axioms

MDPI

Article

Collaborative Search Model for Lost-Link Borrowers Information Based on Multi-Agent Q-Learning

Ge You [1], Hao Guo [2,*], Abd Alwahed Dagestani [3,*] and Ibrahim Alnafrah [4]

[1] School of Literature and Media, Nanfang College Guangzhou, Guangzhou 510970, China; youg@nfu.edu.cn
[2] School of Management, Wuhan Textile University, Wuhan 430200, China
[3] School of Business, Central South University, Changsha 410083, China
[4] Graduate School of Economics and Management, Ural Federal University, Yekaterinburg 620002, Russia; ibrahimnafrah@gmail.com
* Correspondence: hguo@wtu.edu.cn (H.G.); a.a.dagestani@csu.edu.cn (A.A.D.)

Abstract: To reduce the economic losses caused by debt evasion amongst lost-link borrowers (LBs) and improve the efficiency of finding information on LBs, this paper focuses on the cross-platform information collaborative search optimization problem for LBs. Given the limitations of platform/system heterogeneity, data type diversity, and the complexity of collaborative control in cross-platform information search for LBs, a collaborative search model for LBs' information based on multi-agent technology is proposed. Additionally, a multi-agent *Q*-learning algorithm for the collaborative scheduling of multi-search subtasks is designed. We use the *Q*-learning algorithm based on function approximation to update the description model of the LBs. The multi-agent collaborative search problem is transformed into a reinforcement learning problem by defining search states, search actions, and reward functions. The results indicate that: (i) this model greatly improves the comprehensiveness and accuracy of the search for key information of LBs compared with traditional search engines; (ii) during searching for the information of LBs, the agent is more inclined to search on platforms and data types with larger environmental rewards, and the multi-agent *Q*-learning algorithm has a stronger ability to acquire information value than the transition probability matrix algorithm and the probability statistical algorithm for the same number of searches; (iii) the optimal search times of the multi-agent *Q*-learning algorithm are between 14 and 100. Users can flexibly set the number of searches within this range. It is significant for improving the efficiency of finding key information related to LBs.

Keywords: lost-link borrowers; multi-agent; collaborative search; cross-platform; multi-source data

MSC: 68T05; 68T37

Citation: You, G.; Guo, H.; Dagestani, A.A.; Alnafrah, I. Collaborative Search Model for Lost-Link Borrowers Information Based on Multi-Agent Q-Learning. *Axioms* **2023**, *12*, 1033. https://doi.org/10.3390/axioms12111033

Academic Editors: Jun Huang, Yueyuan Zhang, Yuan Sun and Yankai Li

Received: 2 August 2023
Revised: 23 October 2023
Accepted: 1 November 2023
Published: 3 November 2023

1. Introduction

In recent years, incidents of loss of connection and runaways from loan customers have occurred frequently, which causes great economic losses for the country and investors. Finding lost-link borrowers (LBs) and recovering overdue loans has become an urgent problem for credit institutions. The core issue of loan recovery is the focus of LBs. When a borrower chooses to run away or lose contact, timely access to the information of the LB becomes the key to finding their whereabouts. However, the information of LBs is usually distributed on multiple internet platforms/systems, which include social platforms (e.g., WeChat, Weibo, QQ, and TikTok), E-commerce platforms (e.g., Shopping, car rental, matchmaking), major bank apps, government apps, peer-to-peer (P2P) lending platforms and so on. Meanwhile, it comprises numerous data types. Thus, it is an interesting and important research topic to efficiently obtain valuable information about LBs from complex data on multiple platforms.

The key to solving the problem of searching for LB information on multiple platforms is to achieve cross-platform and multi-source information searching. The method of collaborative searching is usually used to deal with the problem of cross-platform and multi-source information searching [1]. In this method, the problem of multi-source data processing is solved by multi-task unit cooperative control. In detail, the system determines the search objective according to the user's needs, formulates the corresponding search subtasks by considering the environmental states of each platform, and then initiates distributed searches on multiple platforms simultaneously. Each search result is processed during the search process, and the descriptive characteristics of the search objective are updated synchronously. However, the following three difficulties arise in cross-platform information collaborative searching: (1) The information on LBs is usually scattered on multiple internet platforms. These platforms are often heterogeneous, meaning that a single search task cannot run simultaneously among multiple platforms. Formulating specific search subtasks for each platform according to search requirements is necessary, and then performing a distributed collaborative search. (2) The information of the LBs is on multiple internet platforms, and each platform has multiple data types, such as text, picture, audio, and video. The diversity of platform data types determines the diversity of data types for LBs. (3) To achieve cross-platform information collaborative searching, it is necessary to coordinate multiple searching subtasks and data-analyzing subtasks globally to determine when and what data filtering conditions are used on each platform. Determining when to process which type of data on which platform in the search is a complex process, and a multi-task unit coordinated control mechanism needs to be developed for coordinated control.

Multi-agent technology is an artificial intelligence technology that uses multiple agents to form an organic whole that cooperate to complete a task together [2,3]. Multi-agent technology effectively addresses dynamic and complex real-time problems. In recent years, multi-agent technology has been used to solve the problem of collaborative searching for multi-task units. For example, a multi-agent collaborative "infotaxis" strategy is presented, which uses the relative entropy of the system to synthesize a suitable search strategy for the team [4]. Vasile's work combines the fundamental heuristics underneath monotonic basin hopping within the general scheme of multi-agent collaborative searching. The basic idea is that the local search performed by each agent in a multi-agent collaborative search can be substituted with an iteration of basin hopping [5]. Kim et al. presented a collaborative web agent designed to enable across-user collaboration in web searching and recommendation [6]. In the work of Birukou et al., a multi-agent recommendation system called "Implicit" has been developed, which supports web searches for groups of people or communities [7]. Shimoji and Sakama considered a problem wherein multiple agents search for target objects in a field, and presented an experimental study on collaborative searching by distributed autonomous agents [8]. To synthesize the spatio-temporal sensing capabilities of a group of agents and optimize the search time, a collaborative "infotaxis" strategy has been proposed by extending the single-agent infotaxis to a multi-agent system [9]. According to the study by Perez-Crespo et al. [10], a metasearch engine based on software agents was developed for collaborative contexts. The metasearch engine allows group members to share a web search process. To address the problem under real-time, multi-source, and data-rich conditions, a new multi-source information search model based on multi-agent collaboration is put forward in the study [11]. Vasile et al. present an overview of multi-agent collaborative searching (MACS) for multi-objective optimization, and analyzed different heuristic local searches [12]. Koval et al. presented a novel collaborative approach for exploring and covering multi-agent coordination in unknown complex indoor environments [13].

In multi-agent collaborative searching, multi-agents can make optimal or near-optimal scheduling decisions using reinforcement learning [14,15]. Scheduling multi-agents using *Q*-learning has been studied in some works in the literature. For instance, an ensemble imitation learning multi-trick multi-agent deep deterministic policy gradient (EILMMA-

DDPG) has been developed, and the proposed algorithm complies with an ensemble imitation learning policy [16]. Zhou et al. proposed a novel learning architecture that consists of several learning modules. Every learning module includes a *Q*-learning module and an ASPL (Action Selection Priority Level) module to determine the action selection priority level [17]. Sethi et al. presented a novel deep reinforcement learning-based IDS that employs deep *Q*-network logic in multiple distributed agents and uses attention mechanisms to detect and classify advanced network attacks efficiently [18]. Asghari and Sohrabi combined the coral reefs optimization algorithm and multi-agent deep *Q*-network to reduce the energy consumption of data centers and cloud resources using the dynamic voltage and frequency scaling (DVFS) technique [19]. Moreover, a deep *Q*-learning (DQL) algorithm has been proposed. It is based on a cooperative learning strategy in which all agents perceive a common reward, and thus learn cooperatively and distributively to improve the resource allocation solution through offline training [20]. Messaoud et al. proposed a deep-federated *Q*-learning (DFQL) framework. The simulation results show that the DFQL framework performed more efficiently than traditional approaches [21]. Dou et al. proposed a fast-scene adaptive reinforcement learning (FSACL) algorithm. Compared with the traditional cooperative *Q*-learning (CL) and independent *Q*-learning (IL) algorithms, the FSACL algorithm can obtain a somewhat larger system capacity with less power [22]. Chen et al. presented a novel deep reinforcement learning-based algorithm that combines a graphic convolution neural network with a deep *Q*-network to form an innovative graphic convolution *Q* network that serves as the information fusion module and decision processor [23]. Tampuu et al. designed an independent deep *Q*-learning network (DQN) for each agent in the environment, enabling the agent to learn its strategy independently and maximize the overall reward through the *Q* value [24]. Daeichian and Haghani introduced a multi-agent traffic light adjustment method based on traffic conditions. Fuzzy *Q*-learning and game theory are used to develop strategies based on the previous experience and decisions of neighboring agents [25]. Leng et al. proposed a multi-agent reward iterative fuzzy *Q*-learning (RIFQ) method for multi-agent cooperative tasks. It provides a feasible reward relationship for multi-agents, which makes the training of multi-agents more stable [26].

This work focuses on the solution for cross-platform and multi-source information collaborative searching for LBs based on multi-agent technology. To address these issues, a cross-platform information collaborative search model framework based on multi-agent technology is proposed, and a multi-agent collaborative search control method based on the *Q*-learning algorithm is designed. It provides an effective solution for the multi-source information collaborative search of LBs in the cross-platform environment.

Our contribution lies in our study of the method of searching for LB information through the application of multi-agent *Q*-learning, which expands the application of multi-agent technology to solve the collaborative problem of multiple search subtasks for LBs information, and achieves a cross-platform collaborative search of LBs information. Furthermore, a multi-agent *Q*-learning algorithm based on function approximation is developed to update the description model of the LBs in this paper. In the feedback loop search process, the description model of the LBs is continuously improved, and the comprehensiveness of the multi-source information acquisition of the LBs is improved.

The rest of the paper is organized as follows. Section 2 describes the research problem. Section 3 formulates the multi-agent collaborative search model. Section 4 provides the details of the proposed multi-agent *Q*-learning algorithm. Section 5 presents the simulation results. Section 6 concludes the paper. The acronyms appearing in this paper are presented in Appendix A.

2. Problem Description

The LBs studied in this paper refer to runaway borrowers with incomplete loan procedures, no guarantees, no tracking and supervision of the project after the loan and a low default cost of borrowing, and who shut down their communication devices, such as

mobile phones, in order to avoid repayment [27]. In previous studies, the family mobile social contact big data network and historical daily consumption transaction network records are often adopted to analyze tracking information of LBs. For instance, Pang et al. proposed an information-matching model and multi-angle tracking algorithm for LBs based on a family mobile social-contact big data network and achieved information matching for LBs [28]. Based on the daily consumption transaction network of LBs, a network sorting search rule and algorithm is proposed to track LBs in different address types [29]. However, the above-mentioned studies regarding tracking the information of LBs relied on complete household mobile phone records and consumption records, which had to be historical data from within the last six months; otherwise, the prediction results would be inaccurate. The household mobile phone records and consumption record data of LBs are scattered on multiple data platforms, and complete historical record data are difficult to obtain. Hence, to improve the efficiency and integrity of information searching for LBs, this paper proposes using multi-agent technology to address the problem of cross-platform information collaborative searching for LBs. The research involves the following two brand new concepts:

Definition 1: *Cross-platform information collaborative searching means that the system determines the search objective according to the user's needs and makes corresponding search subtasks based on the environmental status of each platform/system. It then launches distributed searches on multiple platforms/systems simultaneously, and processes each search result during the search process. Moreover, the description characteristics of the search objective are updated synchronously*.

Definition 2: *Cross-platform information collaborative searching for LBs refers to the process of conducting a collaborative search for LBs across multiple platforms. The search process needs to formulate search subtasks for each platform/system wherein the material data of the LBs are located according to the search objective. Although each of the search subtasks independently performs information screening and analysis tasks on their respective platforms, they are closely related*.

The user's search needs to drive the collaborative search process for the LBs across multiple platforms. The specific search process is shown in Figure 1. The search work steps are as follows.

Step 1: The user assigns the task of searching for the name, birthplace, age, ethnicity, telephone number, contact person, contact time, address, and other information of the LBs as needed.

Step 2: The system formulates search subtasks for each platform/system according to the total search task. Specifically, the system formulates the search subtasks of contact information on the telecommunication platform. The system formulates the search subtasks of social contacts on the social platform. The system develops address information search subtasks on the e-commerce platform. The system develops basic information search subtasks such as name, birthplace, age, ethnicity, etc., in the government information platform. The system formulates loan information search subtasks on the P2P platform, and each platform searches for and collects information according to the assigned search subtasks.

Step 3: Since multiple data types exist on each platform/system, the system formulates multiple data analysis subtasks according to the data type. According to the data type, these can be subdivided into text, audio, video, and other data analysis subtasks. A description model of the LBs with a unified data type is formulated to describe the results of the data analysis.

Step 4: Storing the result of data analysis—that is, the descriptive information of the unified data type of LBs—in the storage unit.

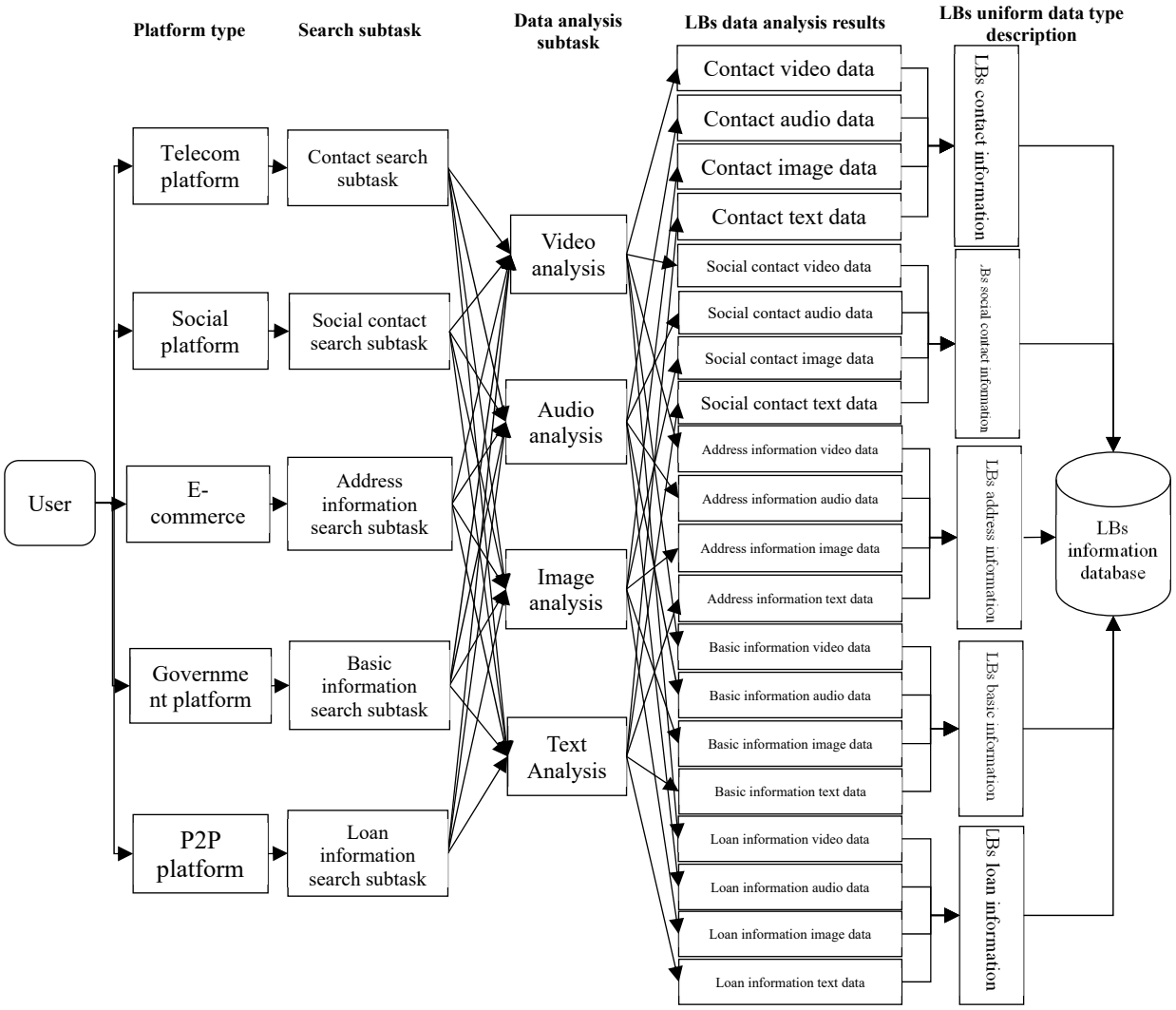


Figure 1. The workflow of the cross-platform collaborative search for LBs' information.

3. Multi-Agent Collaborative Search Model for LBs Information

3.1. Model Framework

A multi-agent model based on reinforcement learning is proposed to achieve the collaborative searching of cross-platform information for LBs. This model employs multiple agents to process information from different platforms and data types, and integrates multiple results as the final description of the LBs. The multi-agent collaborative search model structure for LBs is composed of six main modules: collaborative control agent, data collection agent group, data analysis agent group, descriptive model of LBs, object knowledge base of LBs, and historical information database of LBs. The specific structure of the multi-agent collaborative search model is as follows:

(1) Cooperative control agent—This is set according to the information search objective, description model, and object knowledge base of the LBs determined by the user. The system formulates the search tasks and dynamically deploys the agents in the data analysis agent group to complete the search and data analysis subtasks in different platforms/systems;

(2) Data collection agent group—This consists of the multiple data collection agents responsible for collecting data about LBs from different internet platforms (e.g., telecommunications platforms, social platforms, e-commerce platforms, government information platforms, and P2P platforms);

(3) Data analysis agent group—Multiple data analysis agents are responsible for completing data screening and analysis subtasks on different platforms/systems. The collaborative agent determines each agent's specific tasks according to the total search task. These data analysis agents have different functions and can process data types in parallel;

(4) Descriptive model of LBs—When searching for LBs' information, it is necessary to determine the characteristics of the search object. Therefore, an object description model is used to represent the information model. The collection of searched LBs' information in this paper can be described as name, hometown, age, ethnicity, phone number, contact person, contact time, and address. During the information collaborative search process, the descriptive model of LBs can be gradually updated to include additional information such as academic qualifications, Weibo account, Weibo content, and more;

(5) Object knowledge base of LBs—This refers to storing knowledge about LBs that is summarized based on historical data. This knowledge is mainly used as a reference basis to help the collaborative control agent determine the distribution strategy of search subtasks in each platform and the filter conditions for each search subtask. The knowledge base also guides the material data collection agent in collecting relevant data. In the process of information collaborative searching, as the agent group analyzes the data, new knowledge is continuously obtained, and the knowledge base of LBs can be continuously updated;

(6) Historical information database of LBs—This is responsible for storing the data of LBs collected by the material collection agent and provides the historical data in the database to the agent group for data analysis. The descriptive feature model and object knowledge base of the LBs are updated according to the analysis of the results.

The process of multi-agent collaborative information search is shown in Figure 2. The detailed steps are as follows:

Step 1—The user draws up the objective content for searching for the information of LBs, and assigns the search task to the collaborative control agent;

Step 2—The collaborative control agent initializes the description model of LBs according to the content of the search task. For example, if the search task involves finding the address information of LBs, the collaborative control agent initializes the address information in the description model of the LBs;

Step 3—The collaborative control agent reaches the data collection agent on each platform under the filtering conditions of the data of LBs according to the content of the search task. Using the description model and the characteristics of the LBs in the knowledge base, the collaborative control agent that formulates and assigns search subtasks is formulated and assigned to each agent in the data analysis agent group;

Step 4—The material data collection agent of the LBs collects material data on multiple platforms/systems, and stores the collected material data in the historical material database of LBs;

Step 5—The data analysis agent searches and analyzes the material data of the LBs in the historical material database. The data analysis agent mines valuable information about LBs, completing the data analysis tasks. The information value of LBs is judged by the cumulative sum of environmental rewards when previous predictions are correct. It notifies the collaborative control agent, stores the search results in the LBs' knowledge base, and updates the object description model of the LBs;

Step 6—The cooperative control agent assesses the completion of the search task. If all agents meet the conditions for the end of the search task, the search task is concluded. Otherwise, we go to step 2 until the requirement is met.

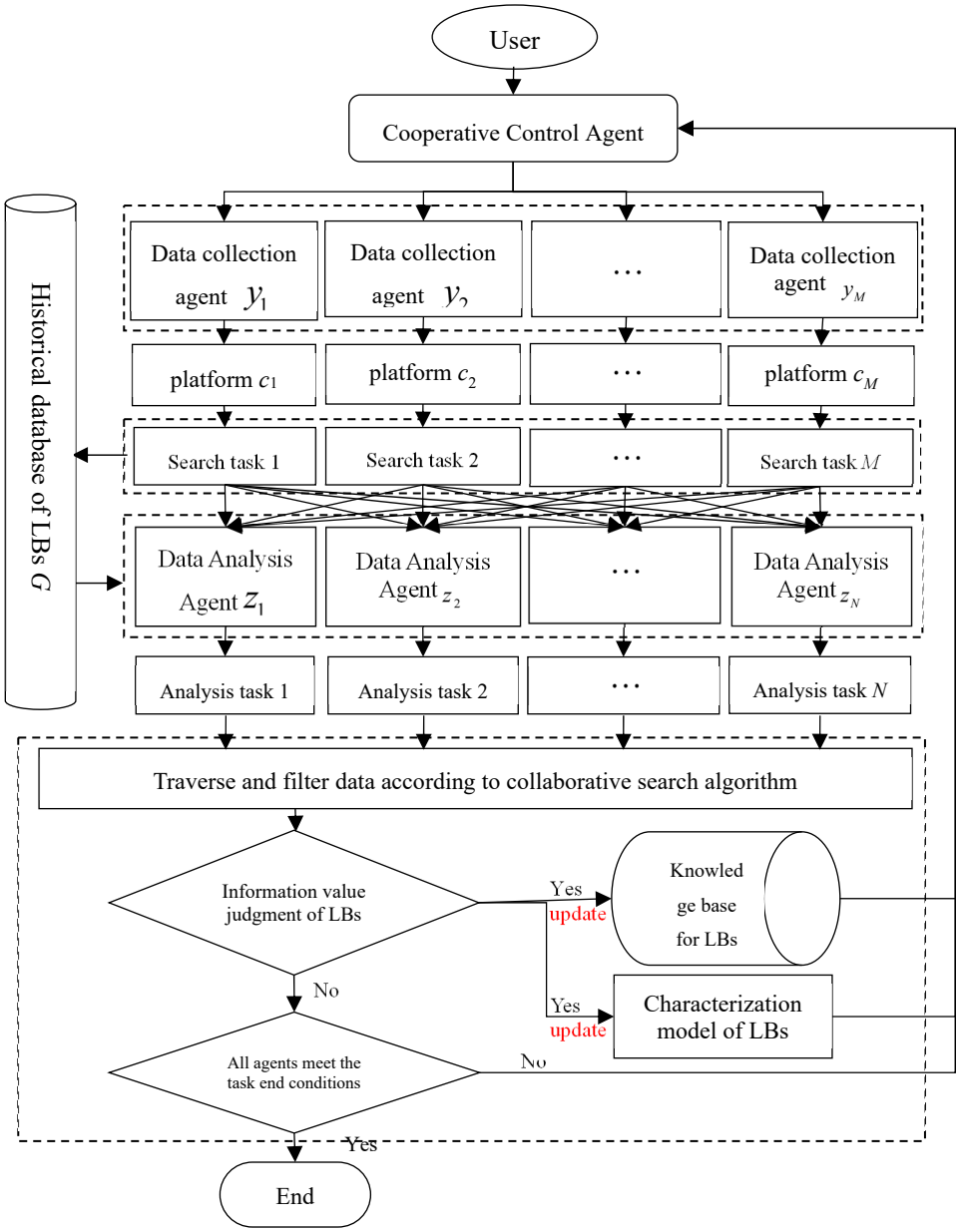


Figure 2. Workflow of multi-agent collaborative information search model.

3.2. Model Mathematical Description

Assumptions are made as follows to facilitate the description of the collaborative search model for LB information.

Assumption 1: *The data of LBs are distributed on multiple internet platforms/systems.* $C = \{c_1, c_2, \cdots, c_M\}$ *represents the collection of these M platforms,* c_i *represents the ith platform, and* $1 \leq i \leq M$.

Assumption 2: *There are N types of data concerning LBs information.* $D = \{d_1, d_2, \cdots, d_N\}$ *represents the collection of data types of LBs,* d_j *represents the jth data type, and* $1 \leq j \leq N$.

Assumption 3: *Historical records of LB information on various platforms and in different data types are found, and the collection of historical LB information is* $G = \{g_1, g_2, \cdots, g_K\}$, *where* $g_1, g_2, \cdots, g_K$ *represent the diachronic record of the information of LBs. Each diachronic record* g_u *contains two attributes, that is,* $g_u = \left\{{g_u}^c, {g_u}^d\right\}$, $1 \leq u \leq K$, ${g_u}^c$ *represents the platform where*

the μth record appears, and ${g_u}^d$ *represents the data type of the occurrence of the μth record, where* ${g_u}^c \in C$, ${g_u}^d \in D$.

Assumption 4: *The description model of the LBs at time t is the collection* $L_t = \{l_{1t}, l_{2t}, \cdots, l_{Wt}\}$, *where* $l_{1t}, l_{2t}, \cdots, l_{Wt}$ *is the attribute of the LBs at time t (e.g., name, hometown, age, nationality, phone number, contact person, contact time, address). The knowledge base is represented by* I_t.

Rule 1: *Since the data information of the LBs is distributed on M internet platforms, M data collection agents are required to collect data in each platform. We assume that the data collection agent set is* $A = \{y_1, y_2, \cdots, y_M\}$, *and the task is scheduled for the agent* y_i *to perform data collection in the platform* c_i, *where* $1 \leq i \leq M$. *Then, the corresponding relationship between the agent and the platform is* $y_1 \to c_1, \cdots, y_M \to c_M$.

Rule 2: *Since the LBs' information has n data types, n data analysis agents must collect data in each platform. We assume that the data analysis agent set is* $B = \{z_1, z_2, \cdots, z_N\}$, *and the task is arranged for the agent* z_j *to analyze data type* d_j, *where* $1 \leq j \leq N$. *Then the corresponding relationship between the agent and the platform is* $z_1 \to d_1, z_2 \to d_2, \cdots, z_N \to d_N$.

Rule 3: *Combining Rule 1 and Rule 2, this paper can use a task chain* $y_i \to c_i \to z_j \to d_j$ *to represent a data collection agent* y_i *that collects data from the platform* c_i. *Then, the data analysis agent* z_j *performs data analysis on data type* d_j. *Then, M data collection agents and N data analysis agents will combine* $M \times N$ *task chains*.

To describe the task status of the multi-agent at time t, this paper uses the grid model for illustration. We assume that six data types on six platforms are used to search for information on LBs, as shown in Figure 3:

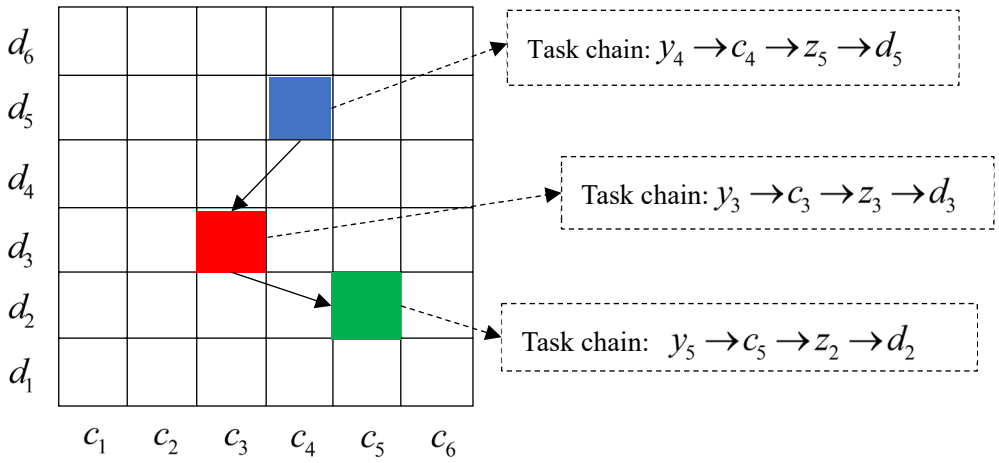


Figure 3. Grid model of multi-agent collaborative information search task status.

The blue area in Figure 3 indicates that the multi-agent at time t analyzes the data type b_5 on the platform c_4. According to rule 3, the task chain corresponding to the blue area is $y_4 \to c_4 \to z_5 \to d_5$. Similarly, the task chains corresponding to the task states of the multi-agents in the red and green areas are $y_3 \to c_3 \to z_3 \to d_3$, $y_5 \to c_5 \to z_2 \to d_2$.

The historical record collection $G = \{g_1, g_2, \cdots, g_K\}$ of the LB information $\mathrm{cal}(G)$ represents the value of the historical information of LBs for each data type in each platform. $\mathrm{cal}(G) \to (c_i, d_j)$ shows that the information value of the LBs in the platform c_i and data type d_j is the highest as regards the calculation result. Then, the search process of the multi-agent at time t can be expressed as:

$$IF\, \mathrm{cal}(G) \to (c_i, d_j)\ THEN\ y_i \to c_i \to z_j \to d_j\ \mathrm{update}(L_t, I_t) \tag{1}$$

where update(L_t, I_t) refers to updating the description model and knowledge base of the LBs at time t. The description model and knowledge base of the LBs at time $t+1$ are the results of the update at time t $(L_{t+1}, I_{t+1}) =$ update(L_t, I_t). Equation (1) indicates that if the highest historical information value of LBs is on platform c_i and data type d_j at time t, the data collection agent y_i will collect data on platform c_i. Further, data analysis agent z_j will perform data analysis on data type d_j. Finally, according to the analysis results, the description model and knowledge base of the LBs are updated.

In the multi-agent collaborative search process shown in Figure 3, the task state jumps from the blue area to the red area, and then from the red area to the green area. The whole process can be described as:

IF cal$(G) \rightarrow (c_4, d_5)$ $THEN$ $y_4 \rightarrow c_4 \rightarrow z_5 \rightarrow d_5$ update(L_t, I_t) Next
IF cal$(G) \rightarrow (c_3, d_3)$ $THEN$ $y_3 \rightarrow c_3 \rightarrow z_3 \rightarrow d_3$ update(L_{t+1}, I_{t+1}) Next
IF cal$(G) \rightarrow (c_5, d_2)$ $THEN$ $y_5 \rightarrow c_5 \rightarrow z_2 \rightarrow d_2$ update(L_{t+2}, I_{t+2}) Next

In summary, the key to achieving a multi-agent collaborative search is to judge the task status of the multi-agent at the next time $t+1$ based on the historical information record and knowledge base of the LBs at time t. The following proposes a multi-agent Q-learning collaborative search algorithm for this problem.

4. Multi-Agent Q-Learning Collaborative Search Algorithm

4.1. Motivations for Q-Learning Algorithm

As mentioned in Section 1, "Introduction", Q-learning is widely used to solve the joint scheduling problem of multiple search subtasks. The following presents the reasons why Q-learning is chosen as the methodology in this paper.

Firstly, in practice, information on LBs is often scattered across multiple internet platforms. For the search of LBs information, the multi-agent needs to perform search subtasks on each platform, and then conduct the collaborative search, while the Q-learning algorithm is popularly used in multi-agent control and scheduling. By properly defining the search action, search state, and return function, the multi-agent collaborative search problem is formulated as a Q-learning problem.

Secondly, in the process of collaborative searching, the multi-agent needs to decide the next search behavior within a relatively reasonable time, and search for as much valuable LB information as possible. The Q-learning algorithm can be used to train agents to make decisions without knowing the environmental model, and to update the description model of the LBs based on function approximation. It can to some extent solve the decision problem when the multi-agent processes which platform/system and which type of data to use.

Thirdly, some researchers who have focused on a similar problem to this paper have demonstrated that the Q-learning algorithm offer potential advantages in solving finite Markov decision processes (MDP). The Q-learning algorithm can ensure convergence to an optimal strategy. For instance, Gulzar et al. [3] and Matignon et al. [30] indicated an outstanding solution could be obtained by applying a Q-learning algorithm to multi-agent collaborative search and control. In this context, this paper leverages and extends their insights to design the Q-learning algorithm discussed here.

4.2. Algorithm Designing

The multi-agent Q-learning algorithm is a reinforcement learning method whereby the agent interacts with the environment to obtain rewards, influencing its future actions. The collaborative search process using multi-agent Q-learning is modeled as a Markov decision process [31,32]. The agent aims to maximize future rewards by making decisions based on the current internal state, external state, and fixed state transition probability of the environment, thereby obtaining immediate rewards. In detail, at each time step t, the controller observes the agent's current search state, denoted as s_t, the action taken by the agent, denoted as a_t, and the information value of the LBs obtained from the search $R(s_t, a_t)$, and makes the system transition to the next search state s_{t+1}; the transition probability is

$P(s_t, a_t)$. For the convenience of calculation, assuming that the occurrence probability of the LBs' data in each platform/system and data type conforms to the normal distribution, then $P(s_t, a_t)$ obeys the normal distribution.

According to the Bellman Equation [33,34], given a search strategy π, we define Q as the search state s_t, and search action a_t and the expectation of the reward discount sum of the subsequent time steps in strategy π. The implementation of the Q-learning method is as follows: At each time step t, we observe the current search state s_t, and select and execute the search action a_t. After observing the subsequent search status s_{t+1}, the value of the LBs information obtained is $R(s_t, a_t)$. According to the algorithmic rules proposed by Watkins [35], in the collaborative search process, the maximum expectation of the cumulative discount of the information value of the LBs that the agent can obtain in the next search action is expressed as Equation (2):

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha[R(s_t, a_t) + \gamma \max P(s_t, a_t)Q_t(s_{t+1}, a_{t+1})] \quad (2)$$

where α represents the learning rate of the agent's search action, $\alpha \in [0, 1]$. γ represents the discount factor, $\gamma \in [0, 1]$, which describes the impact of the search time on the value of the information of the LBs. $Q^{\pi}(s_t, a_t)$ represents the expectation that the agent performs the search action a_t and the subsequent strategy π when searching for the state s_t to obtain the cumulative discount of the information value of the LBs.

Q-learning aims to estimate the Q value under the optimal strategy when the probability and the reward obtained are unknown. Therefore, to facilitate the calculation, let $Q^*(s, a)$ be the maximum expectation of the agent obtaining the discount of the information value of the LBs under the optimal search strategy. The value of the accumulated information of the LB is denoted as $V^{\pi}(s) = Q^{\pi}(s, a)$. According to the mapping of the search strategy and the state action, the optimal search strategy is found; that is $\pi : s \to a$. We then select the search action when changing the search state of the agent in turn to maximize the sum of the rewards obtained, and the optimal search strategy can be obtained as follows:

$$V^*(s) = Q^*(s, a) = \max Q(s, a) \quad (3)$$

The Q value will gradually approach the optimal strategy $Q^*(s, a)$ by iteratively updating the Q value that repeatedly performs actions.

The three elements of state, behavior, and reward function are the core of constructing a multi-agent Q-learning process. In this paper, the system state refers to the current agent search state, action refers to the search action of each agent, and the reward function represents the value of the agent searching to obtain the information of the LBs. The following will explain the agent's search state, search actions, and environmental reward.

4.2.1. Search State

The search state refers to the state the current agent's searches for the LBs' information in each type of data and platform/system. The division of the search state space is the basis for the agent to select collaborative search actions reasonably. In this paper, the search state S of the system characterizes the situation in which the agent searches for information about LBs. Because the data of LBs are distributed in M platforms and N types of data, the search state of the system can be directly represented by the data source and data type $s(c_i, d_j)$ of the LBs discovered by the agent, $i \in [1, M]$, $j \in [1, N]$. The agent has $M \times N$ search states, and each state can jump to each other. The search state collection of the system is expressed by Equation (4):

$$S = \left\{ s(c_i, d_j) \middle| i = 1, 2, 3, \cdots, M; j = 1, 2, 3, \cdots, N; \right\} \quad (4)$$

where $s(c_i, d_i)$ means that the agent searched for valuable information about the LBs in the d_j data type in the platform c_i; assuming $M = 6$ and $N = 6$, the system has 36 search states. As shown in Figure 4, each small square represents the possible search status of the system.

If the current search state of the system is a small blue square $s(c_2, d_3)$, it represents that the agent has searched for the information of the LBs in the data type d_3 in the platform c_2.

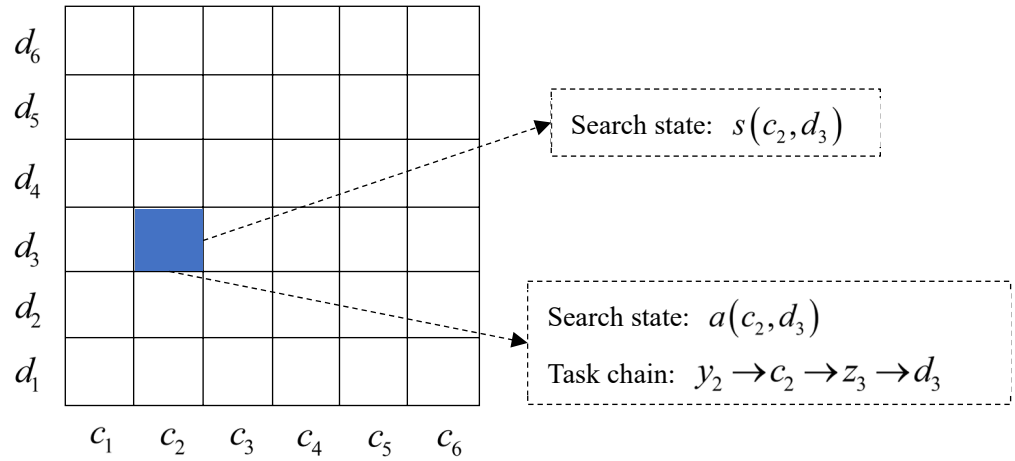


Figure 4. Multi-agent collaborative search state and action.

4.2.2. Search Action

The search action refers to the action of the agent jumping from the current search state to the next search state. $a(c_i, d_i)$ represents that the agent executes data of type d_j in platform c_i. Since this paper assumes that any search state can be arbitrarily jumped to during the collaborative information search process and there are $M \times N$ states in the search state space, the system executes action $a(c_i, d_i)$, and there are $M \times N$ kinds of search actions. Then, the search action set of the system is expressed as Equation (5):

$$A = \{a(c_i, d_j) | i = 1, 2, 3, \cdots, M; j = 1, 2, 3, \cdots, N; \} \tag{5}$$

Assuming $M = 6$ and $N = 6$, the system has 36 search actions in total. The small blue square in Figure 4 can perform 36 search actions. When the agent executes the search action $a(c_2, d_3)$, it represents the agent executing the information search task of LBs in the data type d_2 and the platform c_2. According to rule 3, the search action $a(c_2, d_3)$ corresponds to the task chain $y_2 \rightarrow c_2 \rightarrow z_3 \rightarrow d_3$. That is, the data collection agent y_2 collects data in the platform c_2. Then, the data analysis agent z_3 performs data analysis on the data type d_3.

4.2.3. Environmental Reward

The environmental reward $R(s_t, a_t)$ represents the value of the agent's search action a_t in the search state s_t, and the value of the LB's information obtained by searching in the corresponding platform/system and data type. The larger the environmental reward, the greater the value of the information of the LBs found by the agent.

The setting of the environmental reward can be obtained by evaluating the importance of the data source and data type based on the historical record of the LBs information. For example, the LB is a person who often posts various comments on Weibo, so it is easier for the agent to obtain the valuable information of LBs when processing the text data from the Weibo platform. In other words, when the agent processes the text data from the Weibo platform, it obtains a higher environmental reward. Meanwhile, it can also determine the number of object information items obtained in the data source and data type per unit of time. In this paper, the latter is chosen as the basis for formulating the environmental reward. In detail, according to the historical record of LBs $G = \{g_1, g_2, \cdots, g_K\}$, $1 \leq u \leq K$.

$g(c_i, d_j)$ represents the number of historical records of LBs in platform c_i and data type d_j, then $\sum g(c_i, d_j) = K$. The environmental reward function is expressed as Equation (6):

$$R(s_t, a_t) = \begin{cases} 1 & 0 \le g(c_i, d_j) \le K/(M \times N) \\ 2 & K/(M \times N) < g(c_i, d_j) \le K/M \\ 3 & K/M < g(c_i, d_j) \le K \end{cases} \quad (6)$$

where $g(c_i, d_j)$ represents the number of historical records of LBs in the platform c_i data type d_j. When $g(c_i, d_j) \le K/(M \times N)$, the information value of LBs obtained by the agent in the data type d_j and the platform c_i is 1. Similarly, when $K/(M \times N) < g(c_i, d_j) \le K/M$, the information value of LBs obtained by the agent in the data type d_j and the platform c_i is 2. When $K/M < g(c_i, d_j) \le K$, the information value of LBs obtained by the agent in the data type d_j and the platform c_i is 3. Figure 5 is an example of the environmental reward that the agent executes for the d_j type of data in the platform c_i in the search state s_t with $M = 6$ and $N = 6$.

	c_1	c_2	c_3	c_4	c_5	c_6
d_6	1	2	2	1	2	2
d_5	3	3	3	2	1	3
d_4	2	2	1	3	1	1
d_3	3	2	1	3	1	2
d_2	2	1	2	1	2	3
d_1	1	2	3	2	3	2

Figure 5. Example of environmental reward function.

4.3. Algorithm Steps

In the process of multi-agent collaborative searching, the next search state of the agent is only determined by its current search state, which conforms to the characteristics of the Markov process. The algorithm steps are designed according to the characteristics of the object conforming to the Markov process. Moreover, a two-dimensional table Q is adopted to store the Q value [30,36]. The detailed steps of the algorithm are shown in Figure 6.

The detailed steps of the multi-agent Q-learning collaborative search algorithm are as follows:

Step 1—Initialize the search state of the multi-agent in the system, and let the search state of the agent be (s^-, s^0, s^+), where s^0 represents the state in which the agent recently discovered the information of the LBs. s^- represents the previous search state s^0. s^+ represents the next search state s^0. At the initial moment when multi-agents conduct a collaborative search, the search action a_{t_0} from the action set is selected to search;

Step 2—Initialize the Q table, $Q(i, j, k) = 0$, $(1 \le i, j, k \le M \times N)$. Initialize learning rate α and discount rate γ. Initialize the environmental reward function. In order to facilitate the simulation, it is assumed that the historical record of LBs manually sets the environmental reward function. In practice, it can be obtained according to actual data statistics. Initial transition probability $P(s_t, a_t)$ and $P(s_t, a_t)$ conform to the normal distribution;

Step 3—When the LBs with new information appear in the record $g = \{g^s, g^d\}$, update $s^+ = s(g^c, g^d)$, and meanwhile, update the Q table. According to Equation (5), the next step update formula is $Q(s^-, s^0, s^+) = (1 - \alpha)Q(s^-, s^0, s^+) + \alpha[r(s^+) + \gamma \max Q(s^-, s^0, s^+)P(s_t, a_t)]$;

Step 4—Search states within the system are updated, where $\begin{cases} s^- = s^0 \\ s^0 = s^+ \end{cases}$;

Step 5—Predict the next search action a_t of the agent through the Markov decision process. Then, $s^+ = \max\limits_{1 \le i \le M \times N} \left(\sum\limits_{1 \le j \le M \times N} Q(j, s^0, i) \right)$;

Step 6: The system agent performs the next search action, and processes the data $\{c, d\}$ corresponding to the state s^+. The corresponding equation is:

$$\begin{matrix} c = \begin{cases} \frac{s^+}{M} - \left\lfloor \frac{s+}{M} \right\rfloor \times M & \frac{s^+}{M} - \left\lfloor \frac{s+}{M} \right\rfloor \times M \neq 0 \\ M & \frac{s^+}{M} - \left\lfloor \frac{s+}{M} \right\rfloor \times M = 0 \end{cases} \\ d = \left\lfloor \frac{s+}{M} \right\rfloor \end{matrix} \tag{7}$$

where $\lfloor \rfloor$ represents rounding down, waiting for a new record of LBs information;

Step 7—Repeat steps 3 to 6 until the Q matrix value converges.

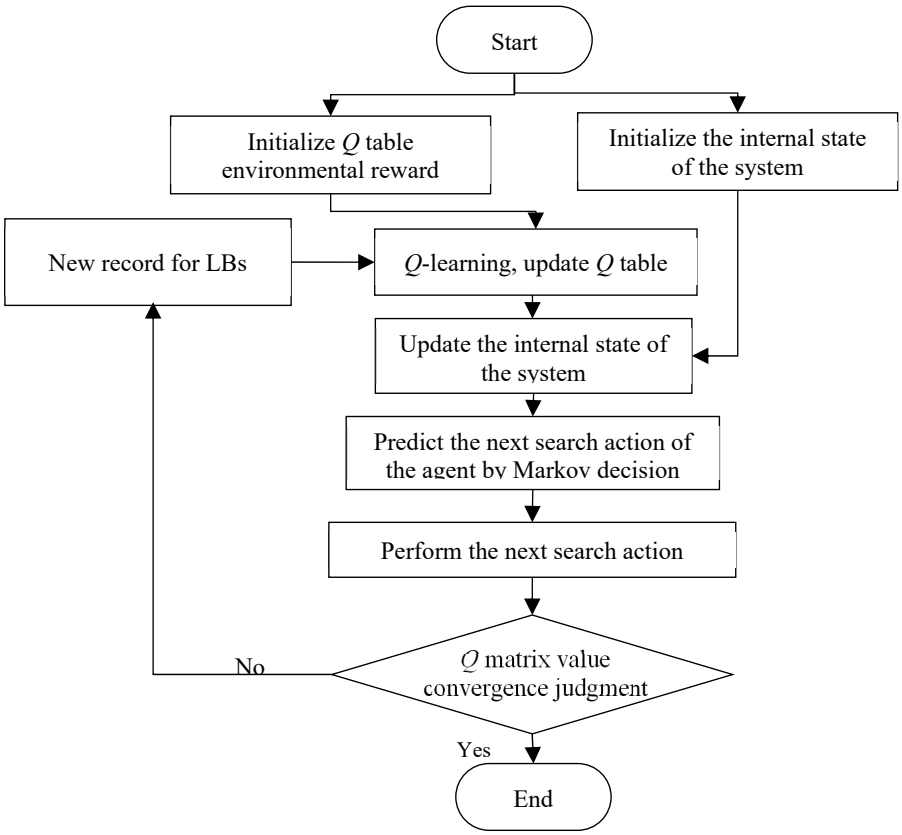


Figure 6. Flowchart for multi-agent Q-learning collaborative search algorithm.

The pseudo-code of the multi-agent Q-learning algorithm is briefly introduced in Figure 7.

Input: Reward function R; Set of actions A; Start state s_0; Discount factor γ; Learning rate α
Output: Strategy π
Begin:
1: Choose episode size epi_size
2: Create an initial replay memory G to capacity N
3: Create an initial action-value function Q =0
4: Create an initialize transition probability P randomly
5: **for** (episode =1 to epi_size)
6: Create an initial sequence s_1={x_1}
7: **for** (t = 1 to T)
8: With probability P select a random action a_t
9: Choose a_t in emulator and observe reward R_t and image x_t+1
10: Set $s_{t+1} = s_t, a_t, x_{t+1}$.
11: update $Q(s^-, s^0, s^+) = (1-\alpha)Q(s^-, s^0, s^+) + \alpha\left[r(s^+) + \gamma \max Q(s^-, s^0, s^+)P(s_t, a_t)\right]$
12 Calculate Q matrix value in action a_t
13: Every steps reset $Q = Q$
14: **End**
15 Return to line 6
16: **End**
17: Until the Q matrix value converges

Figure 7. The pseudo-code of the multi-agent Q-learning algorithm.

4.4. Algorithm Complexity Analysis

When analyzing the algorithm steps, it can be seen that the calculation burden of the algorithm is mainly concentrated in step 5. The maximum value of the Q matrix is found when predicting the next action through the Markov decision process. Since the agent has $M \times N$ search states and $M \times N$ search actions, the complexity of Markov decision is $T(n) = (M \times N) \times (M \times N) = O(M^2N^2)$. Therefore, the total complexity of the algorithm is $O(M^2N^2)$, and it can be seen that the complexity of the algorithm is at the polynomial level, which can meet the needs of real-time processing.

5. Simulation Analysis

In order to verify the performance of the multi-agent collaborative search algorithm, this paper simulates the algorithm. All simulations were implemented using Matlab 2014 on a Windows PC (AMD A10-9600P RADEON R5, 10 COMPUTE CORES 4C + 6G 2.40 GHz; RAM: 4.00 GB DDR; OS: Windows 10). The algorithm verification process is shown in Figure 8. The record generation module generates Markov object appearance records. The information value and information acquisition rate statistics module calculate the value and information acquisition efficiency of LBs information obtained by different algorithms according to the environmental reward function. To evaluate the algorithm's performance, two performance indicators are defined: information value and information acquisition rate. (i) Information value refers to the cumulative sum of environmental rewards when previous predictions are correct. (ii) Information search rate refers to the ratio of the information value obtained by the algorithm to the number of searches. This paper compares the collaborative search performances of the Q-learning algorithm, the transition probability matrix algorithm, and the probability statistical algorithm. The transition probability matrix algorithm uses object appearance records to count the transition probability matrix of the object between states and determine the next action according to the maximum transition probability value of the current state. The probabilistic statistical algorithm uses the historical appearance record of the object in each state to determine the probability of the object appearing in that state. The next action is to dispatch a data processing agent to the data platform and data type corresponding to the state with the highest probability of occurrence.

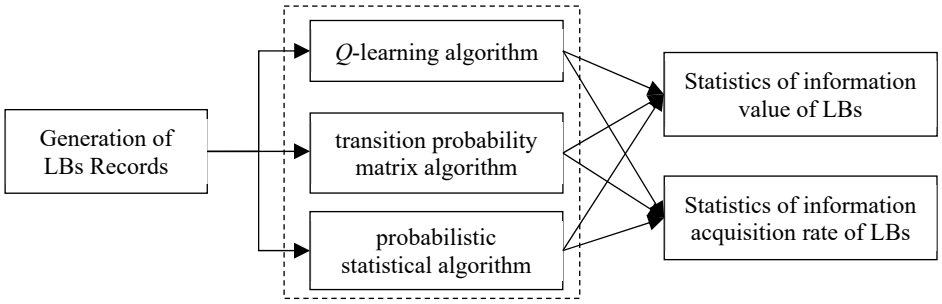


Figure 8. Algorithm verification process.

The parameters for the following experiment are: $M = 6$, $N = 6$, $K = 95$, $\alpha = 0.1$, $\gamma = 0.8$. Initial Q matrix value $E_Q = 0$, the occurrence probability of LBs information in each platform and data type conforms to the normal distribution. The detailed parameter settings are shown in Table 1, and then the position where the information of the LBs appears is generated and a step action is predicted.

Table 1. The initial value of each parameter.

Parameter	The Initial Value of Each Parameter
Data sources of LBs	$M = 6$, $N = 6$, $K = 36$, initial Q matrix value $E_Q = 0$, the occurrence probability of the LBs information in each data source and data type conforms to the normal distribution.
The historical record of the information of the LBs in each platform and data type	$g(c_1,d_1) = 17$, $g(c_1,d_2) = 3$, $g(c_1,d_3) = 0$, $g(c_1,d_4) = 0$, $g(c_1,d_5) = 1$, $g(c_1,d_6) = 1$, $g(c_2,d_1) = 2$, $g(c_2,d_2) = 4$, $g(c_2,d_3) = 4$, $g(c_2,d_4) = 1$, $g(c_2,d_5) = 1$, $g(c_2,d_6) = 0$, $g(c_3,d_1) = 1$, $g(c_3,d_2) = 10$, $g(c_3,d_3) = 0$, $g(c_3,d_4) = 3$, $g(c_3,d_5) = 3$, $g(c_3,d_6) = 2$, $g(c_4,d_1) = 6$, $g(c_4,d_2) = 1$, $g(c_4,d_3) = 1$, $g(c_4,d_4) = 0$, $g(c_4,d_5) = 3$, $g(c_4,d_6) = 0$, $g(c_5,d_1) = 0$, $g(c_5,d_2) = 1$, $g(c_5,d_3) = 2$, $g(c_5,d_4) = 0$, $g(c_5,d_5) = 18$, $g(c_5,d_6) = 0$, $g(c_6,d_1) = 0$, $g(c_6,d_2) = 4$, $g(c_6,d_3) = 2$, $g(c_6,d_4) = 0$, $g(c_6,d_5) = 1$, $g(c_6,d_6) = 3$
Other parameters	$\alpha = 0.1$, $\gamma = 0.8$

According to Table 1, the amount of information appearing on various platforms and data types of LBs is counted. According to Equation (6), the environmental reward of multi-agent collaborative searching is calculated as shown in Figure 9a. To illustrate the impact of the environmental reward, the environmental rewards in Figure 9b are all set to 1, as the experimental reference group.

1	2	1	1	1	2
1	1	1	1	3	1
2	1	1	1	2	1
1	2	1	2	2	1
1	2	2	1	1	1
3	2	1	1	1	1

(**a**)

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

(**b**)

Figure 9. The settings of the two environmental reward functions. (**a**) is a setting with different environmental reward functions, and (**b**) is a setting with same environmental reward functions.

5.1. The Impact of Environmental Reward Function on Search Trajectory

In order to explore the impact of the environmental reward on the search trajectory of multi-agents, two sets of experiments are set up according to the two environmental rewards given in Figure 9, and the number of searches is set to 10. The calculated multi-agent search trajectory is shown in Figure 10. Figure 10a is a multi-agent search trajectory drawn using the environmental reward of Figure 9a. Figure 10b is a multi-agent search trajectory drawn using the environmental reward of Figure 9b.

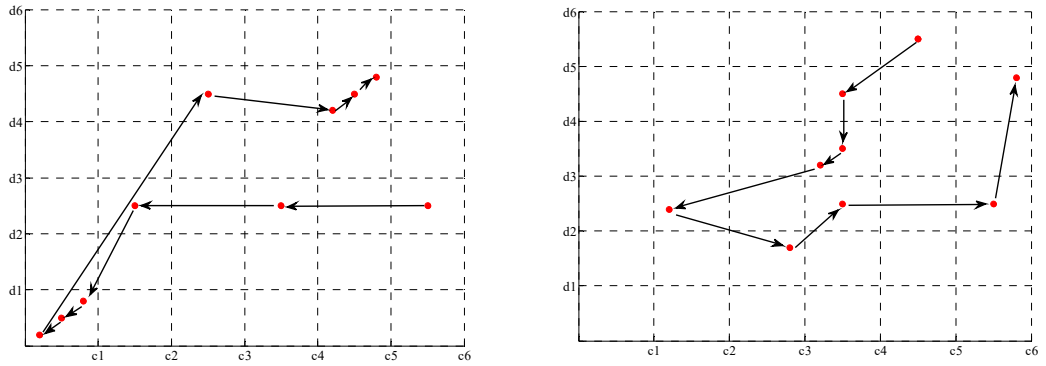


Figure 10. Multi-agent collaborative search trajectory. (**a**) is a multi-agent search trajectory drawn using the environmental reward of Figure 9a, (**b**) is a multi-agent search trajectory drawn using the environmental reward of Figure 9b.

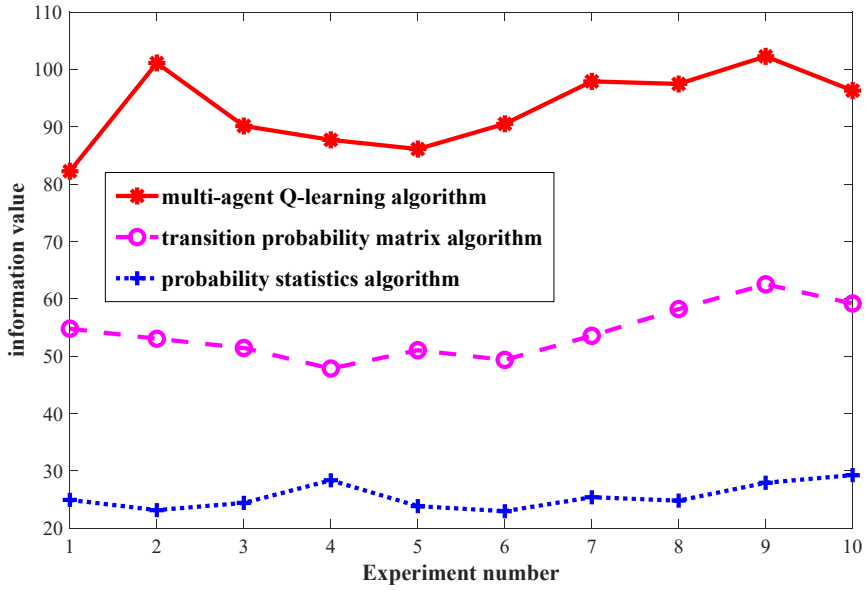
It can be seen from Figure 10a that the multi-agent randomly selects some areas to search at the beginning. With subsequent Q-list learning, the agent concentrates on searching in certain areas. The agent searches more in the platform c_1 data type d_1 and the platform c_5 data type d_5, accounting for 60% of the total search times. Moreover, the final search area of the multi-agent converges to platform c_5 and data type d_5. The corresponding task chain is $y_5 \rightarrow c_5 \rightarrow z_5 \rightarrow d_5$. Figure 10b shows that the multi-agent has been searching randomly selected areas and has not concentrated on a certain area to search. Compared with Figure 9a, it can be seen that the environmental reward in the platform c_1 data type d_1 and the platform c_5 data type d_5 are both at the maximum value of 3. The environmental rewards of all regions in Figure 9b are the same, so the environmental reward has a greater impact on the selection of the multi-agent search area. In the search process, multi-agents tend to search on platforms and data types with larger environmental reward functions.

5.2. The Impact of Environmental Reward on Algorithm Performance

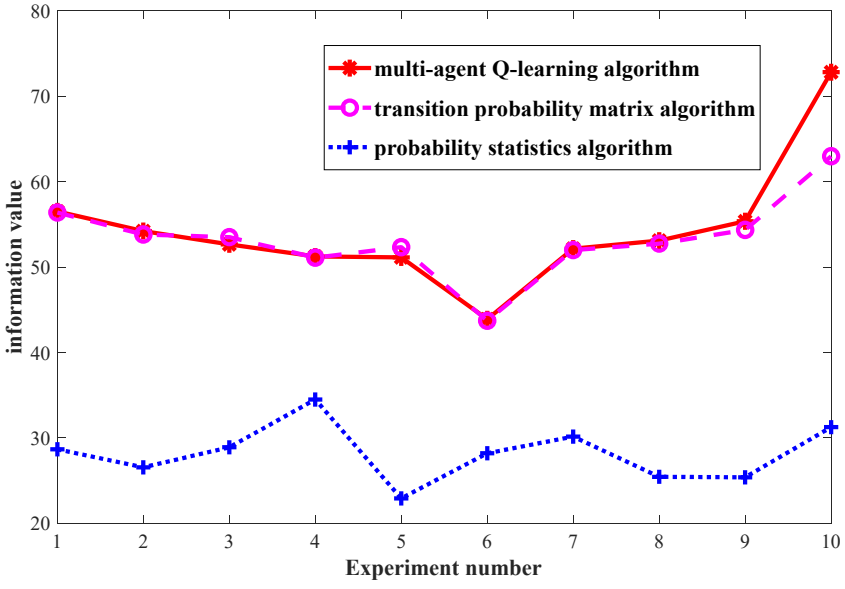
The environmental reward is one of the main differences between the Q-learning algorithm and the transition probability algorithm in this paper. To illustrate the impact of the environmental rewards, experiments are carried out according to the two environmental reward function settings given in Figure 9. Figure 11a is the information value calculated through the environmental reward function of Figure 9a. Figure 11b is the information value calculated through the environmental reward function of Figure 9b. Each method generates and predicts 1000 times and obtains statistics of the information value of LBs. Each experiment gives 10 experimental results. The specific statistical results are shown in Figure 11.

In Figure 11a, in the case of the same number of searches, the ability of the Q-learning algorithm in this paper to obtain the information value of the LBs is stronger than those of the transition probability matrix algorithm and the probability statistics algorithm. This is because this paper divides the Q-learning algorithm's environmental reward function into three levels to emphasize its advantages, leading to a 60% to 70% improvement in the overall performance of the transition probability matrix algorithm. This demonstrates the

significant impact of the environmental reward function in this paper on the algorithm's performance. In Figure 11b, the experimental results show that the performances of the *Q*-learning search algorithm and the transition probability matrix algorithm are the same, but both are significantly better than the probability statistics algorithm. This is because when all environmental reward function values are set to 1, the *Q*-learning algorithm in this paper degenerates into a transition probability matrix algorithm. Essentially, both algorithms make predictions for the next action based on the probability transition matrix of the Markov model. The slight difference in the individual experimental results in Figure 11b is caused by the difference in the amount of data required for the first training of the two. The *Q*-learning algorithm needs three records to generate the first *Q* value, while the transition probability matrix algorithm only needs two records.



(**a**)



(**b**)

Figure 11. Comparison of the impact of environmental reward function on algorithm performance. (**a**) is the information value calculated through the environmental reward function of Figure 9a, and (**b**) is the information value calculated through the environmental reward function of Figure 9b.

5.3. Algorithm Performance Analysis

Figures 12a and 12b, respectively, show the convergence trend of the three algorithms' information value and information acquisition rate. Figure 12a is a real-time statistic of the of information acquisition value by three algorithms for LBs. One experiment is carried out, and the number of searches is 1000. Figure 12b shows the real-time statistics of the three algorithms in terms of the information acquisition rate of LBs. One experiment was carried out, and the number of searches was 1000. The small graph in Figure 12a is a trend graph of the information acquisition rate of the first 100 searches.

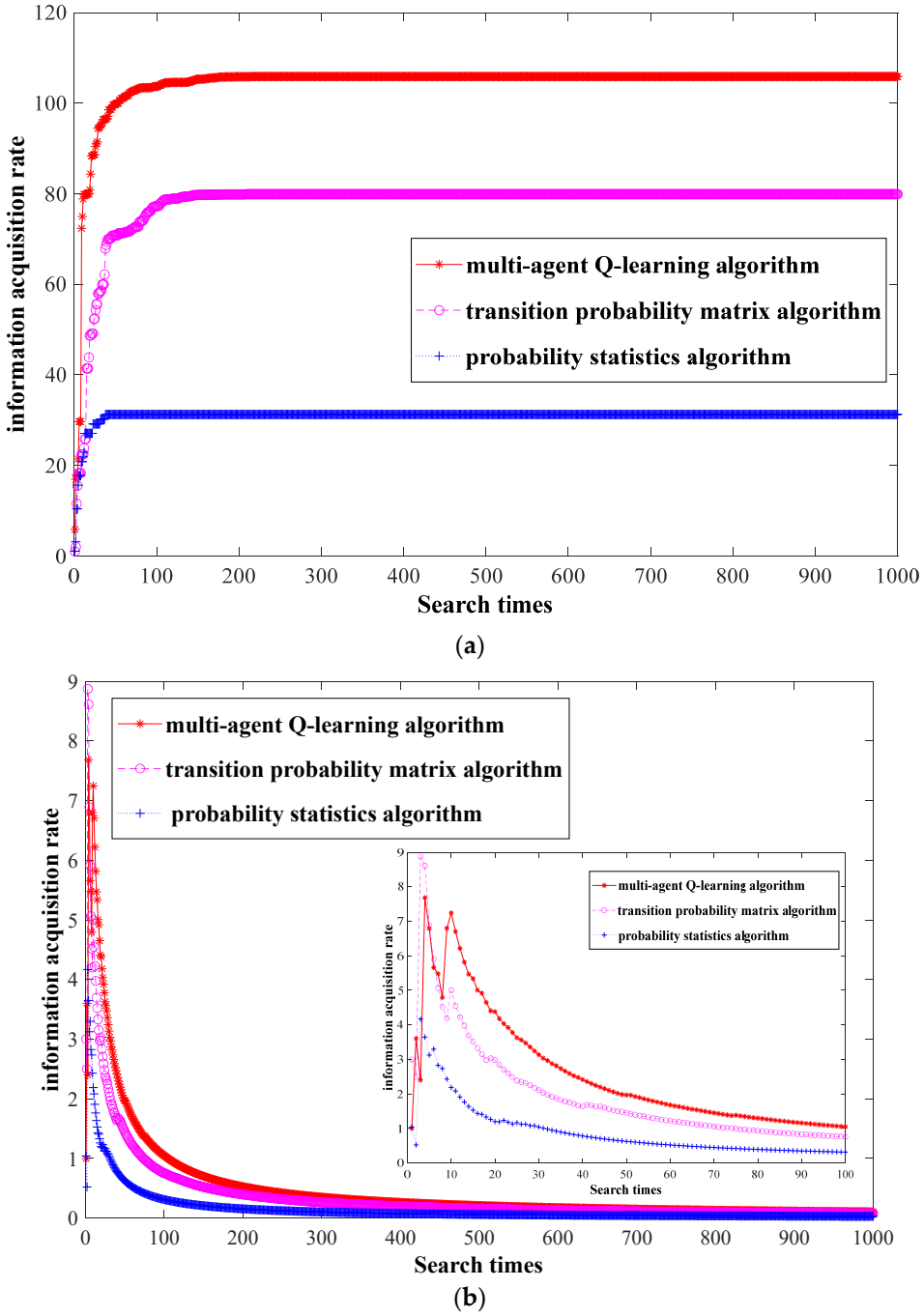


Figure 12. Algorithm convergence comparison. (**a**) is a real-time statistic of the information acquisition value by three algorithms for LBs, and (**b**) is the real-time statistics of the three algorithms in terms of the information acquisition rate for LBs.

It can be seen from Figure 12a that as the number of searches increases, the capacity of the three algorithms to obtain information about LBs gradually increases and finally shows a trend of convergence. The *Q*-learning algorithm achieves the highest value in terms of obtaining information about missing borrowers, and its convergence speed is faster than those of the transition probability matrix algorithm and the probability statistics algorithm. When the number of searches is about 50, the value of algorithmic information acquisition reaches about 90, and it continues to grow after that. When the number of searches is about 100, the growth rate slows down and stabilizes. In addition, the performances of the two algorithms after convergence decrease in the order of transition probability matrix algorithm and probability statistical algorithm. It can be seen from Figure 12b that the information acquisition rates of the three algorithms all increase stepwise first, then gradually decrease, and finally show a trend of convergence. When the number of searches is less than 14, the information acquisition rate of the algorithm fluctuates sharply as the number of searches increases. When the number of searches is greater than 14, the information acquisition rate of the algorithm continues to decrease. After the number of searches reaches about 100, the rate of decline gradually slows down and finally shows a trend of convergence. This change occurs because, at the beginning of the search, the information value of the LBs obtained by the unit of search times is more valuable. However, as the number of searches increases, the value of the information obtained by the number of searches per unit of LBs gradually decreases and eventually converges to 0, but it is always greater than 0. The other two algorithms have the same inflection point. The information acquisition rate of the two algorithms decreases in the order of transition probability matrix algorithm and probability statistical algorithm after convergence. Therefore, when the number of searches is greater than 15, the performance of the *Q*-learning algorithm in this paper is better than those of the other two algorithms.

This article also compares the trends of the average running times of the three algorithms. It can be seen from Figure 13 that the times taken by the *Q*-learning algorithm, transition matrix algorithm, and probability statistics algorithm in this paper are not much different with the same number of searches, and the running time gradually increases with the increase in the number of searches.

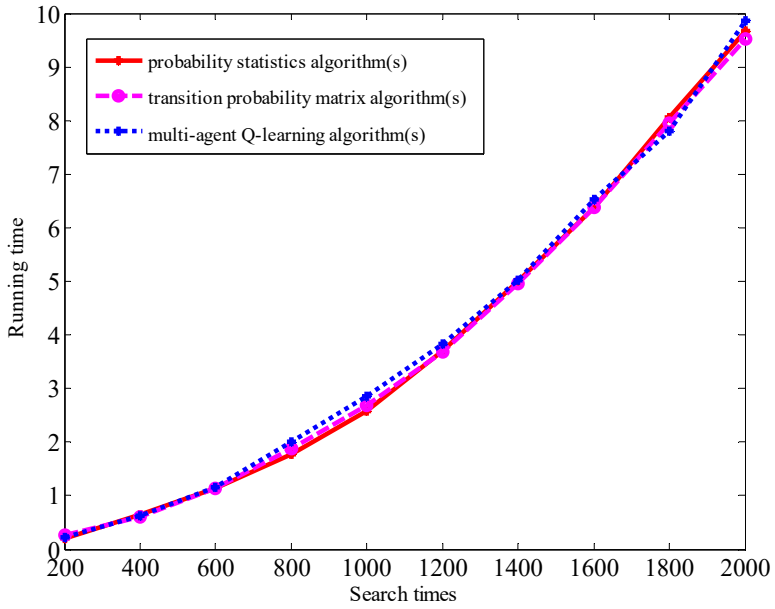


Figure 13. The change trend of the average running times of the three algorithms.

This paper also compares the running times of three algorithms across 1000 searches (See Table 2). We performed 10 sets of experiments for each algorithm, set the number of searches for each set to 1000, and counted the average running times of the 10 sets

of experiments. The statistical results show that the average running time of the three algorithms is about 2.6 s, which shows that the search speeds of the three algorithms are almost the same. However, in the same running time, the *Q*-learning algorithm's value and LBs information acquisition rate are significantly higher than those of the transition probability matrix algorithm and the probability statistical algorithm. This shows that the algorithm in this paper has a better performance advantage in terms of controlling multi-agent information collaborative searches.

Table 2. Running time statistics of each algorithm (1000 search times).

Algorithm Type	1	2	3	4	5	6	7	8	9	10	Average Time/s
Q-learning algorithm	2.787	2.685	2.759	2.563	2.558	2.640	2.601	2.708	2.701	2.654	*2.6656*
Transition probability matrix algorithm	2.984	2.518	2.563	2.541	2.820	2.756	2.679	2.569	2.702	2.584	*2.6716*
Probability statistical algorithm	2.630	2.675	2.530	2.615	2.561	2.637	2.644	2.609	2.635	2.656	*2.6192*

Through the above-mentioned experimental verification, it has been found that the characteristics of the multi-agent *Q*-learning algorithm are as follows: (1) When searching for the information of LBs, the multi-agent *Q*-learning algorithm has a stronger ability to acquire information than the transition probability matrix algorithm and the probability statistical algorithm for the same number of searches. (2) The information value acquisition rate of the multi-agent *Q*-learning algorithm fluctuates sharply with the increase in the number of searches, and then shows a marginal decreasing trend. When the number of searches is greater than 14, the algorithm's information acquisition rate begins to decline. After the number of searches reaches about 100 (100 is also the inflection point of the convergence of the value of the algorithm in obtaining information), the rate of decline gradually slows down, and finally converges. Therefore, users can set the optimal number of searches to between 14 and 100 according to their needs when searching for LBs information.

6. Conclusions and Future Research

This paper proposes a collaborative search model for LBs information based on multi-agent *Q*-learning to address cross-platform collaborative searches in a multi-source and diverse data environment. A multi-agent *Q*-learning collaborative search algorithm has been designed to coordinate multiple search subtasks. The *Q*-learning algorithm, based on function approximation, was used to update the descriptive model of LBs. By reasonably defining search actions, search states, and reward functions, the problem of the collaborative control of multiple search subtasks was here transformed into a problem of *Q*-learning, achieving a cross-platform multi-source information collaborative search. The conclusions are as follows:

(1) Compared with traditional search engines, this model focuses on LBs. In the feedback loop search process, the descriptive model of the LBs can be continuously improved, and the information of the LBs can be obtained from multi-source data. This greatly improves the comprehensiveness and accuracy of the search for key information regarding LBs;

(2) In the search process, multi-agents are more inclined to search on platforms and data types with larger environmental rewards. In other words, multi-agents are more inclined to perform search tasks on platforms and data types that have greater information value for LBs;

(3) The multi-agent *Q*-learning algorithm that designs the environmental reward can significantly improve the efficiency of information searching for LBs. Furthermore, it can acquire information value more easily than the transition probability matrix and probability statistical algorithms;

(4) The simulation results show that the optimal search times of the multi-agent *Q*-learning algorithm are between 14 and 100. Users can flexibly set the number of searches to within this range when searching for LBs information. This is significant for improving the efficiency when searching for key information about LBs.

The limitations and valuable topics are discussed as follows. Firstly, although the multi-agent collaborative information search model can improve the intelligence, comprehensiveness, and flexibility of the LBs' information search, there is a lack of timeliness due to the method used for the online collection of elements and offline cycle analysis. Besides this, the model does not analyze the difference in the processing power of the agent group in detail. In particular, the multi-agent *Q*-learning collaborative search algorithm proposed in this study regards the learning process of the agent as a Markov decision process. In other words, the decision of the agent depends only on the current state of the environment, so if there is a temporal correlation between the states, then the learning effect is not good. It might be useful to consider the epsilon-greedy technique as a part of the *Q*-learning strategy, which could be studied in the future. Additionally, other mechanisms (e.g., how to improve social awareness by adding some entities like the Twitter API, etc.) portrayed by Kumar et al. [37] and principles of machine learning [38] are also suggested to extend the current studies. The possible future directions for research in this area include exploring a multi-agent collaborative information search model considering the improvement of the environmental reward function and the action value function, and the joint scheduling of multiple search subtasks based on deep *Q*-learning, multi-stage *Q*-learning, and fuzzy *Q*-learning.

Author Contributions: G.Y. designed the algorithm and drafted the manuscript. H.G. conducted the simulation analysis and co-drafted the manuscript. A.A.D. reviewed and edited the manuscript. I.A. supervised the research and provided constructive suggestions to improve the research. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the 2022 Young Innovative Talents Project of Guangdong Colleges and Universities (Grant no. 2022KQNCX138), 2023 Guangdong Province Education Science Planning Project (Higher Education Special, Grant no. 2023GXJK615), 2022 The Teaching Quality and Teaching Reform Project of Guangdong Province (Grant no. GDJG2208), 2022 Research project of Guangdong Undergraduate Open Online Course Steering Committee (Grant no. 2022ZXKC579), The 14th Five-Year Plan for the development of philosophy and social sciences in Guangzhou (Grant no. 2023GZGJ103).

Data Availability Statement: The data used to support this research article are available from the first author upon request.

Acknowledgments: The authors thank the anonymous reviewers for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. A List of Acronyms

The acronyms appearing in this paper are listed in Table A1.

Table A1. A list of acronyms that appear in the paper.

Acronym	Full Name
LBs	lost-link borrowers
MSC	mathematics subject classification
P2P	peer-to-peer
EILMMA-DDPG	ensemble imitation learning multi-trick multi-agent deep deterministic policy gradient
ASPL	action selection priority level
IDS	intrusion detection system
DVFS	dynamic voltage and frequency scaling

Table A1. *Cont.*

Acronym	Full Name
DQL	deep Q-learning
DFQL	deep-federated Q-learning
FSACL	fast-scene adaptive reinforcement learning
CL	cooperative Q-learning
IL	independent Q-learning
DQN	deep Q-learning network
RIFQ	reward iterative fuzzy Q-learning
PC	personal computer
MDP	Markov decision processes

References

1. Hertzum, M.; Hansen, P. Empirical studies of collaborative information seeking: A review of methodological issues. *J. Doc.* **2019**, *75*, 140–163. [CrossRef]
2. Yu, W.; Wang, H.; Hong, H.; Wen, G.H. Distributed cooperative anti-disturbance control of multi-agent systems: An overview. *Sci. China Inf. Sci.* **2017**, *60*, 110202. [CrossRef]
3. Gulzar, M.M.; Rizvi, S.T.H.; Javed, M.Y.; Munir, U. Multi-agent cooperative control consensus: A comparative review. *Electronics* **2018**, *7*, 22. [CrossRef]
4. Hajieghrary, H.; Hsieh, M.A.; Schwartz, I.B. Multi-agent search for source localization in a turbulent medium. *Phys. Lett. A* **2016**, *380*, 1698–1705. [CrossRef]
5. Vasile, M. A memetic multi-agent collaborative search for space trajectory optimization. *Int. J. Bio-Inspir. Com.* **2009**, *1*, 186–197. [CrossRef]
6. Kim, B.M.; Li, Q.; Howe, A.E.; Chen, Y.P. Collaborative web agent based on friend network. *Appl. Artif. Intell.* **2008**, *22*, 331–351. [CrossRef]
7. Birukou, A.; Blanzieri, E.; Giorgini, P. Implicit: A multi-agent recommendation system for web search. *Auton. Agents Multi-Agent* **2012**, *24*, 141–174. [CrossRef]
8. Shimoji, R.; Sakama, C. Multiagent Collaborative Search with Self-Interested Agents. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Singapore, 6–9 December 2015; pp. 242–249.
9. Song, C.; He, Y.Y.; Ristic, B.; Li, L.; Lei, X.K. Multi-agent collaborative infotaxis search based on cognition difference. *J. Phys. A-math. Theor.* **2019**, *52*, ab5088. [CrossRef]
10. Perez-Crespo, C.F.; Perez-Crespo, M.M.; Costaguta, R. A metasearch engine that streamlines collaborative searches. *Campus Virtuales* **2018**, *7*, 81–93.
11. Chu, Y.; Xu, Z. Multi-source information search method based on multi-agent collaboration. *Comput. Eng.* **2015**, *41*, 193–198.
12. Vasile, M.; Ricciardi, L.; Schutze, O.; Trujillo, L.; Legrand, P.; Maldonado, Y. Multi Agent Collaborative Search. In Proceedings of the NEO 2015: Results of the Numerical and Evolutionary Optimization Workshop, Tijuana, Mexico, 23–25 September 2015; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; Volume 663, pp. 223–252.
13. Koval, A.; Mansouri, S.S.; Nikolakopoulos, G. Multi-Agent Collaborative Path Planning Based on Staying Alive Policy. *Robotics* **2020**, *9*, 101. [CrossRef]
14. Zhou, T.; Tang, D.B.; Zhu, H.H.; Zhang, Z.Q. Multi-agent reinforcement learning for online scheduling in smart factories. *Robot. Comput.-Integr. Manuf.* **2021**, *72*, 102202. [CrossRef]
15. Jing, G.S.; Bai, H.; George, J.; Chakrabortty, A.; Sharma, P.K. Learning Distributed Stabilizing Controllers for Multi-Agent Systems. *IEEE Control Syst. Lett.* **2022**, *6*, 301–306. [CrossRef]
16. Li, J.W.; Yu, T.; Yang, B. Coordinated control of gas supply system in PEMFC based on multi-agent deep reinforcement learning. *Int. J. Hydrogen Energy* **2021**, *46*, 33899–33914. [CrossRef]
17. Zhou, T.; Hong, B.R.; Shi, C.X.; Zhou, H.Y. Cooperative Behavior Acquisition Based Modular Q Learning in Multi-Agent System. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 8, pp. 205–210.
18. Sethi, K.; Madhav, Y.V.; Kumar, R.; Bera, P. Attention based multi-agent intrusion detection systems using reinforcement learning. *J. Inf. Secur. Appl.* **2021**, *61*, 102923. [CrossRef]
19. Asghari, A.; Sohrabi, M.K. Combined use of coral reefs optimization and multi-agent deep Q-network for energy-aware resource provisioning in cloud data centers using DVFS technique. *Clust. Comput.-J. Netw. Softw. Tools Appl.* **2022**, *25*, 119–140. [CrossRef]
20. Mlika, Z.; Cherkaoui, S. Network slicing for vehicular communications: A multi-agent deep reinforcement learning approach. *Ann. Telecommun.* **2021**, *76*, 665–683. [CrossRef]
21. Messaoud, S.; Bradai, A.; Ben Ahmed, O.; Quang, P.T.A.; Atri, M.; Hossain, M.S. Deep Federated Q-Learning-Based Network Slicing for Industrial IoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5572–5582. [CrossRef]
22. Dou, Z.; Si, G.Z.; Lin, Y.; Wang, M.Y. A power allocation algorithm based on cooperative Q-learning for multi-agent D2D communication networks. *Phys. Commun.* **2021**, *47*, 101370. [CrossRef]

23. Chen, S.K.; Dong, J.Q.; Ha, P.; Li, Y.J.; Labi, S. Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Comput. Aided Civ. Inf. Eng.* **2021**, *36*, 838–857. [CrossRef]
24. Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; Vicente, R. Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE* **2017**, *12*, e0172395. [CrossRef] [PubMed]
25. Daeichian, A.; Haghani, A. Fuzzy Q-Learning-Based Multi-agent System for Intelligent Traffic Control by a Game Theory Approach. *Arab. J. Sci. Eng.* **2018**, *43*, 3241–3247. [CrossRef]
26. Leng, L.X.; Li, J.C.; Zhu, J.H.; Hwang, K.S.; Shi, H.B. Multi-Agent Reward-Iteration Fuzzy Q-Learning. *Int. J. Fuzzy Syst.* **2021**, *23*, 1669–1679. [CrossRef]
27. Pang, S.; Yang, J. Social reputation loss model and application to lost-linking borrowers in an internet financial platform. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 1193–1203. [CrossRef]
28. Pang, S.L.; Wang, J.Q.; Xia, L.H. Information matching model and multi-angle tracking algorithm for loan loss-linking customers based on the family mobile social-contact big data network. *Inform. Process. Manag.* **2022**, *59*, 102742. [CrossRef]
29. Pang, S.L.; Wang, J.Q.; Yi, X.S. Application of loan lost-linking customer path correlated index model and network sorting search algorithm based on big data environment. *Neural Comput. Appl.* **2023**, *35*, 2129–2156. [CrossRef]
30. Matignon, L.; Laurent, G.J.; Le Fort-Piat, N. Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *Knowl. Eng. Rev.* **2012**, *27*, 1–31. [CrossRef]
31. Zhang, Z.; Wu, F.; Qian, B.; Hu, R.; Wang, L.; Jin, H. A Q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation. *Expert Syst. Appl.* **2023**, *234*, 121050. [CrossRef]
32. Ahmed, M.; Khoo, H.; Ng, O. Discharge control policy based on density and speed for deep Q-learning adaptive traffic signal. *Transp. B Transp. Dyn.* **2023**, *11*, 1707–1726. [CrossRef]
33. Ni, X.R.; Hu, W.; Fan, Q.C.; Cui, Y.B.; Qi, C.K. A Q-learning based multi-strategy integrated artificial bee colony algorithm with application in unmanned vehicle path planning. *Expert Syst. Appl.* **2023**, *236*, 121303. [CrossRef]
34. Amhraoui, E.; Masrour, T. Smooth Q-Learning: An Algorithm for Independent Learners in Stochastic Cooperative Markov Games. *J. Intell. Robot. Syst.* **2023**, *108*, 65. [CrossRef]
35. Watkins, C. Learning From Delayed Rewards. *Robot. Auton. Syst.* **1989**, *15*, 233–235.
36. Lee, D.; Hu, J.H.; He, N. A Discrete-Time Switching System Analysis of Q-learning. *Siam J. Control Optim.* **2023**, *61*, 1861–1880. [CrossRef]
37. Kumar, N.; Gupta, M.; Sharma, D.; Ofori, I. Technical Job Recommendation System Using APIs and Web Crawling. *Comput. Intell. Neurosc.* **2022**, *2022*, 7797548. [CrossRef]
38. Kumar, N.; Aggarwal, D. LEARNING-based Focused WEB Crawler. *IETE J. Res.* **2023**, *69*, 2037–2045. [CrossRef]