results by examining two modified networks.

4.1 General Computational Results

Table 2 displays the overall success rates of the algorithms. In the table, there are two categories of query failures. The first is when there is no path between the asker and available experts. All the failures in using BFS belong to this category. The second is when the algorithm cannot find available experts even when there are paths. For expertise location, we are primarily concerned with this type of failure. (The adjusted rate in the table shows the successful rate of a query presuming the first type of failure does not occur.)

From the table, we can see that these algorithms are reasonably successful. They can all find a qualified expert for most of the queries in this network. (Note with N=30,000, all differences are statistically significant. We omit p-values from our discussion except where important.)

Algorithm	BFS (b)	RWS (r)	WTS (w)	STS (s)	BCS (h)	ISS (i)	CSS (c)	HDS (d)
Success Rate (%)	97.9	94.7	96.2	95.8	97.1	97.1	97.1	97.1
Adjusted Rate(%)	100	96.8	98.3	97.9	99.2	99.2	99.2	99.2

Table 2: the success rate of various algorithms

Figure 3 further shows the percentage of successful queries within a given number of search steps using the various strategies. As one can see in the figure, for different search lengths, the rank of these algorithms changes very little. Although HDS and BCS are a little slower than BFS[5], they are still very fast and successfully finish 80% of the queries within six steps. CCS and ISS can still finish more than 60% queries, WTS can find 55%, but RWS and STS can only find about 40% within six steps.

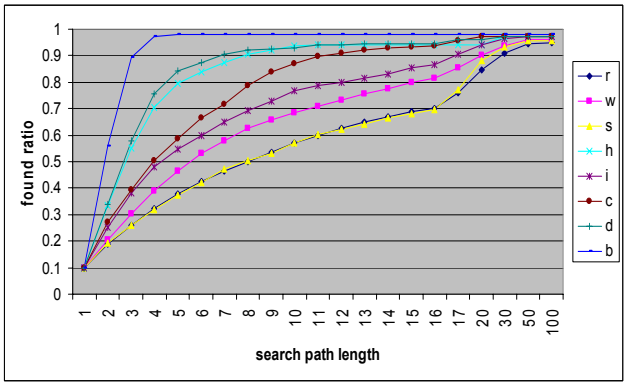


Figure 3: Percentage of succeed queries within different search length using various strategies

[5] Note in regarding the speed of BFS, we use the depth of the search instead of the number of people used in the query.

We can also see that when targets are far away from the askers, there is much less difference among these strategies.

4.2 Comparison of Social Costs

4.2.1 Number of People Used Per Query

Figure 4 shows the distribution of the number of people used per query using the different algorithms. As the system becomes less completely automatic, this value becomes increasingly important to people's user experience. Measures for these values are shown in table 3. Compared to the BFS broadcasting, HDS, BCS, and CSS strategies bother many fewer people. ISS and WTS are also clearly better.

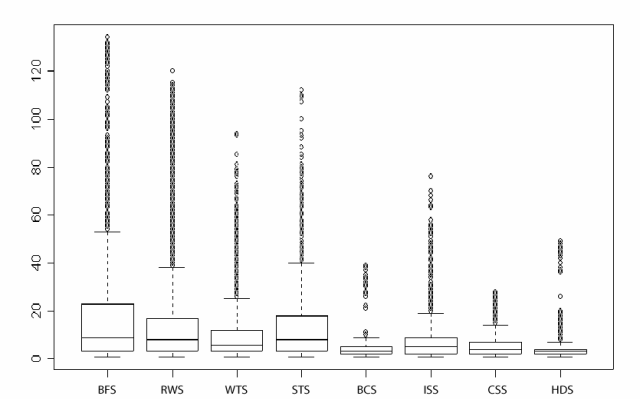


Figure 4: Distribution of Number of People used using various strategies

Algorithm	BFS (b)	RWS (r)	WTS (w)	STS (s)	BCS (h)	ISS (i)	CSS (c)	HDS (d)
Median	9	8	6	8	3	5	4	3
Max.	134	117	94	112	39	76	28	49

Table 3: Number of People used using various strategies

In Figurer 4, also note that there are a lot of outliers: Some queries used a lot of people before finding a desired target. Regarding the worst queries, as shown in Table 3, CSS handles them best and BFS handles them worst.

Based on these results, we can see that in this network, HDS, BCS, and CSS clearly have advantages over BFS and RWS regarding the number of people used per query. Also, ISS and WTS are better, but less so. Yu and Singh considered ISS to be very promising, but here we have found it less so than HDS, BCS, and CSS. Considering their performance as shown in Figure 3, HDS, BCS, and CSS could be promising algorithms to replace BFS when the speed and depth of searching chain are not the most important factors while the number of people being bothered is. The other interesting finding is that STS is obviously worse than WTS. The implications of this finding will be discussed later.

4.2.2 Depth of Query Path

Figure 5 shows how the depth of the query is distributed for each algorithm. This measures how long a query is in the social network; when designing a system, one would like to minimize these values. Except BFS, which we already knew always found the closest target, the result is not very different from measuring

the number of people used per query. We checked the number of paths tried for the various algorithms (except BFS) and found that most successful queries are finished using only one path. This indicates that at least in this social network, there is little need to send queries simultaneously to multiple users to achieve a successful result. This implied in our dataset, therefore, the two measures of depth of query path and number of people used would have the nearly same value.

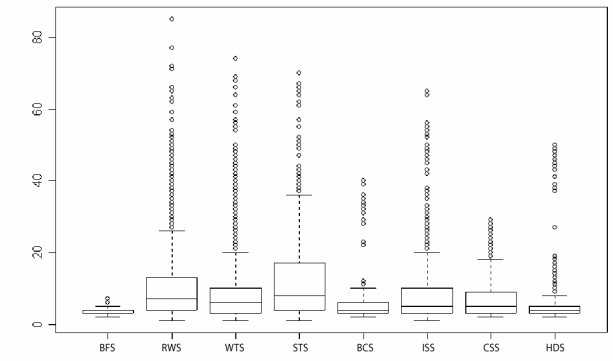


Figure 5: Distribution of depth of query chain using various strategies

4.2.3 Labor distribution

Figure 6 and Table 4 show the labor distribution (i.e., how distributed the search is) for these strategies. We can see that when using BCS, HDS, and CSS, most people are used less frequently, but some users are used extremely frequently. This indicates that referring is mainly loaded on very few members of the network. ISS is a little more balanced than these three algorithms, and BFS bothers people much more frequently than the other strategies. We will further discuss what strategies bother people more in section 4.3.1

Algorithm	BFS (b)	RWS (r)	WTS (w)	STS (s)	BCS (h)	ISS (i)	CSS (c)	HDS (d)
Median (%)	19.1	8.2	2.2	6.3	0.6	3.5	0.7	0.5
Max(%)	60.1	25.1	48.4	33.9	61.1	23.2	45.2	63.4

Table 4: Distribution of Labor using various strategies

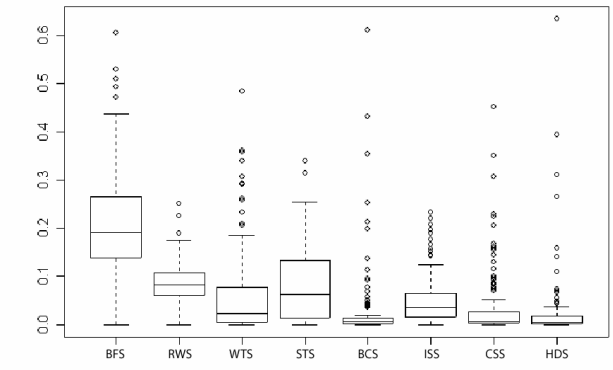


Figure 6: Distribution of a user's frequency of being used using various strategies.

4.3 Impact of Social Characteristics

We briefly looked at how different social characteristics influence the performance of these algorithms. Based on the findings from the previous section about social costs, we mainly discuss the impact of two characteristics of the social network: user's out-degree and tie strength.

4.3.1 Impact of User's Out-degree

Figure 7 displays correlations between a user's out-degree and frequency of being used using various strategies.

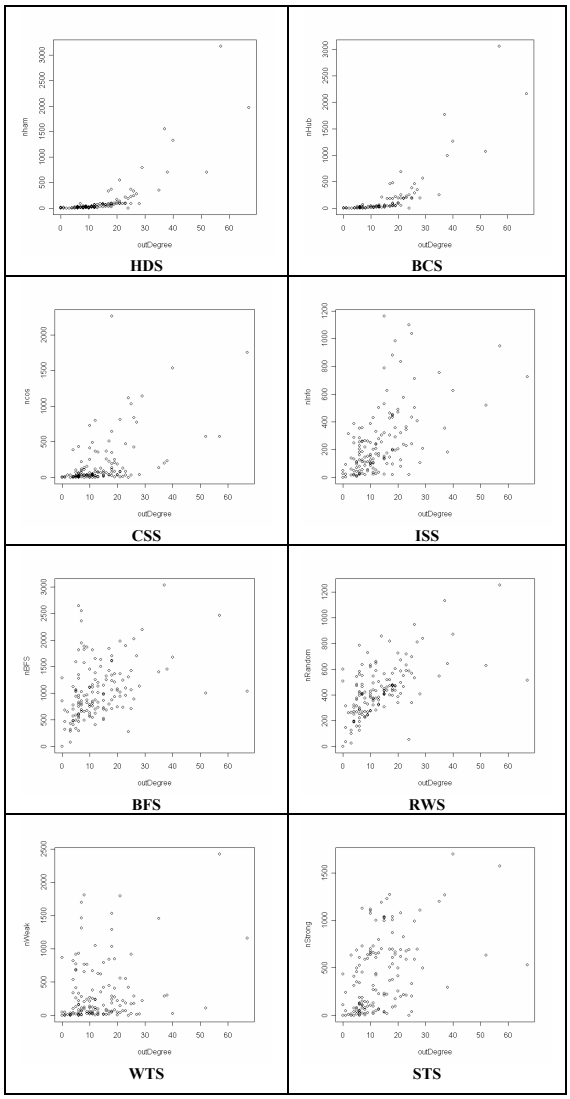


Figure 7: Correlation between a user's frequency of being used and his out-degree using an algorithm

Surprisingly, out-degree is important even when the algorithms are not explicitly designed with this in mind. This echoes findings regarding the importance of position in networks and node centrality in organizations (e.g., Burt [7]). We can see that when HDS and BCS are used, the relation looks close to exponential. People used most frequently are those highly connected people. This is not a surprise, since this is how these two algorithms are defined. More importantly, we checked the social status of those

frequently used people and found that the CEO, CIO, and the president of the company are central nodes (or social network hubs). This strongly suggests that if a similar algorithm and system are not totally automatic, they will not be practical in this organization. CSS is designed to decrease the impact of people's out-degree; thus, the correlation in its case is weak. However, it still uses a lot of highly connected users.

As well, there is an intermediate correlation when using RWS. This indicates that random walk is actually not random. As Newman [20] pointed out, nodes with high in-degrees have a high probability of being picked by other nodes in a random walk in a network. We found that there is a correlation between a user's in-degree and out-degree in our network, thus explaining the result here. The case of BFS is similar to RWS. People with high in-degrees also have a high chance being searched during the whole simulation process.

An interesting finding is that there seems some correlation even when IIS is used. The adjusted r-square is 0.31, $p<.001$. This indicates that the IIS strategy is not independent from the network structure. For instance, people have more social connections may have more diverse knowledge. This relationship is worth further investigating in the future.

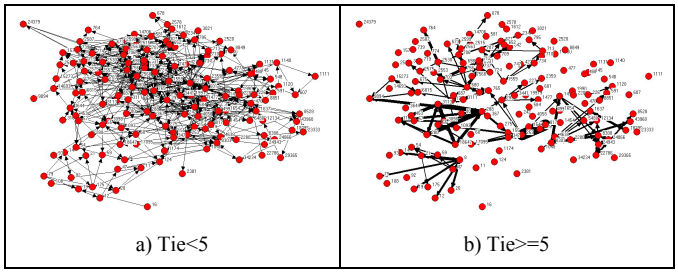
There is no clear correlation when WTS and STS are used.

4.3.2 Impact of Weak Ties

As described in previous findings, WTS seems more effective than STS. It spreads a query faster and bothers fewer people. To explore the reason for this difference, we visualized the distribution of these two types of ties into two network views, as shown in Figure 8. From these two views, we can see that weak ties are evenly distributed but strong ties form several local clusters. Thus, it seems the weak tie strategy propagates queries relatively evenly to other parts of the network, and the strong tie strategy usually makes local loops when forwarding the messages.

From this point of view, we can see that strong ties are not useful for seeking new information. However, we noted the motivational advantages of using strong ties. Any algorithms using strong ties, or thresholds for interpersonal association, may need to consider disjoint subgraphs.

The other interesting question is: since the different strengths of ties are not evenly distributed in this social network, what is the impact to other information searching algorithms? We further discuss this issue in section 4.4.1.



a) Tie<5 b) Tie>=5

Figure 8: Layered network with various tie strength

4.4 Sensitivity Analysis

As we discussed earlier, the availability of high degree nodes and weak ties are important for searching algorithms we evaluated in this paper. However, different networks will have different degree distributions: the Enron data set only presents a single case and it is a very dense social network. Even within other social networks, the availability of weak ties is not stable and changes frequently [12]. To evaluate how these algorithms will accommodate to changes of density and tie strength, we carried out two sensitivity analyses using modified networks. The first redefined weak ties and the other removed users with varying out-degrees.

4.4.1 Removing Weak Ties

We first modified the network by removing ties that had less than 5 messages[6]. The result network is the one shown in Figure 8b (density=0.041, average shortest path=3.435).

We then ran the same simulation on this modified network with the same settings. Because of the changed cut point (or threshold for weak ties), note that the operationalization of "weak" tie here is not the same as in the previous simulation.

Table 5 shows the successful rate of this simulation. As can be seen, compared to original network, about 23% more queries cannot be finished because the network became less connected. More interestingly, as can be seen from the adjusted rate, there is a clear performance drop for RWS, WTS, STS, and ISS. This indicates that RWS, WTS, STS, and ISS are sensitive to weak ties and related network structure changes while BCS, HDS, and CSS are less so.

Algorithm	BFS (b)	RWS (r)	WTS (w)	STS (s)	BCS (h)	ISS (i)	CSS (c)	HDS (d)
Success rated (%)	76.3	44.0	40.0	45.8	73.2	57.6	73.3	72.8
Adjusted Rate(%)	100	57.6	52.4	60.1	95.9	75.5	96.1	95.5

Table 5: the success rate of all algorithms in modified network

Furthermore, we can see that performances of HDS, CSS, and BCS are also affected. Figure 9 shows the changes of average path length of successful queries in the modified network. Compared to little change in BFS strategy, the changes in HDS, CSS, and BCS are noticeable.

[6] We also tried other thresholds for "cuts". A threshold of 5 was selected because it changed the network enough but still kept the network roughly connected. It is also close to the cut point that Adamic et al. used in their simulation.

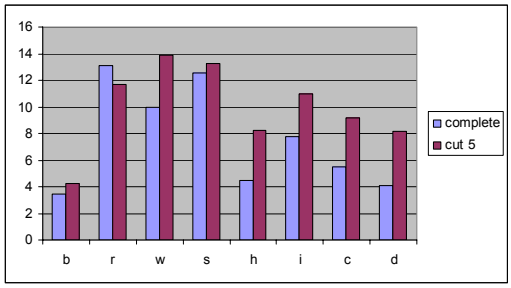


Figure 9: Comparison of average path length of successful queries

Above all, results in this modified network suggest that weak ties are really important information channels. They should not be simply ignored in designing social network based systems or in doing email based social network analysis.

4.4.2 Removing Users with High Out-Degree

For the second sensitivity analysis, we modified the original network by removing the 10 users who had the highest out-degrees. Most of them are also the most frequently used users in the original simulation. In this modified network, the average shortest path length became 2.754 and density became 0.076.

Table 6 shows the success rates in this simulation. Surprisingly, we find that the performances of BCS, HDS, and CSS are not affected at all. Actually, their relative performances got better with regard to the adjusted success rates.

Algorithm	BFS (b)	RWS (r)	WTS (w)	STS (s)	BCS (h)	ISS (i)	CSS (c)	HDS (d)
Succeess rated (%)	81.9	76.4	79.0	73.1	81.7	81.5	81.8	81.5
Adjusted Rate(%)	100	93.2	96.4	89.3	99.7	99.5	99.9	99.4

Table 6: the success rate of all algorithms in modified network

However, in Figure 10, which shows the changes in average path length of successful queries with this modified network, we can see that BFS is the only one that is not clearly affected. BCS, HDS, and CCS algorithms are affected much more. This suggests their sensitivity to those highly connected nodes. As well, the performance rank of these algorithms did not change.

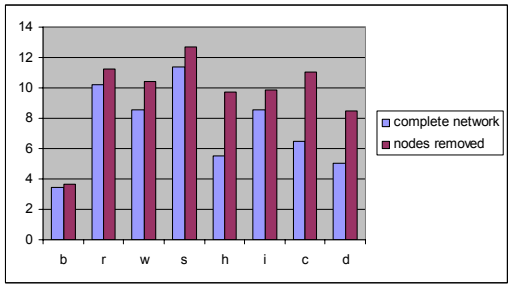


Figure 10: Comparison of average path length of successful queries

Interestingly, although designed from different perspectives, these algorithms still are affected by the change of network characteristics. Note these changes result from losing some specific ties or people who are particularly useful for the strategies used. A good example is the ISS strategy: While its design does not consider the effect of weak ties (as in Yu and Singh), the removal of weak ties changes its performance considerably.

5. SUMMARY

Searching within social networks has gained more theoretical support over the last decade with a better understanding of network dynamics and structure. However, compared to approaches automatizing the small world problem, we know relatively little about searching for expertise in social networks

Searching for expertise is not only affected by the graph characteristics of the network, such as the degree distribution, but also social characteristics of the network, such as people's social interactions and expertise. A human social network is not simply a graph structure, it also includes different social characteristics.

We used a simulation on an organization's email data set, compared three families of searching strategies that utilize both graph and social characteristics of the derived social network, and then explored the algorithms' tradeoffs and social characteristics. Our results indicate these characteristics can affect the searching process in important ways:

- The relative rank of different algorithms changes little when examining social costs.
- The Information Scent Strategy's advantage (IIS), surprisingly, is not obviously better than out-degree based strategies (BCS and HDS). IIS's performance is close to the Weak Tie Strategy (WTS). Furthermore, we actually found that it also tends to use high out-degree nodes more frequently than low out-degree nodes
- As Granovetter suggested, when compared to the Strong Tie Strategy (STS), the Weak Tie Strategy (WTS) is better. Furthermore, when the weak ties are removed, we also found that performance of IIS also decreased considerably. This indicates weak ties are likely to be critical for automated or augmented expertise finding.
- Our findings confirmed that out-degree based strategies, such as BCS and HDS, in networks like Enron's social network, have a clear advantage over other strategies. However, a very few nodes turn out to be very key in affecting the performance of such social network searching. .
- Simulation, in combination with carefully considered data and analysis, can be very useful in exploring the complex relations among different strategies, social costs, and social characteristics of networks.

As a first-step study, our findings can provide insights for designing future social network based information searching systems. They also open up some interesting avenues for further research. We plan to further look at how the information scent strategy (ISS) really works and its correlation with degree distributions. Then, based on that work, we will try exploring some mixed, dynamic, and learning strategies. We are planning to extend our simulation to examine people's availability and related

issues by using data from people's email exchange patterns. If possible, we are also planning to run the simulations on other data sets.

6. ACKNOWLEDGMENTS

This work was supported in part by the University of Michigan CARAT Fellowship Program and a grant from National Science Foundation (IIS-0325347). The authors would like to thank Judy Olson, George Furnas, Marshall Van Alstyne, and the Group05 reviewers for their various thoughts, discussions, and help.

7. REFERENCES

[1] Ackerman, M. S., Pipek, V., Wulf, V. *Sharing Expertise: Beyond Knowledge Management*, MIT Press, Cambridge MA, 2003.

[2] Ackerman, M.S., Boster, J., Lutters, W., McDonald, D. Who's there? The knowledge mapping approximation project, in Ackerman, M. S., Pipek, V., Wulf, V. *Sharing Expertise: Beyond Knowledge Management*, MIT Press, Cambridge MA, 2002.

[3] Adamic, L.A., and Adar. E. How to search a social network. *Social Networks*, 27(3), 2005, 187-203.

[4] Adamic, L.A., Lukose, R.M., Puniyani, A.R., and Huberman, B.A. Search in power-law networks. *Physics Review E*, 64(46135), 2001.

[5] Axelrod, R. Advancing the Art of Simulation in the Social Science, *Simulating Social Phenomena*, 1997.

[6] Bernard, H. R., Killworth, P. D., McCarty, C. Index: An informant-defined experiment in social structure. *Social Forces,* 61 (1), 1982, 99-133.

[7] Burt, R.S. The network structure of social capital. *Research in Organizational Behavior*. JAI Press, 2000, forthcoming.

[8] Cohen, W. Enron Email Dataset, *http://www-2.cs.cmu.edu/~enron/*

[9] Dodds, P. S., Muhamad, R., Watts, D. J. An Experimental Study of Search in Global Social Networks. *Science,* 301, 2003, 827-829 .

[10] Nardi, BA., Whittaker, S., and Schwarz, H. It's not what you know, it's who you know: work in the information age. *First Monday*, 5, 2000.

[11] Foner, L. Yenta: A multi-agent, referral-based matchmaking system. *In Proceedings of the 1st International Conference on Autonomous Agents*, 1997, 301-307.

[12] Granovetter, S. The strength of weak ties. *American Journal of Sociology,* 78, 1973, 1360-80.

[13] Hamming, R.W. Error-detecting and error-correcting codes, *Bell System Technical Journal,* 29(2), 1950, 147-160.

[14] Hutchins, E. *Cognition in the Wild*, MIT Press, 1995.

[15] Kautz, H., Selman, B., and Shah, M. The hidden Web. *AI Magazine*, 18(2), 1997, 27-36.

[16] Killworth, P., and Bernard, H. Reverse small world experiment. *Social Networks,* 1, 1978, 159-192.

[17] Kleinberg, J. Navigation in a small world. *Nature,* 406, 2000, 845.

[18] McDonald, D. W. and Ackerman, M.S. Expertise Recommender: A Flexible Recommendation Architecture. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW '00)*, 2000, 231-240.

[19] Milgram, S. The small-world problem. *Psychology Today,* 1, 1967, 62-67.

[20] Newman, M.E.J. A measure of betweenness centrality based on random walks, Arxiv preprint cond-mat/0309045, 2003.

[21] Russell, S., and Norvig, *P. Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.

[22] Whittaker, S., Jones, Q., Terveen, L, Contact Management: Identifying Contacts to Support Long-Term Communication. *Proceedings of the ACM Conference on Computer Supported Cooperative Work.*, 2002, 216-225.

[23] Streeter, L.A. and Lochbaum, K.E., Who Knows: A System Based on Automatic Representation of Semantic Structure. RIAO, 1988, 380-388.

[24] Travers, J., Milgram, S., 1969. An experimental study of the small world problem. *Sociometry,* 32, 425-443.

[25] Wasserman, S., Faust, K., Iacobucci, D, and Granovetter, M. *Social Network Analysis: Methods and Applications*, Cambridge University, 1994, 130-142.

[26] Watts, D. J., Dodds, P. S., Newman, M. E. J. Identity and search in social networks. *Science,* 296, 2002, 1302-1305.

[27] Wegner, B., Erber, R., and Rayomond, P. Transactive Memory in Close Relationships, *Journal of Personality and Social Psychology*, 61 (6), 1991, 923-929.

[28] Yang, S. B., and Garcia-Molina, H. Improving search in peer-to-peer networks. *In Proceedings of 22nd International Conference on Distributed Computing Systems*, 2002, 5-14.

[29] Yates, R.A, Ribeiro, B. *Modern Information Retrieval*. ACM Press/Addison-Wesley, 1999.

[30] Yu, B., and Singh, M.P. Searching Social Networks, *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003, 65-72.

[31] Yu, B., Venkatraman, M., and Singh, M.P. An Adaptive Social Network for Information Access: Theoretical and Experimental Results, *Journal of the Applied Artificial Intelligence*, 17 (1), 2003, 21-38.