Search in Social Networks: Designing Models and Algorithms that Maximize Human Influence

Ericsson Santana Marin
Institute of Computer Science
Federal University of Goiás
Campus II, Goiânia, Goiás 74001970
Email: ericssonmarin@yahoo.com.br

Cedric Luiz de Carvalho
Institute of Computer Science
Federal University of Goiás
Campus II, Goiânia, Goiás 74001970
Email: cedric@inf.ufg.br

***Abstract*—The ease with which people can use today's technology to form connections has generated an unprecedented situation for society: an era of global connectivity. This ease of connecting has increased the number of people using social networks, making the search for connections in this kind of network extremely complex. In this paper, grounded in concepts from Network Science and Artificial Intelligence, we report on models we have constructed and on algorithms aimed at producing a search engine integrated into social networks environments. The contribution of this engine is its ability to evaluate the numerous paths that connect source and target people, opting for the path where the interpersonal influence through the path is maximized. This increases the probability of finding reliable connections. In order to simulate its operation, we implemented this search engine in a multiagent application whose test performance produced results that exceeded expectations.**

I. INTRODUCTION

If the current era of human history was characterized in a simple way, its description might be the most highly connected period thus far. Interpersonal connections have become so explicit that today almost all people are familiar with the online social networks, like Facebook [1], which recently surpassed the mark of one billion active users [2].

In this context, emerged a science responsible for understanding the consequences of all this interconnectivity, through an interdisciplinary approach focused on intensive study of complex networks. Grounded in sociological research and in Graph Theory, Network Science has produced significant works that have subverted previously defined concepts, presenting revealing features about the social universe.

Network Science adopts a vision of society as a dynamic network and understands that people's success is conditioned in the structure of the networks they form, and the ability of these networks to be navigated [3]. It is this perspective of navigation that creates the possibility of conducting search in social networks, the main topic to be discussed in this paper.

A. *Searches in Social Networks*

Essentially a social network is a set of people and their connections to each other. Social networks evolve organically from the tendency of people make their friendships, and these choices put individuals in specific locations in the networks.

According to the physicist Mark Buchanan [4], the concept of a networked society was evidenced in the 80's, when people realized that the best strategy to find information about abstruse matters was to explore the links of their social networks, searching for partners. "People extended antennas to their friends and acquaintances, and hoped that at some point in the chain, some preciousness appeared in their paths".

Finding a person, who could provide desired information has become even more important today when problems, which no clear solutions, have to be solved quickly. Defining which person should be looked for and what is the best path to that person can be extremely complex, especially if done online, because the number of connections grows much more quickly.

This faster growth occurs because the core of human connectivity is not simply dyads, or connections formed by direct contacts. Instead these dyads aggregate to form huge webs of social ties that go far beyond direct connections, resulting in exponential branching [5]. Thus, paths of length two in social networks are already extremely difficult to search manually and longer than this is generally impractically.

In addition, an important factor in evaluating paths in such networks is influence [6], [7]. This type of social prestige is present in most friendships and may be paramount in creating connections. Different levels of influence exist in different paths connecting two people and influence can be effected by various factors along path (e.g., length).

A major difficulty occurs in the search process when solving a global problem using only local information [3]. It is hard to know which is the path of greatest influence between a source and target, because at each node, a new decision must be made and there is no clear way of evaluating the options.

In order to mitigate this problem, we present the design of models and algorithms that enable the creation of an intelligent engine for performing searches in social networks. The idea is to create computational agents able to quickly traverse huge networks. These agents operate in parallel using various approaches, such as Dijkstra's Algorithm, Ant Colony Optimization, and Genetic Algorithms. A multiagent application called Fluzz was designed to simulate this search engine and, based on the Network Science's findings, the agents are able to classify all paths connecting the source person to the target, where the best path is the one in which the influence probability among individuals is maximized.

To facilitate the use of our system, we added mapping

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.203

and visual transformation resources to the application. These display the stored networks, enabling users to evaluate the quality of the paths the agents find. In simulations performed in Fluzz, the agents were able to locate the best paths to the targets in satisfactory time, using their different skills cooperatively. Our hope is that all this practical and theoretical apparatus can be integrated into future online social networks, with our new model helping people find new connections.

The remainder of this paper is organized as follows: section II shows how the Network Science is able to map social networks. Section III introduces the path-finding algorithms used for creating the computational agents of our search engine. Section IV presents the related work. Section V presents the search engine, detailing the models and algorithms developed. Section VI introduces the Fluzz application, responsible for simulating the search engine proposed. Finally, section VII considers the contributions of this research.

II. NETWORK SCIENCE: MAPPING SOCIAL NETWORKS

From works based on decades of theory and experiments in fields such as mathematics, sociology, and computer science, scientists are learning how to map all kinds of networks, starting to visualize patterns not previously perceived [4].

This knowledge is leading to some remarkable findings such as: why most of the wealth ends up in the few individuals hands [8]; how a population of 7 billion people may be separated by only six degrees [9]; how individuals using local knowledge can find partners on social networks through a chain of acquaintances [10]; why people may become obese, lonely, or happy because of individuals they do not know [5]; why one innovation quickly spreads through a social network while another similar one falls by the wayside [3].

Adopting a network vision as an integral part of an evolving and continuously self-reconstituting system, so called Network Science (also called Science of Networks or Network Theory) offers a deeper perspective on the fundamental importance of connections in the world [3], [11], [4]. Besides understanding the organizational patterns of complex systems, Network Science tries to discover how to influence these patterns.

The search process in social networks is one important aspect of this field. The main idea is to help people to explore social networks, expanding their ability to obtain important information, using the influence factor on their social circle. In this research, most of theoretical concepts are imported from Social Network Analysis (SNA) [12], whose purpose is analyze the structural relations between social actors.

There are many important works using SNA in search algorithms [13], [14]. The Web itself was conceived from a proposal of hypertext, a linked network of texts. The largest and most used Web search engine today, Google [15], uses linkage analysis between Webpages as a main factor in defining the order in which retrieved pages are presented [16].

In a different approach, online social networks provide information about the relationships between individuals, which can be used in search. The combination of search and social networks is revolutionary because it uniquely incorporates the way people think and interact. The most common reason to search socially, instead using a classic search engine, is that participants trust more in the responses provided by their friends, rather than the opinions of strangers [17].

For this reason Network Science research has become even more relevant, producing central works for several approaches to social search [18], [19]. In the following we will briefly discuss some of these works which are advancing SNA and are essential for the social model proposed in this paper.

A. Six Degrees of Separation

In 1967, the psychologist Stanley Milgram conducted an experiment to investigate the hypothesis which stated that the world, seen as a huge network of social relations, was "small" [20]. The small-world problem, as it became known, ensured that anyone could be reached through a network of friends in just a few steps. Milgram's goal was to estimate the "distance" between any two people in the United States, since at that time this measure was typically estimated in the hundreds.

The experiment involved sending letters to randomly selected people, asking them to try to contact a target person through their friendship network. Surprisingly, 43 of the 160 letters arrived at their destinations, allowing Milgram to calculate the average number of intermediate persons, on that occasion 5.5, rounded to the famous six degrees of separation. In 2003, the researchers Peter Dodds, Roby Muhamad, and Duncan Watts replicated Milgrams experiment on a global scale using e-mail as a mode of communication [21]. They recruited thousands of volunteers to send a message to the targets, and roughly six steps were needed on average to send the email to each target, confirming Milgram's estimate.

The six degrees of separation suggested two conditions: first, global society can be navigated by social links from one person to another; second, in a network with 7 billion people, any pair of nodes is on average six links away from any other [3]. These are the conditions that enable searching in social networks with high probability of finding relevant results.

B. Three Degrees of Influence

The fact that people are connected with six degrees of separation does not mean that these connections dominate all others. The researchers Nicholas Christakis and James Fowler showed that influence diffusion in social networks obeys the rule of three degrees of influence [5]. Everything a person does tends to reverberate through the network, exerting an impact that dissipates at every separation, until the probable limit of degree three, no longer having a noticeable effect.

According to the researchers, there are three possible reasons for this influence be limited. First there is deterioration in the reliability of data during transmission. Second, the influence may decrease due to an inevitable transformation in network, given the constant turnover in social ties through the paths. Third, evolutionary biology can predominate because in the past, there were no greater distances than degree three [5]. Thus, although six degrees of separation applies to the degree of connection, three degrees of influence applies to the degree of contagion, allowing to measure human influence.

C. Social Distances

Interested in understanding how people compute social distances, Watts, Dodds and Newman observed that they often do it violating the mathematical condition known as the triangular inequality [3]. According to this condition, the length of any side of a triangle is always smaller or equal to the sum of the lengths of the other two sides, as shown in Figure 1.

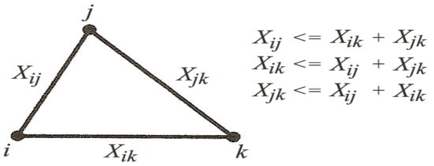

j
X_{ij}
X_{jk}
i
X_{ik}
k
$X_{ij} <= X_{ik} + X_{jk}$
$X_{ik} <= X_{ij} + X_{jk}$
$X_{jk} <= X_{ij} + X_{ik}$

Fig. 1. Triangular Inequality.

To understand this violation, it is necessary to verify how distances can be measured in society. The first way is the number of links on the shortest path between two points. In the case of the points *i* and *j* in Figure 1, this path would be X_{ij}. But this is not the definition that people typically use when considering how far away they are from someone. For the researchers, individuals tend to identify themselves in terms of contexts, making use of multiple dimensions [22].

In an exemplification given by Watts in Figure 2 [22], the same individuals of Figure 1, *i* and *k*, can perceive themselves as close to *j*, where *i* is close in a one dimension *d*, e.g., geography, and *k* close in another dimension, e.g., occupation.

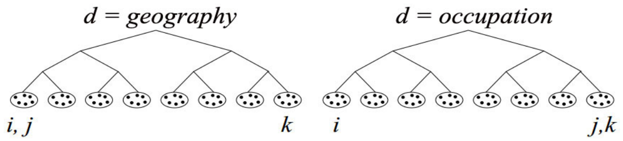


Fig. 2. Social Multiple Dimensions.

These multidimensional nature of social identity can explain the violation of the triangular inequality [22]. Figure 2 shows that the node *i* recognizes *k* as distant because there is no dimension in which both are present. Hence, although the triangular inequality dictates that the cost of the path (*i*,*k*) is less than that of the path (*i*,*j*,*k*), as shown in Figure 1, this is exactly the opposite of the way people calculate the distance. As the node *i* is closely linked to *j* and the node *j* is closely linked to *k*, the distance from *i* to *k*, is smaller if it is traveled by the path (*i*,*j*,*k*) because *i* and *k* do not share a dimension to perceive themselves as close. Consequently, the more contexts people share, the more chances they have to connect.

D. Weak Ties: Social Bridges

Inspired by the work of mathematician Anatol Rapoport [23], the sociologist Mark Granovetter investigated how the six degrees of separation were possible. For Granovetter, a parameter disregarded by scientists was a crucial piece to this puzzle: the strength of social ties [24]. Overall, he called strong ties those that exist between family members or good friends, while weak ties connected just acquaintances [24]. Granovetter's insight was to realize that, with the evolutionary dynamics of networks, strong ties tend to appear in triangles being therefore responsible for social clustering. Hence, incomplete triangles should be rare in this context, being found mainly in groups dominated by weak ties [24].

From this perspective, if a strong tie of a social network is removed, it will cause little effect in the number of degrees of separation. As this kind of tie almost always appear in triangles, it would still be possible to find a path from one point to another of the broken link, in just two steps, by moving along two remaining edges of the triangle. However, when weak ties are removed, the inevitable consequence is network fragmentation. For Granovetter, weak ties facilitate the existence of small, connected worlds.

In the 90's, grounded in Milgram's and Granovetter's findings, Duncan Watts and Steven Strogatz offered to scientists a formal way to understand how the six degrees of separation were possible proposing the small-world networks [9]. In this network model they presented how a few long-distance connections can make all the difference in a highly clustered network. Over the same period, Albert Barabási and Réka Albert showed how the historical mechanism in which the rich get richer, plays a similar role to long-distance connections, proposing a different network model called scale-free networks [8]. Both models were first attempts to explain the small distances between individuals as a result of a complex organizational behavior, disregarding the notion that social networks are formed by a purely random process.

E. Strong Ties: Social Capital

The consequences of clustering in society produces social capital, which is the ability of people to work together based on trust [25]. If weak ties keep networks connected, the social capital, created by the clustering of strong ties, provides a context of individuals firmly set. This enables the creation of social support produced by cooperative social chains [5].

As evidence of this phenomenon, researchers from the University of Chicago presented the results of a study that provided a description of the relational behavior of individuals. Table I shows who introduced couples in different relationship types [26]. About 61% of people met their partners through someone who they knew before, while only 39% met via self-presentation. Thus, although there was the possibility of strangers meeting, most people found partners by meeting individuals to whom they were indirectly connected.

Another study examined a network of word-of-mouth recommendations for three piano teachers [6]. Most recommendations occurred among close friends who were directly connected and positive references spread mostly to students within three degrees of the teachers. These studies demonstrate that people rely on friends and family for any kind of relationship, since socially mediated presentation is more informative.

TABLE I
INTRODUCTIONS IN SOCIETY

	Close Relation.		Distant Relationship			
Type of Relation.	Family	Friend	Work	School	Neighbor	Self.
Weddings	15%	35%	6%	6%	1%	32%
Cohabitation	12%	40%	4%	1%	1%	36%
Partnerships	8%	36%	6%	4%	1%	42%
Short-term partner.	3%	37%	3%	4%	2%	47%

III. PATH-FINDING ALGORITHMS

A path-finding algorithm searches a graph from a source vertex and explores adjacent nodes until the target be reached, usually looking for the shortest path [27]. For this last case, the problem is usually known as “Shortest Path Problem”, having become one of the most studied real problems in optimization, especially when the contexts are social networks [28], [17]. Several approaches for solving this type of problem have been developed recently and improved by the scientific community. In this section, we will briefly describe the three approaches used for developing the proposed search engine: Dijkstra’s Algorithm, Ant Colony Optimization and Genetic Algorithms.

A. Dijkstra’s Algorithm

The Dijkstra’s Algorithm [29], published by Edsger Dijkstra in 1959, addresses the issue of the shortest path in a directed or undirected graph whose edges contain no negative weights. This algorithm is used to determine the shortest path from one node to another node or to all the other nodes of the network.

Possessing a greedy strategy, Dijkstra’s Algorithm always makes the best decision in the moment. This strategy works because each subpath of a shortest path is also a shortest path. Hence, is possible to find the shortest path between two nodes, corresponding to the optimal solution of the problem, by determining all the intermediates shortest paths.

Dijkstra’s Algorithm uses a recursive formula in which in each iteration, the neighbors of a vertex are checked to determine what is the best option to expand the search. The node whose cumulative weight is the lowest among the candidates, and consequently all the neighbors of this node, is the next node analyzed. The search stops when the target node is reached, or when there are no more nodes to be analyzed.

B. Ant Colony Optimization

The foraging behavior of many ant colonies is based on a type of communication known as stigmergy [30], which is a indirect exchange of messages, performed by the ants through a volatile chemical substance called pheromone. While they walk from the nest to the food sources and vice versa, ants deposit pheromone on the surface, forming a chemical trail that will guide the others ants of the colony.

Over time, the initial stochastic fluctuations of choosing a path to the food source are reduced and a second mechanism plays an important role. As the ants that have chose the shortest path are faster than the others, this path receives a greater amount of pheromone. The result of this positive feedback is that after some time, the whole colony converges to the same route, guided by its high level of pheromone [31].

The importance of this finding is that high levels of pheromone are synonymous with short paths. So inspired by the foraging behavior of ants, in the 90’s, Marco Dorigo proposed the metaheuristic Ant Colony Optimization (ACO), which uses artificial ants to build solutions for optimization problems. These artificial beings exchange information about the quality of its solutions through a communication scheme similar to that used by real ants. Recently, different versions of algorithms have been proposed using ACO such as: Ant system (AS), Ant Colony System (ACS), Max-Min Ant System and ASrank [32]. These versions have obtained results for some optimization problems that are among the best heuristics currently available, including shortest path problems [32].

C. Genetic Algorithms

In the 60’s, John Holland investigated machines that learned more organically, exploring how simple rules could lead to complex behaviors. The idea was to create a software capable of unlimited learning [33]. The biggest insight of Holland’s was to use the power of natural selection. He took the evolutionary Darwinian logic and the genetics of Mendel and turned them into code, naming his creation of Genetic Algorithm [34].

Genetic Algorithms preform a stochastic, metaheuristic search that presents biological evolution as a technique for solving optimization problems. Individual generated by this algorithm, called chromosomes, represent a potential solution to a problem and evolve to find an individual that is, in fact, a solution. In the algorithm, these chromosomes compete with each other and the more adapted, at the expense of weaker candidates, are selected to be crossed in the next generation, mimicking Darwin’s theory of evolution. Consequently, each new generation produces more adapted chromosomes, implying a convergence to the problem solution. The population average fitness increases with each iteration and the process is repeated several times to enable the discovery of better results.

Because of their proven qualities, Genetic Algorithms have recently been integrated into techniques for solving the shortest path problem [35], mainly because they can adapt better to unexpected situations during search.

IV. RELATED WORK

In this section, a brief description of three applications related to this work is presented. The main feature examined is the ability of these applications to perform social searches and return shortest paths connecting source and target people. It is important to note that none of these applications uses information about mutual friends for modeling social networks, with potential violations of the triangular inequality. Furthermore, none of them uses a multiagent system with different pathfinding methods working cooperatively.

Thanks to *Kevin Bacon Game*, whose goal is to find the shortest path between any actor and the referenced movie star, in 1997, Glen Wasson and Brett Tjaden realized that distance calculation between actors was a viable project to be developed [11]. After obtaining access to the Internet Movie Database [36], Wasson and Tjaden installed the site *The Oracle of Kevin*

Bacon [37]. Providing the names of any two actors, the site presents the shortest path between them, listing the chain of movies and actors by which they are connected.

Referral Web aims to create an interactive tool to help people explore their social networks, so that they can quickly find short referral chains between themselves and experts on arbitrary topics [38]. The application uses the co-occurrence of names in close proximity in any documents publicly available on the WWW as evidence of a direct relationship, creating weighted edges that indicate the degree of association between the individuals. Consequently it allows users to search for chains in the networks to either named individuals or, more generally, to people who are likely to be experts on a given topic, graphing them in its visual interface.

The *SmallBlue* project within IBM aims to analyze the corporation's social network, for locating experts on given topics using queries keywords [28]. It checks the user's email outbox and outgoing IM chat transcripts for returning a relevance-ranked list of people, showing how to reach them based on the personal interactions among individuals. The search engine aggregates the results for all the keywords and ranks them according to relevance weighting and aggregated social-network structure. Then, it displays the minimal number of intermediate people to contact the target person, with an option to filter those who are one, two, or three degrees apart.

V. The Project

All works previously mentioned produced by the Network Science and Artificial Intelligence form the conceptual basis for our design of a new social search engine, to be integrated into any social network application. The main idea was to create computational agents that are grounded in social models, giving them the facility to evaluate and classify the various paths that connect a given pair of people in a network. Using these agents, an application can then return paths where interpersonal influence is maximized. Each path represents chains formed by a set of sequentially adjacent individuals by their "friendship" relation, as shown in Figure 3.

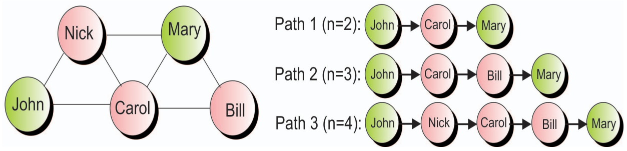

Nick
Mary
John
Carol
Bill
Path 1 (n=2): John → Carol → Mary
Path 2 (n=3): John → Carol → Bill → Mary
Path 3 (n=4): John → Nick → Carol → Bill → Mary

Fig. 3. Paths Representing Social Chains.

As social networks often violate the triangular inequality, the counts on the prestige of friends through a paths can become the differential to create new relationships [6], [7], [17]. In addition to the influence exerted by an individual on people that surround him, each intermediate link on a path, can significantly contribute to strengthen this influence. Therefore, we believe that returning a relevant social path can help users to strategically use the influences throughout this chain to create a social channel to the desired individual. To achieve this, we consider in the agent's model two features for maximizing the probability of interpersonal influence: minimizing the distance and maximize the strength of the ties.

First, by minimizing path length, we ensure the possibility of finding paths with the lowest degree of separation. According to the Christakis and Fowler [5], the farther apart two people are on the network, the less chance there is of influence. Further, the researchers found that, in society, the limit of human influence is degree three. For longer distances, such influence stops having noticeable impact, and the paths become unreliable. Hence, the distance minimization process is concentrated at the region bounded by paths of length three. These paths may represent the major difference between an attempt at direct contact and contact mediated by friendships.

The second feature of our model is the weight of connections representing the strength of the ties, which should be maximized. This measure is considered highly complex [6], as it depends on variables inherent to each relationship. Nevertheless, according to Granovetter [24], the strength of relational ties can be classified roughly into two categories: strong ties and weak ties. This fact combined with Watts' discovery [22] about the multiple social dimensions, enabled us to create a metric to estimate the strength of ties.

According to Watts, people share multiple social dimensions, and the more dimensions they share, the more chance there is that they will meet each other [22]. Using this reasoning, we concluded that, the more dimensions people share, the more intense is the referent friendship. The amount of shared contacts cannot directly determine the amount of shared dimensions, but it is a good indicator of tie intensity. The reason is that more shared contacts increases the chances of different dimensions being present. Therefore, we use shared contacts as an estimate of the weight of social ties. This implies that the weights are non-negatives.

A. Search Engine's Architecture

In this subsection we present the search engine's architecture, detailing all its components: *AgentMain*, *Search Agents*, and *Data Repository*, as shown in Figure 4.

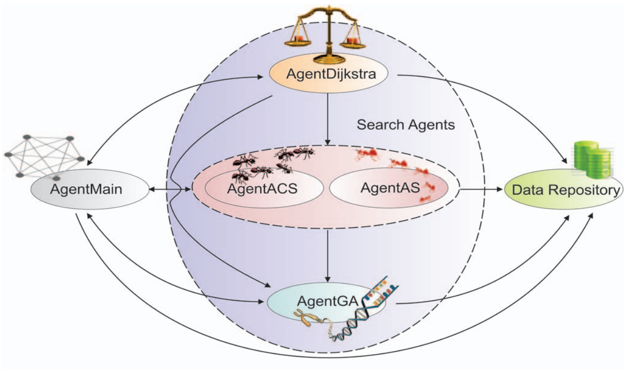


Fig. 4. Search Engine's Architecture.

1) AgentMain: This agent starts the search process by sending a message to the first search agent. Then, a series of communications is performed between the agents, which cooperatively search for a solution. As soon as one of them reaches the target a message is returned to *AgentMain* which graphically presents the path found.

2) Search Agents: There are four agents that work cooperatively. *AgentDijkstra* is the first to be activated. Its name is due to the algorithm it uses to navigate the networks, Dijkstra's Algorithm [29]. The next agents to be activated are *ACS* and *AS*, which initiate a search at the same time as *AgentDijkstra*. The strategies they use follow respectively the Ant Colony Optimization's metaheuristics *Ant Colony System* [32] and *Ant System* [32]. In addition, there is still the possibility of *AgentGA* being activated by any of these second level agents. If this occurs, the *AgentGA*, which implements a Genetic Algorithm [35], also participates in the search process.

AgentDijkstra is solely responsible for establishing the zone of influence, i.e., for returning paths up to degree three, because, such paths if present, are potentially optimal solutions to the problem. As the number of each person's connections in society is relatively small, even with an exponential growth of this number, the number of the paths of up to length three can still be analyzed quickly by this agent.

In order to define a qualitative analyze of the paths, we created the *Bestway* model to integrate the evaluation logic of the *AgentDijkstra*. The model produces a weighted graph using the following three factors (discussed in order of importance):

- **Minimize the weak ties in the human influence zone:** In our social model, ties are classified as strong or weak according to the number of their shared contacts, where the threshold between strong and weak is defined by the user. The first factor to be checked is the number of weak ties in the zone of human influence. Despite the importance of weak ties for providing rapid access to distant network points, in our perception, they correspond to unreliable and low-intensity relationships. Therefore, the goal in our search is to minimize weak ties in paths up to length three which decreases the chances of breaking a social chain. In fact, this might have been the main reason for most of the social chains of the Milgram's and Watts's experiments have not succeed. We believe that weak ties represent something near to absence of shared social dimensions, being responsible for the violation of the triangular inequality. So, the main idea is to attempt to reach the target person using up to three strong ties, because they represent the trusted people [6], [39].

  Figure 5 illustrates two situations where the number of weak ties dictates which is the best path. In the picture on the left side, three paths separate the vertices 1 and 5. If the user has defined that a weak tie corresponds to links with weights smaller than 5, the path (1,5) has one weak tie, the path (1,4,2,5) has two weak ties, and the path (1,3,5) does not have weak ties. As all these paths are in the zone of human influence, the agent determines that the path (1,3,5) is the best choice because it has the smallest number of weak ties. In the diagram on the right of the figure, two paths separate the vertices 1 and 3. The path (1,4,2,5,3) has no weak ties, while the path (1,4,3) has one. Nevertheless, the path (1,4,2,5,3) has more than three degrees of separation, so the path (1,4,3) is chosen.

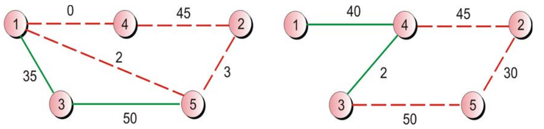


Fig. 5. First Factor of the *Bestway* Model.

- **Minimize the degree of separation:** When the above factor is not able to distinguish between two paths, a second criterion is used to minimize the distance across the network. Figure 6 illustrates two different situations of this analysis. The diagram on the left shows two paths that separate the vertices 1 and 5. Using the analysis of the number of weak ties (again assuming the weak tie threshold is 5), both paths (1,4,2,5) and (1,3,5) have one weak tie and are in the zone of human influence. This situation forces the agent to consider the degree of separation, which indicates that the path (1,3,5) should be chosen. In the diagram on the right, both paths separating vertices 1 and 6, (1,4,2,5,6) and (1,4,3,2,5,6) are outside the zone of influence and will be differentiated only by the degree of separation, making the path (1,4,2,5,6) preferred, even with one more weak tie.

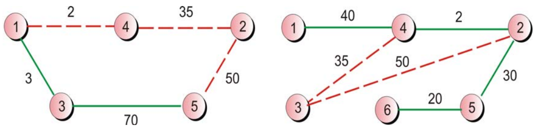


Fig. 6. Second Factor of the *Bestway* Model.

- **Maximize the path weights:** If there is still a situation where two different paths exhibit the same amount of weak ties and the same degree of separation, maximum path weight is used. The goal in this case is to select people with more intense friendships that will probably be more apt to cooperate with each other. This process is illustrated in Figure 7, where there are three best paths of degree 2 that connect the vertices 1 and 5. In this example, there are no weak ties along these paths. However, the weight of the path (1,3,5) is 70, the weight of the path (1,4,5) is 65, and the weight of the path (1,2,5) totals 75. Hence, the path (1,2,5) is preferred.

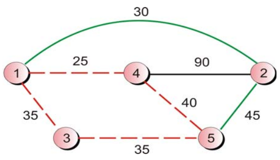


Fig. 7. Third Factor of the *Bestway* Model.

Figure 8 illustrates these factors being used in combination, resulting in the *Bestway* model.

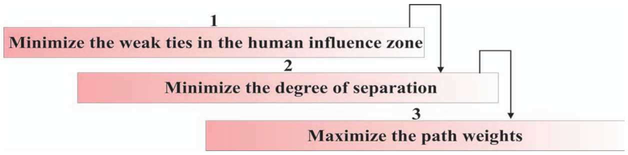

1
Minimize the weak ties in the human influence zone
2
Minimize the degree of separation
3
Maximize the path weights

Fig. 8. *BestWay* Model.

As predicted by the Network Science, after three degrees of separation, influence in a social network becomes negligible. Therefore, finding a person four degrees away has no practical significance, being a better choice try direct contact rather than to intermediate. Nevertheless, a long path provides a clue because any connection created by its intermediate individuals can push the distance between source and target into the zone of influence. This is a reason why long paths should not be disregarded. In online social networks, for instance, knowing who are the individuals of a friendship chain, enables the recognition of close people in real life, which are momentarily distant online. Consequently, strategic connections can be made by the source person, shortening this distance.

The problem that emerges when it is necessary to search beyond the limits of human influence is the branching factor. Depending on the network, the number of links from this region can represent a vast search space. Although *AgentDijkstra* is efficient, after this level, we have found that it is better to have the other agents cooperating in the search process.

According to our architecture, the *AgentACS* and *AgentAS* only contribute their information, when the *AgentDijkstra* crosses the boundary of three degrees. At this point, if *AgentACS* or *AgentAS* find a solution with four degrees they send a message to the *AgentMain* with their solution, which extinguishes all active agents and returns the path found.

Ant behavior provides an excellent metaheuristic when applied to optimization problems. Thanks to feedback from pheromone's deposits, artificial ants are able to navigate through large networks, and find short paths quickly. Beyond three degrees of separation, it becomes less important to locate optimal paths. So, the use of metaheuristic optimization works well in this context. Hence our model finds best paths of up to length three, and near best paths when the degree is greater.

Finally, the last search agent, the *AgentGA*, can be activated by either of these agents based on ant colonies. Its main goal is to produce a diversification of the paths previously found by the ants, which can converge to some local optimum. Consequently, if this agent, who also works beyond the three degrees of separation, finds a solution before the others, the process will end exactly how described above for the ants.

The *AgentGA* uses as metaheuristic the Genetic Algorithms with a fundamental definition for composing its initial population. To avoid problems of creating infeasible paths, all chromosomes from the *AgentGA* population are originally ants from the *AgentACS* or *AgentAS*, as illustrated in Figure 9.

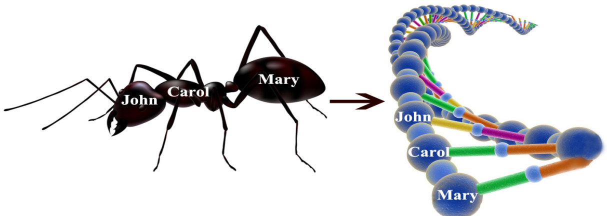


Fig. 9. Representing Ants and Chromosomes.

Hence, both ants and chromosomes are represented by integers which correspond to the vertices of a path, and as soon as the first two ants are formed, they are encoded as chromosomes subject to the process of genetic improvement.

Whenever *AgentDijkstra* enters in a new level of distance, a message is sent by it to all the other search agents, let them aware of the required distance at each time.

3) Data Repository: The *Data Repository* must be provided by the application that implements the search engine created, being responsible for storing and maintaining the integrity of all relational information of the networks created.

VI. The Fluzz Application

As a way to simulate the proposed search engine, we built the Fluzz application. The simulation results can be graphically analyzed in the Fluzz interface because visual mapping and transformation capabilities were incorporated to it. The development platform we used was JavaSE (Java Platform Standard Edition) version 7.0 [40]. The data generated by the simulations are stored in the database provided by PostgreSQL version 9.0 [41]. The framework used for implementing the cooperative agents was JADE (Java Agent Development Framework) version 7.0 [42]. Finally, the graphic composition of the networks is performed by the framework JUNG (Java Universal Network/Graph Framework) version 4.2 [43].

A. Simulating the Search Engine

To initiate the search procedure, the user must input to the Fluzz upper panel the source and target vertices and the threshold value corresponding to a weak/strong tie boundary value. Then, the button *Search Best Path* should be pressed. The *AgentMain* compute all the paths lengths and starts the search agents. As soon as *AgentMain* receives a message, it processes the response returning the solution.

By implementing the search engine in Fluzz, additionally we produce resources for creating multidirectional searches, allowing that multiple individuals could be targets at the same time. There are on the upper panel of the application three fields prepared to receive information relating to individuals, such as personal and professional characteristics. When at least one of these attributes is filled, the *AgentMain* identifies that the search should include as targets all the vertices that have the attributes informed. The best path found by the agents, among all individuals filtered, is then returned.

The *AgenteDijkstra* stores the evaluation of each traveled subpath in the following matrix: $[S, D, W]$, where S = amount of weak ties, D = degree of separation and W = path's weight. So, this agent uses a *binary heap* [29] to sort this matrix, and the best subpaths are those which have (in order of importance): the smallest S, the smallest D, the highest W. If the *AgenteDijkstra* does not find the target person within three degrees of separation, it restarts the search and continues until the end without using S, for analyzing potential paths previously disregarded within this zone. This procedure ensures that paths up to length three are always chosen first, and after that, the paths with the lowest degree of separation.

For the *AgentACS* and *AgentAS* we implemented variations of algorithms *Ant Colony System (ACS)* and *Ant System (AS)*. The artificial ants are created sequentially, being the pheromone globally upgraded in the edges, always after the creation of two new ants. Each of these agents represents a different ant colony, however only one pheromone trail was created making the colonies share their learnings.

Another important detail is how the agents manipulate pheromones. *AgentACS* performs its deposit in an elitist way by only allowing the best ants to make deposits, while *AgentAS* allows deposits from all ants. As to pheromone evaporation, only *AgentACS* performs this procedure, processing a small reduction of this substance on the edges traveled by its ants.

For the ant creation, we stipulated a limit value of 100 cycles, which means that no path can be created with more than 100 vertices. This generation process is implemented using Depth-First Search (DFS) [29]. The ants use a tabu list to check for vertices already visited and if one of them reaches a dead end, it *backtracks* [29] to the last valid configuration.

Unlike the original algorithm, in our implementation ants use only pheromone information for orientation, since the accumulated weight of connections after the third degree of separation is disregarded. Moreover, *AgentACS* does not do exploitation as originally proposed for *ACS* algorithm.

Finally, the *AgentACS* and *AgentAS* are responsible for starting *AgentGA* when the first two ants are created. They send to this agent all previously generated ants, which are encoded as chromosomes and stored in a repository divided into three sections: the general section, which comprises all the chromosomes created, the elitist section, which stores the best chromosomes, and the mutant section, which includes chromosomes genetically modified. We attribute to these three sections the same chance of being chosen by the selection process of the Genetic Algorithm. Thus, as the last two sections have relatively few chromosomes, there is a greater chance these chromosomes be recombined in an elitist process.

Once two chromosomes are selected they are recombined through the process *NBX (Node Based Crossover)* [35], which enables the crossing only if the two chromosomes have at least one pair of the same gene, excluding the source and target vertices. After performing the crossing, a repair function is used to remove potential loops in the paths. After the crossing, chromosomes can still be submitted to the mutation operation with probability of 1% to occur in each gene. The exchange of genetic material is carried out by consulting the list of adjacent vertices of the previous gene chosen. Hence, a partially valid chromosome is created, being subsequently completed with the genetic material of another compatible chromosome.

Figures 10(a),(b),(c) and (d) illustrate the search engine operation, using four different types of network (created randomly, small-world, scale-free, and a mix of these models) arranged with 100 nodes and 197 links. Figure 10(a) shows that *AgentGA* was responsible for the solution, because it has found a path between the vertices 95 and 74 before the other agents. Figure 10(b) indicates that *AgentAS* obtained the best performance, presenting the path between the vertices 54 and 36. Figure 10(c) shows that *AgentACS* was the victor, returning the path between the vertices 62 and 45.

All the solutions presented by these agents are possible because the *AgentDijkstra* had already focused its search at greater than three degrees of separation. Since *AgentACS*, *AgentAS* and *AgentGA* had already found a solution in the last level searched by the *AgentDijkstra*, they could present it first. This did not occur with the exemplification shown in Figure 10(d), where *AgentDijkstra* was the winner.

Fluzz has a resource on its left panel to identify the agent responsible for the solution. As shown in Figures 10(a),(b),(c) and (d), the application indicates who was the victorious agent, painting the respective marker with a red color. In addition, it is in this marker that the partial solutions of the agents are presented, being possible to check the degree of separation in which each one is located.

On the Fluzz right panel, there is a space dedicated to the description of the path found, displaying the vertices that form it, the degree of separation between the source and target, the cumulative weight, and the amount of weak ties present. Fluzz also provides a visualization of the solutions with the paths found being highlighted graphically in the application's rendering visual component by painting them purple.

We made twenty measurements to compare the agents performance during the search process. Table II presents the amount of times that *AgentACS*, *AgentAS* or *AgentGA* were responsible for determining the solution in different networks.

TABLE II
Wins of the *AgentACS*, *AgentAS* e *AgentGA* Faced *AgentDijsktra*

	Network Size and Density			
N. Nodes	N. *Links*	Wins	N. *Links*	Wins
1000	1997	7x	4985	5x
2000	3997	6x	9985	4x
5000	9997	3x	24985	1x
100000	199997	2x	499985	1x
200000	399997	1x	999985	1x
500000	999997	0x	2499985	0x

As shown in Table II, as the network becomes larger or denser, the metaheuristic agents have their performance hampered, with *AgentDijkstra* increasingly being victorious. This occurs because there are more paths in the network, which increases the chance of these agents wander for a while, until they start converging on a solution. Nevertheless, besides returning some solutions, the metaheuristic agents provide an important resource for the search engine: diversification.

Although it is possible to create a randomized version of

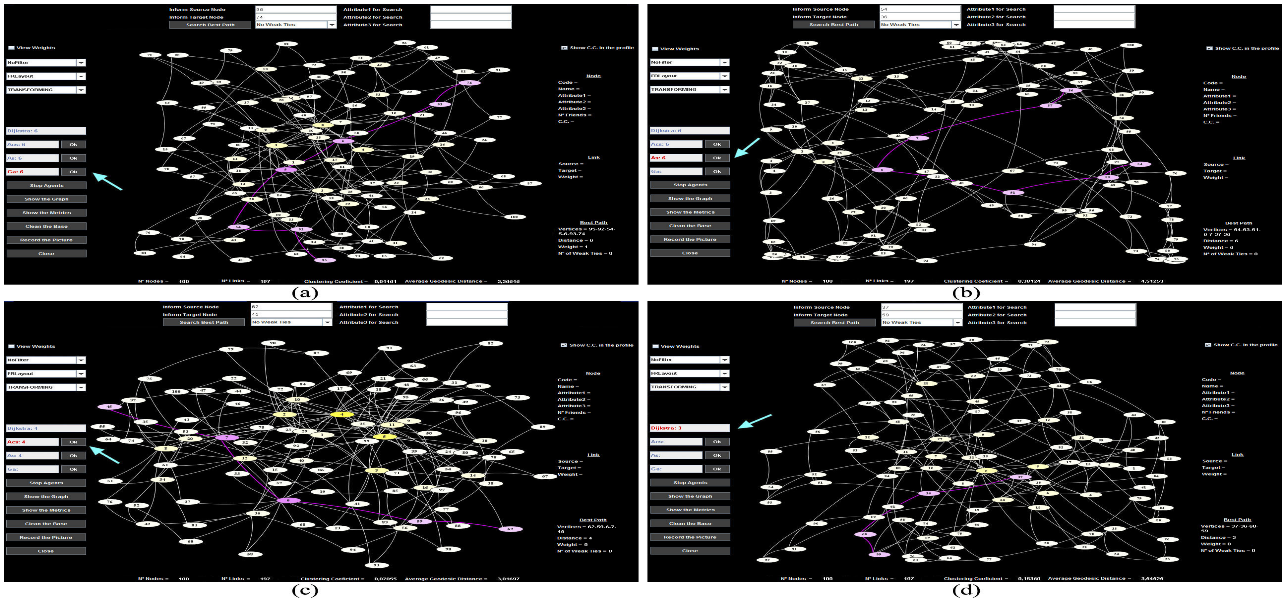


Fig. 10. Search Process.

Dijkstra's Algorithm that could make *AgentDijsktra* return different solutions for the the same search, using the stochastic nature of metaheuristic agents becomes more effective when the subject is diversification. It occurs because these agents can search different points of the network simultaneously, identifying completely different paths for the same search.

Thus, even if the *AgenteDijkstra* finds a solution before its partners, the Fluzz application maintains these other search agents working, in case the user decides to check the solutions provided by the metaheuristc agents. If otherwise, users can use the button *StopAgents* to stop the search process. We also created buttons next to each metaheuristic agent. When they are pressed, they interrupt the search process returning the solution found by the respective agent at that moment.

Through Fluzz, we performed several simulations of the search engine, using networks of 100 to 500.000 vertices with different densities. The *Search Agents* processed the information correctly, producing the expected results according to the models, and keeping a search average time of a few seconds.

B. Filtering Results

We implemented some filters to visualize the solutions. Figure 11(a) shows a search conducted with no filter applied, which occurred by selecting the option *NoFilter*. In Figure 11(b), the filter *SPFilter(Shortest Path Filter)* was selected, and only the components belonging to the path returned are presented. Another filter is the *SPFFilter (Shortest Path and Friends Filter)* used in Figure 11(c). In addition to the components belonging to the path found, all contacts of each node from this path were presented. Finally, we created the filter *SPFCFilter (Shortest Path and Friends in Common Filter)*. As seen in Figure 11(d), this filter adds to the subgraph existing shared contacts between any vertices of the path found.

VII. Conclusion

In this paper we proposed the creation of a search engine able to return a path between two people in social networks. The novelty of the mechanism is its ability to distinguish the various possibilities and choose the one where the interpersonal influence through the path is maximized. In our model the use of influence is a key factor in the connective process and was the main social feature analyzed in this work.

The main idea is to enable a new method that can extend the conventional means by which people form connections, bringing them closer to their potential partners in a more natural and efficient way. The models proposed are based on the recent sociological studies conducted by Network Science and were integrated into the search engine to enable the quantification of human influence.

Then we defined a search engine architecture using various techniques of Artificial Intelligence, and finally we created a multiagent application in order to simulate the engine. As a contribution, the models and algorithms produced in this work could be used freely, especially on the popular online social networks, further increasing the human interconnection.

References

[1] M. Zuckerberg and S. Sandberg, “Facebook,” http://www.facebook.com.

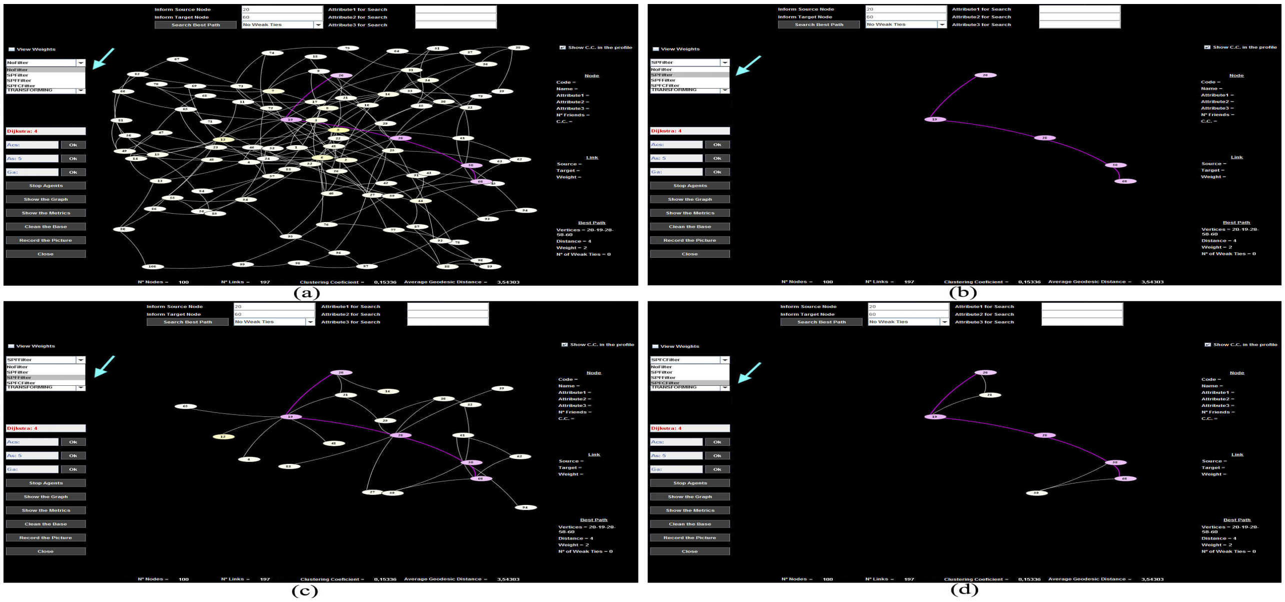
[2] “BBC - News Technology,” http://www.bbc.co.uk/news/technology-19816709.

[3] D. J. Watts, *Six Degrees: The Science of a Connected Age*. Norton, 2003.

[4] M. Buchanan, *Nexus: Small Worlds and the Groundbreaking Science of Networks*. W.W. Norton, 2003.

[5] N. A. Christakis and J. H. Fowler, *Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives*. Little, Brown and Co., 2009.

[6] J. J. Brown and P. H. Reingen, “Social Ties and Word-of-Mouth Referral Behavior,” *The Journal of Consumer Research*, vol. 14, no. 3, pp. 350–362, 1987.



(a)

Inform Source Node 20
Inform Target Node 60
Search Best Path
No Weak Ties
Attribute1 for Search
Attribute2 for Search
Attribute3 for Search
View Weights
Show C.C. in the profile
NoFilter
Dijkstra: 4
Acs:
As: 5
Ga:
Node
Code =
Name =
Attribute1 =
Attribute2 =
Attribute3 =
Nº Friends =
C.C. =
Link
Source =
Target =
Weight =
Best Path
Vertices = 20-19-28-58-60
Distance = 4
Weight = 2
Nº of Weak Ties = 0
Nº Nodes = 100
Nº Links = 197
Clustering Coeficient = 0,15336
Average Geodesic Distance = 3,54303

(b)

Inform Source Node 20
Inform Target Node 60
No Weak Ties
SPFilter
Dijkstra: 4
As: 5
Best Path
Vertices = 20-19-28-58-60
Distance = 4
Weight = 2
Nº of Weak Ties = 0
Nº Nodes = 100
Nº Links = 197
Clustering Coeficient = 0,15336
Average Geodesic Distance = 3,54303

(c)

Inform Source Node 20
Inform Target Node 60
No Weak Ties
SPFFilter
Dijkstra: 4
As: 5
Best Path
Vertices = 20-19-28-58-60
Distance = 4
Weight = 2
Nº of Weak Ties = 0
Nº Nodes = 100
Nº Links = 197
Clustering Coeficient = 0,15336
Average Geodesic Distance = 3,54303

(d)

Inform Source Node 20
Inform Target Node 60
No Weak Ties
SPFCFilter
Dijkstra: 4
As: 5
Best Path
Vertices = 20-19-28-58-60
Distance = 4
Weight = 2
Nº of Weak Ties = 0
Nº Nodes = 100
Nº Links = 197
Clustering Coeficient = 0,15336
Average Geodesic Distance = 3,54303

Fig. 11. Filtering Paths.

[7] J. H. Fowler and N. A. Christakis, "Cooperative Behavior Cascades in Human Social Networks," *Proceedings of the National Academy of Sciences*, 2010.

[8] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[9] D. J. Watts and S. H. Strogatz, "Collective dynamics of'small-world'networks." *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.

[10] J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective," in *in Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000, pp. 163–170.

[11] A. L. Barabási, *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume, 2003.

[12] S. B. Seidman and B. L. Foster, "Sonet-1: Social Network Analysis and Modeling System," *Social Networks*, vol. 2, no. 1, pp. 85–90, 1979.

[13] X. Shi, "Social Network Analysis of Web Search Engine Query Logs," School of Information, University of Michigan, Tech. Rep., 2007.

[14] D. E. Rose and D. Levinson, "Understanding User Goals in Web Search," in *Proceedings of the 13th international conference on World Wide Web*, ser. WWW '04. ACM, 2004, pp. 13–19.

[15] Google Inc., "Google," https://www.google.com.br/.

[16] S. Brin and L. Page, "The Anatomy of a Large-scale Hypertextual Web search Engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 107–117, Apr. 1998.

[17] B. Yu and M. P. Singh, "Searching Social Networks," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '03. ACM, 2003, pp. 65–72.

[18] A. Mislove, K. P. Gummadi, and P. Druschel, "Exploiting social networks for internet search," in *In Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets-V)*, 2006.

[19] S. H. Lee, P.-J. Kim, Y.-Y. Ahn, and H. Jeong, "Googling Social Interactions: Web Search Engine Based Social Network Construction," *PLoS ONE*, vol. 5, no. 7, pp. e11 233+, Jul. 2010.

[20] S. Milgram, *The Small World Problem*. Psychology Today, May 1967.

[21] P. S. Dodds, R. Muhamad, and D. J. Watts, "An Experimental Study of Search in Global Social Networks," *Science*, vol. 301, no. 5634, pp. 827–829, Aug. 2003.

[22] D. J. Watts, P. Dodds, and M. Newman, "Identity and Search in Social Networks," *Science*, vol. 296, pp. 1302–1305, 2002.

[23] A. Rapoport, *Mathematical Models of Social Interaction*, ser. Handbook of Mathematical Psychology. Wiley, 1963, vol. 2.

[24] M. S. Granovetter, "The Strength of Weak Ties," *American Journal of Sociology*, vol. 78, pp. 1360–1380, May 1973.

[25] F. Fukuyama, *Trust: The Social Virtues and The Creation of Prosperity*. Free Press, 1995.

[26] E. Laumann, *The Social Organization of Sexuality: Sexual Practices in the United States*, ser. Studies in Crime and Justice Series. University of Chicago Press, 1994.

[27] D. Brackeen, B. Barker, and L. Vanhelsuwé, *Developing Games in Java*, ser. NRG Series. New Riders, 2004.

[28] C.-Y. Lin, N. Cao, S. X. Liu, S. Papadimitriou, J. Sun, and X. Yan, "SmallBlue: Social Network Analysis for Expertise Search and Collective Intelligence," in *Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on*, 2009, pp. 1483–1486.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, September 2001.

[30] P. Grassé, *Les Insectes et Leur Univers*, ser. Les Conférences du Palais de la découverte. Université, 1949.

[31] S. Goss, S. Aron, J. Deneubourg, and J. Pasteels, "Self-organized Shortcuts in the Argentine Ant," *Naturwissenschaften*, vol. 76, no. 12, pp. 579–581, 1989.

[32] M. Dorigo, G. Di Caro, and L. Gambardella, "Ant Colony Optimization: A New Meta-Heuristic," in *Proceedings of the Congress on Evolutionary Computation*, vol. 2. IEEE Press, Jun-Sep 1999, pp. 1470–1477.

[33] S. Johnson, *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*, ser. Touchstone Book. Scribner, 2002.

[34] J. Holland, *Emergence: From Chaos to Order*, ser. Popular science / Oxford University Press. Oxford University Press, 2000.

[35] C. W. Ahn and R. S. Ramakrishna, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, December 2002.

[36] Amazon.com, "The Internet Movie Database," http://www.imdb.com.

[37] G. Wasson and B. Tjaden, "The Oracle of Kevin Bacon," http://oracleofbacon.org.

[38] H. Kautz, B. Selman, and M. Shah, "The Hidden Web," *AI MAGAZINE*, vol. 18, pp. 27–36, 1997.

[39] X. Shi, L. A. Adamic, and M. J. Strauss, "Networks of Strong Ties," *Physica A: Statistical Mechanics and its Applications*, vol. 378, no. 1, pp. 33–47, 2007.

[40] Oracle, "JavaSE - Java Platform, Standard Edition," http://www.oracle.com/us/technologies/java/overview/index.html.

[41] T. P. G. D. Group, "PostgreSQL," http://www.postgresql.org/.

[42] T. I. Lab, "JADE - Java Agent Development Framework," http://http://jade.tilab.com/.

[43] T. J. F. D. Team, "JUNG - Java Universal Network/Graph Framework," http://jung.sourceforge.net/.