



Oracle Recognition of Oracle Network Based on Ant Colony Algorithm

Xianjin Shi^{1,2} and Xiajiong Shen^{1,3*}

¹School of Computer and Information Engineering, Henan University, Kaifeng, China, ²Henan Center for Educational Technology, Zhengzhou, China, ³Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, China

Recent studies have shown that compared with traditional social networks, networks in which users socialize through interest recommendation have obvious homogeneity characteristics. Recommending topics of interest to users has become one of the main objectives of recommendation systems in such social networks, and the widespread data sparsity in such social networks has become the main problem faced by such recommendation systems. Particularly, in the oracle interest network, this problem is more difficult to solve because there are very few people who read and understand the Oracle. To address this problem, we propose an ant colony algorithm based recognition algorithm that can greatly expand the data in the oracle interest network and thus improve the efficiency of oracle interest network recommendation in this paper. Using the one-to-one correspondence between characters and translation in Oracle rubbings, the Oracle recognition problem is transformed into character matching problem, which can skip manual feature engineering experts, so as to realize efficient Oracle recognition. First, the coordinates of each character in the oracle bones are extracted. Then, the matching degree value of each oracle character corresponding to the translation of the oracle rubbings is assigned according to the coordinates. Finally, the maximum matching degree value of each character is searched using the improved ant colony algorithm, and the search result is the Chinese character corresponding to the oracle rubbings. In this paper, through experimental simulation, it is proved that this method is very effective when applied to the field of oracle recognition, and the recognition rate can approach 100% in some special oracle rubbings.

Keywords: weak relationships, oracle social networks, interest-based recommendations, ant colony algorithms, heterogeneous networks

OPEN ACCESS

Edited by:

Shudong Li,
Guangzhou University, China

Reviewed by:

Yu Wu,
Aerospace Information Research
Institute (CAS), China
Xiaohua Hu,
Drexel University, United States

*Correspondence:

Xiajiong Shen
shenxj@henu.edu.cn

Specialty section:

This article was submitted to
Social Physics,
a section of the journal
Frontiers in Physics

Received: 31 August 2021

Accepted: 04 October 2021

Published: 08 November 2021

Citation:

Shi X and Shen X (2021) Oracle
Recognition of Oracle Network Based
on Ant Colony Algorithm.
Front. Phys. 9:768336.
doi: 10.3389/fphy.2021.768336

INTRODUCTION

Recommendation systems, as an effective means of information filtering, can help users discover the content they are more interested in from a large amount of information [1–4]. Most traditional recommendation algorithms perform collaborative filtering recommendations with strong relational objects that are closely related to the target users and have good interpretability. However, strong relational recommendations often lead to single recommendations due to their own small circle and more content of invalid information generated than quality information content. In contrast to strong relationships, weak relationships have richer semantic information and stronger information influence. Therefore, recommendations based on weak relationships can get rid of the problems of single recommendation type and duplication of recommendation items brought by traditional recommendations based on strong relationships.

Social networks are usually regarded as homogeneous or heterogeneous structural networks [5, 6], where homogeneous social networks have a single type of nodes and relationships with strong asymmetry and small group size, which is not conducive to eliminating users' perception bias [7]. Oracle social network is a classical homogeneous networks. And because of the presence of a large number of oracle images in the oracle network, which are generally in the form of CAPTCHA, the task of recommendation in oracle social networks is a very difficult task. Therefore, the data sample for this interest-based social network recommendation is very small, especially for the oracle-based interest-based social network. Since reading oracle texts requires a lot of time and expertise, the data samples in oracle-based interest-based social networks often do not meet the quantitative requirements needed for network recommendation algorithms. and in oracle interest social networks, oracle pictures are generally presented in the form of CAPTCHA, which becomes more difficult to recognize oracle CAPTCHA compared to traditional oracle recognition.

However, there exist a large number of deciphered oracle rubbings, and these topographies corresponding to the translated text would largely advance the recommendation work in oracle-based interest-based social networks if they could be used for training of oracle CAPTCHA recognition. However, these deciphered topographies are not labeled with the correspondence between each character and the translated text, so they cannot be directly used for training.

Based on the above problems, this paper proposes an oracle rubbings character recognition method based on ant colony algorithm. The oracle characters in the bones and the characters of the interpreted text can be corresponded one by one, laying the foundation for the establishment of an oracle bone character library. The following is the contribution of this paper:

- Oracle bones are used as the identification unit and the use of manual annotation for generating oracle character sets for identification is skipped, thus manpower loss is reduced and work efficiency is improved.
- The euclidean distance-based ant colony algorithm is improved to character matching degree based ant colony algorithm, and the improved ant colony algorithm is applied to oracle bones character recognition to improve the recognition accuracy. The recognition rate reaches 100% in some rubbings with more standardized oracle character arrangement.

In *Related Works*, the principle of topology-based oracle character recognition is elaborated; in *Oracle Captcha Recognition*, how the traditional ant colony algorithm can be improved and applied to oracle character recognition is shown; in *Experimental Design and Analysis of Results*, the topology-based oracle character recognition proposed in this paper is simulated and modeled; in *Conclusion*, the whole paper is summarized and the direction of future work is described.

RELATED WORKS

Interest Network Recommendation Research

Network recommendation is to use the interaction data generated by the whole user group when browsing the Internet to filter the huge amount of information and provide the target users with the required information. Network recommendation not only saves its own operation cost, but also improves the user's loyalty and satisfaction. The first step in the construction of a web recommendation system is data input. The dramatic increase in the volume of Internet data and the number of users has greatly increased the cost of explicit scoring data, and the problem of data sparsity has become more serious, leading to a decrease in recommendation accuracy and user satisfaction.

User-project implicit interaction data for the purpose is an ideal solution to the problem [8, 9]. At present, the collection of user browsing behavior data is mainly divided into three ways, namely server-side, client-side and server-client. Firstly, the quantitative collected browsing behavior data is analyzed to get the implicit interest score of users to the project, and then the personalized recommendation list is obtained by using recommendation algorithms such as collaborative filtering according to users' different personalized needs and application scenarios. Finally, the results are pushed to the target users on the page in the form of ranked lists, images, links, etc.,

The popularity of artificial intelligence and big data has made applied research in the field of personalized recommendations even hotter. RecSys has placed increasing emphasis on modeling and analyzing user interaction behavior in recent years, and the 2017 Xing Social Network Job Recommendation Challenge generated job recommendations for job seekers by fusing user browsing behavior data with user and project attribute data [10]. A "Million User Playlist Dataset (MPD)" was provided by the Spotify online music platform in 2018, which is a dataset consisting of 1,000 sparse playlists, including title and meta and metadata, as well as playlist data from users' collections, to recommend 500 ordered waitlists to users the data set consists of 1,000 sparse playlists, including title and metadata, as well as playlist data from users' collections [11, 12]. In online global hotel search platform of [13], an anonymous dataset of nearly 16 million session interactions was provided, which involves over 700,000 users who visited the trivago website during the week of November 2018 for the purpose of providing online travel recommendation questions. A large data set of approximately 160 million public tweets provided by Twitter in 2020 public dataset containing retweet, like, comment, and retweet-plus-comment social browsing behavior data, user data, and Tweet data. The data set will contain data on social browsing behavior of retweets, likes, comments and retweets plus comments, user data and tweet data, and the data set will be protected by user privacy regulations for deleted tweets and user profiles. The data set of deleted tweets and user profiles is updated according to user privacy regulations, making it necessary for the challenger to must continuously retrain the data test set for

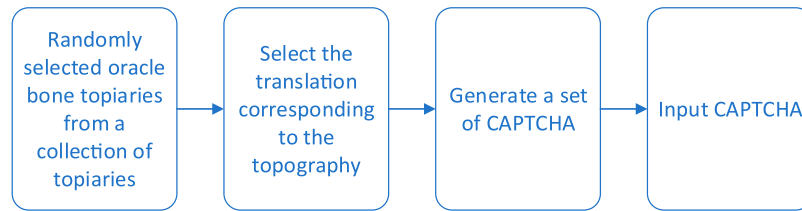


FIGURE 1 | Oracle-based CAPTCHA generation process.

predicting user browsing behavior Prediction of engagement [14]. Gao et al. solved the cold start problem caused by sparse user rating data by calculating the similarity between active users and a small number of expert users based on user collaborative filtering as the trust value [15]. Han et al. propose a collaborative filtering recommendation algorithm that incorporates item features and mobile user trust relationships to mitigate the effects of scoring data sparsity on collaborative filtering algorithms and improve the accuracy of mobile recommendations. Compared with traditional recommendation algorithms, the added “trust” mechanism effectively alleviates the data sparsity problem, but fundamentally still uses similarity as a criterion to measure the trust value, and does not change the subject of the recommendation, and the recommendation duplication problem is not solved. In contrast to strong relationships, weak relationships can be two-way or one-way, because they do not require strong interaction, so they are less costly to maintain. Relying on the power of “weak relationship” can expand the selection range of recommendations, increase the novelty of recommendations, reduce nagging content and make recommendations more clear [16].

Oracle-Based CAPTCHA Scheme

Due to the development of deep learning, both the text-based CAPTCHA scheme and the image-based CAPTCHA scheme mentioned above have different degrees of security risks. To improve the security line of CAPTCHA, researchers of text information systems have proposed an Oracle-based CAPTCHA scheme with the flow shown in **Figure 1**.

The captcha generation starts with a random selection of oracle rubbings, and a library of oracle rubbings is created in advance, each of which contains the corresponding translation, and the oracle rubbings contain annotation boxes for all oracle characters. **Figure 2** shows an example of saving to the oracle rubbings library.

As shown in **Figure 3**, the order of the oracle bone marker box starts from the top right and ends at the bottom left. Similarly, the order of the transliteration is also from the top right, “申(Shen)” is the first character in the topography, and “莫(Mo)” is the last character in the topography. The “□” represents the unrecognizable oracle bone character. Because of hand-carving, the shape of the same character in the Oracle rubbings is also very different, so the deep neural network still has a big defect in recognizing Oracle. This also provides a great security guarantee for this verification code solution. And for

some text information systems with confidentiality requirements, using the captchas scheme based on Oracle can filter out some irrelevant personnel and improve the security of the system.

ORACLE CAPTCHA RECOGNITION

Using the correspondence between Oracle topology and interpretation text, this paper proposes an Oracle captcha recognition model based on ant colony algorithm. This model have both lower computing time and higher accuracy compared with the deep learning model.

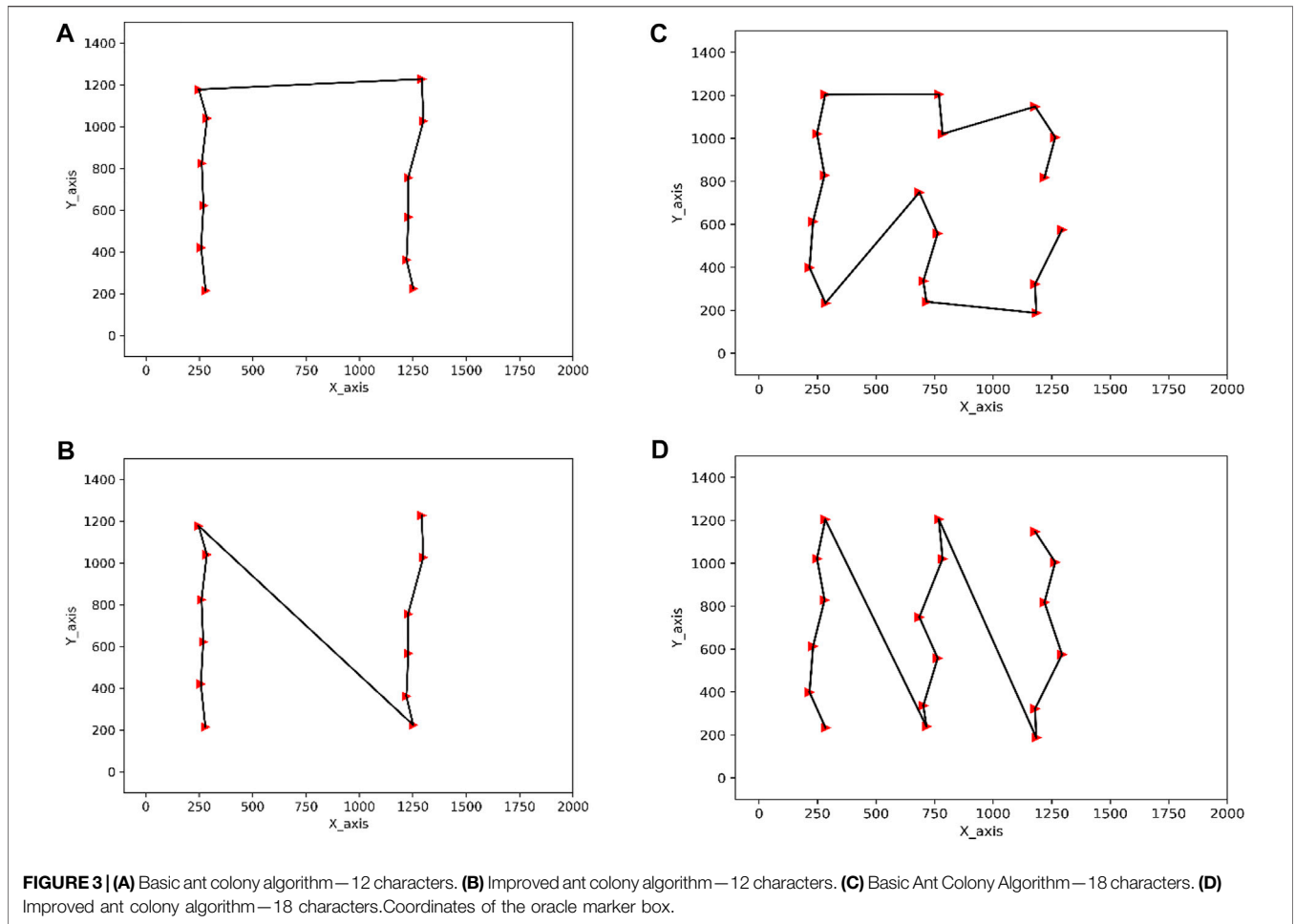
Prerequisites and Conventions

Convention 1. Assume that all oracle bone inscriptions in oracle rubbings can be detected and annotated by applying oracle bone detection tools.

Constraint 2. Assume that the coordinates of the center point of the oracle markup box are the coordinates of the markup box. This is shown in **Figure 4**. The coordinates of this markup box are (123, 456).



FIGURE 2 | Example of storage in the oracle topography library.



Constraint 3. The distance of the markup box distance is the distance from the center of the markup box.

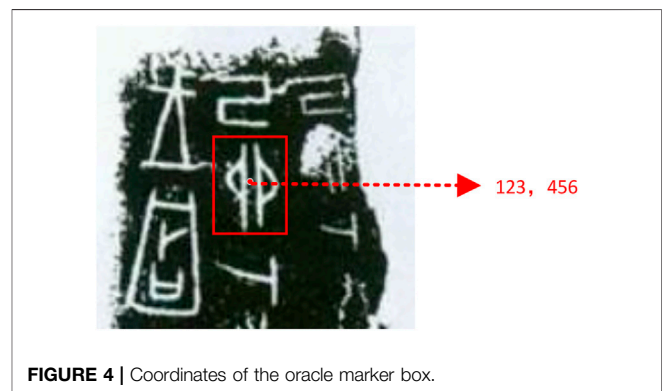
Constraint 4. The closer the distance between two oracle characters marker boxes in each line of an oracle rubbings, the higher the probability that these two oracle characters translations are adjacent to each other.

Identification Steps

The oracle bone writing is generally shown in **Figure 5**, and the writing is not uniform in any way. For researchers who do not know the oracle rubbings, they can only compare them one by one according to the interpretation. However, regardless of the rules of oracle bone writing, they all have a similar rule: the closer the distance between two oracle bone marker boxes, the higher the probability that the two oracle bone interpretations are adjacent to each other. Based on this law, this paper transforms the oracle bone captcha recognition problem into an optimization problem that can be solved using heuristic search algorithms.

The identification steps are as follows.

Step 1. Use the oracle bone detection algorithm according to convention 1 to detect all oracle bone characters in the oracle



rubbings, extract the coordinates of all characters in the oracle rubbings and label them.

Step 2. Identify each oracle bone character in the tops according to the interpretation and the existing oracle bone character database, and the results of identification are arranged in descending order in this paper, and the top 3 are taken as the final results of identification. Here, the recognition results are named as matching degrees.



FIGURE 5 | Oracle rubbings.

Step 3. Calculate the distance between every two labeled boxes according to convention 3.

Step 4. Using the improved ant colony arithmetic, the oracle characters corresponding to the oracle rubbings translation are searched with the matching degree obtained in step 2 as the primary constraint and the distance obtained in step 3 as the secondary constraint, and the final search result is the recognition result of this paper.

Model

In this section, the oracle recognition problem is abstracted into a mathematical model for description. In this paper, all the labeled boxes on a piece of oracle rubbings are regarded as the vertices of the graph, and the coordinates of the center point of the labeled boxes are regarded as the coordinates of the vertices in the graph. Then they are defined as follows.

Definition 1. Figure $G = (V, W, D)$, the $V = \{v_1, v_2, v_3, \dots, v_n\}$ denotes the graph G the set of vertices in the graph, and $W = \{w_1, w_2, w_3, \dots, w_n\}$ denotes the set of matches between the vertices in graph G and the corresponding interpretations, and $E = \{d_{12}, d_{13}, d_{14}, \dots, d_{(n-1)n}\}$ denotes the set of distances between the vertices in the graph, where d_{12} denotes the distance between the first vertex and the second vertex. In this paper, we call the graph G as the rubbings graph.

Definition 2. Assume the existence of paths $P = \{v_1, v_2, v_3, \dots, v_n\}$, in which the order of the nodes is the same as the order of the oracle rubbings, then the path P is the Best Path.

According to definition 1 and definition 2, this paper transforms the Oracle recognition problem into finding the

best path problem in graph G . The order of nodes in this path is the same as the order of the oracle interpretation. A formal language is used to describe it as follows.

It is known that a certain oracle rubbing has n oracle rubbings, the set $V = \{v_1, v_2, v_3, \dots, v_n\}$ denotes the coordinates of each character in the oracle rubbings and $o(v_i, v_j) = c$, $1 \leq i < j \leq n$, $0 \leq c \leq 1$ indicates that the probability that the next Oracle character next to v_i on the Oracle rubbing is v_j is c . The sequence of a group of nodes is required to be $P = \{v'_1, v'_2, \dots, v'_n\}$, and satisfies $\max(\sum_{i=1}^n o(v'_i, v'_{i+1}))$, then P is the Best Path.

How to solve for the best path? Exhaustive enumeration is the simplest and the most accurate method. Suppose an oracle rubbings have n characters, then the size of the solution space is $S = (n-1)!/2$ and the solution with the highest matching degree is the best path solution. But when $n = 10$, we have $S \approx 1.8 \times 10^4$; when $n = 20$, we have $S \approx 6.0 \times 10^{16}$; when $n = 30$, we have $S \approx 4.4 \times 10^{30}$. In the face of such a large solution space, it is impractical to use the exhaustive method.

In general, all problems that can be computed using polynomial-time algorithms and thus obtain results, we call P-problems [17–20]. All problems that can be solved in polynomial time to verify whether the solution of a problem is correct or not, we call NP problems, and it is important to note that the time used to solve the solution of the NP problem itself is unbounded. If there exists an NP problem where the time required to solve this NP problem itself is super-polynomial time, then we call this problem an NP-hard problem. Solving the optimal path problem is an NP-hard problem.

And a common method for solving NP-hard problems is to use heuristic search methods, of which the ant colony algorithm [21–23] is relatively common. In this paper, we propose a method for solving the optimal path based on the ant colony algorithm.

Ant Colony Algorithm

Ant colonies release a substance called pheromone in the process of foraging, and the more concentrated pheromone on a path means that more ants are walking on that path. Based on this principle, Maro Dorigo et al. proposed a bionic algorithm to simulate the foraging behavior of ants and named it the ant colony algorithm. The ant colony algorithm has two key steps: state transfer and pheromone update.

Let α denotes the information heuristic factor, which will affect the search range of the ant colony. With the increase of α , the search range of the ant colony will increase, but the randomness of the corresponding path selection will decrease. A decrease of α will make it easier to get a locally optimal solution. Let β denote the desired heuristic factor, and β affects the convergence speed of the algorithm. As the expected heuristic factor increases, the convergence speed of the algorithm will increase, but it will be easier to obtain the local optimal solution. Let τ_{ij} denote the path (i, j) the pheromone concentration of the path, d_{ij} denote the Euclidean distance of the path (i, j) and η_{ij} denote the heuristic function. Furthermore, let the set $J_k(i)$ denote the first city i that has not been visited by an ant. Then the path transfer formula of the ant colony algorithm is

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{\mu \in J_k(i)} [\tau(i, \mu)]^\alpha [\eta(i, \mu)]^\beta}, & j \in J_k(i) \\ 0, & j \notin J_k(i) \end{cases} \quad (1)$$

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (2)$$

Let ρ denote the information volatility factor, then $1 - \rho$ denotes the residual factor. As ρ increases, it will cause some paths to be abandoned for search, which may contain valid paths and affect the search of optimal values. And as ρ decreases, it will cause repeated search and affect the convergence speed of the algorithm. Then the pheromone update formula at time $t + 1$ is

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^M \Delta\tau^k(t) \quad (4)$$

where $\Delta\tau_{ij}(t)$ denotes the pheromone concentration when the k -th ant circulates at time t . Let Q be the pheromone intensity, and L_k represents the total length of the path taken by the k -th ant, then

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L_k} \quad (5)$$

Find the best path based on ant colony algorithm

Based on the theory of the traditional ant colony algorithm, this paper designs an algorithm for finding the best path on Oracle rubbings according to the characteristics of Oracle recognition [24–26]. The transfer rules and pheromone update rules of the ant colony algorithm are mainly changed.

As mentioned above, the optimization goal of the best path is $\max(\sum_{i=1}^n o(v'_i, v'_{i+1}))$, the same Convention 4 indicates that the closer the two Oracle characters are, the greater the probability that they are adjacent in the translation. However, when there is a line break in the Oracle rubbings, agreement 4 will not be met. Therefore, it is necessary to integrate when looking for the best path. Consider the distance relationship between the matching degree of the Oracle rubbings and the translation and the Oracle characters. According to (Eq. 1), this paper proposes the state transition formula for finding the best path as:

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta o(i, j)}{\sum_{\mu \in J_k(i)} [\tau(i, \mu)]^\alpha [\eta(i, \mu)]^\beta o(i, \mu)} & j \in J_k(i) \\ 0, & j \notin J_k(i) \end{cases} \quad (6)$$

$$\eta_{ij} = \frac{1}{d_{ij}o(i, j)} \quad (7)$$

where $o(i, j)$ denote Probability that the next node of the current node i is the j -th node. Then we have

$$o(i, j) = \frac{1}{2}(w_i + w_j) \quad (8)$$

where w_i denote the match between the oracle bone topos and the oracle bone translation of the first i character in the translation.

Since the pheromone concentration affects the paths chosen by the ant colony, the pheromone update rule needs to be modified as well. The goal of the modification is to increase the pheromone concentration on the paths with larger matches and decrease the pheromone concentration on the paths with smaller matches. The pheromone update rules are as follows.

$$\Delta\tau_{ij}^k(t) = \frac{Q \times o(i, j)^2}{L_k} \quad (9)$$

Implementation of improved ant colony algorithm

Input: Oracle rubbings $G = (V, W, D)$; Parameters α, β, ρ, m , maximum number of iterations $iter_max$.

Output: The best path $P = (V', W', D')$.

Step 1: Initialize $iter_max, \alpha, \beta, \rho, m$, and initialize W according to oracle identification algorithm.

Step 2: The roulette algorithm is used to select m nodes as the initial positions of the ants, and the initial positions are added to the taboo table.

Step 3: According to the state transfer formula, the probability of all possible arrival points is calculated, and the roulette algorithm is used to select the next node and add the next node to the forbidden table.

Step 4: The matches of the nodes on the path taken by all ants are summed and the total match W_{sum} is updated.

Step 5: Update the value of ρ to update the global pheromone. Step 6: Determine if the maximum number of iterations is reached. If the maximum number of iterations is reached, the algorithm runs to the end and outputs the best path currently searched. If the maximum number of iterations is not reached, another iteration is added 1 and go to Step 3 to continue running the algorithm.

Based on the above analysis, this paper gives a pseudo-code example of the improved ant colony algorithm, as shown in **Algorithm 1**.

Line 1 indicates initialization $\alpha, \beta, \rho, m, iter_max, w_{sum}, W, Q$. Lines 2-4 indicate assigning a corresponding match value to each oracle character to be recognized. Line 5 indicates the use of the roulette wheel algorithm to select m The initial positions of the ants are selected using the roulette algorithm and the initial positions are added to the forbidden table. Line 6 indicates that the maximum iteration of the ant colony algorithm cannot be larger than $iter_max$ and rows 7-9 indicate that the probability that all ants may reach each point is calculated according to the state transfer (Eq. 1). Line 10 indicates that the roulette wheel algorithm is used to select the next node and add the next node to the forbidden table. Lines 11-16 indicate that the matches of the nodes on the path taken by all ants are summed up. Lines 17-24 indicate that the best path is selected W_{best} , and update the total matching degree W_{sum} . Line 25 indicates that the update of ρ according to (Eqs 3, 4). Line 26 indicates that the global pheromone is updated according to Eq. 9.

Algorithm 1 | The improved ant colony algorithm

Input: W, V	
Output: W_{best}	
1	Initialization $\alpha, \beta, \rho, m, iter_max, w_{sum}, W, Q$
2	for $v_i \in V_i$, do
3	$v_i = w_i$
4	end for
5	use Roulette algorithm selects m vertices and add tabu list
6	for $i = 0, i < iter_max; i++$ do
7	for $w_i \in W_i$ do
8	recalculate w_i according to Eq. (1)
9	End for
10	use Roulette algorithm to select the next vertices and add a tabu list
11	for $v_i \in V_i$, do
12	$w_{temp} = 0$
13	if the ants reach over v_i then
14	$w_{temp} += w_i$
15	end if
16	end for
17	if $w_{temp} > w_{sum}$ then
18	$w_{sum} = w_{temp}$
19	for $v_i \in V_i$, do
20	if the ants reach over v_i then
21	w_i adds W_{best}
22	end if
23	end for
24	end if
25	update ρ according to Eq. (3) and Eq. (4)
26	update Q according to Eq. (9)
27	end for

TABLE 1 | Parameter settings of ant colony algorithm.

Parameters	Basic ant colony algorithm	Improved ant colony algorithm
m Number of ant colonies	20	20
α	1	1
β	2	2
ρ	0.1	0.1
$iter_max$	100	100
Q Pheromone intensity	1	1
w	[0.9, 1) or (0, 0.5]	[0.9, 1) or (0, 0.5]

EXPERIMENTAL DESIGN AND ANALYSIS OF RESULTS

Experimental Environment and Data Set

The experimental setting for this paper is as follows.

Operating system	Windows 10
Python version	3.7
CPU	Intel i5-8300H
Memory	16GB
GPU	Quadro P4000
Video Memory	8GB
Deep Learning Framework	Tensorflow 2.1

Since there is no research on oracle CAPTCHA recognition and there is very little data about it, this paper uses simulation to conduct experiments. Firstly, the arrangement of oracle characters on the real oracle CAPTCHA is simulated on a 400* 240-pixel image and a series of coordinates are generated. This paper assumes that the recognition efficiency based on the individual characters of oracle is very high, so the corresponding matching degree is assigned as [0.9, 1), while the non-corresponding oracle match is assigned as (0,0.5]. The number of characters generated is divided into 2 categories: 12, 18.

Analysis of Experimental Results

The experimental parameters are set as shown in **Table 1**. The experimental results are shown in **Figure 3**.

From **Figure 3**, it can be seen that the improved ant colony algorithm, which is more in line with the recognition law of Oracle CAPTCHA, has a higher accuracy of recognition. **Table 2** gives a comparison of the convergence speed of the basic ant colony algorithm and the improved ant colony algorithm.

It can be seen from the above table that the improved ant colony algorithm has greatly reduced the number of iterations compared with the basic ant colony algorithm. However, due to the increase of the amount of calculation in each iteration, the total time of the improved ant colony algorithm increases compared with the basic ant colony algorithm, but the increase range is very limited.

TABLE 2 | Convergence speed of ant colony algorithm.

Parameters	Basic ant colony algorithm	Improved ant colony algorithm
Optimal number of iterations—12 characters	31	16
Optimal number of iterations—18 characters	52	29
System runtime—12 characters	1.21s	1.26s
System runtime—18 characters	2.31s	2.48s

CONCLUSION

The oracle based social network is a typical homogeneous network with strong asymmetry and small group size, and the recommendation task in oracle social network is a very difficult task because of the existence of a large number of oracle pictures in the oracle interest network, which generally exist in the form of CAPTCHA. Since oracle researchers have deciphered a very large number of oracle topographies, if we can use the deciphered oracle topographies to expand the data samples in oracle social networks, it will greatly improve the accuracy of recognizing oracle CAPTCHA in oracle social networks, and thus improve the accuracy of oracle network recommendation problem. Therefore, this paper proposes an oracle CAPTCHA recognition model based on ant colony algorithm. Firstly, the matching degree parameter is introduced based on the feature of higher recognition of single character rate of oracle topology. Secondly, the path selection rules and pheromone update rules of the ant colony algorithm are modified according to the matching degree. Finally, simulation experiments are conducted by simulating different classes of oracle verification codes. The experimental results show that the improved ant colony

algorithm has better performance in the problem of identifying the best path, and it converges faster, with fewer iterations, and is more accurate than the traditional ant colony algorithm. This paper provides a new exploration direction for the recommendation problem in the oracle network, but there are few reference factors for the algorithm change at this stage, and there is still room for improvement in parameter optimization and other aspects.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

SHI and SHEN contributed to the conception or design of the work; All authors contributed to article revision, read, and approved the submitted version.

REFERENCES

- Longe OB, Robert ABC, Onwudebelu U. *Checking Internet Masquerading Using Multiple CAPTCHA challenge-response Systems*. International Conference on Adaptive Science & Technology (2009). p. 244–9. doi:10.1109/ICASTECH.2009.5409718
- Yan J, El Ahmad AS. Captcha Robustness: A Security Engineering Perspective. *Computer* (2011) 44(2):54–60. doi:10.1109/mc.2010.275
- Saini B, Bala A. A Review of Bot protection Using CAPTCHA for Web Security. *IOSR J Comp Eng* (2018) 8(6):36–42. doi:10.9790/0661-0863642
- Hernández C, Barrero D, Li S. *An oracle-based Attack on CAPTCHAs Protected against oracle Attacks*. arXiv preprint: arXiv:1702.03815 (2017).
- Yan J, El Ahmad AS. Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms. In: *Twenty-Third Annual Computer Security Applications Conference*. Springer (2007). p. 279–91. doi:10.1109/ACSAC.2007.47
- Yan J, El Ahmad AS. A Low-Cost Attack on a Microsoft Captcha. In: *Proceedings of the 15th ACM Conference on Computer and Communications Security*. Association for Computing Machinery (2008). p. 543–54. doi:10.1145/1455770.1455839
- Lupkowski P, Urbanski M. SemCAPTCHA-user-friendly Alternative for OCR-Based CAPTCHA Systems. *Proc Int Multiconference Comp Sci Inf Tech* (2008) 3:325–9. doi:10.1109/IMCSIT.2008.4747260
- Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J. GroupLens. *Commun ACM* (1997) 40(3):77–87. doi:10.1145/245108.245126
- Nichols D. Implicit Rating and Filtering. In: *Proc of the 5th DELOS Workshop on Filtering and Collaborative Filtering* (1997). p. 10–2.
- Peter K, Yashar D, Farshad B, Jens A, Gerard L, Philipp M. Proceedings of the Workshop on ACM Recommender Systems Challenge. In: *RecSys Challenge '17: ACM Recommender Systems Challenge 2016 Workshop*. Boston, United States: RecSys Community (2016).
- Peter K, Yashar D, Farshad B, Jens A, Gerard L, Philipp M. Proceedings of the Workshop on ACM Recommender Systems Challenge. In: *RecSys Challenge '17: ACM Recommender Systems Challenge 2017 Workshop*. Como, Italy: RecSys Community (2017).
- Peter K, Yashar D, Farshad B, Jens A, Gerard L, Philipp M. Proceedings of the Workshop on ACM Recommender Systems Challenge. In: *RecSys Challenge '18: ACM Recommender Systems Challenge 2018 Workshop*. Vancouver, Canada: RecSys Community (2018).
- Peter K, Yashar D, Farshad B, Jens A, Gerard L, Philipp M. Proceedings of the Workshop on ACM Recommender Systems Challenge. In: *RecSys Challenge '19: ACM Recommender Systems Challenge 2019 Workshop*. Copenhagen, Denmark: RecSys Community (2019).
- Peter K, Yashar D, Farshad B, Jens A, Gerard L, Philipp M. Proceedings of the Workshop on ACM Recommender Systems Challenge. In: *RecSys Challenge '19: ACM Recommender Systems Challenge 2020 Workshop*. [Online, Worldwide] RecSys Community (2020).
- Gao F, Huang M, Zhang T. Collaborative Filtering Recommendation Algorithm Based on User Characteristics and Expert Opinions. *Comp Sci* (2017) 44(2):103–6. doi:10.11896/j.issn.1002-137X.2017.02.014
- Han Z, Ghen Y, Liu W, Yuan B, Li M, Duan A. Research on Node Influence Analysis in Social Networks. *J Softw* (2017) 28(1):84–104. doi:10.13328/j.cnki.jos.005115
- Tian Z, Gao X, Su S, Qiu J, Du X, Guizani M. Evaluating Reputation Management Schemes of Internet of Vehicles Based on Evolutionary Game Theory. *IEEE Trans Veh Technol* (2019) 68(6):5971–80. doi:10.1109/TVT.2019.2910217

18. Tian Z, Su S, Shi W, Du X, Guizani M, Yu X. A Data-Driven Method for Future Internet Route Decision Modeling. *Future Generation Comp Syst* (2019) 95: 212–20. doi:10.1016/j.future.2018.12.054
19. Gu Z, Wang L, Chen X, Tang Y, Wang X, Du X, et al. Epidemic Risk Assessment by A Novel Communication Station Based Method. *IEEE Trans Netw Sci Eng* (2021) 1:1. doi:10.1109/TNSE.2021.3058762
20. Gu Z, Li H, Khan S, Deng L, Du X, Guizani M, et al. IEPSBP: A Cost-Efficient Image Encryption Algorithm Based on Parallel Chaotic System for Green IoT. *IEEE Trans Green Commun Netw* (2021) 1:1. doi:10.1109/TGCN.2021.3095707
21. Gu Z, Hu W, Zhang C, Lu H, Yin L, Wang L. Gradient Shielding: Towards Understanding Vulnerability of Deep Neural Networks. *IEEE Trans Netw Sci Eng* (2021) 8(2):921–32. doi:10.1109/TNSE.2020.2996738
22. Zhang L, Huang Z, Liu W, Guo Z, Zhang Z. Weather Radar Echo Prediction Method Based on Convolution Neural Network and Long Short-Term Memory Networks for Sustainable E-Agriculture. *J Clean Prod* (2021) 298: 126776. doi:10.1016/j.jclepro.2021.126776
23. Zhang L, Xu C, Gao Y, Han Y, Du X, Tian Z. Improved Dota2 Lineup Recommendation Model Based on a Bidirectional LSTM. *Tinshhua Sci Technol* (2020) 25(6):712–20. doi:10.26599/TST.2019.9010065
24. Zhang L, Huo Y, Ge Q, Ma Y, Liu Q, Ouyang W. A Privacy Protection Scheme for IoT Big Data Based on Time and Frequency Limitation. *Wireless Commun Mobile Comput* (2021) 2021:1–10. doi:10.1155/2021/5545648
25. Han D, Chen J, Zhang L, Shen Y, Gao Y, Wang X. A Deletable and Modifiable Blockchain Scheme Based on Record Verification Trees and the Multisignature Mechanism. *CMES-Computer Model Eng Sci* (2021) 128(1):223–45. doi:10.32604/cmesci.2021.016000
26. Lv L, Zheng C, Zhang L, Shan C, Tian Z, Du X, et al. Contract and Lyapunov Optimization-Based Load Scheduling and Energy Management for UAV Charging Stations. *IEEE Trans Green Commun Netw* (2021) 5(3):1381–94. doi:10.1109/TGCN.2021.3085561

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Shi and Shen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.