IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 20, 2020, accepted August 24, 2020, date of publication August 27, 2020, date of current version September 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019797

Dynamic Weighted Heuristic Trust Path Search Algorithm

RU KONG AND XIANGRONG TONG
School of Computer and Control Engineering, Yantai University, Yantai 264005, China
Corresponding author: Xiangrong Tong (xr_tong@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572418 and Grant 61972360.

ABSTRACT Trust propagation is being increasingly adopted to assist recommendation systems in providing more reliable information, upon which, users can make more accurate decisions. Optimal trust path search integrating trust value and path length plays a critical role in trust propagation, but suffers from insufficient performance regarding search accuracy and time. Generally, the quality of trust propagation is affected by the path length, and the longer the path is, the worse the trust quality is. However, the length of the path is not the unique crucial factor. Some longer paths with greater trust values may be more credible. In addition, the A^* algorithm can find an optimal solution, but it expends much time to distinguish some similar paths. The A^* algorithm is improved and a dynamic weighted heuristic trust path search algorithm is proposed. According to the six-degree space theory, the paths are extended to six-degree admissible trust paths. Then, according to the depths of the nodes in the search path, it relaxes the evaluation function $f(n)$ by devising a dynamic weighted factor w, inserts all nodes satisfied specific conditions into the FOCAL list. Furthermore, it sets the secondary heuristic factor and selects the nodes with the minimum heuristic factor value to reach the target node, and outputs the optimal trust path. Experiments on the public Advogato and FilmTrust datasets demonstrated that the proposed algorithm could efficiently identify the reliable trust paths and predict trust value with high accuracy and reduced computational complexity. The proposed algorithm could be applied to recommendation systems in the future.

INDEX TERMS Bounded-suboptimal solution, heuristic search, recommendation system, trust path.

I. INTRODUCTION

Trust recommendation systems are important applications based on social networks [1], [2], which combine the trust relationship between users to recommend items to users. However, in current trust recommendation systems, the critical problems of some trust algorithms such as the length of paths and propagation method have not been effectively resolved, which makes it impossible to give accuracy and highly-quality recommendations. Many scholars have put forward different trust propagation algorithms [3], [4] to find the optimal trust path and obtain more accurate trust recommendations. Therefore, it is significant to find a more reliable trust path and build a more reasonable algorithm for trust prediction [5].

To measure the trust value between two users, the length of the trust path becomes an issue [3]. Studies have shown that integrating all possible paths to evaluate the trust value can provide more accurate trust inference results, but this approach is very time-consuming in large social networks. A number of studies have considered various strategies to alleviate the path length. Examples include the following: setting the search width [6] and limiting the search depth [7]. However, these strategies are still very time-consuming and may lead to few paths or even only one path, which reduces the coverage of the trust path and the accuracy of trust propagation.

In real life, when two people are unfamiliar, they generally turn to a third user who is familiar with both users to introduce them. Compared with direct neighbors, two-hop users with higher trust find it easier to establish a better trust relationship quickly, three-hop users with higher trust find it easier to establish trust relationships than two-hop users, and so on. One of the motivations of this article is that according to the six-degree space, one user can find any users in the world only after at most six connections, which means that it only needs to consider the trust relationships within six hops. In contrast, it is unnecessary to consider the trust relationships outside six

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

hops when finding the admissible trust paths because the trust decay is too high.

Another important issue in inferring trust values is how to choose the optimal trust path from among the admissible trust paths. Compared with the traditional random search algorithm, the heuristic search algorithm [8] can accelerate the search process. As the most classic path search algorithm, the A^* algorithm [9] aimed at combining a mathematical method and a heuristic method to find the optimal solution. However, the memory consumption and the run time required by the A^* algorithm exponentially increase as the search proceeds, which makes searching inefficient.

Therefore, the A^* algorithm is improved through adding a constraint factor w and a new H_f (secondary heuristic factor, H_f). On the one hand, this article emphasizes the node depth because nodes with different depths play different roles in the search path. Because of the influence of the node depth on the search complexity and efficiency, a node with two depths of hops is nearest to the initial node and has the most evaluation value. Thus, the limitation of the evaluation function $f(n)$ should be less. The evaluation function $f(n)$ should be more strictly constrained due to the distance among nodes with greater depths. On the other hand, the MSE (Mean Squared Error, MSE) indicates the fluctuations of the value on the path. Setting MSE as the H_f would ensure that the changes of trust of the paths less than MSE, and avoid the possibility of extremely values in the path. This is in line with practical applications.

In summary, this article proposes the DWHS (Dynamic weighted heuristic trust path search algorithm, DWHS). First, it finds all admissible trust paths based on the six-degree space theory. Second, by considering the constraint factor w, the node depth and the H_f, the A^* algorithm is improved and can better find the optimal trust path. Finally, the predicted trust value is obtained based on the propagation function. Compared with the other algorithms, it can not only provide users with accurate and high-quality recommendations, but also improve the efficiency of recommendations.

The main contributions of this article are as follows.

1. According to the different weights of the nodes at different depths in the trust network, the DWHS is proposed based on the A^* algorithm. The DWHS finds a reliable trust path as the optimal choice, improves the accuracy of trust prediction and reduces the search time.

2. The H_f which is the MSE could ensure that the changes in the selected paths are less than the MSE, that is, the fluctuations on the paths tend to be stable.

3. By conducting comparative experiments on the Advogato and FilmTrust datasets, the experimental results verify that the DWHS effectively improves the accuracy of trust inference, and it is significantly better than other algorithms.

The reminder of the paper is structured as follows. Section II gives a brief introduction of the related background, which is divided into two parts. The first part is mainly about the trust algorithms. The second part introduces some basic knowledge of the heuristic search algorithms. Section III defines a description of the problem. Section IV states the details of the DWHS. The specific process and full pseudocode are presented. Section V presents several groups of compared experiments. Section VI presents some findings and conclusions.

II. BACKGROUND

The work concerns both trust algorithms and heuristic search algorithms. Therefore, in this section, the state-of-the-art trust algorithms and heuristic search algorithms are briefly summarized.

A. TRUST ALGORITHMS

Trust is essential to reduce uncertainty and enhance collaboration in many practical applications, including social networks [10], [11] large-scale Internet of Things systems [12], [13], peer-to-peer networks [14], and wireless sensor networks [15]. In these applications, trust inference is widely used as a mechanism to establish trust between unknown users.

Golbeck and Hendler [7] proposed the TidalTrust model that infers trust values. These researchers believe that the accuracy of the trust prediction will decrease along with the length of the trust path. When the trust value is fixed, a shorter path length is more reliable; and when the path length is fixed, the path with a higher trust value is more reliable. Based on the above two rules, an adaptive trust threshold is calculated to ignore the noncritical paths, and the trust value is inferred. This trust threshold allows the TidalTrust model to prune unimportant paths for trust propagation. The adaptive threshold is calculated by the BFS (breadth first search, BFS). In most cases, the calculated threshold is relatively high.

Comparatively, the MoleTrust model proposed by Massa and Avesani [6] considers all paths that are satisfied with the presumed maximum path length and threshold. First, the rings in the trust network are removed and the trust network is converted into a directed acyclic graph. Then, the paths from the initial user to the target user are discovered and the trust values are aggregated by calculating the weighted average method. In [6] and [7], the threshold can lead to trust prediction with low effectiveness.

The neighborhood-aware trust network method proposed by Jiang *et al*. [16] aims to measure the trust among users to solve the problem of failed trust propagation. This method takes into account the domain perception influence of users, uses directed multiple graphs to model the multiple trust relationships among users in heterogeneous trust networks, and then designs a domain-aware trust metric to measure the relationship between users' degree of trust. Mao *et al*. [17] combined the similarity of weighted interest topics and trust propagation to find the strong trust paths between two users. Weighted interest topics are used to measure the semantic similarity between users, and the heuristic rules of ''small world'' theory are used to constrain the traversal depth.

Ghavipour and Meybodi [18] considered the change of trust in the trust transmission process, proposed a heuristic algorithm based on learning automata, and used an improved collaborative filtering aggregation strategy to infer the trust. On this basis, Ghavipour and Meybodi [19] also proposed the dynamic algorithm based on learning automata, which uses distributed learning automata to spread random trust. The purpose of both is to learn to discover reliable paths between users in social networks. In addition, some scholars also apply trust to data collection. In [13], Jiang *et al*. proposed a trust-based unmanned aerial vehicle energy efficient data collection scheme, which collects only trusted data and improves the quality of data collection, and used it to plan a flight path. In [14], Ren *et al*. used the idea of machine learning to select trusted data reporters to collect data, and they proposed a trust-based minimum cost quality perception data collection scheme to optimize data collection to maximize the data coverage and minimize the budget in malicious networks.

B. HEURISTIC SEARCH ALGORITHMS

Heuristic search algorithms have been applied in many fields. These algorithms can rely on some natural rules and even experiences to plan specific search strategies. Compared with traditional random search algorithms, these algorithms apply a strategic method to make the search process more efficient in a huge state space.

Hart *et al*. first proposed the A^* algorithm for robot path-finding in 1968. In the A^* algorithm, for an evaluated node n, $f(n) = g(n) + h(n)$ is defined as the evaluation function, and it uses the heuristic function to find the node with the minimum cost to expand. $f^*(n) = g^*(n) + h^*(n)$ represents the minimum cost of the shortest path from initial node s to target node t via node n, where $g^*(n)$ means the minimum cost of the shortest path from initial node s to node n, and $h^*(n)$ means the minimum cost of the shortest path from node n to the given target t. Correspondingly, $f(n)$, $g(n)$ and $h(n)$ are their estimates.

In practical applications, when the problem space graph is very complex and large, the A^* algorithm needs to spend many time to find an optimal solution. By contrast, it is better to accept a suboptimal solution, which is considerably faster than the A^* algorithm. Pohl proposed two extensions of the A^* algorithm, weighted A^* algorithm [20] and dynamic weighted A^* algorithm [21], which are faster than the A^* algorithm but might find a suboptimal path. He intended to relax the conditions of the A^* algorithm by introducing a fixed factor and a dynamic factor to increase the weight of the heuristic value $h(n)$ for weighted A^* algorithm and dynamic weighted A^* algorithm, respectively. Pearl and Kim [22] proposed an approach called A^*_ε algorithm in which it adds a FOCAL list maintaining a subset of the nodes from the OPEN list. This subset is the set of nodes whose cost does not deviate excessively from the minimal cost of the nodes under the control of a factor greater than $1+w$.

Yiu *et al*. [23] proposed an evolutionary heuristic algorithm A^* for multiweighted heuristic functions. This algorithm minimizes the workload of the heuristic function design through the genetic algorithm to optimize the search performance of the A^* algorithm. Stern *et al*. [24] attempted to find the possible optimal solution and introduced the concept of the probably bounded-suboptimal search algorithm. This search algorithm accepts two parameters, δ and ε, and outputs the solution, that is, the probability is at least $(1 - \delta)$ and the cost is at most $(1+\varepsilon)$ times the optimal solution. In addition, many scholars have improved the A^* algorithm and conducted corresponding studies on robot path-finding to avoid obstacles [25]. In [25], a heuristic search-based planner was used to solve the obstacle avoidance path planning problem of a manipulator in narrow space, and an improved algorithm based on stasis detection was proposed. By introducing a variety of inadmissible heuristics, the algorithm can effectively avoid the problem of search stagnation caused by inconsistent heuristics.

III. PROBLEM DEFINITION

Trust inference aims to predict the trust rating from one user to another without direct interaction experiences. The trust ratings among users are usually inferred from the trust propagation among their mutual friends. In this section, some definitions closely related to the trust path are defined, and the purpose and method of the research are introduced.

Definition 1 (Trust): Trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome [6]. Let $\tau_{ij} \in [0, 1]$ represent the trust value between v_i and v_j. If $\tau_{ij} = 1$, it represents that v_i completely trusts v_j. Furthermore, if $\tau_{ij} = 0$, then it represents that v_i completely distrusts v_j.

Definition 2 (Trust Network G): Take a trust network $G = (V, E)$, where $V = (v_s, \ldots, v_t)$ is the set of nodes and $E \subseteq \{(v_i, v_j): i \neq j \text{ and } v_i, v_j \in V\}$ is the set of edges. Two nodes v_i and v_j are called adjacent if $(v_i, v_j) \in E$.

Definition 3 (Trust Path): For $\exists P = (v_s, \ldots, v_i, v_j, \ldots, v_t)$, if $\forall v_i$ and $\forall v_j$ are adjacent, it is said that P is a trust path.

Definition 4 (Trust Propagation): For $\forall v_s$ and $\forall v_t$, if $s \neq t$ and $v_s, v_t \in V$, $\exists P$, trust propagation can be completed, and the final trust value τ_{st} is calculated.

The problem that should be studied is solving the optimal trust propagation path and calculating the final trust value in the trust network. Based on the two factors of the path length and heuristic search, the DWHS is proposed. The algorithm includes three subtasks: first, all admissible trust paths should be found; second, the proposed heuristic algorithm is realized to find the optimal trust path; and third, the final trust value is obtained based on the propagation function.

IV. OVERVIEW OF THE DYNAMIC WEIGHTED HEURISTIC TRUST PATH SEARCH ALGORITHM

In the DWHS, the input is a trust network $G = (V, E)$, the initial node v_s and the unconnected target node v_t.

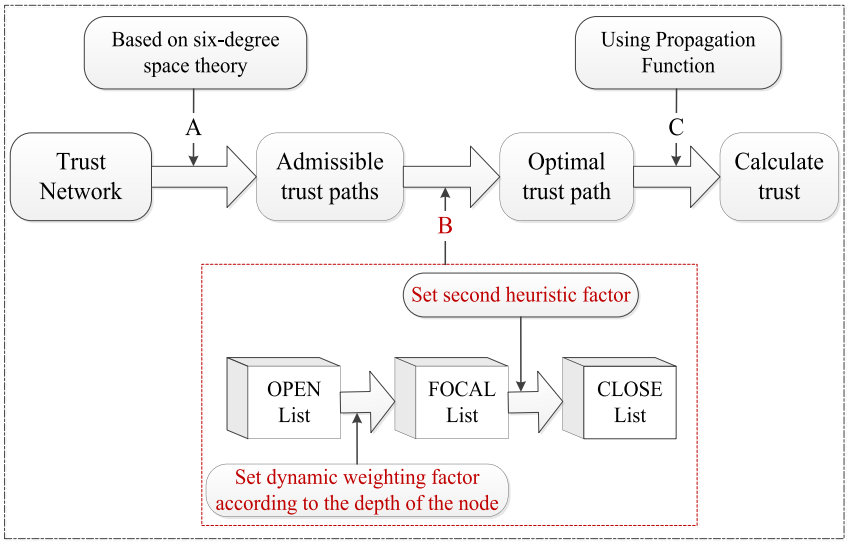


FIGURE 1. Whole architecture of the DWHS, the red mark represents the main contributions.

The output is the final estimated trust value τ_{st}. Figure 1 shows the framework of the DWHS, which consists of the following:

A. According to the six-degree space theory, the BFS is used to find the trust paths with less than or equal to six hops from the initial node to the target node, and obtain the admissible trust paths;
B. According to the depths of the nodes, the improved A^* algorithm is adopted to apply the corresponding dynamic weight to the function values and select the reliable trust path as the optimal choice;
C. The trust propagation function is used to calculate the final predicted trust value τ_{st}.

A. FINDING ALL ADMISSIBLE TRUST PATHS

The objective of this step is to discover all admissible trust paths from the initial node to a given target node. The previous studies have shown that an algorithm using all trust paths to evaluate trust can improve the quality of the trust inference. However, finding all trust paths between two nodes has exponential time complexity. Since online social networks are usually massive in size, it is impractical to calculate trust using all trust paths.

According to the six-degree space theory, each person can find any person in the world after at most six connections. Therefore, this article aims to find the trust path from the initial node to the target node with less than or equal to six hops and to use the BFS to find all admissible trust paths that meet the requirements. During this process, the algorithm will also remember the values on each hop and calculate the depth of each node and the MSE of each path. Once the upper limit of the number of path hops is set as six, the algorithm will stop searching in the other longer paths and check all trust paths at this level.

B. IMPROVING THE A* ALGORITHM TO FIND AN OPTIMAL TRUST PATH

The heuristic search A^* algorithm can prune the impossible path by using heuristic information, reduce the complexity of the problem and obtain the optimal trust path. However, the A^* algorithm expends many time to distinguish similar paths, and there is not enough memory or running time to find the optimal solution. Furthermore, nodes with different depths have different weights in the trust network. As the search path deepens, the functional values need to be dynamically weighted. Therefore, the DWHS traverses the trust network and uses an improved heuristic algorithm to select the optimal trust path for trust propagation, which enables it to have better performance than that of the A^* algorithm. Figure 2 shows the process of improving the A^* algorithm to find an optimal trust path.

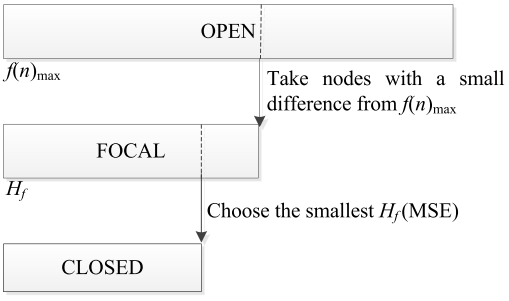


FIGURE 2. Process of improving the A^* algorithm to find an optimal trust path, where the nodes in the OPEN list are sorted according to $f(n)$,$f(n)_{max}$ represents the largest sum of the weights; next, the nodes that meet the conditions are inserted into the FOCAL list and sorted according to H_f which is MSE; finally, the node with the smallest H_f is inserted into CLOSED list.

First, similar to the A^* algorithm, $f(n) = g(n) + h(n)$ is also used as the evaluation function, $g(n)$ as the sum of weights from the initial node v_s to the intermediate node n, and $h(n)$

as the sum of weights from the intermediate node n to the target node v_t. The evaluation function $f(n)$ is sorted by the first heuristic process $h(n)$ in descending order in the OPEN list.

Second, the A^* algorithm only selects the node with the largest sum of the weights $f(n)_{\max}$ in the OPEN list to be inserted into the CLOSED list for expansion. Unlike the A^* algorithm, the DWHS adds a FOCAL list to store the nodes with a small difference from $f(n)_{\max}$, and then selects the node meeting conditions to be inserted into the CLOSED list. These properties would avoid the comparison of the minimum difference values, greatly save search time and improve the search efficiency. Therefore, according to the depths of the nodes in the trust network, the dynamic weighting condition w is used and definition 5 is given as follows.

Definition 5 (FOCAL List):

$$\text{FOCAL} = \{n | f(n) \geq \left[w + (1 - w) * \frac{d(n)}{N} \right] f(n)_{max} \}$$

where FOCAL $\subseteq$ OPEN means to insert all the nodes that meet the conditions from the OPEN list into the FOCAL list.

In definition 5, $f(n)_{\max}$ represents the largest sum of the weights in the OPEN list. Parameter w represents the dynamic weighting condition. When $w = 1$, it is the classical A^* algorithm, and the obtained solution is the optimal solution. More generally, to alleviate the running time of the A^* algorithm, w is set in the range of (0, 1), which means that certain errors are allowed to obtain a feasible suboptimal solution. In summary, the larger that w is, the smaller the error that the algorithm can tolerate. The feasible value of w is obtained by experiment. $d(n)$ represents the depth of node n. N is the length of the optimal solution. In general, N is not known in advance, but it can be represented by an upper limit on the length of the admissible paths. $d(n)/N$ represents the ratio of the node depth. The closer node n is to the initial node, the smaller the value is.

Therefore, the smaller that the weight $[w+ (1-w)d\ (n)/N]$ is, the smaller the restriction. In contrast, when node n is further away from the initial node, the limitation on the evaluation function $f(n)$ will be greater, and the range of the nodes to be considered will be smaller. It would effectively reduce the computational complexity. Therefore, different weighted values can be dynamically applied to nodes at different depths by dynamically relaxing the optimal solution $f(n)$. As the node depth increases, the limitation of $f(n)$ become stricter, which means that the further the node is from the initial node, the lower its reference value will be until it reaches the target node.

Third, the H_f (numerically equal to the MSE) is used to select the nodes with the minimum H_f value from the FOCAL list to insert into the CLOSED list. When the MSE is small, it means that the weight on the path is very stable. Thus, the change of the path weight is less than the MSE, which is beneficial to the final effectiveness. It should be noted that some nodes may exist on multiple different paths, which means that one node may have multiple H_f values. The minimum H_f value of those paths is selected as the second standard in this situation.

Finally, the successor of the node in the CLOSED list is expanded, and the previous work is repeated until the target node is expanded and the algorithm ends.

Algorithm 1 DWHS

1: **Input:** v_s, v_t, Admissible trust paths, MSE of the weight of each path, w.
2: **Output:** Optimal trust path.
3: set OPEN list, FOCAL list, CLOSED list;
4: add v_s to OPEN list;
5: **if** ($v_s == v_t$)
6: end;
7: **else**
8: OPEN$\leftarrow$ getCHILDNODE(v_s);
9: sort($f(n)$); /* In descending order */
10: **If** $f(n) \geq \left[w + (1 - w)\frac{d(n)}{N} \right] * f(n)_{max}$
11: FOCAL$\leftarrow$ node v_i;
12: **end if**
13: sort(H_f); /*In ascending order of MSE */
14: CLOSED $\leftarrow$ getBESTNODE(v_j);
15: expand node v_j;
16: until reach v_t;
17: **end if**

The pseudo code for the DWHS is shown in Algorithm 1. The main contributions of the DWHS lie in the addition of lines 10-14. The traditional A^* algorithm often spends a lot of time to distinguish many similar paths and the amount of memory as well as the run time is exponential as searching goes further, this makes searching inefficient. Unlike the A^* algorithm, the DWHS dynamically relaxes the optimal solution $f(n)_{\max}$ according to the depths of the nodes and inserts all nodes that meet the definition 5 into the FOCAL list, then, it sets a new H_f and sorts the FOCAL list, selects the node with the minimum H_f value to insert into the CLOSED list, until the target node v_t is reached and the optimal trust path is output. Although the DWHS introduces the FOCAL list and parameter w, it saves time to distinguish similar paths, and improves the search efficiency. Experiments give a detailed process description.

C. CALCULATING THE TRUST VALUE

Some propagation operations are selected to calculate the trust propagation in a single chain. Even if users do not have any direct experience, trust propagation can help users evaluate other users.

Two widely used trust propagation functions are the Min-function and the Multi-function. The Min-function calculates the trust value as the minimum trust value among all trust values.

$$\tau_{st} = \min\{\tau_{si}, \tau_{ij}, \cdots, \tau_{kt}\} \quad (1)$$

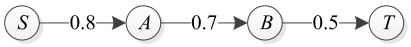


FIGURE 3. Trust propagation.

Meanwhile, the Multi-function is used to calculate the trust value using the product of the trust in the path.

$$\tau_{st} = \tau_{si} * \tau_{ij} * \cdots * \tau_{kt} \tag{2}$$

As shown in Figure 3, S's trust rating for A is 0.8, A's trust rating for B is 0.7, and B's trust rating for T is 0.5. Then, S's trust rating for T can be calculated as follows:

1) Min-function:

$$\tau_{st} = \min\{0.8, 0.7, 0.5\} = 0.5$$

2) Multi-function:

$$\tau_{st} = 0.8^{*}0.7^{*}0.5 = 0.28$$

It can be seen from the above example that the trust value assigned by the Min-function does not decrease as the path length increases. In contrast, the Multi-function causes the trust value to decrease too fast when the path length is very long. Therefore, compared with the Multi-function, the Min-function is favorable for the final trust prediction. Jiang *et al*. [4] showed that the Min-function has more advantages than the Multi-function after some experiments. Therefore, the Min-function is adopted in the subsequent experiments.

V. EXPERIMENTS

There are some commonly-used datasets for evaluating trust value such as Advogato and FilmTrust. These two datasets are selected to design and implement several experiments to evaluate the efficiency and performance of the DWHS.

A. EXPERIMENTAL DESIGNS

The evaluation method, datasets, evaluation metrics and comparison experimental algorithm are introduced in this section.

1) EVALUATION METHODS

The leave-one-out method is usually used to train and test machine learning classifiers and is adopted to validate the effectiveness of the DWHS. More precisely, one direct trust value between two nodes is randomly hidden and treated as the real value. Then, the trust value between those two nodes is inferred through trust network with the DWHS.

2) DATASETS

The Advogato and FilmTrust datasets were selected to evaluate the accuracy of the DWHS. Users can authenticate each other at four different trust levels in Advogato: Observer, Apprentice, Journeyer and Master. To have confidence in the true value, the experiments assign 0.2 to the Observer, 0.4 to Apprentice, 0.6 to Journeyer, and 0.8 to master. FilmTrust allows users to maintain lists of friends and evaluate the degree of the ratings or comments on a movie. Users rate their friends on a sequence from 1 to 10 (1: least trustworthy, 10: most trustworthy). The data are normalized and processed accordingly, that is, the trust value $\tau_{ij} \in [0, 1]$.

3) EVALUATION METRICS

The accuracy metrics are shown in Table 1, which include the MAE (Mean Absolute Error, MAE), Precision, Recall and F-$score$. Let T_A be the number of edges through which v_s directly trusts v_t, and T_B be the number of edges that v_s estimates that trust v_t through the DWHS.

TABLE 1. Accuracy metrics.

Metrics	Computing equations
MAE	$\mathrm{MAE} = \frac{\sum_{e_{ij} \in E} \lvert value_{infer} - value_{real} \rvert}{\lvert E \rvert}$
Precision	$Precision = \frac{T_A \bigcap T_B}{T_B}$
Recall	$Recall = \frac{T_A \bigcap T_B}{T_A}$
F-$score$	$F\text{-}score = \frac{2 \times Recall \times Precision}{Recall + Precision}$

4) METHODS FOR COMPARISON

To demonstrate the accuracy of the DWHS, the comparison experiments are conducted with the WHST proposed by Wei and Tong [26], the classic TidalTrust proposed by Golbeck *et al*., the MoleTrust proposed by Massa *et al*., and the A^* algorithm proposed by Hart.

B. RESULTS AND ANALYSIS OF EXPERIMENTS

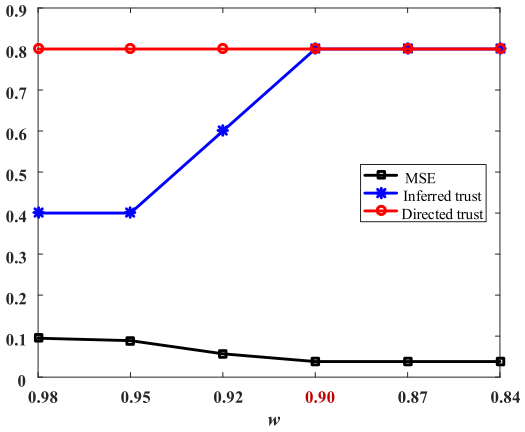
In order to evaluate the performance of the proposed algorithm, some experiments and corresponding analysis are provided in this section.

1) PARAMETER W SETTING

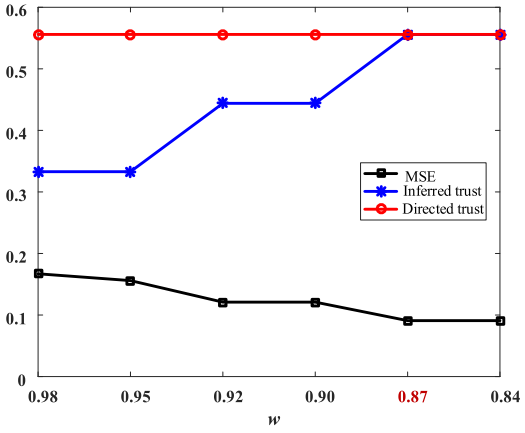
This experiment aims to study the influence of parameter w in the DWHS. Different results are generated with different w for the selection of the candidate nodes in the FOCAL list, which affect the final trust prediction. Figure 4 shows the effect of different w in the same trust network.

In the Advogato datasets, when the parameter $w \in [0.95, 0.98]$, no node with a small difference from the optimal node was inserted into the FOCAL list. Therefore, the prediction result was not ideal. When $w \in (0, 0.9]$, the MSE did not change. This finding indicated that there were already optimal nodes in the FOCAL list. If parameter w was continuously reduced, the comparison of some different H_f and time would increase, but the experiment results would not change at all. Therefore, the effect of the trust prediction is better when w is 0.9 in the Advogato datasets.

Similarly, in the FilmTrust datasets, nodes with a small difference from the optimal nodes were not inserted into the FOCAL list when parameter $w \in [0.95, 0.98]$. When $w \in (0, 0.87]$, the MSE did not change. It indicated that the optimal nodes already existed in the FOCAL list, and the experiment results would not change at all. Therefore, $w = 0.87$ has a better effect on trust prediction.



(a)The effect of parameter w on the Advogato datasets.



(b)The effect of parameter w on the FilmTrust datasets.

FIGURE 4. Impacts of parameter w in the DWHS. In (a), when w is 0.9, the effect of the trust prediction is better in the Advogato datasets. In (b), $w = 0.87$ has a better effect on trust prediction in the FilmTrust datasets.

Therefore, if parameter w is too high, it may not include the optimal or suboptimal trust path. If parameter w is too low, it can include the optimal or suboptimal trust path and meanwhile many nodes would be selected from the OPEN list to insert into the FOCAL list due to the relaxed restriction. This would lead to the rise of the time complexity because it requires comparing nodes to pick one with the minimum MSE. Furthermore, with the increase of the restriction factor w, since the node with the minimum MSE has been included, the results of the experiment do not change. According to different cases, parameter w would take a responding value. In conclusion, the trust value with a small fluctuation which means a small MSE value is beneficial to improving the accuracy of the trust prediction. Because there is a small possibility of noise in the path, the value of parameter w is usually closely related to the difference in the trust value. This is in line with practical applications of the recommendation systems.

2) COMPARISON EXPERIMENTS

The performance and efficiency of the DWHS and other trust inference algorithms in inferring trust values were evaluated according to the evaluation metrics. The comparison methods that are selected are as follows: the A^* algorithm, the WHST, TidalTrust and MoleTrust.

Among them, the A^* algorithm is a classical heuristic search algorithm. The WHST combines the A^* algorithm with trust inference, zooms in on the maximum value of the trust path in a fixed weighted way, and introduces the trust decay factor d. In the comparison experiments, $w = 0.9$ and $d = 0.5$ are the best values for the Advogato datasets according to the experiment experience. Meanwhile, $w = 0.85$ and $d = 0.5$ have the best effect for the FilmTrust datasets. TidalTrust and MoleTrust are classic trust propagation algorithms. TidalTrust can automatically generate a trust threshold according to the strength of the trust path while MoleTrust needs a predefined trust threshold and maximum depth. Based on the value from the experimental experience, the trust thresholds of Advogato and FilmTrust are set as 0.5 and 0.3 in the compared experiments, respectively, and the maximum depth is6.

The experiment randomly selects 2000 data from the Advogato and FilmTrust datasets, respectively. Therefore, the above four algorithms and two datasets are selected for the comparison experiments in this article, which can verify the performance and efficiency of the DWHS.

From Figure 5, for the single-path trust prediction, the A^* algorithm can always find the optimal trust path and expand the optimal nodes every time. Therefore, the DWHS and A^* algorithm have the same MAE and *F-score* on the Advogato datasets, which indicates that the solution obtained by the DWHS is optimal. Meanwhile, the A^* algorithm cannot accept the suboptimal solution. This algorithm always fluctuates among the nodes limited in a small difference, which leads to the algorithm having a longer running time than that of the DWHS. Moreover, in the FilmTrust datasets, although the MAE of the DWHS is slightly worse than that of the A^* algorithm, the suboptimal solution is more quickly obtained, and the running time is significantly less than that of the A^* algorithm, which improves the search efficiency dramatically.

The WHST fixes the weight of the zoom function. In contrast, the DWHS assigns nodes at different depths different weights. When the search path is deeper, the weights can be dynamically adjusted. Therefore, except for the running time, the experimental results show that the DWHS is better than the WHST. Since the WHST only looks for the optimal path among the shortest paths, the running time is slightly better than that of the DWHS.

For multipath trust prediction, TidalTrust and MoleTrust may lead to the loss of the optimal paths by limiting the trust paths. The DWHS takes into account the distribution of the trust value of the whole path and avoids the robustness of the weighted average function adopted by TidalTrust and MoleTrust. Therefore, each metric of the DWHS is clearly better than TidalTrust and MoleTrust.

Therefore, compared with other algorithms, when it is applied to the recommendation systems, the DWHS can not

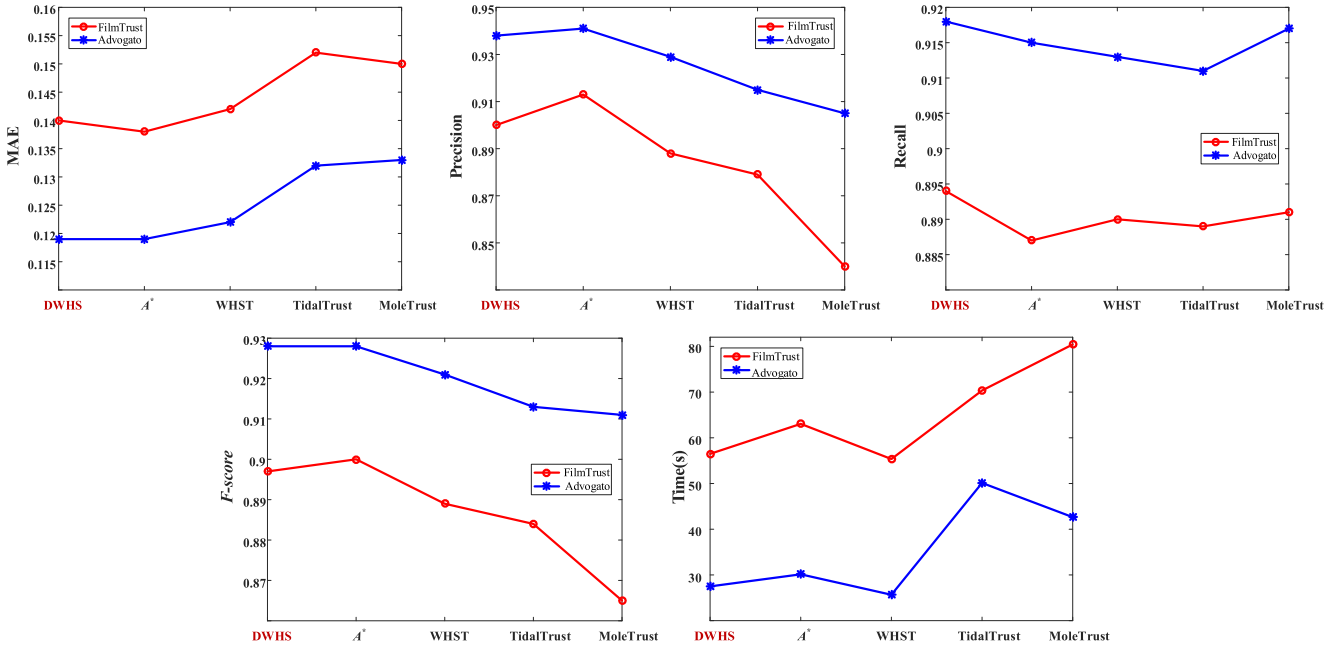


FIGURE 5. Performance of the DWHS under comparison based on two datasets, evaluated based on five different metrics.

only provide accurate and high-quality recommendations, but also improve the efficiency of recommendations.

C. SUMMARY OF EXPERIMENTS

Experiments show that the DWHS can effectively improve the performance and efficiency of trust prediction. The path length is limited to the range of six hops according to the six-degree space theory to find the admissible trust paths. The value of parameter w is usually closely related to the difference in the trust values among paths. By adjusting parameter w, the fault-tolerance effect can be dynamically changed. This path is robust and has a relatively high trust value, which is beneficial to trust inference. In addition, through some comparison experiments, the DWHS has more advantages than other algorithms, and the searching time is relatively reduced.

Therefore, the DWHS can help users make accurate decisions by predicting the trust among users. It can be used in many applications, such as recommendation systems, which provide users with high-quality recommendations on different categories of products or services according to their preferences. It not only saves search time and improves search efficiency, but also improves the accuracy of prediction.

VI. CONCLUSION

With the widespread application of recommendation technology in e-commerce systems, increasing attention has been devoted to the research on the precision and quality of trust recommendation systems. This article draws on the heuristic search A^* algorithm, comprehensively considers the role of the node depth in trust path, and proposes the DWHS. According to the depths of the nodes on the search path, the dynamic weighting condition w is used to dynamically relax the evaluation function $f(n)$, and then the secondary heuristic factor is used to select nodes to ensure the stability of trust paths so as to predict trust with high accuracy. The experimental results show that the DWHS has better performance than those of existing algorithms.

The DWHS could be further improving through more advance dynamic factors in the future and applied to recommendation systems. Additionally, it can be extended to find two or more trust paths, and when one of the trust paths fails, the suboptimal path will be used as the alternate path for trust propagation. It should be used to design a new framework to find multiple reliable paths and incorporate them in the main future work.

ACKNOWLEDGMENT

The authors are grateful to all of the reviewers for suggestions and insights that improved the paper.

REFERENCES

[1] J. Wu, X. Li, F. Chiclana, and R. Yager, "An attitudinal trust recommendation mechanism to balance consensus and harmony in group decision making," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 11, pp. 2163–2175, Nov. 2019.
[2] Y. Li, J. Liu, and J. Ren, "A novel learning model based on trust diffusion and global item for recommender systems," *IEEE Access*, vol. 7, pp. 170270–170281, 2019.
[3] Y. A. Kim and H. S. Song, "Strategies for predicting local trust based on trust propagation in social networks," *Knowl.-Based Syst.*, vol. 24, no. 8, pp. 1360–1371, Dec. 2011.
[4] W. Jiang, G. Wang, and J. Wu, "Generating trusted graphs for trust evaluation in online social networks," *Future Gener. Comput. Syst.*, vol. 31, pp. 48–58, Feb. 2014.
[5] X.-R. Tong, W. Zhang, and Y. Long, "Transitivity of agent subjective trust," *J. Softw.*, vol. 23, no. 11, pp. 2862–2870, Jan. 2014.

[6] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on Epinions.com community," in *Proc. 20th Nat. Conf. Artif. Intell.*, Jul. 2005, pp. 121–126.
[7] J. Golbeck and J. Hendler, "FilmTrust: Movie recommendations using trust in Web-based social networks," in *Proc. CCNC. 3rd IEEE Consum. Commun. Netw. Conf.*, Jan. 2006, pp. 282–286.
[8] H. N. Nsaif Al-Sammarraie and D. N. A. Jawawi, "Multiple black hole inspired meta-heuristic searching optimization for combinatorial testing," *IEEE Access*, vol. 8, pp. 33406–33418, 2020.
[9] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
[10] K. Gu, L. Wang, and B. Yin, "Social community detection and message propagation scheme based on personal willingness in social network," *Soft Comput.*, vol. 23, no. 15, pp. 6267–6285, Aug. 2019.
[11] Y. Xu and F. Zhang, "Detecting shilling attacks in social recommender systems based on time series analysis and trust features," *Knowl.-Based Syst.*, vol. 178, pp. 25–47, Aug. 2019.
[12] T. Li, W. Liu, T. Wang, Z. Ming, X. Li, and M. Ma, "Trust data collections via vehicles joint with unmanned aerial vehicles in the smart Internet of Things," *Trans. Emerg. Telecommun. Technol.*, no. 7, pp. 1–24, 2020, doi: 10.1002/ett.3956.
[13] B. Jiang, G. Huang, T. Wang, J. Gui, and X. Zhu, "Trust based energy efficient data collection with unmanned aerial vehicle in edge network," *Trans. Emerg. Telecommun. Technol.*, no. 8, pp. 1–32, 2020, doi: 10.1002/ett.3942.
[14] Y. Ren, Z. Zeng, T. Wang, S. Zhang, and G. Zhi, "A trust-based minimum cost and quality aware data collection scheme in P2P network," *Peer Peer Netw. Appl.*, 2020, doi: 10.1007/s12083-020-00898-2.
[15] T. Wang, H. Luo, X. Zeng, Z. Yu, A. Liu, and A. K. Sangaiah, "Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 19, 2020, doi: 10.1109/TITS.2020.2997377.
[16] C. Jiang, S. Liu, Z. Lin, G. Zhao, R. Duan, and K. Liang, "Domain-aware trust network extraction for trust propagation in large-scale heterogeneous trust networks," *Knowl.-Based Syst.*, vol. 111, pp. 237–247, Nov. 2016.
[17] C. Mao, C. Xu, and Q. He, "A cost-effective algorithm for inferring the trust between two individuals in social networks," *Knowl.-Based Syst.*, vol. 164, pp. 122–138, Jan. 2019.
[18] M. Ghavipour and M. R. Meybodi, "Trust propagation algorithm based on learning automata for inferring local trust in online social networks," *Knowl.-Based Syst.*, vol. 143, pp. 307–316, Mar. 2018.
[19] M. Ghavipour and M. R. Meybodi, "A dynamic algorithm for stochastic trust propagation in online social networks: Learning automata approach," *Comput. Commun.*, vol. 123, pp. 11–23, Jun. 2018.
[20] I. Pohl, "Heuristic search viewed as path finding in a graph," *Artif. Intell.*, vol. 1, nos. 3–4, pp. 193–204, Jan. 1970.
[21] I. Pohl, "The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving," in *Proc. 3rd Int. Joint Conf. Artif. Intell.*, San Mateo, CA, USA: Morgan Kaufmann, 1973, pp. 12–17.
[22] J. Pearl and J. H. Kim, "Studies in semi-admissible heuristics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 4, pp. 392–399, Jul. 1982.
[23] Y. F. Yiu, J. Du, and R. Mahapatra, "Evolutionary heuristic A* search: Heuristic function optimization via genetic algorithm," in *Proc. IEEE 1st Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Sep. 2018, pp. 25–32.
[24] R. Stern, G. Dreiman, and R. Valenzano, "Probably bounded suboptimal heuristic search," *Artif. Intell.*, vol. 267, pp. 39–57, Feb. 2019.
[25] K. Mi, J. Zheng, Y. Wang, and J. Hu, "A multi-heuristic A* algorithm based on stagnation detection for path planning of manipulators in cluttered environments," *IEEE Access*, vol. 7, pp. 135870–135881, 2019.
[26] T. Wei and X. Tong, "Robust trust path generation based on weighted heuristic search," *J. Nanjing Univ.*, vol. 54, no. 6, pp. 111–120, 2018.

RU KONG was born in 1996. She is currently pursuing the master's degree with the School of Computer and Control Engineering, Yantai University. Her main research interests include multi agent systems and propagation models of trust.

XIANGRONG TONG was born in 1975. He received the Ph.D. degree in computer science and technology from Beijing Jiaotong University. He is currently a Full Professor with Yantai University. His research interests include computer science, intelligent information processing, and social networks. He was a member of CCF.