

Research Article

Efficient Data Transmission for Community Detection Algorithm Based on Node Similarity in Opportunistic Social Networks

Aizimaiti Xiaokaiti,^{1,2} Yurong Qian ,^{1,2} and Jia Wu ³

¹Software College, Xinjiang University, Urumqi 830000, China

²Key Laboratory of Signal Detection and Processing in Xinjiang Uygur Autonomous Region, Xinjiang University, Urumqi 830046, China

³School of Computer Science and Engineering, Central South University, Changsha 410083, China

Correspondence should be addressed to Yurong Qian; qyr@xju.edu.cn and Jia Wu; jiawu5110@163.com

Received 11 March 2021; Revised 13 April 2021; Accepted 17 April 2021; Published 29 May 2021

Academic Editor: Fei Xiong

Copyright © 2021 Aizimaiti Xiaokaiti et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of 5G era, the number of messages on the network has increased sharply. The traditional opportunistic networks algorithm has some shortcomings in processing data. Most traditional algorithms divide the nodes into communities and then perform data transmission according to the divided communities. However, these algorithms do not consider enough nodes' characteristics in the communities' division, and two positively related nodes may divide into different communities. Therefore, how to accurately divide the community is still a challenging issue. We propose an efficient data transmission strategy for community detection (EDCD) algorithm. When dividing communities, we use mobile edge computing to combine network topology attributes with social attributes. When forwarding the message, we select optimal relay node as transmission according to the coefficients of channels. In the simulation experiment, we analyze the efficiency of the algorithm in four different real datasets. The results show that the algorithm has good performance in terms of delivery ratio and routing overhead.

1. Introduction

With the booming of information technology and the popularization of wireless network equipment [1], people have a growing demand for the network. As a fresh type of self-organizing network [2], an opportunistic social network has attracted researchers' attention [3]. There is no complete end-to-end path between nodes in opportunistic social networks [4]; it uses the encounter opportunities brought by node movement to communicate hop by hop [5]. At present, opportunistic social network has widespread use in various fields, such as mobile phones [6], handheld electronic devices [7], vehicular networks with mobile intelligent devices on the road [8], wildlife tracking [9], and network transmission in remote areas [10].

The traditional social network method to deal with data transmission faces significant challenges [11], which will become an obstacle to the information exchange and sharing [12]. To enhance data transmission in a 5G wireless network

[13], we should design a more convenient model to achieve data forwarding flexibly [14]. The user terminal equipment needs to transmit a large amount of data and needs to calculate these intensive tasks [15]. To enhance wireless devices' computer ability, mobile edge computing (MEC) is proposed [16–18]. Because the mobile edge server locates at the edge of the wireless network and closer to the users, it can efficiently provide the surrounding users' services and integrate the concept of opportunistic social networks into mobile edge computing, to reduce the consumption of source nodes [19].

However, each node has many social attributes [20]. They represent the relationship among different users, and the connections between nodes in the same community are more than closer [21]. So, the network nodes can be divided into communities by their different attributes to improve the algorithm's performance [22]. The existing algorithms do not fully consider nodes' characteristics, so there is a large space for improvement in community detection accuracy

and efficiency [23]. That is why it is necessary to propose an efficient community detection algorithm.

Opportunistic social network uses the strategy of “storing-carrying-forwarding” to handle the energy consumption problem in the data transmission process [24]. Messages are forwarded through encounter opportunities produced by node movement. In this paper, the network topology attributes and social attributes are used to measure the similarity between nodes, and the hierarchical clustering method effectively divides the community [25]. In the process of data transmission, if the mobile device does not have a suitable transmission target, the message will occupy a lot of cache, and the data transmission in the community is likely to wait a long time and cause the delay in transmission [26]. After dividing the community, we need to further establish the weight distribution between nodes and community to reduce the time complexity and overhead cost and construct a set of candidate relay nodes based on the relationship between information forwarders and adjacent nodes. From the perspective of minimizing bit error rate, the channel coefficients of the two channels from the source node to the relay node and the relay node to the destination node are analyzed. This must select the optimal relay node from the set of candidate relay nodes as transmission. In summary, we propose an efficient data transmission strategy for community detection in opportunistic social network using mobile edge computing combined with network topology and social attributes. The transmission strategy is divided into two periods: the initialization period and the routing period.

The contributions of this research study are as follows:

- (1) Initialization period: using network topology attributes and social attributes to measure the similarity between nodes, a community detection algorithm is proposed through hierarchical clustering.
- (2) Routing period: based on the relationship between the message forwarder and the adjacent nodes, a set of candidate relay nodes is constructed. By analyzing the channel coefficients of the source node to the relay node and the relay node to the destination node, a method for selecting the optimal relay node is proposed.
- (3) Simulation results show that the algorithm EDCCD proposed in this paper has good performance such as delivery ratio, routing overhead, and average end-to-end delay in different real datasets.

2. Related Works

Many researchers have conducted research on routing and forwarding algorithms in opportunistic social networks and proposed very effective approaches in different application scenarios in recent years. Many research methods have focused on algorithm research. Routing algorithms can be roughly divided into two sorts: existing social-ignorant algorithms and existing social-aware algorithms [27].

Existing social-ignorant algorithms mean that social message relating to nodes will not make adaptable messaging

decisions in the process of data transmission. Vahdat and Becker [28] proposed the epidemic routing algorithm. Epidemic algorithm is essentially a flooding algorithm, and each node forwards information to all its neighbors. However, there are a lot of message copies in the network, which will consume many network resources. Sisodiya et al. [29] proposed a flood routing algorithm, that is, spray and wait algorithm, which divides the information forwarding process into two steps. The first step is to copy the message and the transmission process is in the second step. It can easily lead to ultratransmission delay and data redundancy.

Sharma et al. [30] proposed a routing protocol named MLProph, which uses machine learning (ML) algorithms, namely, decision trees and neural networks, to determine the probability of successful message delivery, but this algorithm has great limitations. Tang et al. [31] proposed a scheme based on reinforcement learning (RL), which can apply to opportunistic routing transmissions that require high reliability and low latency. However, this opportunistic routing scheme can only be used for specific scenarios and is not for all networks. Wu et al. [32] proposed the algorithm that adjusts the cache by analyzing the importance of message propagation. This algorithm has a small routing overhead, but to avoid deleting the cached data, the data shares by adjacent nodes will cause data redundancy.

Social-aware algorithms refer to the social relationship between nodes to measure the transmission relevance between nodes. Yan et al. [33] established an effective data transmission strategy (ENPSR), which uses the priority of nodes and social relationships in opportunistic social networks. Obtain the data transmission priority by measuring the social attributes and historical information of the node. Then use the forecast plan to determine the appropriate message delivery decision. Wu and Chen [34] proposed an optimal routing scheme for cooperative nodes based on opportunistic network features. This scheme can use in social networks. By reliability, availability, and weighting factors are used as the weights of human activities to obtain the optimal cooperative node, but the algorithm has a high routing overhead. Drăgan et al. [35] proposed that nodes can be divided into several communities according to their intimacy and the time together. This community detection method does not fully consider all of the nodes in the community.

Zeng et al. [36] proposed a social-based clustering and routing scheme, in which each node selects the nodes with close social relationships to form a local cluster, but this can cause data redundancy issues. Liu et al. [37] proposed an algorithm using node similarity (FCNS) based on fuzzy routing and forwarding. This algorithm has good performance in data transfer ratio and routing overhead but high transmission delay. Niu et al. [38] proposed a predictive and extended routing protocol, which uses Markov chain as a node mobility model to realize the social characteristics of nodes. It does not consider node communication between different places, and nodes just upload and send message in the same place.

Because the abovementioned traditional methods do not fully consider node characteristics and other problems, this

paper proposes a model that combined with the network topology and social attributes to detect community and analyze the channel coefficients of source node to relay node and relay node to destination node to select optimal relay node as information transmission in opportunistic social networks. This model can effectively handle the challenge of improving data transmission and has good performance of low delay and low routing overhead.

3. Model Design

In opportunistic social networks, we can define the topological structure $G = (V, E, w)$, where V is the node of the network and E is the edge set in the network reflecting the relationship between the nodes. $E = \{(m, n) | m \in V, n \in V\}$, m and n are nodes, and w is the weight of the edges of node m and node n . On the basis of the division of the community, we make $C = \{C1, C2, \dots, Cn\}$, which require more edges between vertices in each community subgraph. We consider that there will be differences between nodes and the number of encounters between nodes to weight each edge. This paper proposes to measure the similarity between different nodes in terms of network topology attributes and social attributes. The greater the similarity is between nodes, the more likely they are to belong to the same community.

Firstly, we must reasonably define the similarity between nodes. For a real social network, and considering the network topology, we also need to consider the social attributes between nodes. We must collect the data of the node, and the process is shown in Figure 1. The nodes information collection method is that the base station collects all node information in the area within a period of time. When the node has a transmission task, request the probability table of the source node and the destination node from the base station that has collected the information and use edge computing to transmit decision information to reduce node's workload. Because many communities can usually only share messages based on one or two nodes, there must be enough cache to improve data transmission efficiency. The node requires obtaining the position, speed, and moving direction of itself and the destination node. However, the encounter of nodes in opportunistic social networks is random. Combining the characteristics of node movement to calculate the probability of node encounters, in this paper, PE_{mn} means the probability of nodes m and n meeting in a period of time t , and the node meeting interval time obeys the exponential distribution; then the probability of node m and node n meeting within the sensing range is

$$PE_{mn}(t) = 1 - e^{-\lambda_{mn}(t)}, \quad (1)$$

where m is the source node, n is the destination node, $\lambda_{mn} = (1/\Delta t_{mn})$ is the encounter strength of node m and node n , and Δt_{mn} is the average time between node m and node n :

$$\Delta t_{mn} = \frac{1}{n} \sum_{k=0}^n (t_{mn}^{k+1} - t_{mn}^k), \quad (2)$$

where t_{mn}^k is the time of the k th encounter, and we define $t_{mn}^0 = 0$. In short, combine the formula to get

$$PE_{mn}(t) = 1 - e^{-\lambda_{mn}(t)} = 1 - \exp\left[\frac{-n(t_{\text{init}} - t_0)}{\sum_{k=0}^n (t_{mn}^{k+1} - t_{mn}^k)}\right], \quad (3)$$

where $t_r = t_{\text{init}} - t_0$ is the remaining time to live of the message, t_{init} is the initial time to live of the message, and t_0 is the current time the message has been alive.

Secondly, construct the encounter probability matrix. The number of encounters between nodes to a certain extent only reflects the number of encounters of the node in a period of time.

$$MT = \begin{pmatrix} PE_{11} & \dots & PE_{1n} \\ \vdots & \ddots & \vdots \\ PE_{m1} & \dots & PE_{mn} \end{pmatrix}, \quad (4)$$

Use the number of encounters between nodes to weight each edge. $w = \{w_{mn} | e_{mn} \in E\}$ is the set of edge weights, where w_{mn} is the number of encounters between two nodes. $MT = \{PE_{mn}(t)\}$ represents the $n * n$ encounter matrix of node m and node n in a period of time t .

$$w_{mn} = PE_{mn}(t) \times \sum_{\text{on}}^n w_{m,\text{on}}, \quad (5)$$

where $w_{m,\text{on}}$ represents the number of encounters the node m has met with other nodes within a certain period of time.

In opportunistic social networks, network topology attributes reflect the status of the network. It requires more edges between the vertices in each community subgraph.

- (1) The strength of nodes describes how close the node is to the surrounding network, and the node strength is equal to the degree of the node, that is, the number of neighbor nodes. The defined formula is

$$ND_{(m,n)} = \frac{|NC_m - NC_n|}{|NC_m + NC_n|}, \quad (6)$$

where $ND_{(m,n)}$ is the node connection strength between node m and node n . NC_m is the set of neighbor nodes connected to a node m in current times, and NC_n is the set of neighbor nodes connected to a node n in current times. We have to consider that two nodes may share a set of similar neighbor nodes, so the higher the relationship between them, the higher the probability of data transmission.

- (2) The direct connection strength represents the influence of the direct connection between two nodes. When there is an edge between two nodes, the edge weight measures the strength of the connection between them. We define the sum of the weights of all edges adjacent to node m as $s(m) = \sum_{n \in \theta(m)} w_{mn}$, where $\theta(m)$ is the set of neighbor nodes of m . For any $\theta(m)$, there is a relationship between node m and node n . So the formula for direct connection strength is as follows:

$$DC_{(m,n)} = \frac{w_{mn}}{s(m) + s(n) - w_{mn}}, \quad (7)$$

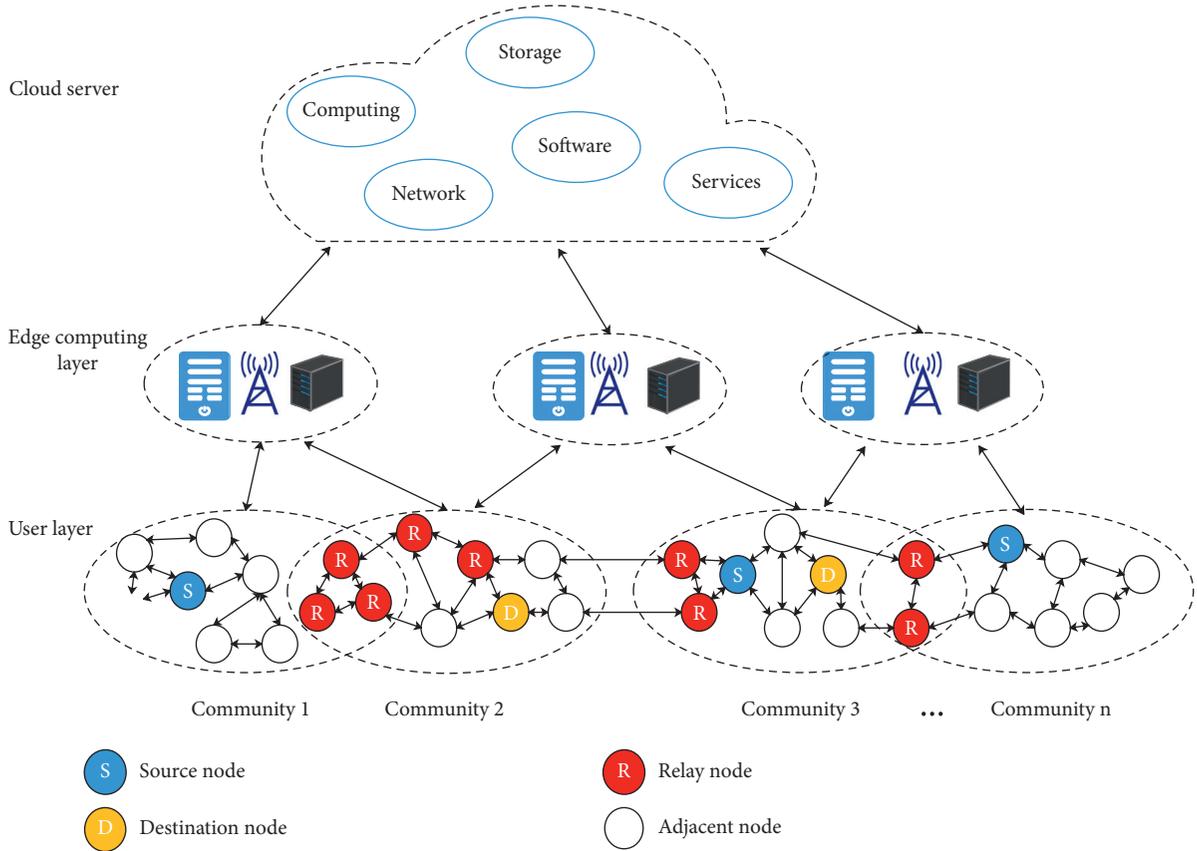


FIGURE 1: Schematic diagram of community node information.

where $DC_{(m,n)}$ is the strength of the direct connection between two nodes and is also the ratio of the weight of the two nodes to the weight of their adjacent edge.

- (3) The indirect connection strength indicates the influence of the indirect connection between two nodes; just as when node m and node n have a common adjacent node p , then node m and node n also have a certain chance to connect. The more adjacent nodes that two nodes have in common, the closer the two nodes are. So the formula for indirect connection strength is as follows:

$$IC_{(m,n)} = \sum_{p \in s(m) \cap s(n)} \frac{w_{mp} + w_{np}}{s(m) + s(n) - w_{mn}}, \quad (8)$$

where w_{mp} and w_{np} represent the connection strength between node m and node n through node p , and the indirect connection strength between nodes is the sum of the strengths of all common neighbor connections. That is to say, the more common adjacent nodes the two nodes have, the greater the indirect connection strength is.

In the network topology attributes, we classify the possible relationships between two nodes into the following four types, where we use $SMD_{tp(m,n)}$ to express topological similarity between node m and node n .

- (a) No direct and no indirect connection:

$$SMD_{tp(m,n)} = \alpha ND_{(m,n)}. \quad (9)$$

- (b) Indirect but no direct connection:

$$SMD_{tp(m,n)} = \eta IC_{(m,n)} + \alpha ND_{(m,n)}. \quad (10)$$

- (c) Direct but no indirect connection:

$$SMD_{tp(m,n)} = \varphi DC_{(m,n)} + \alpha ND_{(m,n)}. \quad (11)$$

- (d) Direct and indirect connection:

$$SMD_{tp(m,n)} = \varphi DC_{(m,n)} + \eta IC_{(m,n)} + \alpha ND_{(m,n)}, \quad (12)$$

where α is the coefficient of the strength of node, φ is the coefficient of the direct connection strength, and η is the coefficient of the indirect connection strength. The higher the topological similarity between nodes, the greater the chance of communication between nodes, which can improve data transmission efficiency.

The social attributes between nodes measure the social similarity between two nodes.

- (1) The geographic relevance of nodes: the node has mobile characteristics; the mobile node's trajectory information is used to analyze the geographic location correlation of the node. The trajectory

information refers to the geographic location information of the sensing area. The sensing area is the area where the node can transmit messages within a certain range. Specifically, in the time period T , if the nodes' geographical locations are close, it means that the probability of node information transmission is high; that is to say, the probability of meeting in the same area will also be increased. The geographical correlation between nodes can be expressed as

$$GD_{(m,n)} = \frac{\sum_{m=1}^k \sum_{n=1}^k A(R_m^r, R_n^r)}{T}, \quad (13)$$

where $GD_{(m,n)}$ is the geographic relevance of nodes, $A(R_m^r, R_n^r)$ represents the similarity function of node m and node n at position A , R_m^r represents rm th trajectory information of node m , and R_n^r represents rn th trajectory information of node n .

$$A(R_m^r, R_n^r) = \max\{E_m^r, E_n^r\} - \min\{Q_m^r, Q_n^r\}, \quad (14)$$

where $\max\{E_m^r, E_n^r\}$ takes the maximum value between E_m^r and E_n^r , E_m^r is the time when node m enters the sensing area for the rm th time, and E_n^r is the time when node n enters the sensing area for the rn th time. $\min\{Q_m^r, Q_n^r\}$ represents take the minimum value between Q_m^r and Q_n^r , Q_m^r is the time when node m quits the sensing area for the rm th time, and Q_n^r is the time when node n quits the sensing area for the rn th time.

- (2) The interesting relevance of nodes: users with common interests will visit the same business. Naturally, mobile users with the same interests will spend more time and energy communicating together. The information transmission between nodes will be carried out between mobile users with the same interest in the time period T . The interesting relevance between nodes can be expressed as

$$IR_{(m,n)} = \frac{\sum_{k=1}^{k_n} T_{m,n}^k}{\sum_{k=1}^{k_n-1} T_{m,otn}^{k-1}}, \quad (15)$$

where $IR_{(m,n)}$ represents the interesting relevance between node m and node n . $T_{m,n}^k$ represents the ratio of time occupied by node m and node n during the k th transmission of information in time period T . $T_{m,otn}^{k-1}$ represents the ratio of the time occupied by node m and other nodes except node n in the k -1th transmission information in time period T .

- (3) The separating time relevance of nodes: two nodes can make a connection and communicate. The average interval between two nodes can be defined as the time interval when two nodes meet each other. If there is no communication for a long time, the relationship between the two nodes is not close enough. Conversely, a shorter separation means that the two nodes are closely related. The separating time relevance of nodes can be expressed as

$$AS_{(m,n)} = \frac{T_{m,n}^k - T_{m,n}^1}{k \times T}, \quad (16)$$

where $AS_{(m,n)}$ represents the separate time relevance of node m and node n to convey information. $T_{m,n}^k$ is the time of the k th transmission of information in the time interval T . $T_{m,n}^1$ is the time of the first transmission of information in the time interval T .

Through the above calculation of social attribute values, we can quantify the relationship between node m and node n . $SR_{(m,n)}$ represents the similarity of social attributes as follows:

$$SR_{(m,n)} = \beta GD_{(m,n)} + \delta IR_{(m,n)} + \rho AS_{(m,n)}, \quad (17)$$

where β is the coefficient of the geographic relevance of nodes, δ is the coefficient of the interesting relevance of nodes, and ρ is the coefficient of the separating time relevance of nodes. The higher the node's social attribute value, the higher the closeness between the nodes and the higher the probability of encountering communication, which will improve the efficiency of information transfer between nodes.

Node similarity is affected by the network topology and social attributes. $NS_{(m,n)}$ represents the similarity between node m and node n . Correspondingly, in this paper, we define node similarity to be composed of network topology and social attributes, and the node similarity formula is

$$NS_{(m,n)} = \phi SMD_{tp(m,n)} + (1 - \phi) SR_{(m,n)}. \quad (18)$$

Through the above description, we can know the relationship between nodes more accurately. The higher the node similarity, the more frequent the communication between nodes. Source node can accurately find the relay node and then transmit information to the destination node by establishing a community [39]. The information transmission in this process is more efficient, and the time delay reduces.

The nodes within the same community are closely connected. Community detection is essentially the clustering of nodes with a tight structure in the network. This paper uses a hierarchical clustering algorithm to divide the community. Lead in modularity Q , which is used to measure the degree of community division. The fast unfolding algorithm considering data scale, running time, and other aspects of the community division results is ideal. The algorithm is stable and will continuously merge nodes to construct new graphs, which significantly reduces the calculation amount. The algorithm steps are as follows:

Step 1: initialize and calculate the node similarity; divide each node into the community where the adjacent node is located. As shown in Figure 2, the source node S is in community one. We try to move the node S to community two and community three. Calculate the corresponding modularity value, and move the node S to the corresponding community with the largest change value. We lead in modularity Q to measure the

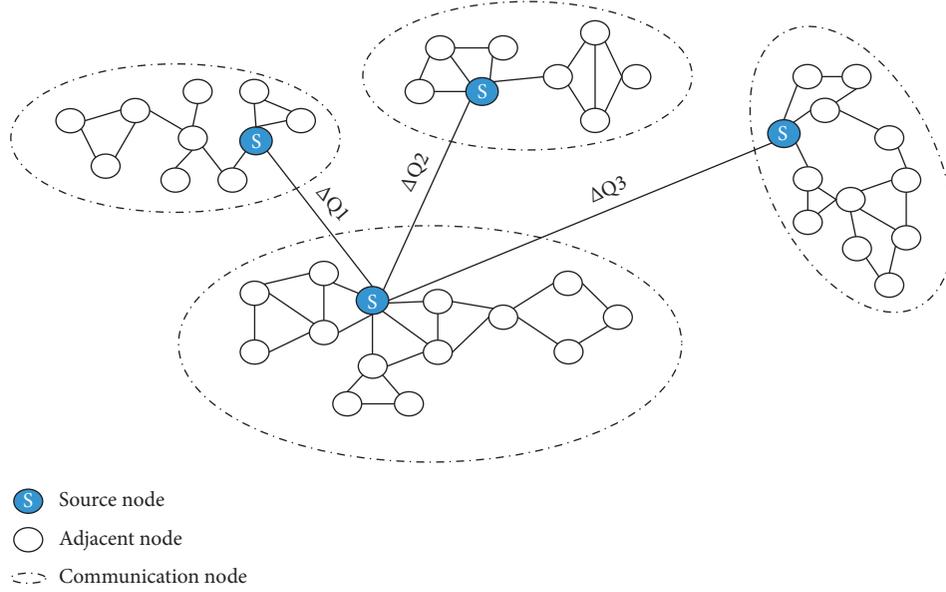


FIGURE 2: Schematic diagram of nodes moving between communities.

degree of community division. The specific calculation formula is as follows:

$$Q = \sum c \frac{\sum \text{in}}{2l} - \left(\frac{\sum \text{tot}}{2l} \right)^2, \quad (19)$$

where Q is the modularity, $\sum \text{in}$ represents the number of connections within the community, $\sum \text{tot}$ represents the sum of degrees of all nodes in the community, and l is the sum of weights in the network.

Step 2: select each node one by one, and calculate the modularity gain divided into the community where the adjacent point is located. ΔQ represents modularity gain, and the calculation formula is as follows:

$$\Delta Q = \left[\frac{\sum \text{in} + 2k_{m,\text{in}}}{2l} - \left(\frac{\sum \text{tot} + k_m}{2l} \right)^2 \right] - \left[\frac{\sum \text{in}}{2l} - \left(\frac{\sum \text{tot}}{2l} \right)^2 - \left(\frac{k_m}{2l} \right)^2 \right], \quad (20)$$

where $k_{m,\text{in}}$ is the sum of weights from node m to the community and k_m is the sum of the weights of node m . After calculating the modularity gain, we have to determine whether it is a positive number; if it is a positive number, it will be divided into the corresponding community; otherwise, no division will be made.

Step 3: repeat Step 2 until the node's community no longer changes.

Step 4: construct a new graph; each point in the new graph is each community divided in Step 3; continue to execute until the community structure does not change.

This paper roughly divides the above algorithm steps into two stages:

Stage 1: divide each node into the community where the adjacent node is located so that the modularity value becomes more immense.

Stage 2: the communities divided in the first stage are aggregated into one point, and the network is reconstructed until the structure of the network no longer changes.

This paper draws on the hierarchical clustering idea of the fast unfolding algorithm. We use network topology attributes and social attributes to express node similarity and comprehensively calculate node similarity to update network weights. In the first stage of node merging, we form an initial community to merge and improve the overall modularity and then calculate modularity gain; if ΔQ is positive then the two communities are merged; otherwise they will not be merged. The modularity gain is calculated repeatedly, and the final division result is output.

Nodes have the characteristics of random movement, and it is vital to establish a community. In opportunistic social network, many communities can usually deliver messages based on only one or two nodes. If these nodes do not have enough cache or overhead, data transmission in the community is likely to wait a long time. Therefore, after we divide the community, we need to establish further the weight distribution between the nodes and the community reconstruction so as to reduce the time complexity and overhead cost better. Below we will prove the changes in the community of the source node during the movement.

We define at time t , Q is the degree of modularity of the community, E_{we} is the total weight of weight, E_{wec} is total weight of the edges of community c , D_m is the degree of node m in community c , and ΔE_{we} is the increment of edge weight.

$$Q(t) = \frac{E_{wec}}{E_{we}} - \frac{D_m^2}{4E_{we}^2}. \quad (21)$$

Proposition 1. *In opportunistic social networks, the weight of the edge made by a node with other adjacent nodes in the network increases; the community relevance also will increase.*

Proof. With time t , the modularity in the community is $Q(t)$.

When the time increases to $t + 1$, the modularity change in the community can be expressed as

$$\begin{aligned} Q(t+1) &= \frac{E_{wec} + \Delta E_{we}}{E_{we} + \Delta E_{we}} - \frac{(D_m + 2\Delta E_{we})^2}{4(E_{we} + \Delta E_{we})^2} \\ Q(t+1) - Q(t) &= \frac{E_{wec} + \Delta E_{we}}{E_{we} + \Delta E_{we}} - \frac{(D_m + 2\Delta E_{we})^2}{4(E_{we} + \Delta E_{we})^2} - \left(\frac{E_{wec}}{E_{we}} - \frac{D_m^2}{4E_{we}^2} \right) \\ &= \frac{(4E_{we}^3 E_{wec} + 4E_{we}^2 \Delta E_{we} E_{wec} + 4E_{we}^3 \Delta E_{we} + 4E_{we}^2 \Delta E_{we}^2) - (E_{we}^2 D_m^2 + 4E_{we}^2 \Delta E_{we}^2 + 4E_{we}^2 D_m \Delta E_{we})}{4E_{we}^2 (E_{we} + \Delta E_{we})^2} \\ &\quad - \left(\frac{(4E_{we}^3 E_{wec} + 4E_{we} \Delta E_{we} E_{wec} + 8E_{we}^2 \Delta E_{we} E_{wec}) - (D_m^2 E_{we}^2 + D_m^2 \Delta E_{we}^2 + 2D_m^2 E_{we}^2 \Delta E_{we})}{4E_{we}^2 (E_{we} + \Delta E_{we})^2} \right) \\ &\geq \frac{4E_{we}^3 \Delta E_{we} - 6E_{we}^2 D_m \Delta E_{we} + 2E_{we}^2 D_m \Delta E_{we} - 2E_{we}^2 D_m \Delta E_{we} + (D_m \Delta E_{we})^2}{4E_{we}^2 (E_{we} + \Delta E_{we})^2} \\ &= \Delta E_{we} \frac{4E_{we}^3 \Delta E_{we} - 6E_{we}^2 D_m + 2E_{we}^2 D_m - 2E_{we}^2 D_m \Delta E_{we} + D_m^2 \Delta E_{we}}{4E_{we}^2 (E_{we} + \Delta E_{we})^2} \\ &= \Delta E_{we} \frac{(2E_{we}^2 - 2E_{we} D_m - D_m \Delta E_{we}) \times (2E_{we} - D_m)}{4E_{we}^2 (E_{we} + \Delta E_{we})^2}. \end{aligned} \quad (22)$$

We can get $\Delta E_{we} > 0$, so we just need proof $(2E_{we}^2 - 2E_{we} D_m - D_m \Delta E_{we}) \times (2E_{we} - D_m) > 0$.
In other words,

$$\begin{cases} 2E_{we}^2 - 2E_{we} D_m - D_m \Delta E_{we} > 0, \\ 2E_{we} - D_m > 0, \end{cases}$$

$$\begin{cases} 2E_{we}^2 - 2E_{we} D_m - D_m \Delta E_{we} > 0, \\ 2E_{we} - D_m > 0, \\ \Delta E_{we} > 0, \end{cases}$$

$$\begin{cases} 0 < \Delta E_{we} < 2E_{we} \left(\frac{E_{we}}{D_m} - 1 \right), \\ 2E_{we} \left(\frac{E_{we}}{D_m} - 1 \right) > 0, \\ D_m < 2E_{we}, \end{cases}$$

$$\begin{cases} 0 < \Delta E_{we} < 2E_{we} \left(\frac{E_{we}}{D_m} - 1 \right), \\ D_m < E_{we}. \end{cases} \quad (23)$$

It is known that $2Q$ is the total of nodes in the network, and no community in the network appears more than $2Q$. In short, we are aware that increasing the weight can increase the community's relevance in opportunistic social networks. For this paper, the weight will affect the community's relevance in opportunistic social networks, and the proposition holds. \square

Proposition 2. *If the weight of an edge of two communities increases, node m is in community A, U_{commB}^- will be increased, and U_{commA}^+ will be decreased. The community corresponding to the node m will change, and the weight of an edge between the node and the community is ΔE_{we} ($\Delta E_{we} > 0$); if the weight of the edge can be changed, the result of the community will also change.*

Proof. Before the weight changes, for node m ,

$$\begin{cases} U_{\text{comm } A}^+ = e_A^m - \frac{D_m(D_A - D_m)}{2E_{\text{we}}}, \\ U_{\text{comm } B}^- = e_B^m - \frac{D_m D_B}{2E_{\text{we}}}. \end{cases} \quad (24)$$

After the weight changes, for node m ,

$$\begin{cases} \overline{U}_{\text{comm } A}^- = e_A^m - \frac{(D_m + \Delta E_{\text{we}})(D_A - D_m)}{2(E_{\text{we}} + \Delta E_{\text{we}})}, \\ \overline{U}_{\text{comm } B}^- = (e_B^m + \Delta E_{\text{we}}) - \frac{(D_m + \Delta E_{\text{we}})(D_B + \Delta E_{\text{we}})}{2(E_{\text{we}} + \Delta E_{\text{we}})}, \end{cases} \quad (25)$$

$$\begin{aligned} \overline{U}_{\text{comm } B}^- - U_{\text{comm } B}^- &= \frac{2E_{\text{we}}\Delta E_{\text{we}} + D_m D_B}{2E_{\text{we}}} - \frac{D_m D_B + \Delta E_{\text{we}} D_B + \Delta E_{\text{we}} D_m + \Delta E_{\text{we}}^2}{2(E_{\text{we}} + \Delta E_{\text{we}})} \\ &\geq \frac{2E_{\text{we}}\Delta E_{\text{we}} + D_m D_B}{2(E_{\text{we}} + \Delta E_{\text{we}})} - \frac{D_m D_B + \Delta E_{\text{we}} D_B + \Delta E_{\text{we}} D_m + \Delta E_{\text{we}}^2}{2(E_{\text{we}} + \Delta E_{\text{we}})} \\ &= \frac{2E_{\text{we}} - D_B - D_m - \Delta E_{\text{we}}}{2(E_{\text{we}} + \Delta E_{\text{we}})} \cdot \Delta E_{\text{we}}. \end{aligned}$$

Because $E_{\text{we}} > 0$, when $2E_{\text{we}} = \sum di$,

$$2E_{\text{we}} > 2E_{\text{we}} - D_B - D_m - \Delta E_{\text{we}} \cdot \overline{U}_{\text{comm } B}^- - U_{\text{comm } B}^- > 0. \quad (26)$$

All in all, if the weight of one side increases, then for node m $U_{\text{comm } B}^-$ increases. Then,

$$\overline{U}_{\text{comm } A}^+ - U_{\text{comm } A}^+ = \Delta E_{\text{we}} (D_A - D_m) \frac{D_m - E_{\text{we}}}{2E_{\text{we}}(E_{\text{we}} + \Delta E_{\text{we}})}. \quad (27)$$

Because $\Delta E_{\text{we}} > 0$, $E_{\text{we}} > 0$, $D_A - D_m > 0$, for all edges in the network $\sum D_i > D_m D_m - E_{\text{we}} < 0$, $\overline{U}_{\text{comm } A}^+ - U_{\text{comm } A}^+ < 0$.

If the weight of one side increases, then $U_{\text{comm } A}^+$ decreases.

If the weight of an edge of two communities increases, node m is in community A , then $U_{\text{comm } B}^-$ will increase and $U_{\text{comm } A}^+$ will decrease. \square

Proposition 3. *If node m and node n are connected, and one of the nodes has one and only one edge, when the weight between node m and node n drops, the community will not divide.*

Proof. Let us assume that the community is divided; then the following three conditions must be met:

$$\begin{cases} E_{\text{we},m} + E_{\text{we},n} < E_{\text{we}}, \\ \frac{u_m}{E_{\text{we}}} - \frac{D_i^2}{4E_{\text{we}}^2} + \frac{u_n}{E_{\text{we}}} - \frac{D_j^2}{4E_{\text{we}}^2} < \frac{D_i + D_j + w_{mn}}{E_{\text{we}}} - \frac{(D_i + D_j)^2}{4E_{\text{we}}^2}, \\ w_{mn} > \frac{D_i D_j}{2E_{\text{we}}}. \end{cases} \quad (28)$$

As the weight changes, the formula can also be expressed as

$$\begin{cases} E_{\text{we},m}^* + E_{\text{we},n}^* > E_{\text{we}}^*, \\ w_{mn} < \Delta E_{\text{we}} + \frac{D_i D_j + D_m \Delta E_{\text{we}} + \Delta E_{\text{we}}^2}{2(E_{\text{we}} + \Delta E_{\text{we}})}, \end{cases} \quad (29)$$

$$\frac{D_i D_j}{2E_{\text{we}}} < w_{mn} < \frac{D_i(D_j + \Delta E_{\text{we}})}{2(E_{\text{we}} + \Delta E_{\text{we}})} = \frac{D_i D_j + D_i \Delta E_{\text{we}}}{2(E_{\text{we}} + \Delta E_{\text{we}})}.$$

So, it can be seen from the above proof and we conclude that

$(D_i D_j / 2E_{\text{we}}) < w_{mn} < \Delta E_{\text{we}} + ((D_i D_j + D_m \Delta E_{\text{we}} + \Delta E_{\text{we}}^2) / (2(E_{\text{we}} + \Delta E_{\text{we}})))$ is false.

For a node in opportunistic social networks, if it has only one edge connected to another node, the community will not divide when the weight between the two nodes decreases.

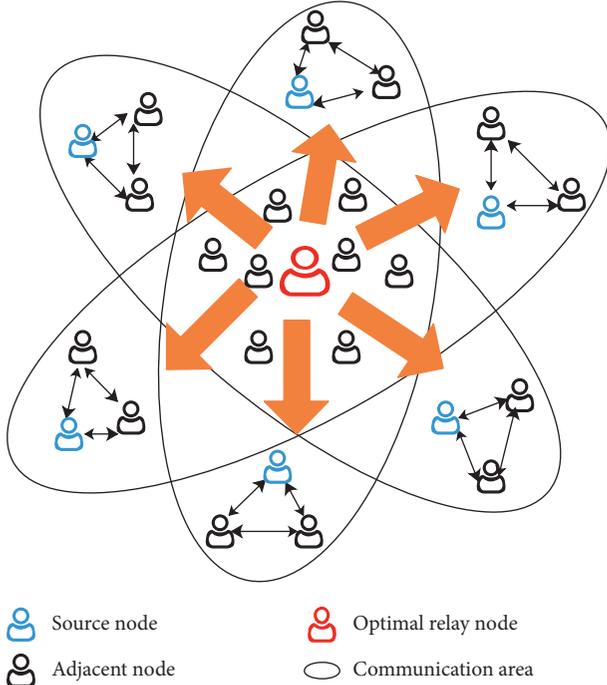


FIGURE 3: Information transfer model between communities.

After community detection, we construct a set of candidate relay nodes according to the relationship between the information forwarder and adjacent nodes. Select the optimal relay node from the set of candidate relay nodes to undertake the transmission task. Therefore, selecting one or more relays among multiple relay nodes to participate in transmission has become our concern. As shown in Figure 3, when the community is established and transmitted between each community, it is necessary to find a reliable relay node to transmit information. To achieve higher efficiency, construct a set of candidate relay nodes from the neighbor nodes of the source node; from the perspective of minimizing the bit error rate, this paper analyzes the channel coefficients of the two segments of the source node to the relay node and the relay node to the destination node and chooses the AF protocol as the relay node's forwarding method, which is suitable for the information transmission process of various channel qualities [40]. Calculate the sum of the channel coefficients of the channel corresponding to each relay node, and find the largest coefficient of the relay node, which is the optimal relay node and will improve the efficiency of information transmission.

Let us suppose there are a source node S , destination node D , and relay nodes R_1, R_2, \dots, R_n , when transferring information between communities. The communication model is as shown in Figure 4. In this case, the channels from the source node to the destination node and the source node to each relay node are all Rayleigh fading channels, which obey the Rayleigh distribution. We assume that the channel coefficient from the source node to the destination node is $C_{s,d}$, the channel coefficient from the source node to the n th relay node is $C_{s,rn}$, and the channel coefficient from the n th relay node to the destination node is $C_{rn,d}$.

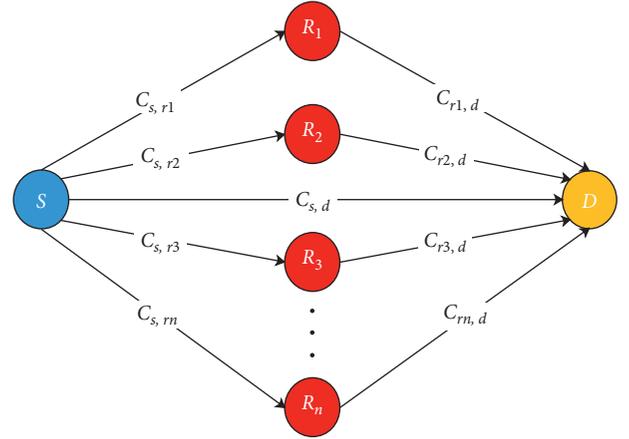


FIGURE 4: The communication model of the source node to the destination node.

The transmit power of the source node is P_1 , and the transmission power of the relay node is P_2 . When there is a direct transmission from the source node to the destination node, the power $P = P_1 + P_2$. When the source node sends information i to the destination node and the relay node is with power P_1 , noise from the source node to destination node is $V_{s,d}$ and noise from the source node to the relay node is $V_{s,r}$. So information received by the relay node and the destination node is as follows:

$$\begin{aligned} RM_{s,d} &= \sqrt{P_1} C_{s,d} \cdot i + V_{s,d}, \\ RM_{s,r} &= \sqrt{P_1} C_{s,r} \cdot i + V_{s,r}. \end{aligned} \quad (30)$$

In the AF protocol, when the relay node receives the signal from the source node and forwards it to the destination node, it will amplify the received signal, and the scaling factor is

$$\chi = \frac{1}{\sqrt{P_1 |C_{s,r}|^2 + N_0}}. \quad (31)$$

We can know that the signal from the relay node to the destination node is $\chi_{s,r}$, and then the information sent by the relay node to the destination node is

$$RM_{r,d} = \sqrt{P_2} C_{r,d} \cdot (\chi_{s,r}) + V_{r,d}. \quad (32)$$

This paper's focus on selecting the optimal relay node is how to find an optimal relay node that makes the channel coefficients of the source node to the relay node and the relay node to the destination node larger.

The channel coefficient matrix from the source node to the relay node is A , and the channel coefficient matrix from the relay node to the destination node is B . Then,

$$\begin{aligned} A_{1 \times n} &= [C_{s,r1}, C_{s,r2}, C_{s,r3}, \dots, C_{s,rn}], \\ B_{1 \times n} &= [C_{r1,d}, C_{r2,d}, C_{r3,d}, \dots, C_{rn,d}]. \end{aligned} \quad (33)$$

We define a threshold for the number of candidate relay nodes ψ and set $\psi \leq 100$; we have to consider the following situations:

- (1) If $n \geq \psi$, compare the channel coefficients of each relay node corresponding to matrices A and B , find the smaller of the two, and store the smaller value in the matrix S .

$$S_{1 \times n} = [C_1, C_2, C_3, \dots, C_n]. \quad (34)$$

Sort the matrix elements S from largest to smallest, select the first m relay nodes with a larger C_i value from them, and store them in the matrix T and $C_i = \min\{C_{s,ri}, C_{ri,d}\}$, where C_i is the smaller value of the channel coefficient of the two channels corresponding to the relay node r_i .

$$T_{1 \times n} = [r_1, r_2, r_3, \dots, r_i, \dots, r_n], \quad (35)$$

where r_i is one of the first m elements in the matrix after sorting. The value of m largely depends on the number of candidate relay nodes n , (m/n) the larger the value, the lower the bit error rate. Bit error rate refers to the index of the accuracy of data transmission within a specified time.

$$\text{SER} = \frac{\text{SER}_{te}}{\text{SER}_T} * 100\%, \quad (36)$$

where SER is bit error rate, SER_{te} is the bit errors in transmission, and SER_T is the total number of codes transmitted. We add the two channel coefficients of these m relay nodes, and the relay node with the largest sum is the optimal relay node as follows:

$$R = \{ri | \max(C_{s,ri} + C_{ri,d}), i = 1, 2, \dots, m\}. \quad (37)$$

- (2) Otherwise, when the number of candidate relay nodes is less than the threshold, we must pay attention to the accuracy of being selected as the optimal relay node; calculate the sum of the channel coefficients of the channel corresponding to each relay node and the relay node with the largest sum, which is the optimal relay node.

Based on the above definition, we propose an efficient data transmission algorithm EDCCD and the algorithm steps are as follows:

Step 1: calculate the encounter probability of node m and node n , construct the encounter probability matrix, and use the number of encounters between nodes to weight each edge.

Step 2: define node similarity, which is composed of network topology attributes and social attributes. Network topology attributes are composed of the strength of node, the direct connection strength, and the indirect connection strength. Social attributes are composed of the geographic relevance of nodes, the interesting relevance of nodes, and the separating time relevance of nodes.

Step 3: use a hierarchical clustering algorithm to divide the community and lead in the modularity Q . The modularity is used to measure the degree of community

division. And the fast unfolding algorithm is used to calculate the node similarity to update the network weight comprehensively.

Step 4: from the perspective of minimizing the bit error rate, after the community is divided into a multihop wireless network, construct a set of candidate relay nodes based on the relationship between the information forwarder and adjacent nodes and select the optimal relay node from the set of candidate relay nodes to undertake the transmission task. Analyze the channel coefficients of the channels from the source node to the relay node and the relay node to the destination node, and select the AF protocol as the relay node forwarding method for routing and forwarding.

To enhance the understanding and readability of the entire algorithm, the specific calculation flowchart of the EDCCD algorithm is shown in Figure 5. Algorithm 1 gives the initialization and community establishment phase of the proposed algorithm, and Algorithm 2 presents the routing and forwarding phase of the proposed algorithm. \square

4. Simulation and Analysis

To assess the performance of the EDCCD, we use a simulation tool called ONE (Opportunistic Network Environment) [41] and we compare with the following four typical routing algorithms.

Spray and wait [29]: this algorithm sprays the copies to the network and waits for these nodes to reach the destination node. The number of copies of the algorithm will affect performance, reduce the message delivery success rate, and increase the delivery delay.

SCR (Social-based Clustering and Routing Scheme) [36]: this algorithm is a useful measurement method of social relations between nodes in mobile opportunistic network, and is a novel social-based clustering and routing scheme.

SECM (status estimation and cache management) [42]: the algorithm uses state estimation and cache management methods to identify surrounding neighbors to evaluate the transmission probability between nodes, to ensure that they have high transmission, and to achieve the purpose of adjusting the cache.

EIMST (effective information transmission based on socialization nodes) [2]: the algorithm is based on social nodes to achieve effective information transmission. According to the defined stop time, when $t < h$, the node forwards the message with the most excellent probability, and when $t > h$, the node stops sending the message.

Download the real datasets from the network repository to experiments. According to the data information required for data transmission in opportunistic social networks, and choose pages-government [43], wiki-elec [44], advogato [45], and slashdot [46] four datasets for simulation experiments. The characteristic information of the four experimental datasets is shown in Table 1.

In the simulation experiment, we set the following metrics according to the characteristics of data transmission. The EDCCD algorithm and the other four algorithms run in

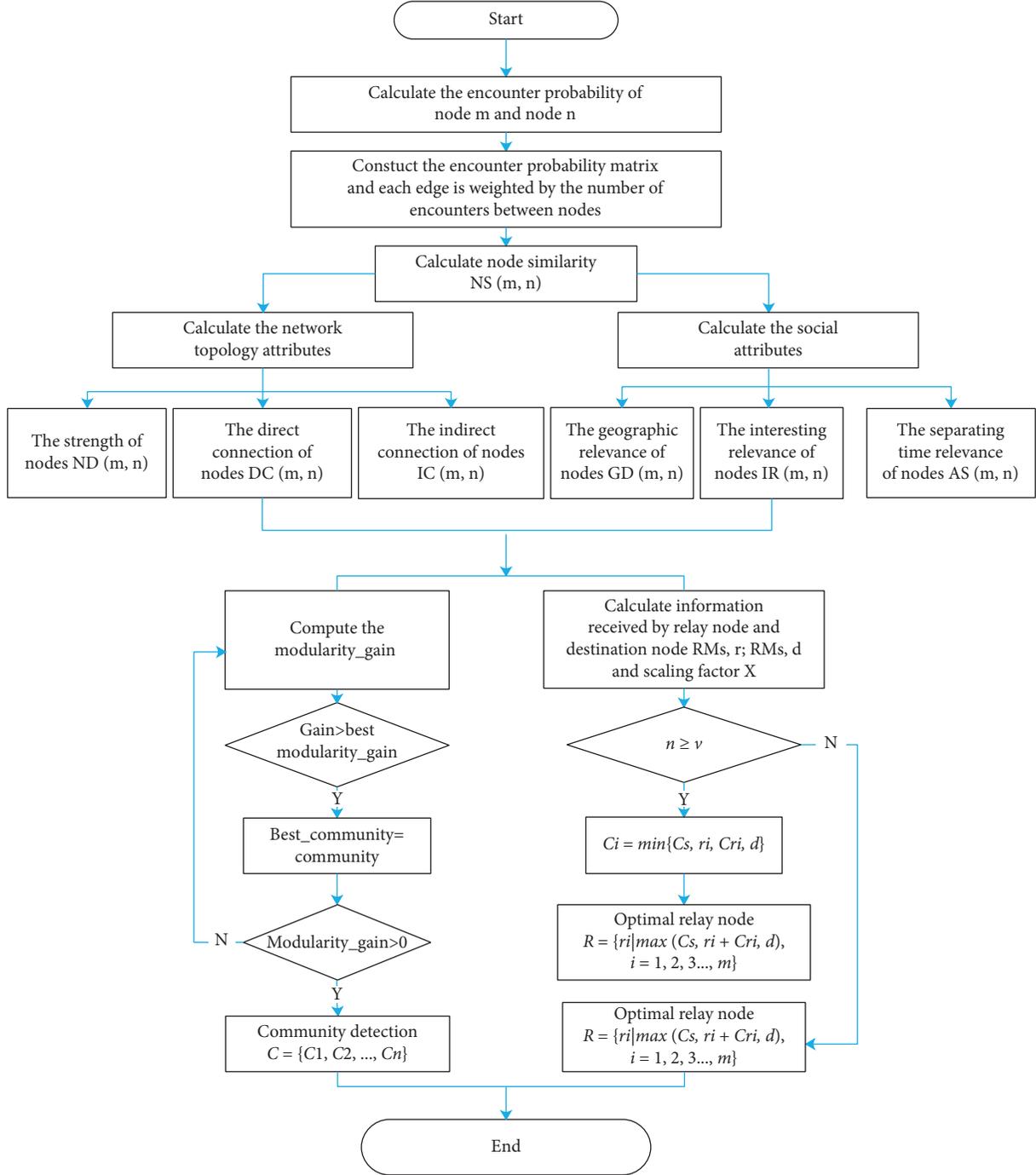


FIGURE 5: The process of EDCCD algorithms.

the same simulation environment to compare their performance.

- (1) Delivery ratio: probability of choosing a suitable node as the next-hop node, represented as follows:

$$D_{\text{node}} = \frac{D_{\text{receive}}}{D_{\text{send}}}, \quad (38)$$

where D_{receive} is the number of messages received by the destination node and D_{send} is the total number of sent messages.

- (2) Routing overhead indicates the overhead between nodes when transmitting information, represented as follows:

$$Rd = \frac{R_{\text{sum}} - R_{\text{suc}}}{R_{\text{sum}}}, \quad (39)$$

where R_{sum} is the total time of the transmission between nodes and R_{suc} is the time to transmit a successful message between nodes.

- (3) Average end-to-end delay: express the delay in selecting the optimal next hop.

```

Input:  $G = (V, E, w)ND_{(m,n)}DC_{(m,n)}IC_{(m,n)}GD_{(m,n)}IR_{(m,n)},AS_{(m,n)}$ 
Output:  $C = \{C1, C2, \dots, Cn\}$ 
(1) Begin
(2) Initialize every node as a cluster;
(3) Calculate the encounter probability and times of node  $m$  and node  $n$  in a period of time  $t$ ;
(4)  $SMD_{tp(m,n)}$  Get Network topology ( $ND_{(m,n)}, DC_{(m,n)}, IC_{(m,n)}$ )
(5)  $SR_{(m,n)}$  Get Social relationship ( $GD_{(m,n)}, IR_{(m,n)}, AS_{(m,n)}$ )
(6)  $NS_{(m,n)} = SMD_{tp(m,n)} + SR_{(m,n)}$ //Compute the node similarity
First_phase:
(7) Initialize (self, nodes, edges):
(8) for ( $i=0; i \leq n; i++$ )
(9) self.communities = { $n1, n2, n3$ };
(10) partition = self.first_phase (network);
(11)  $q = q + \text{self.s\_in}[i]/2l - \text{self.s\_tot}[i]/2l$ ;
(12) End for
(13) Compute modularity_gain (self, node,  $c, k\_i\_in$ ):
(14) return  $2 * k\_m\_in - \text{self.s\_tot}[c] * \text{self.k\_m}[node]/\text{self.m}$ ;
(15) If (gain > best modularity_gain)
(16) best_community = community;
(17) best_partition[best_community].append (node);
(18) self.communities[node] = best_community;
(19) End If
Second_phase:
(20) for ( $i=0; i < \text{partition.length}; i++$ )
(21) Self.communities = (nodes, edges);
(22) In_order (nodes, edges);
(23) If (modularity_gain > 0)
(24) return  $C = \{C1, C2, \dots, Cn\}$ ;
(25) else
(26) return First_phase;
(27) End If
(28) End for
(29) END

```

ALGORITHM 1: Initialize community detection.

$$D_d = \frac{D_{\text{sum}}}{D_{\text{suc}}}, \quad (40)$$

where D_{sum} is the total delay of per node and D_{suc} is the total number of nodes successfully receiving messages.

The correlation between the time and delivery ratio in four different real datasets is shown in Figures 6–9. Figure 6 shows the delivery ratio of spray and wait, SCR, SECM, EIMST, and EDCD algorithms in pages-government dataset. We can infer that when the simulation time is less than one day, the advantages of the algorithm EDCD are not apparent in the four real datasets. However, as the simulation time increases, we can find that the transmission rate of the EDCD algorithm is always bigger than other algorithms. EDCD algorithm divides the community by node similarity, and the effective nodes in the community carry out data transmission, so the data delivery ratio is better than the other four algorithms. The relationship between the delivery ratio and the simulation time in wiki-elec dataset is shown in Figure 7. The SCR algorithms deliver information to nodes, and the community by using the flooding method leads to mass information missing.

The delivery ratio of SECM is 0.65–0.78. EIMST and EDCD algorithm's delivery ratio is higher than the other. EIMST algorithm controls the time interval of delivery information that improves the transmission and receiving of effective information, and its delivery ratio reached 0.66–0.81. Due to the adoption of the EDCD algorithm combining network topology and social attributes, the algorithm's transmission rate is the highest among all algorithms, reaching 0.67–0.84.

The correlation between the delivery ratio and simulation time in advogato dataset is shown in Figure 8. We see that the algorithm with the highest delivery rate is the EDCD algorithm, reaching 0.85–0.88. The spray and wait algorithm uses flooding to transmit information at community nodes, a large amount of information is lost, and the delivery rate is the lowest, only 0.67–0.70. Figure 9 shows the relationship between time and delivery ratio in slashdot dataset. The dataset with the largest number of nodes in the four datasets is slashdot dataset. When the simulation time is less than one and a half days, each dataset's delivery ratio is rising sharply, and the time is up to three days; only the EDCD and EIMST algorithms' delivery ratio is rising. This is because, in slashdot dataset, the two algorithms quantify the social attributes in 5G environment of nodes. On the whole, in the

Input: source node S , relay node R_1, R_2, \dots, R_n , destination node D ; power of source node P_1 ; power of relay node P_2 ;
Output: optimal relay node R ;

```

(1) Begin
(2) Power of Destination node  $P = P_1 + P_2$ ;
(3) Calculate Information received by relay node and destination node  $RM_{s,r}, RM_{s,d}$ ;
(4) Amplify the received signal and calculate Scaling factor  $\chi$ ;
(5) Threshold of number of candidate relay nodes  $\psi$ ;
(6) Function BER = AF_Simulation (max_SNR);
(7) for (snr = 0; snr ≤ max_SNR; snr++)
(8)   for ( $i = 0; i \leq n; i++$ )
(9)      $V = 1/(10^{(snr/10)})$ ; //  $V$  is the variance and the noise energy is normalized
(10)    sig = randsrc(1, N, [0 1]); // Generating binary input sequences
(11)    sig_mod = QpskMapping(sig); // The input binary sequence is QPSK modulated
(12)  End for
(13)  If ( $n > \psi$ )
(14)     $C_i = \min\{C_{s,r_i}, C_{r_i,d}\}$ ;
(15)    Optimal relay node  $R = \{r_i | \max(C_{s,r_i} + C_{r_i,d}), i = 1, 2, \dots, m\}$ 
(16)  else
(17)    Optimal relay node  $R = \{r_i | \max(C_{s,r_i} + C_{r_i,d}), i = 1, 2, \dots, m\}$ 
(18)  End If
(19) End for
(20) END

```

ALGORITHM 2: Routing and forwarding.

TABLE 1: Characteristics of the experimental datasets.

Dataset	Pages-government	Wiki-elec	Advogato	Slashdot
Duration (days)	3.5	11	4	4
Message transmission rate (kbps)	250	250	250	250
Number of experimental devices	7100	8000	5200	70000
Buffer size (M)	5	5	5	5
TTL (time to live)	0.5 days	1day	60 min	2days
Node initial energy (J)	200	200	200	200
The sending frequency of a data packet(s)	35	30	35	25

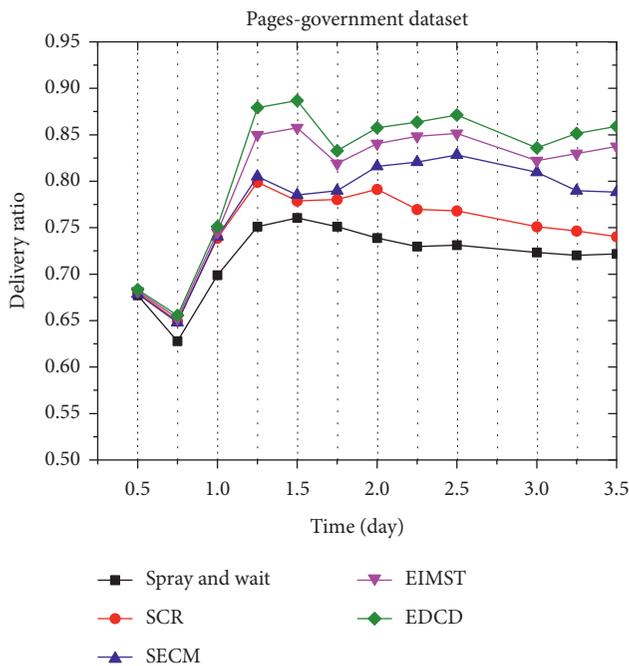


FIGURE 6: Comparison of algorithms delivery ratio in pages-gov dataset.

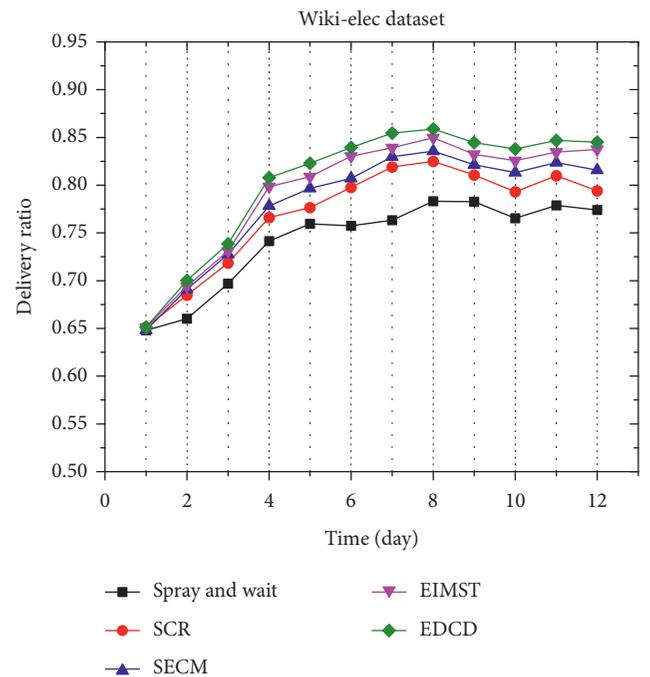


FIGURE 7: Comparison of algorithms delivery ratio in wiki-elec dataset.

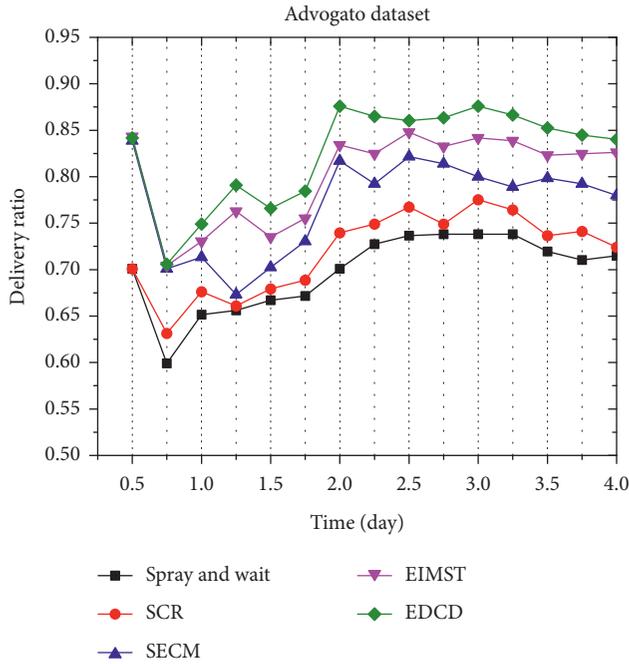


FIGURE 8: Comparison of algorithms delivery ratio in advogato dataset.

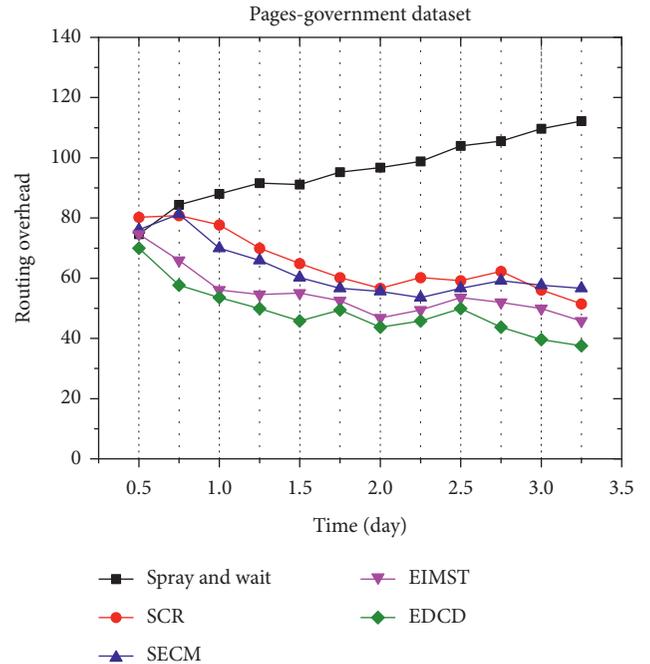


FIGURE 10: Comparison of algorithms routing overhead in pages-government dataset.

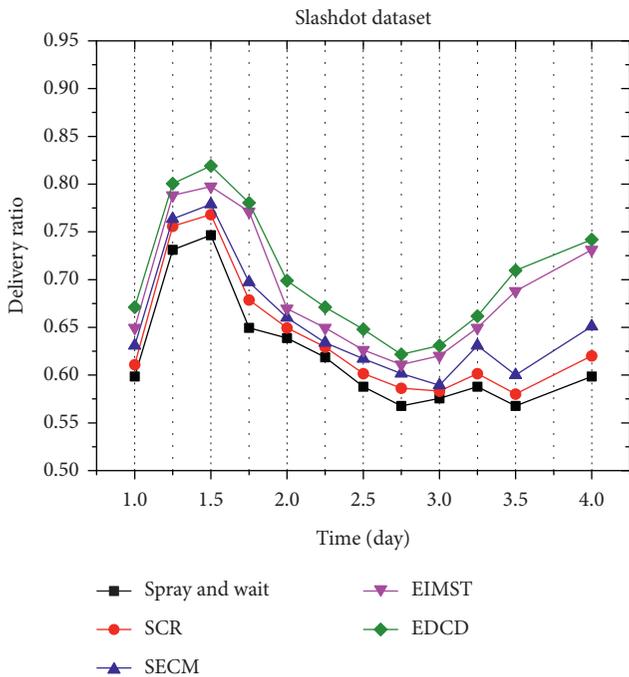


FIGURE 9: Comparison of algorithms delivery ratio in slashdot dataset.

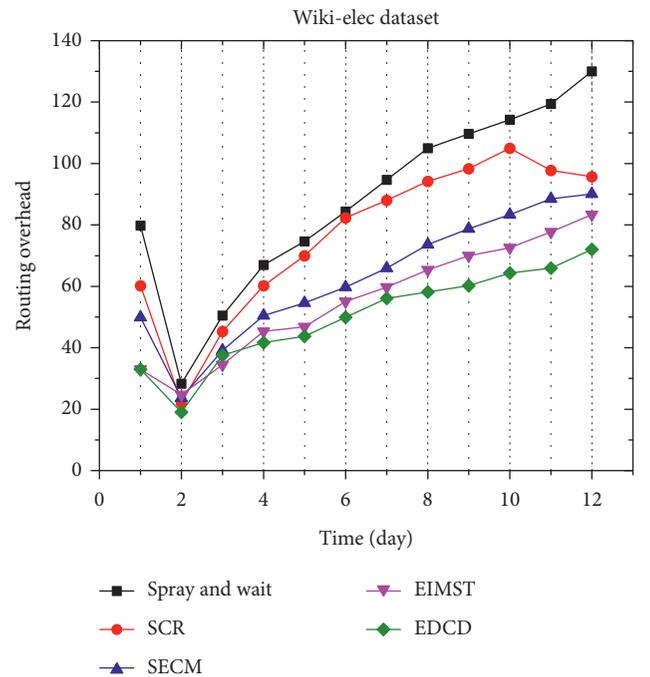


FIGURE 11: Comparison of algorithms routing overhead in wiki-elec dataset.

EDCD algorithm, the delivery ratio is 0.76 on average, which is higher than the other algorithms.

The correlation between the time and routing overhead in four different real datasets is shown in Figures 10–13. The comparison of the routing overhead between these five

different algorithms in pages-government dataset is shown in Figure 10. The average routing overhead of the EDCD algorithm is always kept to the lowest. The algorithm uses the node similarity to divide the community and uses the optimal relay node strategy to forward information. The routing overhead of the EDCD algorithm is maintained between 40 and 65.

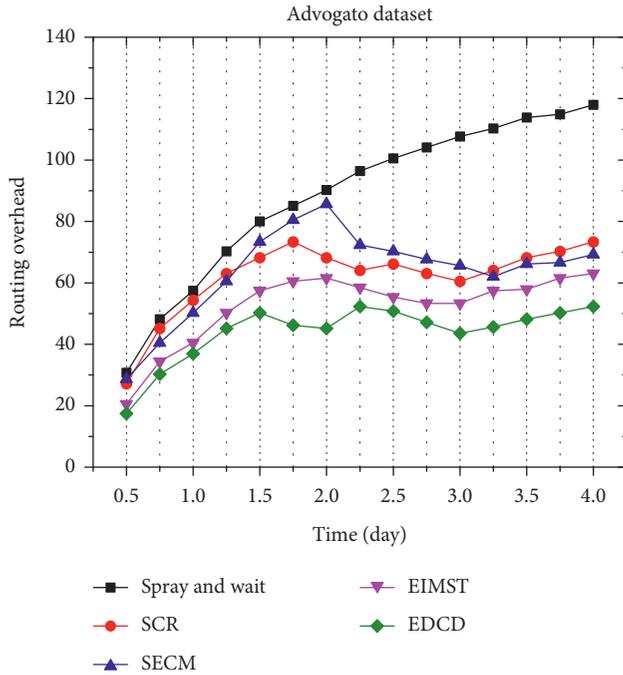


FIGURE 12: Comparison of algorithms routing overhead in advogato dataset.

Figure 11 shows the association between routing overhead and time in wiki-elec dataset. In the spray and wait algorithm, redundant message group copies require a lot of time and resources, which is the main reason for the vast routing overhead. In the SCR algorithm, each node only forwards a copy of the message to the node with the destination node as a cluster member, ignoring the current availability of the next-hop node, which will cause overhead. In the SECM algorithm, because the node injects many redundant data, the overhead will be large. In the EIMST algorithm, information and buffer space can be effectively managed, but it consumes some unavailable node resources. In terms of routing overhead, EDCD always performs best among these five algorithms. Figure 12 shows the relationship between time and routing overhead in advogato dataset. Compared with other algorithms, EDCD algorithms select the optimal relay node and set up the weight distribution between nodes and community to reduce the overhead cost. Regarding the spray and wait algorithms, a lot of redundant information use lot of computing resources. For SCR and SECM algorithms, the cooperation mechanism is conducive to the reasonable allocation of computing resources, so the cost of these two algorithms is in the middle level. EIMST does not fully consider the transmission preference of nodes, so its performance is worse than that of EDCD algorithm.

The relationship between routing overhead and time in slashdot dataset is shown in Figure 13. From the chart, we can see that the routing overhead increases sharply at first, nearly stably by the time it reaches the third day. The routing overhead of the spray and wait algorithm increases dramatically; a large number of data copies are generated in

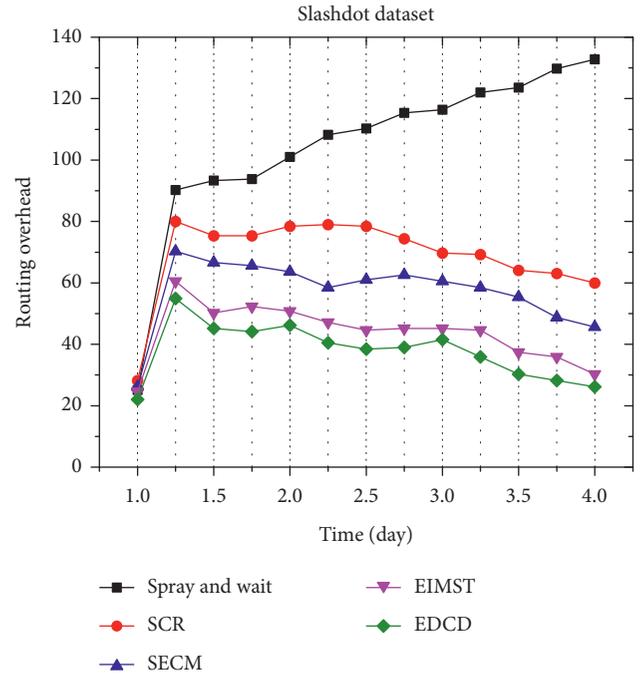


FIGURE 13: Comparison of algorithms routing overhead in slashdot dataset.

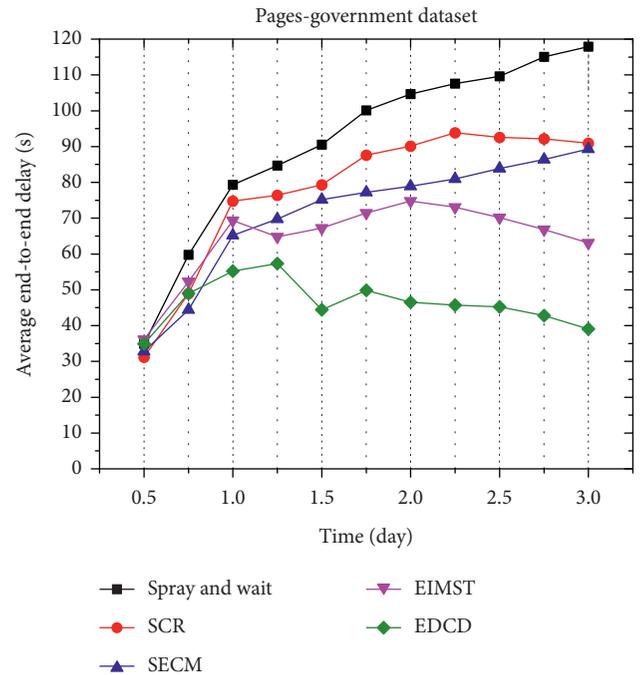


FIGURE 14: Comparison of algorithms average end-to-end delay in pages-government dataset.

slashdot dataset with a large number of nodes, and these need to be processed, so the routing overhead is higher than other algorithms.

The association between the time and average end-to-end delay in four different real datasets is shown in Figures 14–17. The relationship between the average end-to-

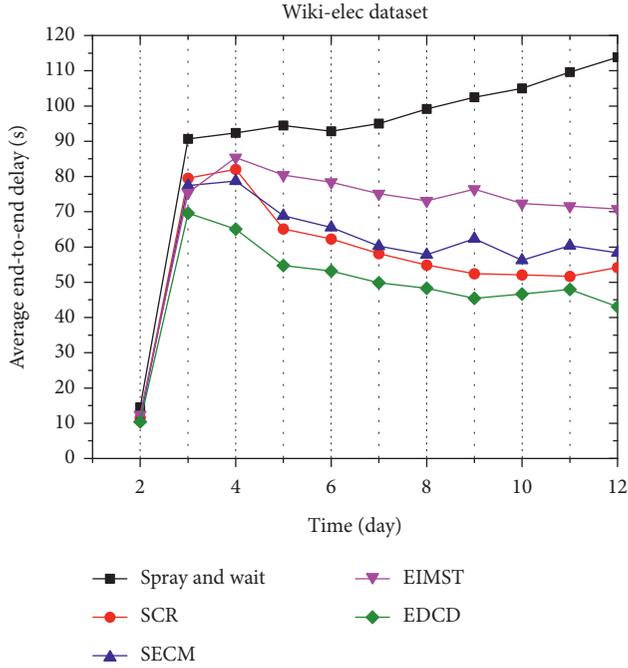


FIGURE 15: Comparison of algorithms average end-to-end delay in wiki-elec dataset.

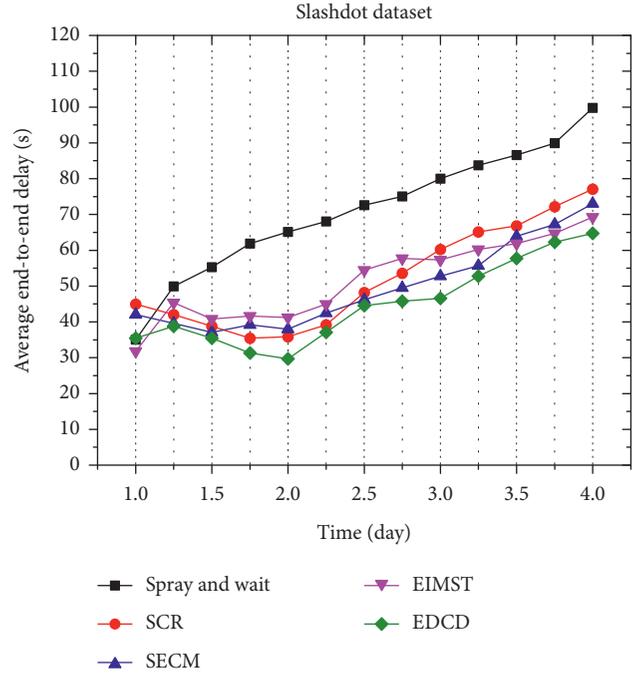


FIGURE 17: Comparison of algorithms average end-to-end delay in slashdot dataset.

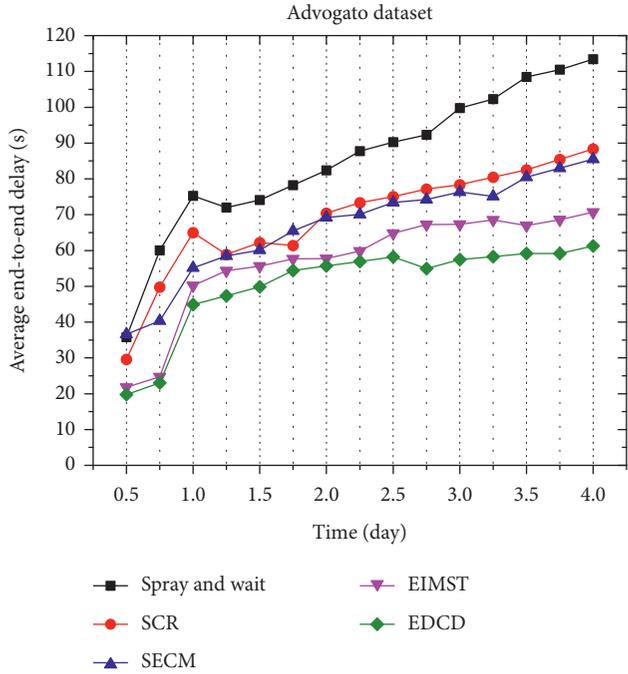


FIGURE 16: Comparison of algorithms average end-to-end delay in advogato dataset.

end delay and time of each algorithm in pages-government dataset is shown in Figure 14. Compared with the other four algorithms, the EDCD algorithm has the lowest average end-to-end delay.

Since the EDCD algorithm proposes a strategy for dividing communities by analyzing the comprehensive

characteristics of nodes, it can reduce inefficient nodes that are not helpful to the transmission process, reducing the average end-to-end delay. The spray and wait algorithm has more message copies, which will cause corresponding delays. The SCR algorithm effectively forwards the copy of the message to the destination node, so the transmission delay is lower than the spray and wait algorithm. SECM algorithm will also increase the cache of node before data transmission, so there will be a corresponding delay.

Figure 15 shows the association between routing overhead and time in wiki-elec dataset. We can see that the EIMST algorithm's delay is higher than that in other datasets but lower in the rest of the datasets. Because the EIMST algorithm applies node based on information management, there are more nodes in the wiki-elec dataset, and the delay increases as the simulation time increases. In short, the average end-to-end delay of the EDCD algorithm in wiki-elec dataset is lower than the other four algorithms.

Figure 16 shows the relationship between average end-to-end delay and time in advogato dataset. To be specific, spray and wait algorithm's maximum delay could reach 95 because this method remarkably increased routing and message forwarding delays. The SCR and SECM algorithms have lower delays than the spray and wait algorithm because both algorithms effectively controlled a lot of message copies. Besides, the SCR algorithm implemented community division and information management. In contrast, the SECM algorithm effectively utilized the cooperation mechanism between nodes to utilize the nodes' cache space reasonably to reduce the delay in the message forwarding process.

The average end-to-end delay of the EIMST algorithm was also significantly lower than the other algorithms.

Figure 17 shows the correlation between the average end-to-end delay and time in slashdot dataset. In a dataset with many nodes, we can see in the figure that the average end-to-end delay of the EIMST algorithm is significantly higher than other datasets. That is why the EIMST algorithm implements community detection. However, the effect is general when processing large amounts of data. The algorithm EDCD proposed in this paper has a lower latency in different real datasets than other algorithms.

5. Conclusions

In this study, an effective data transmission scheme in opportunistic social networks that uses mobile edge computing combined with network topology attributes and social attributes to measure node similarity to divide communities and select the optimal relay node. This algorithm is mainly based on the idea that the closeness between nodes in the community is higher than that exterior in the community and provides a method for selecting the optimal relay node according to the sum of channel coefficients in the process of transmitting information. The simulation experiment results show that the strategy has good performance in different real datasets such as delivery ratio, routing overhead, and average end-to-end delay. The EDCD algorithm can be used to the 5G data transmission scene and can cope with the challenges of stability and continuity required by data in the interactive process through efficient community division and information transmission. In future work, we will enhance the related performance of the algorithm and will further study the security of data transmission in opportunistic social networks.

Data Availability

The data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Science Foundation of China under Grant 61966035 (Research on Super-Resolution Reconstruction of Remote Sensing Images Based on Deep Learning of Spatio-Temporal Spectrum Features), by the Intelligent Multi-Modal Information Processing Project (XJEDU2017T002), by the International Cooperation Project of the Autonomous Region's Science and Technology Department's "Data-driven China-Russia Cloud Computing Sharing Platform Construction" No. 2020E01023.

References

- [1] G. Yu and J. Wu, "Content caching based on mobility prediction and joint user Prefetch in Mobile edge networks,"

- Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1839–1852, 2020.
- [2] J. Wu, Z. Chen, and M. Zhao, "Effective information transmission based on socialization nodes in opportunistic networks," *Computer Networks*, vol. 129, pp. 297–305, 2017.
- [3] J. Wu, Z. Chen, and M. Zhao, "Community recombination and duplication node traverse algorithm in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 940–947, 2020.
- [4] Y. Cai, S. Pan, X. Wang, H. Chen, X. Cai, and M. Zuo, "Measuring distance-based semantic similarity using meronymy and hyponymy relations," *Neural Computing and Applications*, vol. 32, no. 8, pp. 3521–3534, 2018.
- [5] J. Wu, X. Tian, and Y. Tan, "Hospital evaluation mechanism based on mobile health for IoT system in social networks," *Computers in Biology and Medicine*, vol. 109, pp. 138–147, 2019.
- [6] J. Luo, J. Wu, and Y. Wu, "Advanced data delivery strategy based on multiperceived community with IoT in social complex networks," *Complexity*, vol. 2020, pp. 1–15, Article ID 3576542, 2020.
- [7] X. Zhu, Q. Yang, H. Tian, J. Ma, and W. Wang, "Contagion of information on two-layered weighted complex network," *IEEE Access*, vol. 7, pp. 155064–155074, 2019.
- [8] H. Zhang, Z. Chen, J. Wu, and K. Liu, "FRRF: a fuzzy reasoning routing-forwarding algorithm using mobile device similarity in mobile edge computing-based opportunistic mobile social networks," *IEEE Access*, vol. 7, pp. 35874–35889, 2019.
- [9] Y. I. N. Sheng, W. U. Jia, and Y. U. Genghua, "Low energy consumption routing algorithm based on message importance in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 948–961, 2021.
- [10] W. U. Jia, Q. U. Jingge, and Y. U. Genghua, "Behavior prediction based on interest characteristic and user communication in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 1006–1018, 2021.
- [11] E. P. N. Karunanayake, "Optimal relay node placement to improve design optimal relay node placement to improve expected life time in wireless sensor network design," 2020.
- [12] F. Xiong, Y. Liu, and H. F. Zhang, "Multi-source information diffusion in online social networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2015, no. 7, 2015.
- [13] W. Y. B. Lim, "Federated learning in mobile edge networks: a comprehensive survey," *arXiv*, vol. 22, no. 3, pp. 2031–2063, 2019.
- [14] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 744–758, 2017.
- [15] S. Wang, X. Chang, X. Li, Q. Z. Sheng, and W. Chen, "Multi-task support vector machines for feature selection with shared knowledge discovery," *Signal Processing*, vol. 120, pp. 746–753, 2016.
- [16] Z. Gao, J. Meng, Q. Wang, and Y. Yang, "Data offloading for deadline-varying tasks in mobile edge computing," in *Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, pp. 1479–1484, Guangzhou, China, October 2018.
- [17] W. Shi, L. Zhai, M. Ouyang, and J. Zhang, "A mobile edge computing server deployment scheme in wireless mesh

- network,” in *Proceedings of the 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, pp. 25–29, Changchun, China, August 2019.
- [18] A. Adebayo, D. B. Rawat, L. Ni, and M. Song, “Group-Query-as-a-Service for secure low-latency opportunistic RF spectrum access in mobile edge computing enabled wireless networks,” in *Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, Hangzhou, China, July 2018.
- [19] J. Wu, Z. Chen, and M. Zhao, “An efficient data packet iteration and transmission algorithm in opportunistic social networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 8, pp. 3141–3153, 2020.
- [20] F. Xiong, Y. Liu, and J. Cheng, “Modeling and predicting opinion formation with trust propagation in online social networks,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 44, pp. 513–524, 2017.
- [21] Y. He, F. R. Yu, N. Zhao, and H. Yin, “Secure social networks in 5G systems with mobile edge computing, caching, and device-to-device communications,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 103–109, 2018.
- [22] Y. A. N. G. Weiyu, W. U. Jia, and J. Luo, “Effective data transmission and control base on social communication in social opportunistic complex networks,” *Complexity*, vol. 2020, Article ID 3721579, 13 pages, 2020.
- [23] J. Wu, Z. Chen, and M. Zhao, “Weight distribution and community reconstitution based on communities communications in social opportunistic networks,” *Peer-to-Peer Networking and Applications*, vol. 12, no. 1, pp. 158–166, 2019.
- [24] X. Li and J. Wu, “Node-oriented secure data transmission algorithm based on IoT system in social networks,” *IEEE Communications Letters*, vol. 24, no. 12, pp. 2898–2902, 2020.
- [25] X. Zhu, J. Ma, X. Su, H. Tian, W. Wang, and S. Cai, “Information spreading on weighted multiplex social network,” *Complexity*, vol. 2019, Article ID 5920187, 2019 pages.
- [26] Z. Zhang, Y. Liu, X. Chen et al., “Sequential optimization for efficient high-quality object proposal generation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1209–1223, 2018.
- [27] Y. Yan, Z. Chen, J. Wu, L. Wang, K. Liu, and Y. Wu, “Effective data transmission strategy based on node socialization in opportunistic social networks,” *IEEE Access*, vol. 7, pp. 22144–22160, 2019.
- [28] A. Vahdat and D. Becker, *Epidemic Routing for Partially Connected Ad Hoc Networks*, *Handbook of Systemic Auto-immune Diseases*, Elsevier, Amsterdam, Netherlands, 2000.
- [29] S. Sisodiya, P. Sharma, and S. K. Tiwari, “A new modified spray and wait routing algorithm for heterogeneous delay tolerant network,” in *Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 843–848, Palladam, India, February 2017.
- [30] D. K. Sharma, S. K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananeey, and J. J. P. C. Rodrigues, “A machine learning-based protocol for efficient routing in opportunistic networks,” *IEEE Systems Journal*, vol. 12, no. 3, pp. 2207–2213, 2018.
- [31] K. Tang, C. Li, H. Xiong, J. Zou, and P. Frossard, “Reinforcement learning-based opportunistic routing for live video streaming over multi-hop wireless networks,” in *Proceedings of the 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, London, UK, May 2019.
- [32] J. Wu, Z. Chen, and M. Zhao, “Information cache management and data transmission algorithm in opportunistic social networks,” *Wireless Networks*, vol. 25, no. 6, pp. 2977–2988, 2019.
- [33] Y. Yan, Z. Chen, J. Wu, L. Wang, K. Liu, and P. Zheng, “An effective transmission strategy exploiting node preference and social relations in opportunistic social networks,” *IEEE Access*, vol. 7, pp. 58186–58199, 2019.
- [34] J. Wu and Z. Chen, “Human activity optimal cooperation objects selection routing scheme in opportunistic networks communication,” *Wireless Personal Communications*, vol. 95, no. 3, pp. 3357–3375, 2017.
- [35] R. Drăgan, R. I. Ciobanu, and C. Dobre, “Leader election in opportunistic networks,” in *Proceedings of the 2017 IEEE 16th International Symposium on Parallel and Distributed Computing (ISPDC)*, Innsbruck, Austria, July 2017.
- [36] F. Zeng, N. Zhao, and W. Li, “Effective social relationship measurement and cluster based routing in mobile opportunistic networks,” *Sensors*, vol. 17, no. 5, pp. 1109–1119, 2017.
- [37] K. Liu, Z. Chen, J. Wu, and L. Wang, “FCNS: a fuzzy routing-forwarding algorithm exploiting comprehensive node similarity in opportunistic social networks,” *Symmetry*, vol. 10, no. 8, pp. 338–8, 2018.
- [38] J. Niu, J. Guo, Q. Cai, N. Sadeh, and S. Guo, “Predict and spread: an efficient routing algorithm for opportunistic networking,” in *Proceedings of the 2011 IEEE Wireless Communications and Networking Conference*, pp. 498–503, Cancun, Mexico, March 2011.
- [39] Z. Zhang and V. Saligrama, “Zero-shot learning via joint latent similarity embedding,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6034–6042, Seattle, WA, USA, June 2016.
- [40] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. G. Hauptmann, “Semi-supervised multiple feature analysis for action recognition,” *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 289–298, 2014.
- [41] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proceedings of the Second International ICST Conference on Simulation Tools and Techniques*, Rome, Italy, March 2009.
- [42] J. Wu, Z. Chen, and M. Zhao, “SECM: status estimation and cache management algorithm in opportunistic networks,” *The Journal of Supercomputing*, vol. 75, no. 5, pp. 2629–2647, 2019.
- [43] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, “Gemsec,” in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 65–72, Vancouver, Canada, August 2019.
- [44] J. Leskovec, D. Huttenlocher, and J. Kleinberg, “Signed networks in social media,” in *Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI '10*, pp. 1361–1370, Atlanta, GA, USA, April 2010.
- [45] P. Massa, M. Salvetti, and D. Tomasoni, “Bowling alone and trust decline in social network sites,” in *Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 658–663, Chengdu, China, December 2009.
- [46] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, “Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters,” *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.