https://doi.org/10.26599/IJCS.2022.9100022 International Journal of Crowd Science

Commodity Search Based on the Hybrid Breadth-Depth Algorithm in the Crowd Intelligence Based Transaction Network

Zhishuo Liu[1] ✉, Yinan Cheng[1], and Fang Tian[2]

ABSTRACT

Crowd intelligence based transaction network (CIbTN) is a new generation of e-commerce. In a CIbTN, buyers, sellers, and other institutions are all independent and intelligent agents. Each agent stores the commodity information in a local node. The agents interconnect through a circle of friends and construct an unstructured network. To conduct the commodity search task in a network more efficiently and in an energy-saving manner when a buyer presents a commodity demand, a hybrid breadth-depth search algorithm (HBDA) is proposed, which combines the search logic of the breadth-first search algorithm and the depth-first search algorithm. We defined the correlation degree of nodes in a network, optimized the rules of search and forwarding paths using the correlation degree between a node and its neighboring nodes in the circle of friends, and realized the HBDA based on the PeerSim simulation tool and Java. Experimental results show that, in general, the proposed HBDA has a better search success rate, search time, commodity matching degree, and search network consumption over the two blind search algorithms. The HBDA also has good expansibility, thus allowing it to be used for commodity search efficiently with a high success rate in large-scale networks.

KEYWORDS

crowd science; e-commerce; crowd intelligence based transaction network; unstructured network; commodity search; search algorithm

E-commerce has become a mature business model in recent years. It refers to conducting trade activities such as purchasing and selling items electronically, including business to business, business to consumer, consumer to consumer, manufacturers/factories to consumer, online to offline, and government to business. In the e-commerce model, an online platform serves as a center where buyers and sellers go for trading. The platform provides users with various functions, including information registration, commodity recommendation, search, trading, payment, and credit evaluation.

In the work of Chai et al.[1], they proposed that individuals, enterprises, government departments, and other institutions in physical space and other kinds of intelligent equipments and objects, defined as intelligent agents, become more intelligent with the application of big data and intelligence technologies. Yu et al.[2] introduced the concept of collective intelligence from the perspective of crowd science and discussed the application and future potential of collective intelligence. Intelligent agents can be mapped to mirrors in the information space with the help of network technologies and form intelligent digital bodies. These digital bodies can then form a self-organized and ecological transaction network and give rise to many kinds of interactive behaviors, which have the features of personalized and active consumption, centralized and direct circulation, intelligent and decentralized production, and personalized and convenient life, thus forming the crowd intelligence based transaction network (CIbTN), a new generation of e-commerce form that interconnects all things.

The CIbTN in nature has a distributed and unstructured network structure, which means that the operation of this network does not rely excessively on a particular node. The network forms a circle of friends between the intelligent agents, including buyers, sellers, third-party payment institutions, and logistics service providers, according to the correlation degree. Through the circle of friends of different degrees, the CIbTN interconnects everything inclusively. Nodes in different circles of friends can interact or trade with each other, collaboratively making decisions in a more intelligent way[3].

In the CIbTN, buyer and seller nodes both store a series of commodity information related to themselves in a local node. When a buyer node requests information about a certain commodity, it can obtain not only this information from the seller node but also information about the purchase history, commodity evaluation, and sellers' recommendations from the buyer node, thus being able to choose the commodity more diversely and comprehensively.

Commodity search algorithms need to be employed to find the target commodity information efficiently in the CIbTN. The most basic and commonly used algorithms for resource search in an unstructured network are blind search algorithms, including the flooding search algorithm (also known as the breadth-first search algorithm (BFSA)), and random walk search algorithm (also known as the depth-first search algorithm (DFSA)). However, these search algorithms have the shortcomings of long search time, low search efficiency, unguaranteed search success rate, high storage costs, and tendency to occupy large amounts of network broadband resources, as summarized by Thampi and Sekaran[4] and Anthony et al.[5] With the rapid development of the CIbTN in recent years, the number of resources and nodes in the network has considerably increased. Therefore, a significant task is to

1 School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
2 Business Administration Division, Pepperdine University, Malibu, CA 90263, USA
Address correspondence to Zhishuo Liu, zhsliu@bjtu.edu.cn

devise new commodity search algorithms to conduct search tasks in unstructured networks more efficiently and in an energy-saving way, which is the main purpose of this paper.

In this paper, we devise an algorithm that combines the BFSA and the DFSA, and we conduct several simulation experiments to test the performance of the proposed algorithm. The experimental results show that, unlike the two classic blind search algorithms, namely, the flooding algorithm and the random walk algorithm, the search performance of the whole network can be enhanced by employing the proposed hybrid breadth-depth search algorithm (HBDA) with a higher search success rate, lower search time, and lower broadband resource consumption. Thus, the operational cost of the network can be significantly reduced, and the operational efficiency and user experience are greatly improved. We also find that the performance of the HBDA does not decrease as the network scale increases, which means that it has good expansibility that can be used for commodity search in real-world CIbTN application scenarios.

The rest of this paper is structured as follows. Section 1 describes the related works in detail. Section 2 presents the proposed HBDA. Section 3 discusses the results through experimental analyses. Section 4 presents the conclusion.

1 Related Works

A number of algorithms have been developed over the years aiming at solving the commodity information resource search problem in a distributed and unstructured network. Generally, these algorithms can be divided into three types.

1.1 Blind search algorithms

Blind search algorithms are the most basic algorithms for commodity search. Although the idea of these algorithms is simple and they are easy to use, they cannot guarantee the search success rate and efficiency, especially in networks with large and complicated structures.

The flooding algorithm was first proposed in the Gnutella network (gnutella.com)[6]. When a node forwards all search requests to its neighboring nodes for resource search, the neighboring nodes then forward the search requests to all their neighboring nodes until the required resource is found or the time to live (TTL) becomes zero. The advantage of this algorithm is that it can guarantee the success rate of network search, while the disadvantage is that the number of forwarding nodes increases exponentially when the network scale is large. Thus, a large number of useless messages will be generated, and they will consume many broadband resources, resulting in a long search time. Therefore, this algorithm is inefficient and time consuming, thus being unsuitable for application in the CIbTN. Gkantsidis et al.[7] proposed the random walk search algorithm where the node forwards the search request randomly to its K neighboring nodes rather than to all its neighboring nodes as in the flooding algorithm, thereby substantially reducing the number of messages transmitted during the search. However, the randomness in choosing the K nodes in this algorithm reduces the search success rate, and thus, the search time will probably increase. Sugawara[8] and Kalogeraki et al.[9] proposed the modified-BFS search algorithm, which improves the flooding algorithm. In this algorithm, the selection of nodes for request forwarding does not include all neighboring nodes. Instead, it is based on the probability calculated by a proposed equation. The technique of expanding rings was proposed by Yang and Garcia-Molina[10], which reduces the number of unnecessary searches by periodically querying the depth of the search path and taking it as the criteria for terminating the search. When the same search path depth appears, the search process terminates.

1.2 Heuristic search algorithms

Heuristic search algorithms are developed on the basis of blind search algorithms. In combination with other algorithms, these algorithms set reasonable search forwarding rules and heuristic forwarding probability so that they can perform commodity search tasks by using fewer broadband resources in an unstructured network. These algorithms have become a research hot spot in recent years.

In Huang's thesis[11], a dynamic search strategy based on the greedy algorithm was proposed, which combines dynamic programming and greedy algorithm with the search problem. This thesis analogized the search problem to the path selection problem of dynamic optimization. On the basis of the calculated path weights, the nodes will choose the neighboring nodes with smaller weights when forwarding. Therefore, the forwarding is more purposeful and effective, thus improving the search efficiency. The SPUN search algorithm proposed by Himali and Prasad[12] calculates the success probability of neighboring nodes to successfully search different resources by recording the history search behavior of nodes. When forwarding messages in the next search, the first K neighboring nodes with the highest success probability will be chosen, and a specific TTL value will be set as the criteria for ending the search. This algorithm can reduce the blindness of search request forwarding effectively. Nevertheless, this algorithm has low performance when the search success probability is only one of the many factors that affect the search efficiency when the nodes have more attributes. Joseph and Hoshiai[13] proposed a search strategy based on the aggregation of interest domains. A certain interest degree for nodes is defined by comparing nodes' storing resources. Nodes with similar interest degrees are aggregated so that when a resource search request is generated, the node gives priority to its neighboring nodes with similar interest degrees. The algorithm improves the probability of successful searches by changing the spatial location of nodes, which can improve the search efficiency of the network to a certain extent, but the change of node locations in the network consumes a large amount of network resources, leading to a long search time. Using the power-law distribution principle and small-world properties, Tang et al.[14] proposed a search method that combines the meritocratic connection mechanism and the interest decay mechanism in rumor propagation, contributing to a significant reduction in the number of redundant information propagation in the search process, although it shows no significant improvements in the search success rate.

1.3 Search algorithms based on node features

These algorithms are applied to networks with specific structures, holding a certain number of "super nodes" that perform the main information-bearing and resource search tasks. Each super node in the network is connected to a certain number of common nodes, and the super nodes are connected to each other. These algorithms are not widely used because of the limitations of the application scenario.

Leibowitz et al.[15] divided the nodes into super nodes and normal nodes according to their attributes and capabilities. The super nodes perform all the search tasks within the group in the form of topological grouping, which requires them to have high performances. If any of the super nodes malfunction, then the search in the network will be severely affected. Shen et al.[16]

proposed a special index structure and information summarization method, where the index table of a normal node contains all the resource information of itself, while the super node index table contains the resource information of all nodes connected to it so that fast search of resources can be achieved. Xiao et al.[17] explored the relationship between the ratio of super and normal nodes in the network to give a more reasonable ratio, which will improve the search efficiency.

On the basis of previous works, we propose the HBDA, which combines the BFSA and DFSA. Unlike previous works, we present the concept of node correlation degree. With this concept, the search request forwarding between nodes can be closely linked with the node attribute and behavioral features, thus reducing the blindness of search. Combining the BFSA and DFSA integrates the search logic of the two algorithms, thus making the search process in the network more efficient and resource-saving, contributing to better search performance over traditional algorithms.

2 Hybrid Breadth-Depth Heuristic Search Algorithm

2.1 Main idea and aim of the hybrid breadth-depth heuristic search algorithm

2.1.1 Main idea

In the BFSA, when a search request is sent out, it needs to be forwarded to all neighboring nodes, which guarantees the search success rate, but it consumes a large amount of network broadband resources and generates a considerable amount of redundant information, thus resulting in a longer search time and lower search efficiency. In the DFSA, a neighboring node is randomly selected for forwarding each time, which consumes few network broadband resources and enhances the search speed. However, the random search forwarding makes the search too blind, thus resulting in a low search success rate, and generating many useless searches.

In the CIbTN, buyer and seller nodes have attribute and behavioral features. For example, a node has attribute features such as age, education, gender, occupation, hobbies, and income. If some nodes have similar attribute features, then they may like the same commodity, which means that a certain node has a higher probability of finding the required commodity in neighboring nodes with similar features. Nodes record their behavioral features and those of their neighboring nodes, such as the history transmitting time, history search success time, and number of commodities of the neighboring nodes. These features reflect the closeness between a certain node with its neighboring nodes. If a neighboring node has more history transmitting and search success times, then the probability of finding commodity information in this node is higher. On the basis of this condition, we define the attribute correlation features and behavioral correlation features between nodes as the attribute correlation degree and behavioral correlation degree, respectively, and the node correlation degree is the combination of the attribute and behavioral correlation degrees. Before search forwarding is performed, the correlation degree with neighboring nodes should be calculated for each node, and forwarding to neighboring nodes with higher correlation degrees is prioritized to reduce the number of network search message transmissions, improve the search success rate, and reduce the search time.

When a node performs a search task, it is necessary to determine whether the information of the commodity in that node match that of the demanded commodity. To solve this problem, we introduce the idea of the space vector, which calculates the cosine similarity between the demanded commodity information and the commodity information in the node as the matching degree of the search result. Also, a certain matching degree threshold is set. If the threshold is exceeded, then it can be recognized that the commodity resource satisfies the search demand.

2.1.2 Aim

The aims of the HBDA are

(1) To ensure the search success rate while optimizing the search path, reducing unnecessary message forwarding and the occupancy of the entire network broadband, and improving the search efficiency.

(2) To improve the success rate of search and achieve more network searches with less time, while inheriting the speed advantage of the DFSA.

(3) To enhance the extensibility of the search algorithm, i.e., as the number of network nodes increases, the performance of the search algorithm should not decrease.

2.2 Definition

Definition 1 Node attribute correlation degree

An attribute that describes the correlation relationship between the buyer node and seller node attribute features in the CIbTN. The gender, age, education level, and income status are selected to indicate the correlation degree using the calculation method of Euclidean distance.

Definition 2 Node behavioral correlation degree

An attribute that describes the relationship between the history behavioral features of buyer nodes and seller nodes in the CIbTN, and is composed of the history communication frequency between nodes, search success rate, and the total proportion of the number of commodity information, which represents the close relationship between the historical behavior of nodes and neighboring nodes.

Definition 3 Node correlation degree

The combination of the node attribute and behavioral correlation degrees indicates the comprehensive correlation relationship between nodes, and is an important basis for searching and forwarding commodity information resources in the CIbTN.

Definition 4 Commodity matching degree

A measure of whether the commodity information resources in the node match the required search demand commodity information resources. The calculation process of the commodity matching degree is given as follows.

Commodity information resources are composed of keywords and keyword frequencies. Therefore, commodity information keyword feature weights are calculated using the $tf \times idf$ framework, where tf is short for “term frequency”, indicating the number of times a keyword appears in the commodity information, and idf refers to the “inverse document frequency” factor. The more a word appears in the document collection, the weaker its ability to distinguish the difference between documents and the lower the idf value. For example, words such as “你 (you)”, “我 (I)”, and “的 (’s)” in a document cannot be used to distinguish because of their high frequency of occurrence between documents, and the idf value of these words is very low. We use the $tf \times idf$ method to calculate the keyword weights in commodities. See Eqs. (1)–(3).

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

$$idf_i = \log \frac{N}{n_i} \tag{2}$$

$$\omega_{i,j} = tf_{i,j} \times idf_i = \frac{n_{i,j}}{\sum_k n_{k,j}} \log \frac{N}{n_i} \tag{3}$$

where $n_{i,j}$ denotes the number of keywords key_i appearing in commodity D_j, $\sum_k n_{k,j}$ denotes the number of occurrences of all keywords in commodity D_j, N denotes the number of commodities, and n_i denotes the number of commodities containing key_i.

We then use the space vector to calculate the commodity matching degree M_k. See Eq. (4).

$$M_k = \frac{(\boldsymbol{q}\boldsymbol{D}_j)}{|\boldsymbol{q}| \times |\boldsymbol{D}_j|} = \left[\sum_{i=1}^{n} (\omega_{i,j} m_{i,q})\right] \Big/ \left(\sqrt{\sum_{i=1}^{n} \omega_{i,j}^2}\sqrt{\sum_{i=1}^{n} m_{i,q}^2}\right) \tag{4}$$

where the vector $\boldsymbol{q}(m_{i,1}, m_{i,2}, \ldots, m_{i,q})$ refers to the searched commodity. $\boldsymbol{D}_j(\omega_{i,1}, \omega_{i,2}, \ldots, \omega_{i,j})$ refers to the searched commodity in the node, $m_{i,q}$ refers to the weight of the keyword key_i of the searched commodity, and $\omega_{i,j}$ refers to the weight of the keyword key_i of the searched commodity in the node.

Definition 5 Search rule

When a node generates a commodity information searching request, a local node search is conducted first, followed by a network node search. At the same time, a TTL value is given. When a search request is forwarded, TTL is reduced by one, i.e., $\text{TTL} = \text{TTL} - 1$. The search terminates when $\text{TTL} = 0$ or when the required commodity information resources satisfy the termination condition.

Definition 6 Forwarding rule

The node calculates the correlation degree of all its neighboring nodes and selects the top m nodes in the correlation degree ranking before forwarding the search request.

Definition 7 Feedback rule

The node will send the commodity information back to the initial search request node via the original path if it contains the commodity information resources that satisfy the commodity matching degree.

2.3 Calculation of correlation degrees

2.3.1 Attribute correlation degree

When the attribute correlation degree is calculated, the influence factors of node users' age, gender, income status, and education are selected. The buyer node selects these four attribute features as calculation factors, and the seller node selects four attribute features of the main buying group as calculation factors. We define each node as a multidimensional vector space, and each vector space consists of several attribute vectors that represent attribute feature information, denoted by $\boldsymbol{V}_i = (a_{1,i}, a_{2,i}, \ldots, a_{n,i})$. The value of the feature vector represented by each attribute feature information in the vector space is determined by the dimension of the attribute feature information, a natural number starting from zero. The attribute quantitative tables for buyer and seller nodes are shown in Tables 1 and 2.

We calculate the attribute relevance of the buyer node and its neighboring nodes, measured by the Euclidean distance. The eigenvectors of a node in the network and its neighboring nodes are $\boldsymbol{V}_i = (a_{1,i}, a_{2,i}, \ldots, a_{n,i})$ and $\boldsymbol{V}_j = (b_{1,j}, b_{2,j}, \ldots, b_{n,j})$. If the difference in attribute features of the two nodes is larger, the Euclidean distance $|\Delta V|$ of the vector is also larger. Thus, the two-vector Euclidean distance can be used to represent the attribute correlation degree of two nodes i and j. The attribute correlation degree $\text{SimA}(\boldsymbol{V}_i, \boldsymbol{V}_j)$ is calculated by Eqs. (5) and (6).

Table 1 Buyer node attribute quantitative table.

Attribute type	Attribute feature		Feature value
T1	Gender	Male	0
		Female	1
T2	Age	Children	0
		Teenager	1
		Middle-aged	2
		Aged	3
T3	Income	No income	0
		Low income	1
		Medium income	2
		High income	3
T4	Education	High school and below	0
		College	1
		Bachelor	2
		Master's degree	3
		PhD and above	4

Table 2 Seller node attribute quantitative table.

Attribute type	Attribute feature		Feature value
T1	Main gender of frequent buyers	Male	0
		Female	1
		Full coverage	θ
T2	Main age distribution of frequent buyers	Children	0
		Teenager	1
		Middle-aged	2
		Aged	3
		Full coverage	θ
T3	Main income level of frequent buyers	No income	0
		Low income	1
		Medium income	2
		High income	3
		Full coverage	θ
T4	Main education level of frequent buyers	High school and below	0
		College	1
		Bachelor	2
		Master's degree	3
		PhD and above	4
		Full coverage	θ

$$|\Delta V| = \sqrt{(b_{1,j} - a_{1,i})^2 + (b_{2,j} - a_{2,i})^2 + \cdots + (b_{n,j} - a_{n,i})^2} \tag{5}$$

$$\text{SimA}(\boldsymbol{V}_i, \boldsymbol{V}_j) = \begin{cases} 1, & \text{if} |\Delta V| = 0; \\ \frac{1}{|\Delta V|}, & \text{otherwise} \end{cases} \tag{6}$$

2.3.2 Behavioral correlation degree

The behavioral features of each node and its neighboring nodes in the CIbTN, such as the number of commodity types owned, history communications, history search successes, and common neighboring nodes, change sporadically. Hence, each node has a neighboring node behavioral feature information table that records the number of communications between itself and each of its neighboring nodes, the number of search successes, and the commodity. It is dynamically updated after each search is conducted. If a node has more history communication counts and more search successes with a specific neighboring node, then the probability of its next communication and search with that neighboring node is higher, and the greater number of commodities of a node means that the probability of finding the desired commodities with the same communication step is higher. Thus, the behavioral correlation degree $\mathrm{SimB}(\boldsymbol{V}_i, \boldsymbol{V}_j)$ of each node and its neighboring nodes can be calculated based on these historical records. We first calculate the following three rates, namely, $X(i,j)$, $Y(i,j)$, and $Z(i,j)$, and then sum them up to obtain $\mathrm{SimB}(\boldsymbol{V}_i, \boldsymbol{V}_j)$.

$X(i,j)$: The history communication rate, i.e., the rate of the number of total history communications between node i and its neighboring node j to the total number of communications between node i and all its neighboring nodes, is shown in Eq. (7).

$$X(i,j) = \frac{m_{(i,j)}}{\sum_{j=1}^{n} m_{(i,j)} + \sigma} \tag{7}$$

where $m_{(i,j)}$ denotes the number of history communications between node i and its neighboring node j. σ is a random small positive number to prevent $X(i,j)$ from reaching $+\infty$ when the total number of neighboring node communications of node j is 0. σ can be 10^{-11} in the calculation.

$Y(i,j)$: The history search success rate, i.e., the rate of the number of total history search success times between node i and its neighboring node j to the total history search success times between node i and all its neighboring nodes, is shown in Eq. (8).

$$Y(i,j) = \frac{l_{(i,j)}}{\sum_{j=1}^{n} l_{(i,j)} + \sigma} \tag{8}$$

where $l_{(i,j)}$ denotes the number of history search success times between node i and its neighboring node j.

$Z(i,j)$: The rate of the number of commodities of node j, which is a neighboring node of node i, to the total number of commodities of all neighboring nodes of node i, is shown in Eq. (9).

$$Z(i,j) = \frac{o_{(i,j)}}{\sum_{j=1}^{n} o_{(i,j)} + \sigma} \tag{9}$$

where $o_{(i,j)}$ denotes the number of commodities of node i's neighboring node j.

The behavioral correlation degree $\mathrm{SimB}(\boldsymbol{V}_i, \boldsymbol{V}_j)$ is then calculated by Eq. (10).

$$\mathrm{SimB}(\boldsymbol{V}_i, \boldsymbol{V}_j) = X(i,j) + Y(i,j) + Z(i,j) \tag{10}$$

2.3.3 Comprehensive correlation degree

The comprehensive correlation degree is composed of the attribute and behavioral correlation degrees. See Eq. (11).

$$\mathrm{Sim}(\boldsymbol{V}_i, \boldsymbol{V}_j) = \alpha \mathrm{SimA}(\boldsymbol{V}_i, \boldsymbol{V}_j) + \beta \mathrm{SimB}(\boldsymbol{V}_i, \boldsymbol{V}_j) \tag{11}$$

where α and β denote the proportion parameter of the attribute and behavioral correlation degrees, respectively.

2.4 Algorithm data structure and message packet format

2.4.1 Algorithm data structure

Certain data structure tables need to be created in the nodes to satisfy the requirements of node information maintenance and the calculation of correlation and matching degrees in the algorithm.

(1) Index table of local commodity information resources

Index table of local commodity information resources includes commodity name, category, information keyword, keyword weight, and the location of the node where the commodity information is.

(2) Buyer and seller node attribute information tables

The node attribute information tables record the attribute values of the node itself, including information such as gender, age, income level, and education, to facilitate the access and recording of attribute information by neighboring nodes during the search process. It includes the buyer and seller nodes, respectively.

(3) Neighboring node correlation degree information table

By calculating and updating the correlation degree of neighboring nodes each time, this table facilitates the recording and access of the correlation degree.

(4) Neighboring node attribute information table

Neighboring node attribute information table records the attribute information of all the neighboring nodes, including gender, age, income, and education, which are used to calculate the attribute correlation degree.

(5) Neighboring node history behavior information table

Neighboring node history behavior information table records the number of history information transmission, search success, and commodities in the neighboring nodes, which are used to calculate the behavioral correlation degree.

2.4.2 Message packet format

The message packet format table includes four columns, which contain the ID and type of the message packet, the steps of message transmission (hops), and the TTL of the message. Five types of message packets exist in the HBDA.

(1) Add: When a new user registers for the CIbTN, new nodes and node IDs will be generated to facilitate network expansion and updates.

(2) Send: This is used to add requests to nodes in the network as neighboring nodes.

(3) Respond: This is used for nodes to respond to the "Add" requests and share information such as node attributes to already added nodes.

(4) Query: This is used to search for commodity information resources in the CIbTN. When a node receives a search request, it will receive a query message packet that contains new keywords and the keywords' weights for the demanded commodity, and the node will calculate the commodity match degree. If a matching commodity information resource is available, then it will generate a query-hit message packet to return the commodity information resource.

(5) Query-hit: This is used to return a successfully searched commodity information resource for the search request node to download and review the commodity information.

2.5 Algorithm description and flow

The search by the HBDA is performed in two stages: the node commodity information resource search and the network commodity information resource search. When a node in the CIbTN receives a search request for commodity information resources from node S, it first performs a search and calculation of commodity matching degree in a local node commodity information resource list and compares the result with the matching degree threshold. If commodity information meets the demand, then it returns to node S; if not, then the network commodity information resource search will be performed according to the forwarding and search rules. The search terminates when the required commodity information resource is searched or TTL = 0. The flowchart of the algorithm is shown in Fig. 1.

2.5.1 Node commodity information resource search algorithm

(1) Node S generates a search request for commodity information resource $\boldsymbol{q}$, which contains commodity keywords and the frequency of keyword occurrences.

(2) The keyword weights of $\boldsymbol{q}$ are calculated by using Eqs. (1)–(3), and then the keyword weight space vector $\boldsymbol{q}(m_{i,1}, m_{i,2}, \ldots, m_{i,q})$ is generated.

(3) The commodity matching degree M_k between $\boldsymbol{q}(m_{i,1}, m_{i,2}, \ldots, m_{i,q})$ and $\boldsymbol{D}_j(\omega_{i,1}, \omega_{i,2}, \ldots, \omega_{i,j})$, namely the commodity information resource in the node, is calculated by using Eq. (4).

(4) M_k is compared with the commodity matching degree threshold M_0. If $M_k \geqslant M_0$, then the commodity information meets the demand and can be returned to node S.

(5) If all M_k in the searched node is less than M_0, then a value for TTL will be given, and the network commodity search will be conducted.

The pseudo-code is shown in Algorithm 1.

2.5.2 Network commodity information resource search algorithm

If the node search cannot find the required commodity

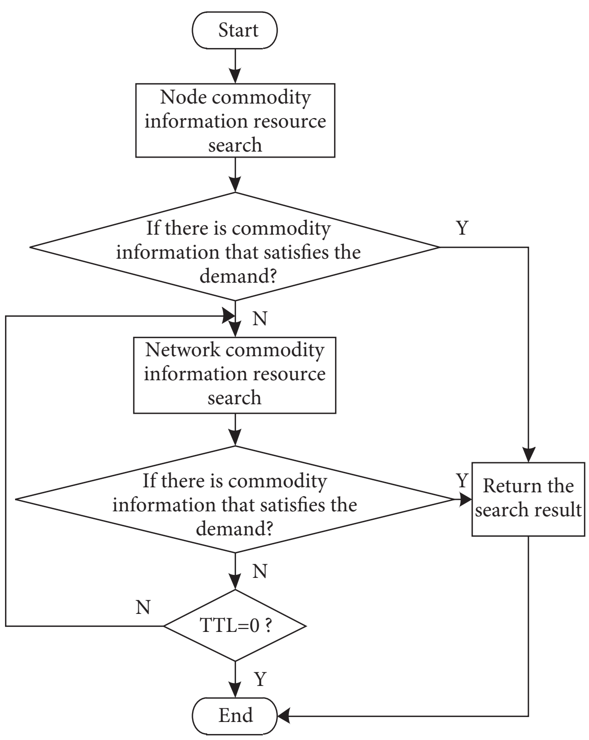

Fig. 1 Flowchart of the HBDA.

Algorithm 1 Node commodity information resource search

Input: The requested commodity information resource $\boldsymbol{q}$
Output: The commodity information resource with the commodity matching degree not less than M_0
Local search $\boldsymbol{q}$ (keys, commodity info) {
Begin
Generate request for commodity information resource $\boldsymbol{q}$ at node S (keys)
According to the keywords and keyword frequencies, calculate the keyword weights in $\boldsymbol{q}$ and obtain the space vector $\boldsymbol{q}(m_{i,1}, m_{i,2}, \ldots, m_{i,q})$
Calculate the commodity matching degree
If ($M_k \geqslant M_0$)
Return D_k (keys, commodity info) // D_k is the searched commodity information
Else
Search in the network neighboring nodes
End
}

information resource, then the network commodity information resource search will be performed.

(1) The neighboring nodes' correlation degree is calculated. Before the search request is forwarded from the local node to the neighboring nodes, the local node reads the attribute and historical behavioral features of all neighboring nodes, and calculates the attribute, behavioral, and comprehensive correlation degrees of neighboring nodes according to Eqs. (6), (10), and (11). The calculated correlation degrees are then recorded in the neighboring node correlation degree information table.

(2) The neighboring nodes are selected for forwarding. The neighboring nodes are sorted according to their correlation degree, and the nodes with the top m correlation degree are selected for search request forwarding. A TTL value is given to the search, and TTL–1 for each search request forwarding in the search. m is an adjustable parameter according to the overall network structure and the number of neighboring nodes.

(3) When the search request reaches the neighboring nodes, the local commodity information resources are searched on m neighboring nodes, the commodity matching degree is calculated, and a comparison is made to check if satisfactory search results are obtained. If so, then the commodity information resources are returned and the search terminates; if not, Step (4) is performed.

(4) TTL = 0 is checked. If true, then the search ends; if TTL is greater than 0, then the search request for forwarding to neighboring nodes is performed; that is, Steps (1)–(3) are repeated until the requested commodity information resources are found or TTL = 0, the search ends.

The pseudo-code is shown in Algorithm 2.

3 Simulation Experiment

PeerSim is used to build the simulation environment of the CIbTN. The network has a certain scale of user nodes, and each node has its own unique node ID. The nodes store random commodity information according to the data structure. Each node can freely join and exit the network, and it has a certain number of neighboring nodes. The user nodes are divided into buyer and seller nodes according to a certain ratio, and the buyer and seller nodes have different numbers of commodity information and attribute features. The search requests are randomly generated on the buyer nodes and are searched using the search algorithm.

Algorithm 2 Network commodity information resource search

Input: The requested commodity information resource $\boldsymbol{q}$ forwarded by neighboring nodes
Output: The commodity information resource with the commodity matching degree not less than M_0
Network search $\boldsymbol{q}$ (keys, commodity info) {
Begin
  Read the local neighboring node attribute and behavioral info tables
  Calculate $\mathrm{Sim}(\boldsymbol{V}_i, \boldsymbol{V}_j)$
  Record $\mathrm{Sim}(\boldsymbol{V}_i, \boldsymbol{V}_j)$ in the local neighboring node matching degree table and sort the neighboring nodes according to their matching degrees
  Forward the search request $\boldsymbol{q}$ to $V_1, V_2, \ldots, V_k$, and give a value to TTL
  While $(m > 0)$ {
  Calculate the commodity matching degree M_k
  If $(M_k \geqslant M_0)$
    Return D_k (keys, commodity info) // D_k is the searched commodity information
    Search ends
      m=m−1
  Else
      m=m−1
  else while
      TTL=TTL−1
  }
  If (TTL > 0)
    Proceed neighboring node forwarding
  Else
    Search ends // No commodity information resource is found
End
}

Java is used to implement the proposed HBDA, together with the flooding algorithm and the random walk algorithm, to conduct a performance comparison of different algorithms. The algorithms are simulated in the PeerSim network environment, the experimental data are collected through the control interface, and the search performances of the three search algorithms are compared. In addition, in the experiments, multiple commodity information resource searches are conducted in one experimental cycle, and each search algorithm runs in multiple cycles. Then, the average results of the multiple experiments are taken to increase the objectivity and credibility.

3.1 PeerSim simulation tool

PeerSim is a software that simulates distributed networks, which can be either structured or unstructured. The CIbTN has a distributed and unstructured network structure. Thus, PeerSim can be used to simulate its buyer and seller nodes and its network structure.

PeerSim has two simulation methods, namely, the cycle-based mode and event-driven mode, each of which has its own characteristics. The event-driven mode allows for control of the transport layer and concurrency of the network. However, in this mode, combining the message forwarding process with other algorithms is difficult. In the cycle-based mode, nodes directly communicate with each other, which has the characteristics of high scalability and low resource consumption. Therefore, it can realize large-scale node simulation (up to 10 million) and it can be combined with other algorithms more easily. Therefore, we adopt the cycle-based mode in the following experiments.

The main interfaces of PeerSim and their functions in the cycle-based mode are shown in Table 3.

3.2 Simulation flow

The flow of implementing the simulated commodity search task by the HBDA is illustrated in Fig. 2, where k is the number of searches, v is the number of successful searches, M_k is the commodity matching degree, and M_0 is the commodity matching degree threshold.

3.3 Algorithm criteria

The search algorithm evaluation criteria are chosen and designed from two dimensions, namely, the user experience and the network performance.

3.3.1 User experience criteria

In the user experience dimension, four evaluation indexes are selected: search success rate, average search time, average number of searched commodity information resources, and average commodity matching degree.

(1) Search success rate

This is the ratio of the number of successful searches to the total number of searches in an experiment. A high search success rate of a search algorithm corresponds to its better search performance in the network and a greater possibility of the buyer nodes when searching for the commodity information resources.

(2) Average search time

This is the ratio of the total search time to the total number of searches in an experiment. A short average search time corresponds to its fast search speed and shorter waiting time of the buyer node during the search process.

(3) Average number of searched commodity information resources

This is the ratio of the total number of commodity information resources returned in an experiment to the number of successful searches. It reflects the number of resource information returned by a successful search of a buyer node. A large criterion corresponds to the user's greater selectivity and reference range of search results and thus a greater possibility that the user will be satisfied with the search results.

(4) Average commodity matching degree

This is the average of all the returned product information resources matched in an experiment. A high-average matching degree of a search algorithm's returned products corresponds to the algorithm's better search matching effect in the network and thus likely higher user satisfaction.

Table 3 Main interfaces of PeerSim.

Name	Function
Node	Protocol container for the nodes, defining a unique node ID for each node in the network and having all the node access protocols
CD Protocol	A specific protocol in the cycle-based mode used for cycle-based operational control of the simulation in this mode
Linkable	Used for node access and information transmission in the network
Control	Used for monitoring the simulation process and collecting the experimental results to serve the analysis of the experimental results

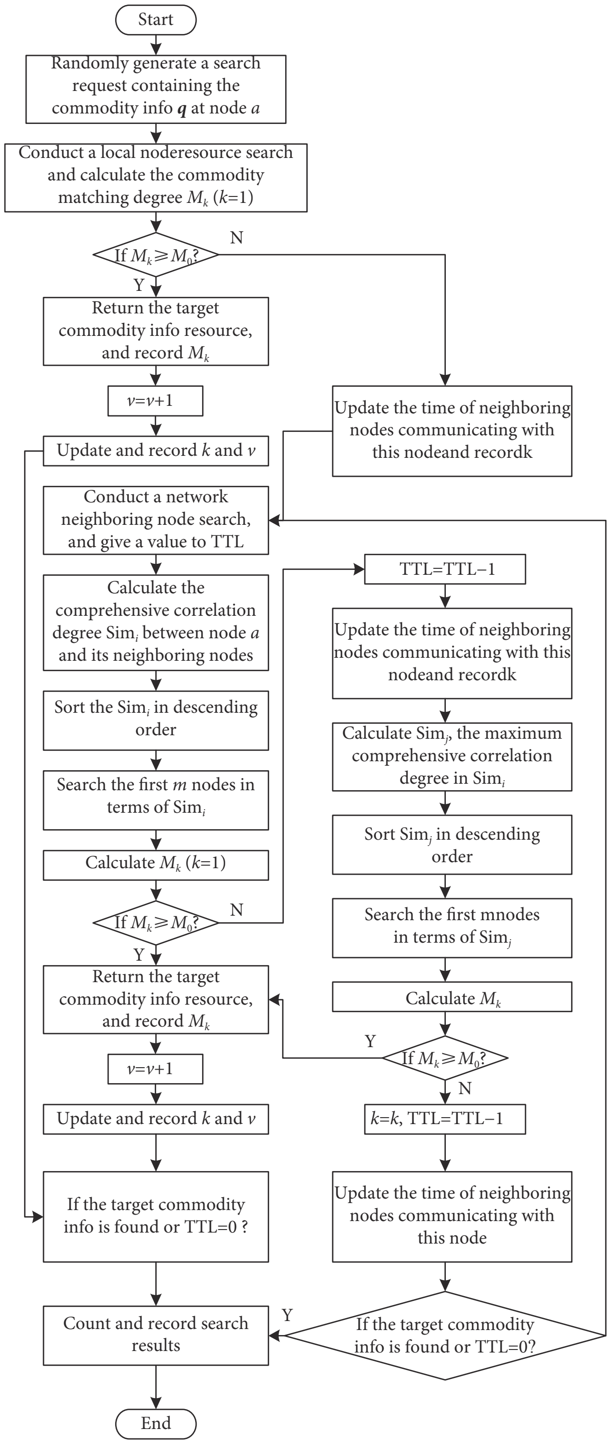

Fig. 2 Flowchart of the simulation process.

3.3.2 Network performance criteria

In the network performance dimension, the average number of message transmission steps and the expansibility of the algorithm are selected.

(1) Average number of message transmission steps

The average number of message transmission steps is the ratio of the total number of message transmission steps to the total number of searches in an experiment, where one message transmission step means one node per search request forwarded. This criterion indicates the network broadband resources the search algorithm takes up during each search. The lower average number of message transmission steps indicates that the search algorithm takes up less network broadband resources during its operation.

(2) Algorithm expansibility

For distributed and unstructured networks, the search algorithm must be able to withstand the increase in the number of nodes in the network. If the search performance is not affected with the increase in the network scale, then the algorithm has good expansibility.

3.4 Parameter setting

Each node in the simulated CIbTN has a unique ID, and the size of the circle of friends for each node is 5. The total number of points in the network is N, and it is set according to the number of nodes demanded in the experiments. Values of the following parameters can be obtained by conducting parameter tuning experiments, as given in Table 4. The parameter configuration of the proposed HBDA is shown in Table 5.

3.5 Analysis of the experimental results

The simulation experiments are conducted by the flooding algorithm, random walk algorithm, and the proposed HBDA. Ten cycles of experiments are conducted using each algorithm, and in each experiment, the search is performed 100 times. The evaluation index for each algorithm is calculated according to the results of each experiment. Then, the average evaluation index of the ten experiments is calculated to obtain the evaluation index of each algorithm, thus increasing the objectivity. The experimental results of the three algorithms are shown in Tables 6–8.

On the basis of the experimental results of each algorithm, the algorithm evaluation indexes are calculated and given in Table 9, including the search success rate, average search time, average commodity matching, average number of search commodity information resources, and average number of message transmission steps.

(1) The results of the search success rate in each cycle are shown in Fig. 3, in which the columns represent the search success rate of each algorithm and the lines represent the average search success rate of 10 cycles for each algorithm. The search success rate of the HBDA is the highest among the three algorithms, with an average of 61%. The search success rate of the flooding algorithm has similar results as the HBDA, reaching 58% on average. The search success rate of the random walk search algorithm is obviously lower than that of the other two algorithms,

Table 4 Parameter configuration of the PeerSim simulation network.

Parameter	Description	Value
N	Number of nodes	10 000
Buyer rate	Ratio of buyer nodes in the network	0.95
Seller rate	Ratio of seller nodes in the network	0.05
Buyer commodity info quantity	Buyer node commodity information quantity	0–10
Seller commodity info quantity	Seller node commodity information quantity	50–100

Table 5 Parameter configuration of the proposed HBDA.

Parameter	Description	Value
α	Coefficient of the attribute correlation degree	0.5
β	Coefficient of the behavioral correlation degree	0.5
m	Number of search forwarding neighboring nodes	3
M_0	Matching degree threshold	0.7
TTL	Time to live	4

Table 6 Experimental results of the flooding algorithm.

Cycle	Total number of searches	Search success count	Average time of each search (ms)	Average commodity matching	Returned commodity information resources	Message transmission steps
1	100	51	5140	0.810	1189	2360
2	100	62	5540	0.802	965	3424
3	100	57	3540	0.799	1201	3091
4	100	57	3280	0.798	1038	3216
5	100	61	2710	0.795	1076	3414
6	100	54	3330	0.795	1002	2953
7	100	63	5960	0.796	988	3309
8	100	55	6780	0.796	1120	3068
9	100	59	4890	0.797	1274	3267
10	100	61	5010	0.798	1035	3390

Table 7 Experimental results of the random walk algorithm.

Cycle	Total number of searches	Search success count	Average time of each search (ms)	Average commodity matching	Returned commodity information resources	Message transmission steps
1	100	23	210	0.803	99	63
2	100	21	320	0.798	304	172
3	100	22	180	0.807	211	218
4	100	17	290	0.803	103	257
5	100	21	370	0.803	87	310
6	100	19	390	0.801	235	324
7	100	18	290	0.800	354	271
8	100	27	310	0.799	201	270
9	100	22	330	0.798	128	291
10	100	17	290	0.798	115	295

Table 8 Experimental results of the proposed hybrid breadth-depth algorithm.

Cycle	Total number of searches	Search success count	Average time of each search (ms)	Average commodity matching	Returned commodity information resources	Message transmission steps
1	100	70	820	0.808	1420	845
2	100	73	730	0.815	1301	1653
3	100	58	1290	0.815	1224	1555
4	100	57	970	0.817	1531	1657
5	100	60	1160	0.816	1464	1626
6	100	55	790	0.817	1358	1512
7	100	64	1440	0.817	1278	1612
8	100	55	1070	0.816	1469	1594
9	100	65	1360	0.816	1328	1475
10	100	53	1020	0.816	1296	1672

with an average of only 20.7%. The HBDA inherits the high search success rate of the flooding algorithm, and the search success rate does not decrease with the reduction of the number of neighboring nodes.

(2) Figure 4 shows the average search time of the algorithms. In the experiments, the average search time of the flooding algorithm is significantly higher than that of the other two algorithms, because of the nature of the algorithm. Its search time is very short because of the randomness of the random walk algorithm. The search time of the proposed HBDA in this paper is also significantly shorter than that of the flooding algorithm. Its search efficiency is about four times higher than that of the flooding algorithm.

(3) The results of the average commodity matching degree are

Table 9 Results of the evaluation index of HBDA, flooding algorithm, and random walk algorithm.

Algorithm	Search success rate (%)	Average time of each search (ms)	Average commodity matching (%)	Average number of searched commodity information resources	Average number of message transmission steps
Flooding	58.00	4618	79.88	19	31.49
Random walk	20.70	298	80.09	9	2.47
HBDA	61.00	1065	81.54	23	15.20

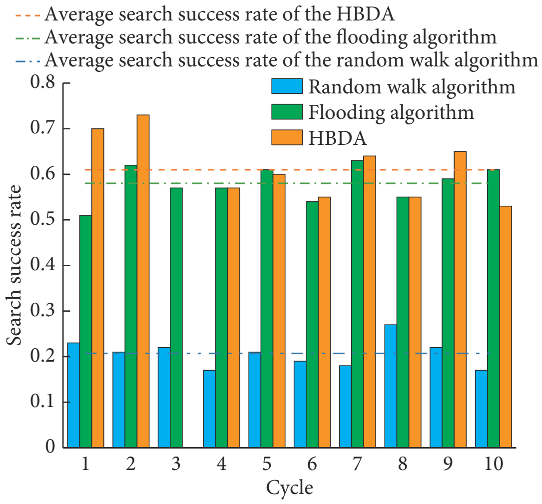

Fig. 3 Search success rate.

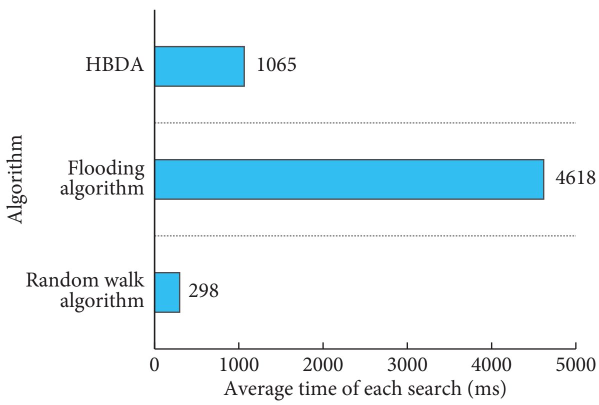

Fig. 4 Average search time.

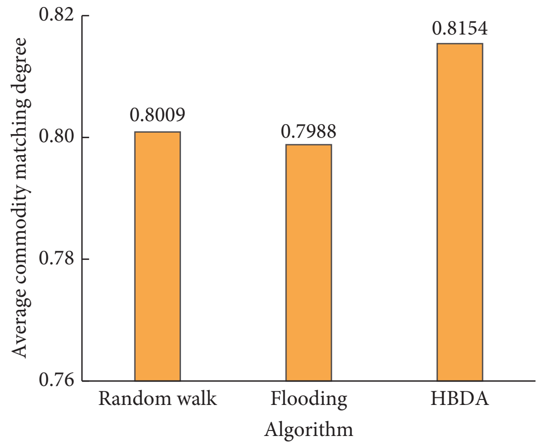

Fig. 5 Average commodity matching degree.

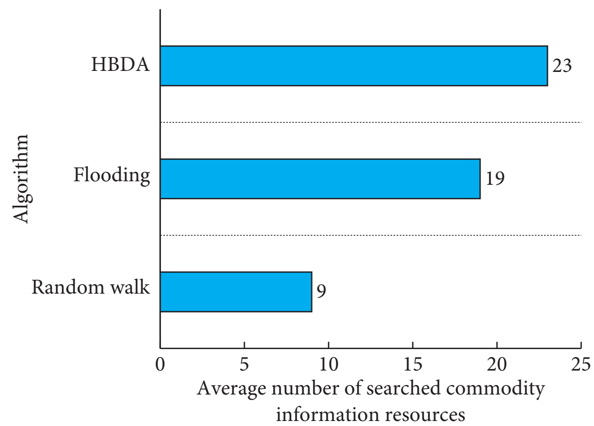

Fig. 6 Average number of searched commodity information resources.

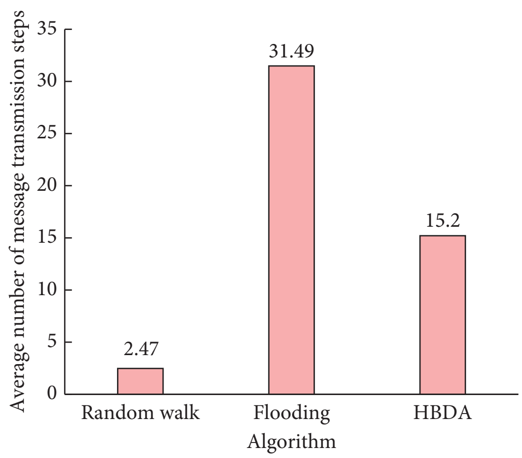

Fig. 7 Average number of message transmission steps.

displayed in Fig. 5. The average matching degrees of the commodities searched by the two blind search algorithms are basically the same, which are nearly 80%. The HBDA increases the commodity matching degree slightly, mainly because the algorithm calculates the attribute and behavioral features of nodes and the keywords of commodity information resources in nodes, and searches among the nodes with similar features. Thus, the commodity information resources are matched to a greater extent, and the requirements of users can be satisfied better.

(4) The average number of searched commodity information resources given by different algorithms is shown in Fig. 6. The HBDA performs better in terms of the number of commodity information resources searched than the two blind search algorithms. Nodes with a high number of commodity information resources and more similar attributes can be searched in priority in the calculation process of the proposed algorithm. Thus, more commodity information results can be found for the user to choose from.

(5) The results of the average number of message transmission steps are given in Fig. 7. The average number of message transmission steps of the search algorithm in a search process represents its consumption of broadband resources in the network. The average transmission steps of the HBDA in one search process is half of that of the flooding algorithm, thus reducing the consumption of broadband resources by about half compared with the flooding algorithm.

(6) To examine whether the performance of the proposed HBDA is susceptible to the network scale, we also conduct the expansibility test experiment, in which we change only the number of network nodes while keeping the rest of the variables unchanged. Figure 8 shows the experimental result.

The result shows that as the network scale enlarges, the search success rate and the average search time of the HBDA fluctuate within a reasonable range rather than show a downward tendency. This finding means that the algorithm has relatively good expansibility, i.e., the search performance of the HBDA will not be affected by the increase in the network scale.

The experimental results show that the proposed HBDA achieves better results in user experience and network performance compared with the two blind search algorithms. The HBDA can achieve a better search success rate and commodity matching degree while using less network broadband resources and searching time. It also has good expansibility, which means that it is suitable to be applied to larger networks. As the number of nodes increases, the scale of the CIbTN continuously enlarges. With its good expansibility, the proposed HBDA will have promising real-life application values in the future.

4 Conclusion

On the basis of the search forwarding logic of the BFSA and DFSA, we proposed the HBDA, a new heuristic algorithm for commodity search in unstructured networks. We defined the concept of node relevance by considering the node's attribute and behavioral features and took advantage of the correlation degree between a node and its neighboring nodes to forward search requests to neighboring nodes with high correlations in priority. As a result, the blindness of the search process, the number of network search message transmissions, and the search time are reduced, and the search success rate is improved.

To verify the search performance of the HBDA, we used PeerSim software to build the CIbTN simulation environment and implemented the HBDA and two blind search algorithms in Java. The experimental results show that the proposed HBDA performs better than the blind search algorithms in terms of search success rate, search time, commodity matching degree, and network consumption. In addition, HBDA has good expansibility. Therefore, it has practical value for performing commodity search tasks in large unstructured networks.

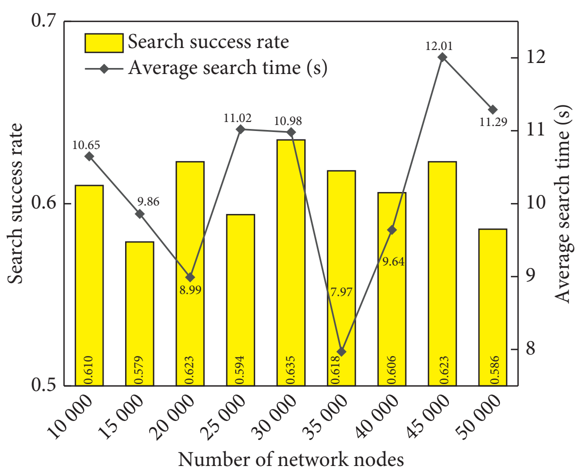

Fig. 8 Experimental results of the expansibility test.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2017YFB1400100).

Dates

Received: 22 March 2022; Revised: 18 May 2022; Accepted: 24 May 2022

References

[1] Y. Chai, C. Miao, B. Sun, Y. Zheng, and Q. Li, Crowd science and engineering: Concept and research framework, *International Journal of Crowd Science*, vol. 1, no. 1, pp. 2–8, 2017.

[2] C. Yu, Y. Chai, and Y. Liu, Literature review on collective intelligence: A crowd science perspective, *International Journal of Crowd Science*, vol. 2, no. 1, pp. 64–73, 2018.

[3] Z. Liu, T. Fang, Y. Dongxin, and N. Kou, Transaction credit in the unstructured crowd transaction network, *International Journal of Crowd Science*, vol. 3, no. 3, pp. 267–283, 2019.

[4] S. M. Thampi and C. Sekaran, Survey of search and replication schemes in unstructured P2P networks, *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 93–131, 2010.

[5] O. Anthony, R. Oluwadoyin, A. Timilehin, and Y. Akinboye, Efficiency analysis of blind tree based search algorithms performance based on time, number of nodes and memory, *International Journal of Scientific & Engineering Research*, vol. 10, no. 7, pp. 663–670, 2019.

[6] Gnutella, http://www.gnutella.org, 2002.

[7] C. Gkantsidis, M. Mihail, and A. Saberi, Random walks in peer-to-peer networks: Algorithms and evaluation, *Performance Evaluation*, vol. 63, no. 3, pp. 241–263, 2006.

[8] J. Sugawara, Proposals and evaluation of modified-BFS using the number of links in P2P networks, *Technical Report of Ieice Ocs*, vol. 104, pp. 5–8, 2005.

[9] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, A local search mechanism for peer-to-peer networks, in *Proc. Eleventh International Conference on Information and Knowledge Management*, McLean, VA, USA, 2002, pp. 300–307.

[10] B. Yang and H. Garcia-Molina, Improving search in peer-to-peer networks, in *Proc. 22nd International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, pp. 5–14.

[11] L. Huang, Research on resource searching in an unstructured P2P network based on dynamic greedy strategy, master dissertation, School of Computer and Information Technology, Beijing Jiaotong University, 2016.

[12] D. M. R. Himali and S. K. Prasad, SPUN: A P2P probabilistic search algorithm based on successful paths in unstructured networks, in *Proc. 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, Anchorage, AK, USA, 2011, pp. 1610–1617.

[13] S. Joseph and T. Hoshiai, Decentralized meta-data strategies: Effective peer-to-peer search, *IEICE Transactions on Communications*, vol. 86, no. 6, pp. 1740–1753, 2003.

[14] D. Tang, M. He, and Q. Meng, Research on searching in unstructured P2P network based on power-law distribution and small world character, *Journal of Computer Research and Development*, vol. 44, no. 9, pp. 1566–1571, 2007.

[15] N. Leibowitz, M. Ripeanu, and A. Wierzbicki, Deconstructing the Kazaa network, in *Proc. Third IEEE Workshop on Internet Applications*, San Jose, CA, USA, 2003, pp. 112–120.

[16] H. Shen, Y. Shu, and B. Yu, Efficient semantic-based content search in P2P network, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 813–826, 2004.

[17] L. Xiao, Z. Zhuang, and Y. Liu, Dynamic layer management in superpeer architectures, *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 11, pp. 1078–1091, 2005.