Peer-to-Peer Netw. Appl. (2011) 4:211–218
DOI 10.1007/s12083-010-0082-2

Survey of search and optimization of P2P networks

Cai Kang

Received: 11 September 2009 / Accepted: 11 June 2010 / Published online: 23 June 2010

Abstract Being highly dynamic and casualness, P2P nodes can organize a special network by themselves, therefore the resource search in a P2P network is a complex problem. A reasonable and efficient resource search algorithm is not only the key to resource sharing for users, but also a critical stage to P2P technology development. Firstly, this paper introduces some basic resource search algorithms on P2P network architecture. Secondly, the paper discusses the advantages and disadvantages of these methods. Above all, the paper focuses on the combination of the P2P resource search with some intelligent optimization algorithms, which have succeeded in many fields. The combination shows a good prospect, although being still immature now.

Keywords P2P · Resource search · Optimization

1 Introduction

The communication mode of P2P makes many computers share information and computing resources directly, so it is not necessary to transmit by servers. For the traditional network method, many clients have to share limited information on a few servers and to share a limited communication bandwidth of the servers. P2P method eliminates the differences between servers and clients. Every person getting online has equal chances to offer his resources to others when sharing others resources. The method makes full use of the memories and computing capability of network computers to improve the efficiency and service quality of network application.

Currently, the technology of P2P has been extensively applied in file exchange, peer-to-peer computing, cooperative work, instant communication, search engine and so on. That many great effective products of P2P have appeared unfolds the great potentialities of business and technology.

That application and users of P2P rapidly increase has raised a new challenge to P2P technology and networks. A peer in a P2P network has double functions as a client and a server, each needing difference processing capability. Peers organize network by themselves, thus being random and dynamic. Some found resources may are invalid while visiting. It is difficult to control the action of peers because peers lack centralizing management. In addition, peers are somewhat anonymous so that they may face some security risks while offering resources. That is the reason why some peers may be selfish and unwilling to share their resource. In contrary, it is possible for a few peers to cheat others with some false or obsolete resources [1, 2]. Therefore, resource discovery in a P2P network is a complicated problem involving the load balance of peers, the dynamic change of resources, the delay of resource discovery, trust value and so on. A reasonable and efficient algorithm for resource discovery is the key to make users successfully share resources and the key of the development of P2P technology.

2 Architectures and basic searching algorithms of P2P network

The searching process of a P2P system is made up of getting the queries from users, locating the position of the resources, returning the results and finally connecting the

C. Kang (✉)
School of Electronics and Information,
South China University of Technology,
Guangzhou, Guangdong 510640, China
e-mail: caikang2009@gmail.com

source with the destination nodes to share the resources. Queries are usually keywords described in regular forms, but maybe defined in different parts in files, or in other forms, such as semantics-based information. According to the different modes of storing resource-index information, we classify existing P2P networks into three architectures: centralized architecture, distributed structured architecture, and distributed unstructured architecture. Different architectures need different search algorithms.

2.1 Centralized architecture and its basic algorithm

In centralized architecture, the search is simple by using special nodes for a central index system which guides available resources in a P2P system. Firstly, the nodes that want to search a resource are allowed to connect to the central index nodes after sending a query. If the central index nodes found the address (in a local index table) of the nodes owning the resource, the address would be returned to the requestor. Otherwise, a failure message would be returned.

When the requestor is connected to the destination node, the operation of sharing resources is to be successful. Take the example of Napster system [3], in this type of P2P system, a document index is stored in a centralized mode, but the storage and service of the document is in a distributed mode. For this method, search is high efficient and there will be fewer search and response messages transmitted in the network. However, the central server can only provides services for limited customers, so it lacks expansibility. Also it is vulnerable and there exists single-point failure problem.

2.2 Distributed structured architecture and its basic algorithm

For the distributed architecture, the index information of a part of resources is reserved in each node. The index tables, the index information of which tells the “direction” of the documents but not the real locations of the documents, are distributed to all network nodes. Distributed search adopts a pure distributed mechanism to transfer messages, and uses the location-based service model to search by keywords. At present, the main approach is the Distributed Hash Table (DHT) technology [4].

Such as Chord [5] and CAN [6] systems are the representatives of this type of model. Chord system, raised by MIT, provides a distributed resource-discovery service, which is suitable to the environment of P2P. This model locates and searches the information resources by maintaining the routing tables of neighbor nodes. The Distributed Hash Routing Table technology only maintains the routing table with the length *logN* to object nodes. However, there are still some problems, such as naming conflicts, poor expansibility and low efficiency of transferring message.

CAN (Content Addressable Networks) project is developed by Berkeley. It adopts *n*-dimensional geometric partition of topology to make resources content-addressable. The method of *n*-dimensional geometric partition makes system easy expansive. However, theoretically, it is more complicated.

In the DHT-based distributed query protocol, several keyword-servers deal with different keywords to reduce the volume of data transmission. DHT technology is suitable for accurate search, but is not available for multiple-keyword search at present.

2.3 Distributed unstructured architecture and its basic algorithm

The search in the distributed unstructured architecture is much difficult than that in other two architectures. Broadcasting search that is the basic algorithm for distributed unstructured architecture, is also called flooding search. In this type of system, there is not only no servers for the storage of the document index information, but also no request for the network topology and placement of the documents. Queries are broadcasted to neighbor nodes of the current node (or within a certain radius) to find source nodes. Gnutella system [7] is one of the typical representatives. In Gnutella network, there is no index information for nodes and there is no place to publicize the information of sharing documents. Just depending on local information, nodes have to select which parts of the network to connect to.

The advantage of the broadcasting search mechanism is simple and robust. It can control the dynamic joining and leaving of nodes to a certain degree of flexibility, and the failure of one node doesn’t influence the realization of the whole searching. But the consequent network flux increases the network load, which makes the load not easy to be measured. Therefore, the mechanism can not be applied to large-scale system. But in a small network, such as the internal network of a company, this mechanism is advisable because it can obtain searching results very fast [8].

3 Advanced searching algorithms

The distributed architecture is fitter for large scale P2P networks than the centralized architecture. However, the search in the distributed architecture is more difficult. At present, the main studies to improve search algorithms focus on the distributed architecture.

3.1 Hybrid P2P search

The hybrid P2P network results from the introduction of Super-Peers on the pure P2P distributed model [9]. It has many advantages, such as centralized quick searching, pure self-organized, load balance and robustness. The hybrid P2P search is considered as the search based on domains. In a system based on the domain model, all peers are divided into super-peers and general-peers. The system arranges all nodes into several domains according to some rules. There is a central control node, which is always a super-peer with high capability, in every domain. As a centre of information searching, the node is responsible for collecting the information of how many peer nodes there are in the domain. All super-peers form a pure P2P sub-network.

We must think about how to form a domain on the net when using the method of domain-based, how to control the amount of Super-Peers, and how to advertise queries and return answers. Literature [10] puts forward a method to divide domains by the interest of every node. The nodes with similar interest were considered as good peers. The good peers would be connected automatically by the system. A node knows the interest of its adjacent nodes when they query each other. The system connects the nodes with similar interest in a domain after calculating the interest of peers. By this information, a node can decide dynamically which nodes to connect with, and when to build or cut a connection. Therefore, the system can easily divide domains again according to the nodes' interest after the structure of the network topology changed.

3.2 Iterative deepening

In literatures [11, 12], we could find out the iterative deepening technique to improve the search in distributed unstructured P2P systems. In the iterative deepening technique, multiple breadth-first searches started to query from the source node. Until either the query is satisfied, or the maximum depth limit D has been reached, querying would not be ended. It is necessary for iterative deepening to set the last depth in the system so that the method has the same performance as the breadth-first search with the depth D. Additionally, a waiting period W that is the time between successive iterations in the policy must also be specified.

3.3 Modified BFS

This solution [13] improved the flooding scheme of the breadth-first search. It is the main difference from the flooding that a searching source just randomly selects a ratio of their neighbors to forward the query, rather than to send to all neighboring nodes. Compared with the flooding, it is an effective attempt to exchange time for memory space. This reduces the average message production, but still contacts a large number of peers [14].

3.4 Directed BFS

In the directed BFS [15], a user sends query to the subset of its neighbor nodes. Choosing which neighbor to forward query depends on historical records of queries. Each node sends a query to the nodes that probably respond to the query, not only to reduce the overhead of query but also to guarantee the quality of the result. For instance, the neighbor nodes which had a good response to query in the past should be chosen to send query. According to the policy, the nodes with good records may perform well in the future. The nodes that have received query message still employ the BFS to forward query to all their neighbor nodes. To select neighbor node intelligently, a node has a simple statistic about its neighbor nodes, for example, the quantity of query results from neighbor nodes, the link state of these neighbor nodes, etc. By analyzing these statistic data, it will gain inspiration and choose the node with best performance.

3.5 Intelligent BFS

To reduce the quantity of queries and peers queried, the intelligent BFS [13] will send query to the nodes which most probably return result. This method will find the target more effectively. When a peer receive a query, if the peer can provide a result, a result file will be sent to the query node, otherwise, the peer will transfer the query to the neighbor which possibly provides the result. In order to prevent message from spreading infinitely in P2P network, a maximum query depth should be set. When a query result is sent to the querying node, the peers on the return path will record the result and those peers providing the result. In order to select some neighbor peers to which to send query, peers need to classify their neighbors by the query results recorded. Each peer needs to maintenance a description which records the results received recently and the neighbor receiving the result. The key of this method is how to choose neighbor and how to classify the neighbors. To compute the level of neighbor, the node receiving a query compares the current query with the previous one to find the similarity between them, and then send the query to the most-probability node. The key of this search mechanism is the method choosing neighbor by similarity.

3.6 Local index

In the local index method [16, 17], Node N has the index data of the nodes that have r hops distance. Here the r is a variable the system sets, denoting the radius of index.

When r is 0, the method degenerates into a breadth-first search. That is to say, every node only has its own data. When a node receives a request, it just lookup the indexes stored in the nodes in r hops distance. In this way, getting search result deals with a few nodes. This method can ensure the quality and quantity of results. At the same time, it can confirm a lower cost. When r is small, the index is small, about 50KB. In the local index method, the system presets a search depth. Note that the different between the local index and the iterative deepening. In the iterative deepening, the depth means the end of an iterative. In the local index, the index of every node has to be maintained. When a node joins, leaves or updates its sharing data, additional measures have to be used to update index information. This method reduces the amount of messages and accelerates searching, although taking a certain cost.

3.7 Random walks search

In the random walks search [18, 19], a requester sends k queries to k neighbor nodes chosen randomly. Each query keeps direct connecting with the requester in the following process, asking whether executing next step. Random walkers will randomly choose next step if the requester allow to continue the random walking, otherwise the search will stop. This mechanism can control the range and width of a search. Because each node keeps connecting with the request node, however, queried nodes will have a great load.

3.8 Search based on mobile agents

The mobile agent is a program that can be transferred from one host to another automatically in an isomerous network and can interact with other agent or resources. Agents are suited to perform information search in the net. Now some researchers from Italy have some forefront studies about combining mobile agents with P2P [20]. One of the ideas is to embed mobile agents in a P2P program. When needing to query, a node sends a mobile agent to its neighbors, the agent records the information about the search. When the agent reached a new host, it will search, in the host, for the resource that the querying node wants. If the host has no the resource, the agent sends searching information to the nodes adjoining the host. Otherwise, the agent returns the address of the found resource to the querying node.

3.9 Routing indices

Routing Indices [21] allow nodes to forward queries to neighbors that are more likely to have answers. The object of the method is to forward queries to neighbors that are more likely to have answers. If a node cannot answer a query, it forwards the query to a subset of its neighbors rather than by selecting neighbors at random or by forwarding the query to all neighbors.

Routing Indices can improve performance by one to two orders of magnitude higher than that of a flooding-based system. However, routing indices require flooding in order to be created and updated, so the method is not suitable for highly dynamic networks [14].

3.10 Adaptive probabilistic search

As the random walks, the adaptive probabilistic search (APS) [22] uses k walkers. However, when retransmitting, the APS chooses the next node according to the neighbors' history rather than randomly choose neighbors. In this mechanism, a peer can efficiently guide walkers to searching according to its' previous search feedback, with just recording a part of the information of neighbors. With high accuracy, low bandwidth-occupied and good robustness in a dynamic environment, the APS search can find more targets. These features result from the learning mechanism of the APS. In the APS, each node maintains a part of the information about each neighbor. The information includes every query of the node and retransmitting indices. If a node has R neighbors, to N targets, it needs $R*N$ memories. Each index indicates the probability of receiving the answer of an object by the node's neighbors. The node retransmits the query for an object to a relevant neighbor according to the probability in the indices. The APS will update these indices periodically. The APS has two strategies to update these indices: optimistic strategy and pessimistic strategy.

3.11 BubbleStorm search

BubbleStorm [23], based on random multi-graphs, is a simple probabilistic search system. The main idea of the method is a flexible and reliable strategy for performing exhaustive search. The search guarantees are tunable.

BubbleStorm system separates query evaluation from network topology. It provides a network-level search strategy which does not compromise scalability, adaptability, or robustness. The search probabilistically guarantees that the application's query evaluator runs on a computer containing the sought data. The application developer is free to use any powerful query evaluator.

4 P2P searches base on intelligence optimization

In fact, P2P search can be considered as a multi-objective optimization problem about sources, paths and load balancing. So broadcasting search is an exhaustive method paying huge cost. The advanced algorithms in Section 3 adopt some heuristic information to reduce the quantity of queries to some extent. These algorithms usually provide a better (or local), but no best (or global), result for a single object about P2P.

In recent years, the computational intelligence has succeeded in many fields. There are some famous optimization algorithms, such as the genetic algorithms, simulated annealing, evolution algorithm, ant colony optimization, and particle swarm optimization. These intelligence algorithms can provide a global optimizing result for multi-object optimization, such as QOS routing. Therefore, the combination of P2P and intelligent optimization shows a good prospect, although the combination is not mature yet.

4.1 Resource search based on genetic algorithms

Genetic algorithms (GAs) that simulate the process of biological evolution have three important advantages: parallel search, robustness and global ability. A P2P resource-search problem could be described as an optimal path problem about data transmission among nodes. When multiple copies existing in the net, the nod, the path to which is the shortest, is the best among all source nodes.

In GAs, the probability by which an individual is possible selected as a parent is proportional to the fitness of the individual. Therefore, selecting a suitable fitness function is one of key factors for resource search. According to different goals, we can define different standards to evaluate paths. For example, low latency or/and few hops.

Considering the practical application of P2P, a common fitness function is shown as follows [24]:

$$f = \frac{1}{\sum_{j=1}^{l-1} C_{g(j),g(j+1)}} \quad (1)$$

where f represents the fitness value of a chromosome, l is the length of the chromosome (i.e., the number of the nodes in a path), $g(j)$ represents the gene (node) of the j-th locus in the chromosome, $C$and is the link cost between nodes.

A P2P resource searching algorithm based on the genetic algorithm (GA) includes following several steps [24]:

(1) Genetic Representation. A chromosome consists of sequences of positive integers that represent the IDs of nodes through which a routing path passes. The gene of first locus is always reserved for the source node. The length of the chromosome is variable.
(2) Create an initial population. A path from source to destination can be constructed by a depth-first search algorithm. The random initialization chooses paths (chromosomes) from all constructed paths. If the algorithm encounters a node for which all of whose neighboring nodes have already been visited, the defective chromosome should be repaired. Setting a flag to avoid the nodes which have already been visited is a good method.
(3) Selection.
(4) Crossover. For route selection, one-point crossover is more suitable than other crossover modes. But the mechanism of the crossover is not the same as that of the conventional one-point crossover. Two chromosomes chosen for crossover should have at least one common gene (node) except for source and destination nodes, but there is no requirement that the common genes are located at the same locus. The crossover does not dependent on the position of nodes in routing paths. The common gene is the crossover point, but the crossing sites of the two chromosomes may be different from each other. For example, two chromosomes *ABCDEF* and *AIDJLF* produce *ABCDJLF* and *AIDEF* by crossover.
(5) Mutation. A gene is randomly selected first, as "mutation point", from the chosen chromosome. One of the nodes, connected directly to the mutation point, is chosen randomly as the first node of the alternative partial-route. The next node is selected randomly from the nodes connected directly to the last node. The process continues till the destination nod is selected.

Note that fitness function and genetic operators are the key factors deciding the efficiency of the P2P resource research algorithm.

The resource search algorithm based on the genetic algorithm can converge to a global solution. For this reason, the algorithm is a nice choice for a practical P2P system. Using GAs to solve the P2P resource search problem can improve the efficiency of the whole net system. Additionally, GAs can be used to decrease bandwidth for queries [25]. However, the genetic algorithm could not make full use of the feedback information of a net yet. When approaching to the final stage, the GA often does a large of redundancy computation, thus increasing cost and time, and reducing the accurate of solution.

4.2 Resource search based on ant colony algorithm

The ant colony algorithm is a kind of bionic optimization algorithms. Ants release a kind of volatile secretion in search, called pheromone, on paths. They transfer information by apperceiving the intensity of pheromone, and tend to move to the direction with stronger intensity, thus forming a positive feedback mechanism, which makes ants find the shortest path, that is, the global solution.

When moving between nodes, ants release some pheromone, which will affect that direction of search at which ants go to the next node. In the P2P system based on the ant colony algorithm [26–29], to search resource, a nod sends a query (as an ant of the ant colony algorithm), which chooses a path with stronger pheromone by probability, releasing new pheromone after passing [30, 31].

Here is the basic mathematical model of ant colony algorithm. Suppose $G=(V, E)$ denotes the topological graph of a net. Where V is a set of vertexes, and E is a set of edges. Every edge (or link) has pheromone $\tau_{ij}(t)$. Ant k (k=1, 2, … m) determines its direction to move by the amount of pheromone of every edge. We use a tabu table k (k = 1, 2, … m) to record the nodes ant k have passed. In the search process, the ant calculates the probability of state transition according to the pheromone and heuristic information of every path.

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \bullet [\eta_{ij}]^{\beta}}{\sum\limits_{s \subset N_k} [\tau_{ij}(t)]^{\alpha} \bullet [\eta_{ij}]^{\beta}} & if\, j \in N_k \\ 0, & otherwise \end{cases} \tag{2}$$

In Formula (2), $P_{ij}^{k}(t)$ denotes the probability of state transition when ant k transfers from state i to state j at time t. N_k represents the nodes that ant k can select at the next step; α, β and η_{ij} are respectively the factor of heuristics information, the constant factor, and the reciprocal of the link cost from the current node i to the neighboring node j.

After each ant walked a step or finished a path, it releases new pheromone on every edge, which means updating the pheromone of every edge in the table. At time t, pheromone is updated according to the following rules.

(1) Part updating [32]

When Ant k passes edge (i, j), pheromone τ_{ij} (t) is updated by the following rules:

$$\tau_{ij}(t+1) = \rho_1 \bullet \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{3}$$

$$\Delta\tau_{ij}(t) = \frac{Q_1}{Cost(i,j)}$$

where ρ_1 is a coefficient (such that 1-ρ represents the evaporation of pheromone). $Cost(i, j)$ is the link cost of edge (i, j). Q_1 is a constant.

(2) Global updating

When ant k successfully finished a path P from source to destination, the pheromone between two neighboring node i and j in the path is updated by formula (4).

$$\tau_{ij}(t+n) = \rho_2 \bullet \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{4}$$

$$\Delta\tau_{ij}^{k}(t) = \frac{Q_2}{Cost_P}$$

Where $Cost_P$ is the total cost that Ant k finish path P.

In an actual P2P system, every ant as a mobile agent can correspond with the nodes visited by the ant, so that the pheromone data stored in the nodes can be updated. Physically, no link (net wire or fiber) can store pheromone data, but a node connected with a link can do this.

The ant colony algorithm, as a recent optimization algorithm, has solved a series of combination problems. Additionally, it is suitable to the multi-agent system. The positive feedback mechanism of the algorithm makes search quick and good. By accumulating and updating the pheromone, the successful rate at which the ant colony algorithm converges to the best path is usually higher than that of simulated annealing, genetic algorithms and other intelligent algorithms, especially in a dynamic net. However, for the P2P optimization, the lack of pheromone during the initial period usually makes the simple ant colony algorithm slow.

4.3 Resource search base on hybrid intelligent algorithm

Compare genetic algorithm with ant colony algorithm, both have different advantages and disadvantages, just as No-Free-Lunch theorem indicates. The ant colony algorithm has the powerful positive feedback capability which can increase the speed of evolution of algorithm to make algorithm convergence as soon as possible in the end, but the lack of the pheromone decreases the speed of evolution of the algorithm in the first. The genetic algorithm has better global capability, but it can't make use of the feedback of a system.

Obviously, combining the advantages of two intelligent algorithms is better than any one of both for the search optimization in P2P [33, 34]. A feasible combination way is to generate the initial distribution of pheromone by a genetic algorithm and to find exact solutions by the ant colony algorithm. For the P2P resource discovery problem, the new method in literature [33], resulting from the combination of the two algorithms, showed better performance in simulated experiments than its two parents did.

The general framework of hybrid algorithm has four parts:

1). The part of preprocessing: analyze constraint conditions and generate P2P network topologic structure.

2). The part of genetic algorithm: Generate several groups of optimizing solutions by a genetic algorithm, and then transfer it into the initial value of pheromone and make prepare for the next part, that is, ant colony algorithm which will get the best resource paths.
3). The part of combination: judge the proper time of combining the genetic algorithm with the ant colony algorithm.
4). The part of the ant colony algorithm: Use the positive feedback and effective convergence features of the ant colony algorithm to find the best resource paths meeting the conditions.

In P2P network structure, the hybrid algorithm rarely traps into a local minimum when routing, searching and solving other communication problems of peer-to-peer, and converges fast.

5 Summary

Resource search is one of the most important problems of a P2P system. A good resource search algorithm can improve not only search efficiency, but also the using rate of bandwidth. The paper introduced some basic resource search algorithms and some improved methods. The present improved methods, mainly focusing on a little mend to the basic algorithms, exist many problems, though they heighten the efficiency to some extent. The combination of the P2P resource search with some intelligent optimization algorithms shows a good prospect, although being still immature now.

References

1. Wallach DS (2002) A survey of peer-to-peer security issues. In: Proceedings of the international symposium on software security, Tokyo, pp 42-57
2. Feldman M, Papadimitriou C, Chuang J, et al (2004) Free riding and whitewashing in peer-to-peer systems. In: 3rd annual workshop on economics and information security, pp 228-236
3. Napster website. http://www.napster.com
4. Ghamri-Doudane S, Agoulmine N (2007) Enhanced DHT-based P2P architecture for effective resource discovery and management. Journal of network and systems management 15(3)
5. Stoica I, Morris R, Karger D, et al (2001) Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the conference on applications, technologies, architectures, and protocols for computer communications, New York, pp 149-160
6. Ratnasamy S, et al (2001) A scalable content-addressable network. In: Proc. of ACM SIGCOMM'01, San Diego California, pp 161-172
7. Gnutella website. http://www.gnutella.com
8. Krishna Ramanathan M, Kalogeraki V, Pruyne J (2002) Finding good peers in peer to peer networks. In: IEEE international symposium on parallel and distributed processing, Ft. Lauderdale, FL, USA, pp 24-31
9. Diaz G, Chen K (2008) Distributed management to services deployment in autonomic networks. In: 3rd international conference on information and communication technologies: From theory to applications, Damascus, pp 1 - 6
10. Tang CQ, Xu ZC, Mahalingam M (2002) PeerSearch: Efficient Information Retrieval in Peer-to-Peer Networks. HPL-2002-198 July 12th, pp 1 - 7
11. Lv Q, Cao P, Cohen E, et al (2002) Search and replication in unstructured peer-to-peer networks. In: Conference proceedings of the 2002 international conference on supercomputing, pp 84-95
12. Yang B, Garcia-Molina H (2002) Improving search in peer-to-peer networks. In: 22nd international conference on distributed computing systems, Vienna, pp 5-14
13. Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D (2002) A local search mechanism for peer-to-peer networks. In: Proceedings of the eleventh international conference on information and knowledge management (CIKM), pp 300-307
14. Tsoumakos D, Roussopoulos N (2006) Analysis and comparison of P2P search methods. In: Proceedings of the 1st international conference on Scalable information systems table of contents, Hong Kong, pp 25-35
15. Yang B, Garcia-molina H (2002) Efficient search in peer-to-peer networks. Technical Report 2001-47, Stanford University, October 2001.
16. Rostami H, Habibi J, Abolhassani H, Amirkhani M, Rahnama A (2006) An ontology based local index in P2P networks. In: *IEEE* proceedings of the second international conference on semantics, knowledge, and grid, pp 11-11
17. Vishnevsky V, Safonov A, Yakimov M, Shim E, Gelman AD (2006) Tag routing for efficient blind search in peer-to-peer networks. In: Proceedings of the 11th IEEE symposium on computers and communications, pp 409 - 416
18. Bisnik N, Abouzeid AA (2007) Optimizing random walk search algorithms in P2P networks. Comput Netw 51(6):1499–1514
19. Bisnik N, Abouzeid A (2005) Modeling and analysis of random walk search algorithms in P2P networks. In: Proceedings. second international workshop on hot topics in peer-to-peer systems, San Diego CA, pp 95-103
20. Mihalyia (2004) Optimizing peer-to-peer networks using mobile agents. In: Proceeding elmar 46th international symposium, pp194-199
21. Crespo A, Gareia-Monila H (2002) Routing indices for peer-to-peer systems. In: Proceedings of the international conference on distributed computing systems, pp 23-32
22. Tsoumakos D, Roussopoulos N (2003) Adaptive probabilistic search for peer-to-peer networks. In: Proceedings of 3rd IEEE International Conference on P2P Computing, pp 102 - 109
23. Terpstra WW , Kangasharju J, Leng C, Buchmann AP, Bubble-Storm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search. In: ACM SIGCOMM 2007: Conference on Computer Communications, Kyoto, pp 49-60
24. Ahn CW, Ramakrishna RS (2002) A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE transactions on evolutionary computation 6(6):566–579
25. Wong WY, Lau TP, King PI (2005) Information retrieval in P2P networks using genetic algorithm. In: Special interest tracks and posters of the 14th international conference on World Wide Web, Chiba, pp 922 - 923
26. Wu CJ, Yang KH, Ho JM (2006) AntSearch: An ant search algorithm in unstructured peer-to-peer networks. In: Proceedings of the 11th IEEE symposium on computers and communications, Cagliari, pp 429 - 434
27. Wu GY, Liu JY, Shen X, Gao LJ, Xu JD, Xi K (2008) ERAntBudget: A search algorithm in unstructured P2P networks.

In: Proceedings of the second international symposium on intelligent information technology application, Shanghai, pp 765 - 769
28. Li JQ, Pan QK, Xie SX (2008) Research on peer selection in peer-to-peer networks using ant colony optimization. In: Proceedings of the fourth international conference on natural computation, Jinan, pp 516-520
29. Cao Y, Li SZ (2006) Research on P2P hybrid information retrieval based on ant colony algorithm. In: Proceedings of the 10th international conference on computer supported cooperative work in design, Nanjing, pp 1048-1052
30. Colorni A, Dorigo M, Manieezzo V (1991) Distributed optimization by ant colonies. In: Proc. of the first european Conf. on artificial life, Paris, pp 134-142
31. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperation agents. IEEE Transaction on System, Man, and Cybernetics-Part B 26(1):28–41
32. Liu P, Gao F, Yang Y (2007) QoS routing algorithm based on the combination of genetic algorithm and ant colony algorithm. Application Research of Computers 24(9):224–227
33. Xiong ZG, Yang PY, Zhang XM, Chen F, Liu PL (2007) Integrating genetic and ant algorithm into P2P grid resource discovery. In: Proceedings of the third international conference on international information hiding and multimedia signal processing (IIH-MSP 2007), Kaohsiung, pp 83-86
34. Dasgupta P (2005) Intelligent agent enabled genetic ant algorithm for P2P resource discovery. Agents and Peer-to-Peer Computing, Lecture notes in computer science 3601:213–220

Cai Kang received the B.S. and M.S. degrees in Computer Science and Engineering from South China University of Technology in 1986 and 1989.Currently he is working towards his Ph.D. degree in School of Electronic and Information at South China University of Technology. His research interests are in the areas of scalable peer-to-peer networks, distributed systems, stream delivery, internet traffic measurement, computer networks and cloud computing. He is a senior member of China Communication Association. Contact him at School of Electronics and Information , South China University of Technology , Guangzhou , Guangdong 510640 , China ; caik@gsta.com