Heuristics for Uninformed Search Algorithms in Unstructured P2P Networks Inspired by Self-Organizing Social Insect Models

Prithviraj Dasgupta and Erik Antonson

Computer Science Department, University of Nebraska, Omaha, USA.
E-mail: pdasgupta@mail.unomaha.edu

Abstract: We consider the problem of rapidly searching for resources or files in a distributed, unstructured, peer-to-peer file sharing network. Unstructured p2p network protocols such as Gnutella use a flooding-based mechanism for resource searching that generates considerable traffic in the network for each search query. When the searching activity by users in a p2p network is high, the traffic generated from the search requests could ensue congestion and result in increased search latency and poor performance in the entire network. To address this problem, we describe a resource search algorithm for p2p networks inspired by the stigmergetic behavior of ants while searching for food. Ants are used to encapsulate a search query initiated by a user in the p2p network. To search for the resource corresponding to their search query among the nodes of the network, each ant associates a certain amount of virtual pheromone with the nodes it visits. Later on, ants searching for resources use the amount and type of pheromone associated by previous ants with each node along their search path to direct the search query towards nodes that have a higher probability of resulting in the success for the search. We have tested our algorithm extensively within a simulated p2p network. Our simulation results show that our ant-based heuristics perform better than a completely uninformed or blind search that requires similar message overhead for each search query. When compared to a flooding-based mechanism, although the ant based search heuristic performs less efficiently under certain circumstances, it is capable of reducing the message overhead per search query by an exponential amount with respect to the flooding-based mechanism.

Keywords: Swarm intelligence, software agents, peer-to-peer networks, resource searching.

1. Introduction

Over the past few years, large scale distributed systems that can dynamically change their configuration over time and exhibit complex interactions between the system's components have emerged as an attractive paradigm for building robust, dynamic and adaptive systems. Such distributed systems have been used to design systems in diverse areas including autonomous multi-robot systems for unmanned search and rescue operations [1], peer-to-peer overlay networks for connecting millions of users for rapid interactions [2], data mining for bioinformatics applications [3], and pricing in online economies. Recently, several researchers have developed self emergent techniques inspired from disciplines such as biology to analyze the interactions between the components of such distributed systems and develop simple rules that control and dynamically adapt the system's behavior [4, 5]. In this paper, we consider the problem of rapidly searching for resources to reduce the latency facing users in a large scale, distributed, unstructured, peer-to-peer (p2p) network. A p2p network is an overlay network of nodes that allows users to share files and resources with one another. One of the major services employed by users in a p2p network is to search for and possibly download resources or files available with users on other nodes. With users and nodes in the order of millions in commercial p2p networks, rapidly searching for resources is a crucial problem that reduces the latency for all the users of the network as well as diminishes the traffic and congestion in the network.

Currently, in unstructured p2p networks, an uninformed search algorithm is used to locate resources being searched for by users across the different nodes of the p2p network. However, uninformed search is inefficient because it generates considerable traffic and congestion in the network through message flooding. In this paper, we describe a heuristics-based algorithm for searching for resources in an unstructured p2p network inspired by the stigmergetic behavior of social insects such as ants. In our algorithm, ants use virtual pheromone to direct a search query towards nodes that are likely to contain the resource being searched for in the search query. Each ant is implemented as a message that encapsulates the search query. An ant visits different nodes while searching for the resource encapsulated within its search query and associates a certain amount of virtual pheromone with the nodes visited by it. An ant searching for a resource later on uses the cumulative amount of pheromone left behind on nodes by previous ants to adapt its search behavior and direct its movement towards nodes that have a higher probability of containing the resource it is searching for. We employ different types of pheromone and different types of ants to improve the efficiency of the p2p search mechanism. Our simulation results show that our ant-based heuristics perform better than a completely uninformed or blind search that requires similar message overhead as our ant based algorithm for each search query. When compared to a flooding-based mechanism, the ant based heuristic evidently performs less efficiently but is able to achieve an exponential reduction in message overhead per search query as compared to the flooding mechanism.

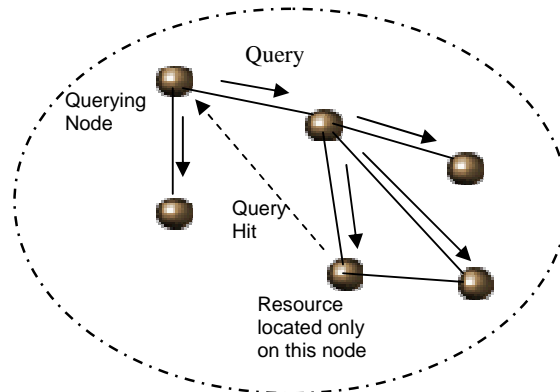## 2. P2P Resource Discovery Protocol

Figure 1 Messages exchanged in the p2p resource discovery protocol.

A p2p network consists of an interconnected collection of nodes. Nodes can join and leave the network dynamically. Each node contains resources that can be downloaded by other nodes in the network. Each node of a p2p network is usually associated with a human user that creates the node. One of the immensely popular applications of p2p networks, popularized by services such as Napster and Kazaa [6, 7, 8, 9], has been to enable file-sharing between the users of the p2p network. In a file sharing p2p network, the resources located on different nodes of the network are media files containing audio, video, image and even textual data. The major operation performed by a user in such a file sharing p2p network is to search for and download files available on other nodes in the network. Each node/user simultaneously allows other users/nodes to access the files present on the node itself. The searching for files (resources) in a p2p network is realized through the p2p resource discovery protocol shown in Figure 1. In the p2p resource discovery protocol, a user at a node wishing to search for a resource creates a search query containing certain unique keywords (e.g. filename or file tags) related to the resource it wishes to search for. The search query is encapsulated within a query message and provided with a search boundary that corresponds to the maximum number of hops the query message should be forwarded for, measured from its source. The *query* message is then forwarded to different nodes in the network in a breadth-first manner originating from the node that initiated the query. When a node in the p2p network receives the query message, a local search is performed among the resources present on that node. If the node contains a resource that returns a match with the keywords contained in the query message, the node sends a *queryHit* message to the node that originated the query, and does not forward the query message to other nodes. On the other hand, if none of the resources on the node that receives the query message returns a match for the keywords in the query message, the query message is forwarded to all the neighbors of that node, provided the search boundary of the query message

has not been reached. After a certain period of time since sending out the search query, the user at the node originating the search query observes all the queryHit messages it has received from other nodes in the p2p network and selects one or more of these nodes to download the resource from.

A potential source of inefficiency in the p2p resource discovery protocol is the enormous amount of messages generated in a p2p network for each search query. For example, if the average degree of a node in a p2p network is denoted by 'd' and the search boundary of a query message is denoted by 'b', each query message results in d^b query messages in the p2p network, in the worst case. With several million users simultaneously using a p2p network, the overall traffic generated from query messages in a p2p network can be overwhelming. This adversely affects the performance in the network by generating enormous amounts of network traffic ensuing congestion. In this paper, we posit that the traffic from query messages in a p2p network can be reduced if the breadth-first traversal used by the query messages in the resource discovery protocol is replaced by a local heuristics-based search that generates less than the exponential number of query messages per search query. Clearly, if we reduce the number of messages generated by a search query, the search query reaches fewer nodes and could potentially result in lower success for the search. Therefore, the heuristics must be carefully designed to compensate for the potential loss in reachability to nodes with an intelligent node selection strategy that forwards to query message to a few neighbor nodes that improve the probability of finding the resource on or along them. To design such a heuristic, we have used the stigmergetic behavior of social insects such as ants in locating objects of interest such as food.

2.1   Ant- Algorithm Based Resource Discovery in P2P Networks

Stigmergy is a process that enables insects to communicate information with each other either directly (e.g., by physical contact) or indirectly (e.g., by depositing chemical trails in the environment)[4]. The communication of information through stigmergy results in a swarm-like collective, emergent behavior between the insects to achieve complex tasks in a collaborative manner. For example, ants searching for food initially explore the environment around their nest. Each ant leaves behind a trail of a chemical substance called pheromone. Pheromone serves as an attractor for ants searching for the food later on. Pheromone also evaporates with time, to model the volatility in the environment and enable ants search for different locations for food as well as to enable the gradual removal of trails after the food at a location is exhausted. When an ant locates food at a particular location, it returns back to the nest while depositing pheromone on the ground. Consequently, the trail that leads from the nest to the food receives the maximum amount of pheromone and ants get attracted towards it. These ants further reinforce the pheromone trail between the nest and the food and enable ants to reach the food from the nest thereafter by following the pheromone trail.

To design the heuristics for the p2p resource discovery protocol based on the stigmergetic behavior of ants, we have modeled the forwarding of a query message as the movement of a virtual ant between the nodes visited by the query message. The stigmergetic behavior of these virtual ants is realized through virtual pheromone associated with the nodes that each ant visits. To record the pheromone information at each node, we maintain a pheromone table within each node. The pheromone table of a node contains the address or identifier of each neighbor of that node and a real value corresponding to the amount of pheromone associated with that neighbor node. Initially, the pheromone value of each neighbor node is initialized to zero. As ants search for resources, they update the values in the pheromone table of each node they visit while searching for resources. When an ant reaches a node, it determines the neighbor node to visit next based on the pheromone value inside the pheromone table of the node.

In a p2p network, nodes can leave and join the network in an ad-hoc manner. Also, a user at a node can dynamically add and remove resources that are stored within the node. This dynamic nature of a p2p network and the resources within its nodes implies that the pheromone values within the pheromone tables of the different nodes must be dynamically updated to enable ants visit newly joined nodes to search for resources within them, as well as to remove outdated trails that lead to removed resources or nodes that have already left the network. Consequently, the ant algorithm needs to balance the exploitation of existing trails to search for resources with exploration of nodes to discover newly added resources. To achieve this balance between exploitation and exploration in our algorithm, we have used two different types of pheromone in our model. The first type of pheromone, called *pheromone*, is used to mark routes that have resulted in successful searches and enables exploitation of existing trails. The second type of pheromone, called *anti-pheromone*, is used to mark nodes that have resulted in an unsuccessful search and enables exploration of new nodes or nodes that did not have resources that were searched for in the past. Based on these two types of pheromone, we have also designed two different types of forward-moving ants for our algorithm, which exhibit different responses to the different types of pheromone. The different ant types used in our algorithm are described below:

- **Forward Foraging Ants.** These ants deposit pheromone at nodes they visit and also get attracted to higher amounts of pheromone and repelled by higher amounts of anti-pheromone. From each node, a forward foraging ant prefers to go to neighbor nodes that have higher amounts of pheromone and lesser amounts of anti-pheromone.
- **Forward Explorer Ants.** These ants deposit anti-pheromone at nodes they visit and also get attracted to higher amounts of anti-pheromone and repelled by higher amounts of pheromone. From each node, a forward explorer ant prefers to go to neighbor nodes that have higher amounts of anti-pheromone and lower amounts of pheromone.

- **Backward Ants** Both types of forward ants become a backward ant when either they discover the resource on a node, or, they reach the search boundary without discovering the resource. Backward ants trace the route taken by their corresponding forward ant in the reverse direction. A backward ant deposits pheromone at each node it visits along its route, if the resource that the corresponding forward ant was looking for was found, and, deposits anti-pheromone at each node it visits along its route, if the resource was not found.

A forward ant contains the search algorithm that is executed by the ant on arriving at a node. The search algorithm performs a linear search for the terms in the user's search query encapsulated by the ant over the resources present on a node and returns a success only if there is a match.

3. Model

Our model of the p2p network comprises a connected network of N nodes. Each node contains certain resources inside a resource table and a resource can be identified on a node with a unique identifier. Nodes join and leave the network at random. Each node maintains a forwarding table containing the addresses of its neighbor nodes determined using the p2p node discovery protocol. Each address in the forwarding table is associated with a normalized weight that represents the probability of ant to migrate to that node. The weight of a node in the forwarding table gets updated when an ant selects it to move to it. Pheromone increases the weight while anti-pheromone decreases it. The use of a single weight attribute to reflect both types of pheromone keeps the ant algorithm simple and also reduces the size of the forwarding table.

A user at a node initiates a search by providing a set of keywords corresponding to the identifier of the resource(s) he or she wishes to locate. The node originating the query creates a forward ant with an empty stack. At each node, the ant selects a neighboring node to move to for the next hop, with a probability given by the weight of the node in the forwarding table. Before migrating to the selected node, the ant updates the weight of the node in the forwarding table according to the ant's type as described below.

Table 1 Parameters used for the ant-heuristics based p2p resource discovery algorithm

Symbol	Parameter
a_n	Number of nodes in the forwarding table of node n
$w_{i,n}^{t}$	Normalized weight associated with neighbor node i of node n at time t
τ_n	Amount of pheromone deposited on node n
τ_0	Amount of pheromone deposited by an ant at the source node of the search
χ_n	Amount of anti-pheromone deposited on node n
χ_0	Amount of anti-pheromone deposited by an ant at the node on which search terminated
$h_{s,n}$	Number of hops made by an ant to reach from the node s on which it started its journey to the current node node n

Forward Foraging Ant. A forward foraging ant starts from its origin with an empty internal stack. The algorithm used by a forward foraging ant at a node n to select a neighbor node i and update the weight associated with node i uses the following parameters shown in Table 1. The update rules for the pheromone at node n are the following:

$$\tau_n = \frac{\tau_0}{[h_{s,n}]^\alpha}$$
$$w_{i,n}^t = w_{i,n}^{t-1} + \tau_n(1 - w_{i,n}^{t-1}) \qquad (1)$$

and,

$$w_{i,n}^t = \frac{w_{i,n}^t}{\sum_{i=1}^{i=a_n} w_{i,n}^t} \qquad (2)$$

The factor α is determined experimentally and it controls the decrease in the amount of pheromone deposited as the ant moves further away from its origin. The second term on the r.h.s of Equation 1 ensures that the amount of pheromone deposited on a node is proportional to its current weight. This prevents excessive pheromone (or anti-pheromone) being deposited on a node whose weight is very high (or low). Equation 2 ensures that the weights of nodes in the forwarding table remain normalized after the weight of a node is updated by an ant. The ant pushes the address of the current node into its internal stack before moving to the selected node.

Forward Explorer Ant. A forward explorer ant works in a manner similar to a forward foraging ant except the following:
- It uses the inverse probability $(1 - w_{i,n}^t)$ to select a node i from the forwarding table of its current node n. This ensures that the probability of

selection of a node by an explorer ant is proportional to the amount of anti-pheromone deposited on it.

- It updates the anti-pheromone at each visited node according to the following equations:

$$\chi_n = \frac{\chi_0}{[h_{s,n}]^\alpha}$$

$$w_{i,n}^t = w_{i,n}^{t-1} - \chi_n(1 - w_{i,n}^{t-1}) \qquad (3)$$

where s is the origin node for the explorer ant.

Backward Ant. When the forward ant locates a resource or reaches its search boundary without locating the resource, it becomes a backward ant. The backward ant inherits the stack from its corresponding forward ant. If the resource was located by the forward ant, the backward ant rewards each node along the reverse route with pheromone using Equation 2. Otherwise, if the search boundary was reached without locating the resource, the backward ant deposits anti-pheromone on each node it visits using Equation 3 to indicate that the node did not lead to a successful resource discovery. For the backward ant, the node s represents the node on which the resource was found (in Equation 2) or the node on which the search boundary was reached without locating the resource (in Equation 3).

4. Simulation Results

We have implemented a Java application to simulate a dynamic p2p network and verify the performance of our ant-based p2p resource discovery algorithms. The Java application implements each node in the p2p network as a thread that is capable of communicating with each other threads (nodes) via message passing. To simulate the dynamic joining and leaving of nodes in the p2p network, we have used the 'churn' parameter that controls the rate at which nodes enter and leave the p2p network. Also, to simulate the availability of resources across the nodes of the p2p network, we have assumed that the probability of locating a resource at a particular node by an ant searching for the resource is a function of the number of resources available at that node. The number of resources available at a particular node is determined using a zipf distribution [10]. The default values of the different parameters used in our simulations are shown in Table 2. In the simulation experiments, we vary different parameters of our ant-based algorithm and in the p2p network and compare the effect of varying the parameters on the success ratio for the search, and, compare the performance of the ant-based algorithm with the two comparison strategies. For all our results, we have used the ratio between the number of successful search queries and the total number of search queries (called success ratio) as a measure of the performance of the search algorithm. All results are averaged over 10 simulation runs.

We have compared the performance of our ant-based algorithm for resource searching with two other search techniques. First, we have used a random or uninformed search mechanism where, at every hop, a search query is forwarded to one of the neighbors of the current node selected at random (denoted by legend R in the graphs of the simulation results). Since a search query is forwarded to only one node at every hop in this random mechanism and our ant-based mechanism, both these mechanisms generates at most d messages for each search query, where d is the search boundary for the search query. For our second comparison strategy, we have used the breadth first search (BFS) strategy currently used for resource searching in unstructured p2p network protocols such as Gnutella (denoted by legend BFS in the graphs of the simulation results). The BFS strategy forwards the search query to all neighbors of a node at every hop. Therefore, the message overhead of the BFS strategy for a single search query is b^d where b is the average node degree in the p2p network and d is search boundary for the search query.

Table 2 P2P Network Parameters for the Simulations

Name	Parameter	Value
N	Number of nodes in the p2p network	{200, 500, 1000}
D	Number of neighbors per node (average node degree)	4
$Churn$	Rate at which nodes join and leaving the network	50
ρ	Resource availability (Probability of a search ending successfully at a node)	0.02
$numberOfAnts$	Number of ants or search queries originated by each node during the lifetime of the simulation	numberOfFiles * ρ / 4 = 25
B	Search boundary for each ant	4
P	Probability of foraging	{0.0, 0.2, 0.4, 0.6, 0.8, 1.0}
χ	Amount of anti-pheromone deposited on the origin node/search boundary	0.6
τ	Amount of pheromone deposited on the origin node/resource node	0.6
α_f	Decrease in the amount of pheromone deposited as the ant moves away from the origin node	4.0
α_b	Decrease in the amount of pheromone deposited as the ant moves away from the search boundary/resource node	4.0

4.1.1.   Varying the Weight of Pheromone

For our first simulation, we observe the effect of varying the amount of pheromone deposited by forward foraging ants on the different nodes visited along their search path (τ) between 0.3 to 0.9. As shown in Figure 2(a), we observe that varying the pheromone weight has a limited effect of about 0.2% in the success

ratio of search queries. The largest amount of pheromone, $\tau = 0.9$, is the most effective overall and this can be attributed to the fact that larger amounts of pheromone require a longer time to decay and are able to sustain successful trails over a longer period of time , thereby causing subsequent ants to follow those successful trails and locating the resources they are searching for. A similar result was obtained on the success ratio by varying the amount of anti-pheromone deposited by explorer ants on nodes along their search path.
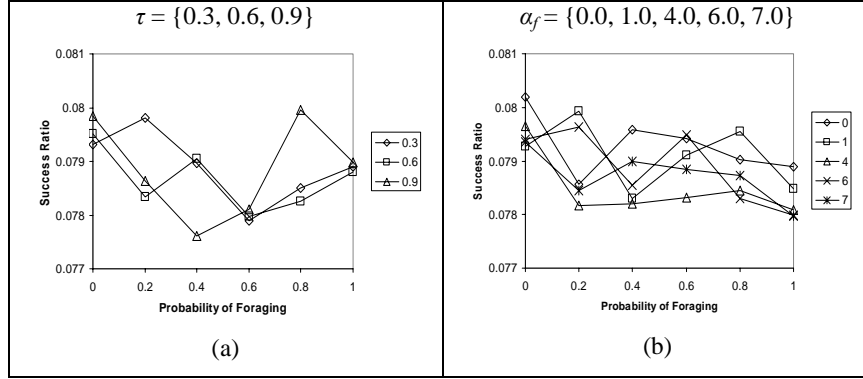

Figure 2 (a) Success Ratios for Varying Pheromone Weights. (b) Success Ratios for Varying Forward Ant Pheromone Decay Rates.

4.1.2.   Varying the Spatial Decay of Pheromone

For our next set of simulations, we varied the decrease in the amount of pheromone deposited by each forward ant as the ant moves away from the origin node (α_f) between 0.0 and 7.0. As shown in Figure 2(b), we observe that varying the forward ant pheromone decay rate has a limited effect of about 0.2% on the success ratio of the search queries. The smallest pheromone decay rate, $\alpha_f = 0.0$ (no decay), is the most effective overall. This behavior can be attributed to the fact that larger values of pheromone decay rate result in pheromone trails decaying rapidly at nodes further away from a successful node. Consequently, with a higher pheromone decay rate ants are not able to locate successful nodes and direct their search effectively when they are even a few hops away from a successful node. A similar result was obtained for the anti-pheromone decay rate (α_b).
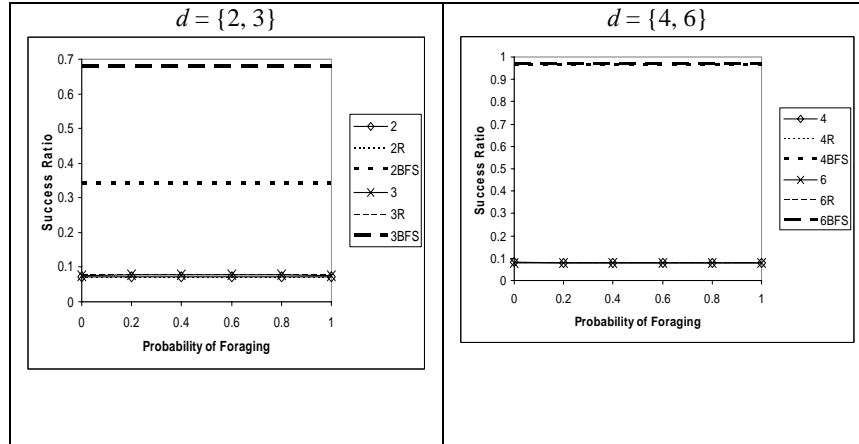
Figure 3 Success Ratios for Varying Neighbors with 200 nodes (left graph) and 1000 nodes (right graph)

4.1.3.   Varying the Number of Nodes and the Number of Neighbors

For the simulations in this group, we varied the number of neighbors per node (d) between two or three (left graph) for a network with 200 nodes, and, between four and six (right graph) for a p2p network with 1000 nodes. As shown in Figure 3, we observe that d = 3 in a network of 200 nodes results in a higher success ratio because the network is more connected. However, in a network of 1,000 nodes changing the degree of each node from 4 to 6 has very limited effect on the success ratio. The reason for this can be attributed to the fact that increasing the node degree in the network beyond an average node degree of 4 does not significantly affect the network diameter and consequently does not improve the success ratio for the search. In fact, for our ant-based algorithm, increasing the node degree beyond a certain value obfuscates efficient trails and reduce the success ratio.

4.1.4.   Varying the Resource Availability

For the next set of simulations, we vary the resource availability (ρ) on a node between 0.005 and 0.1. However, all nodes in the network have the same resource availability. As shown in Figure 4(a), we observe that varying the resource availability has a significant effect of about 36% on the success ratio for the search queries. The largest resource availability, $\rho = 0.1$, is evidently most effective because increasing the number of resources in the network reduces the search time required to locate a resource. In the next set of simulations, we verify the effect of heterogeneous values of the resource availability parameter. ρ is set to different values on different nodes of the network following the values of resource availabilities in actual p2p networks reported in [10]. Here, ρ varies between
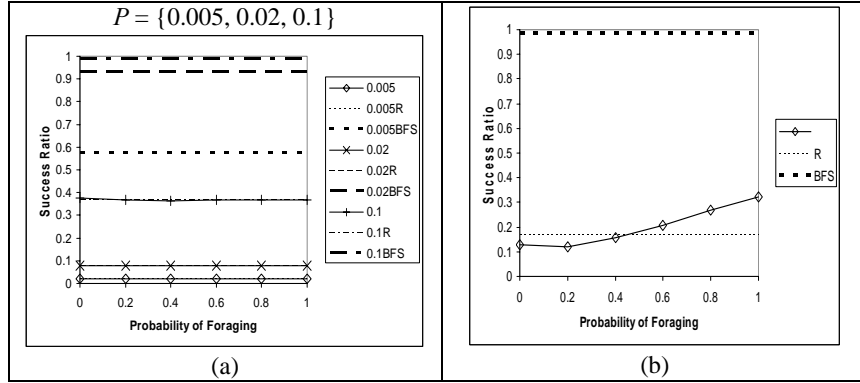
Figure 4 (a) Success Ratios for Varying Resource Availabilities (b) Success Ratios for Heterogeneous Resource Availability

0.0005 (on 30% of nodes), to 0.0055 (on 30% of nodes), to 0.055 (on 30% of nodes), and 0.2525 (on 10% of nodes). As shown in Figure 4(b), we observe that varying the probability of foraging for this p2p network configuration has a significant effect of about 21% in the success ratio of the search queries. The ant-based search behavior was more successful than the random search behavior for probabilities of foraging of 60% and greater. This would suggest that p2p networks having a high degree of heterogeneity will benefit the most from ant-based heuristics for resource discovery within the network. Moreover, it should be noted that a probability of foraging of 100% appears to be the most effective for this type of network.

For the next set of simulations, the resource availability (ρ) on different nodes is allowed to vary between 0.01 and 0.7. The p2p network for this simulation has 1,000 nodes with six neighbors per node. As shown in Figure 5, we observe that varying the resource availabilities for these p2p network configurations has a significant effect of about 93% in the success ratio for the search queries. The ant-based search behavior is more successful than the random search behavior for all the distributions. The ant-based search makes its greatest improvement (up to a 15% improvement over the random search) at aggregate resource availabilities of 4% and 12%. The breadth-first search was significantly more effective than the ant-based search for the heterogeneous resource distributions having fewer resources; but, as the aggregate resource availability for the network simulations increases, the success ratios for the ant-based search approaches the success ratios for the breadth-first search.
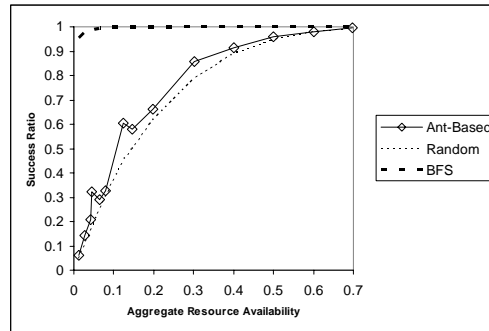
Figure 5 Success Ratios for Various Heterogeneous Resource Availabilities

5. Related Work

Resource discovery is implemented by flooding a resource query across nodes of the network in most commercial p2p systems [7]. However, query flooding produces considerable network traffic by blindly forwarding the query across the network. Improvements to query flooding include p2p stacks [6, 11] that use super-peer nodes, and, dynamic hash table(DHT) based techniques that strategically place resources on nodes using a hash function to improve resource availability and enable rapid lookup[12, 13]. In contrast to our ant-based algorithms, these techniques focus more on resource management and do not incorporate the information obtained from previous resource queries to improve future searches. Ant algorithms have already been applied to several applications [3, 14] including dynamic programming, traveling salesman problem and routing in telecommunication networks [5]. However, resource discovery in p2p networks is different from each of these applications because the node on which the resource will be discovered is not known *a priori* and the topology of the p2p network can change dynamically as nodes join and leave. Extensions to ant algorithms using anti-pheromone for the traveling salesman problem have been studied in [15]. The Anthill framework[2] employs ant-based algorithms for load balancing in a p2p network and ants backtrack along the path they traveled to update routing tables at each node. In contrast our algorithm uses different types of pheromone and ants with different behavior to make p2p resource discovery more efficient.

6.   Conclusions and Future Work

In this paper, we have described an informed search algorithm using an ant-based heuristic for p2p resource discovery. We are currently investigating extensions to the algorithm described in this paper using multiple ants to enable parallel search queries. We are also exploring techniques that enable ants to dynamically change their type and the pheromone deposited by them based on the performance of their search query. Finally, we plan to develop techniques that allow ants to exchange trail information with each other to locate resources rapidly. We envisage that biology inspired emergent algorithms provide a useful direction for further exploring challenges and issues of p2p networks for future research.

References

1.   H. Van Dyke Parunak, and S Brueckner : Swarming Coordination of Multiple UAVs for Collaborative Sensing. In: Proc. 2nd AIAA 'Unmanned Unlimited' Systems Conference, San Diego, CA, (2003).
2.   O. Babaoglu, H. Meling and A. Montresor: Anthill:A framework for the development of agent-based peer-to-peer systems. In: Proc. 22nd International Conference on Distributed Computing Systems(ICDCS), pp. 15-22, (2002).
3.   Abraham, C. Grosan, V. Ramos (eds.): Swarm Intelligence in Data Mining,"Studies in Computational Intelligence , vol. 34, Springer, (2006).
4.   E. Bonabeau. , M. Dorigo, G. Theraulaz: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999.
5.   G. Di Caro and M. Dorigo: AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificial Intelligence Research, vol. 9, pp. 317-365, (1998).
6.   Fast Track, URL http://www.fasttrack.com
7.   Gnutella, URL http://www.gnutella.com
8.   Kazaa, URL http://www.kazaa.com
9.   Napster Inc., URL http://www.napster.com
10. S. Saroiu, P. Gummadi, S. Gribble: Measuring and analyzing the characteristics of Napster and Gnutella hosts. Multimedia Systems, vol. 9 no.2, pp 170-184, (2003).
11. B. Yang and H. Garcia-Molina: Designing a super-peer network. Proc. 19th International Conference on Data Engineering (ICDE), pp. 49-62, (2003).
12. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan: Chord: A peer-to-peer lookup service for internet applications. In: Proc. ACM SIGCOMM Conference, pp. 149-160, (2001).
13. J. Kubiatowicz, et al.: OceanStore: An Architecture for Global-Scale Persistent Storage. In: Proc. ACM ASPLOS, pp. 190-201, (2000).
14. P. Dasgupta: Improving Peer-to-Peer Resource Discovery Using Mobile Agent Based Referrals. In: Proc. 2nd Workshop on Agent Enabled P2P Computing, pp. 41-54, (2003).
15. J. Montgomery, and M. Randall: Anti-pheromone as a tool for better exploration of search space. In: Lecture Notes in Computer Science, vol. 2463, Springer-Verlag, pp. 100-110, (2002).