Flow-Based Clustering on Directed Graphs: 
A Structural Entropy Minimization Approach

YUAN YAO1,2, YICHENG PAN3, SHAOJIANG WANG4, Hongan Wang2,5,6, 
AND WAQAS NAZEER7

1School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China 
2Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China 
3State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China 
4Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China 
5Zhejiang Lab, Hangzhou 311121, China 
6Department of Mathematics, Government College University, Lahore 54000, Pakistan 
Corresponding author: Yicheng Pan (yichengp@buaa.edu.cn)

This research was supported in part by the National Natural Science Foundation for Young Scientists of China Grant No. 61807034, and in part by the National Natural Science Foundation of China Grant No. 61932002.

ABSTRACT In this study, we focus on flow-based clustering on directed graphs and propose a localized algorithm for this problem. Flow-based clustering in networks requires a set of closely related vertices where the flow amount getting into it is larger than that going out of it. It is able to formulate a variety of practical problems, such as fund-raising set detection in financial networks and influential document clustering in citation networks, etc. Methodologically, we propose the new concept of two-dimensional structural entropy on directed graphs, and based on this, a local structural entropy minimization algorithm for detecting the flow-based community structure of networks is designed. We adopt our algorithm for the problem of fund-raising set detection in financial networks, in which vertices represent accounts, edges represent transactions between two accounts, and weights represent money amounts of transactions. In our experiments, the local two-dimensional structure entropy minimization algorithm is devoted to find a fund-raising community which involves a given input account. We conduct experiments on both synthetic and real fund-raising datasets. The experimental results demonstrate that, given a fixed account, our algorithm is able to efficiently locate a fund-raising community (if any) for which the fund flowing into the community is much higher than that flowing out, and the transactions within the community are relatively denser (fund amount based) than that of inter-community. For a synthetic ground-truth fund-raising community, we adjust the parameters to change its fund-raising tendency. The results for the synthetic datasets show that our algorithm obtains higher precision and recall rates as this tendency gets stronger with each single factor varying. For a real fund-raising community embedded in a simulated capital flow network, our algorithm also find it with high precision and recall rates. The experiments for both scenarios verify the effectiveness of our algorithm.

INDEX TERMS Network flows, directed graph, graph mining, structural entropy.

I. INTRODUCTION Using Networks to Represent Complex Systems in various fields has become ubiquitous. Examples include the World Wide Web (WWW), social networks, business collaborations, scientific citations, neural networks, biological networks, and metabolic networks, etc. [1]–[5]. In the last decades, the problem of graph clustering or community detection has attracted huge attention. However, Most of the methods designed for resolving the case of community structure analysis in undirected networks [6]–[10]. Only a limited number of methods were proposed that are based on the directed network issues [11], [12].

A network with unsymmetrical links (flows) in which connections between vertices do not always represent reciprocal relations is commonly considered to be a directed network. It has various applications for many domains. For example, Network flows may represent logistic over transportation systems, car flows over traffic systems and fund flows over financial systems. Cluster or community structure based on network flows not only reveals essential features of network organization but also helps understand the
effect of network topology on the dynamics of the complex systems [2].

A relation between a pair of entities in several real-world networks is by nature directed and a fundamentally different dynamic may be generated when some of these relations have been in the reversed direction. Thus, it is meaningful to absorb all the available information including directional information in the graph during the clustering process. This phenomenon has produced a revival of interest in community detection or clustering problems in directed networks. Furthermore, plenty of research works in several scientific disciplines have proved that considering link direction can shed light on clustering for real-world applications on directed networks. For example, clusters in the hyperlink of the WWW represent web pages with same topic category [13]; Clusters in scientific citation network are used to comprehend the connection patterns between scientific disciplines papers [14]; Cluster analysis in brain networks can help neuroscientists to extract functional subdivisions of the brain [15].

Just like an independent compartment of a graph, the cluster or communities can be considered as a set of vertices which share similar features. In this generic definition, it is important to specify the notion of similarity among the vertices in the directed network. According to the traditional definition of clusters in both directed and undirected networks, similarity is regarded as edge density characteristics, which means the number of edges between vertices assigned to the same clusters is larger then edges between different clusters [16], [17]. Unlike the previous definition mentioned above, Guimerà et al. [18] have defined similarity in the directed network correspond to similar patterns, interpreted as vertices from a cluster have similar topological properties. Rosvall and Bergstrom [14] have adopted an information theory-based method, inducing a flow of information among the entities and the clustering structure depends on how information flows. Then flow-based clusters were proposed under the definition that a community is a group of vertices where a random surfer is more likely to be trapped inside instead of moving out of the group [19]. Although many works enhanced the generalized modularity partition methods through appropriate transformations [20], [21], it seems to be incapable of finding the most natural community structure due to the resolution limit issue [22].

Besides the open issue of how to describe the community in directed network, finding clusters or communities efficiently is a significant issue. In this article, we propose a problem of flow-based clustering on directed graphs. Flow-based clustering in networks requires a set of closely related vertices where the flow getting into this set is larger than that going out of it. Thus, we focus on community detection which is provided with two properties:

1. Vertices in a cluster are relatively densely connected;
2. The total amount of inflows is greater than that of the outflows.

This kind of flow-based clustering can be inferred to represent communities of special significance in real-world networks, such as fund-raising communities in financial networks. Another example is the citation networks, in which if paper \( u \) cites paper \( v \), then there is a directed edge linking from \( u \) to \( v \). A cluster of influential papers with a significant topic will be cited a lot, which means a group of closely related works which have a large number of citations from others to them. So the flow-based clustering problem on citation network requires to find such highly-cited document clusters.

Note that for the community detection problem defined above for directed networks, we cannot partition the given network into communities such that each one is a good cluster, since we cannot guarantee a high quality for every community, because at least, not all communities are able to have higher inflows than outflows simultaneously. So this problem often simulates a community detection problem that asks for the one involving a specific vertex. In this paper, we demonstrate its application in fund-raising community detection, which is quite an important problem in financial network analysis, for example, in the field of fund supervision. For this application, we need to mine a fund-raising community that involves a suspicious account that collects money in public. In China, the private equity fund is a kind of investment behavior that has no legal status. Its contractual dependencies between investors and fund managers are confirmed by moral constraints. Once the confidence of this cooperation has been shattered, investors’ rights will not be protected by the laws. Therefore, it is important to supervise these fund-raising communities in the financial network of China. The tide of funds flows into a commercial institution as financing behavior while money flows out in return for the interest of collecting money.

There are plenty of algorithms and their varieties for clustering on undirected graphs, such as k-means, DBSCAN, etc. For example, in [23], Xiao proposed an evidential fuzzy multicriteria decision making method by integrating Dempster-Shafer theory with belief entropy and the same author in [24] proposed a novel method for multi-sensor data fusion based on a new belief divergence measure of evidences and the belief entropy. We refer [25]–[28] for details about decision making algorithms. However, most of the classic clustering algorithms are either working for symmetric connectivities or global ones that works on the whole network. A straightforward strategy for community detection is exploring the entire network before deciding the best community for the selected vertex. But most existing solutions adopting this global strategy are very costly, as all vertices in the network may need to be visited. Taking into consideration of network scale and the complexity of conditionality and intricate connections between vertices, we need a local strategy. In this paper, we develop a localized algorithm based on structural entropy [19].

In the application of fund-raising community detection, our localized clustering algorithm seeks to find such a collection of accounts that are closely related and rise to huge amounts of money, for the purpose of further risk estimation. We conduct experiments on both synthetic and real fund-raising...
datasets. There are two main parts in each network used in our experiments. The first part is the environment of the capital flow network. It can be typically constructed by using the transaction data in a centralized financial institute, for example, the banks. However, due to the privacy of commercial data, we are not able to obtain it easily. We will simulate it via a random graph model, that is, the directed preferential attachment model with edge weights. The second part is a ground-truth fund-raising community which will be embedded into the capital flow network. We will use both synthetic and real fund-raising structures in our experiments. The goal to construct a synthetic community is to evaluate the effectiveness of our algorithm. There will be several parameters to adjust the community’s fund-raising ability. The experiment results demonstrate that our algorithm performs better for the community with a stronger fund-raising tendency. This is a piece of crucial evidence for the effectiveness of our algorithm. The real fund-raising community comes from a real illegal fund-raising case in China, whose characteristics will be introduced in Section V-A3. The experiment results also verify the effectiveness of our algorithm in terms of high precision and recall rates.

II. OUR METHOD AND CONTRIBUTIONS

We adopt the structural entropy minimization algorithm for the flow-based clustering problem. Structural entropy was raised by Li and Pan [29] for undirected graphs to measure the information embedded in graph structures. The two-dimensional structural entropy can be used to evaluate the clustering quality of a partition [30], [31]. In this paper, we generalize the two-dimensional structural entropy to directed graphs and propose a local algorithm for the flow-based clustering problem. Then we verify our algorithm on synthetic and real datasets for the application of fund-raising community detection problem. We summarize our contributions as follows.

1) Flow-based clustering problem
We propose the flow-based clustering problem on directed graphs for practical applications such as fund-raising detection and influential document clustering, etc. Different from other clustering problems, besides a close connection within clusters, the flow-based clustering problem requires the inflow amount larger than the outflow amount. Although, like the common clustering problem, no formal definition of this new problem is formulated, we can design an algorithm for these two targets to solve this problem.

2) Two-dimensional structural entropy of directed graphs
It is the first time to generalize the concept of structural entropy from undirected graphs to directed ones (Definition (1)). Compared to the definition of structural entropy for undirected graphs that can be expressed with edge weight and vertex degree, the new definition for directed graphs cannot be expressed explicitly, unless the (unique) stationary distribution of random walk is given. The study of structural entropy on directed graphs is of independent interest.

3) Structural entropy minimization approach to flow-based clustering problem
We present a localized algorithm for the flow-based clustering problem. This algorithm aims to minimize the objective function formulated as two-dimensional structural entropy on directed graphs. To find the community that includes a given vertex, it involves recursively and sequentially neighboring vertices into the main set with a greedy strategy to minimize the structural entropy. Since the structural entropy on directed graphs is newly defined, this algorithm is certainly new and non-trivial.

4) Design of synthetic datasets for the fund-raising community in financial networks
In the experiments, we design a model to simulate the fund-raising set with fund flows. Due to the privacy of commercial data, the main purpose of our simulation for such datasets is to verify our algorithm. We conduct four groups of experiments with different parameters of the fund-raising set varying. The four parameters are elaborately designed to evaluate the fund-raising tendency of the set, and thus the results are convincing to demonstrate the effectiveness of our algorithm, which performs better and better in terms of higher precision and recall rates as this tendency gets stronger.

5) Algorithm for the application in practical fund-raising community detection
We propose an algorithm for the fund-raising community detection problem which is of great importance in financial network study. This is a new method for fund flow analysis in terms of clustering on directed graphs. Since the community properties are presumed based on network flows, our algorithm does not need to extract any characteristics of accounts except transaction objects and fund amounts, and all the operations are done on the capital flow network. This preserves the original information of the data to the greatest extent.

The rest of the paper is organized as follows. In Section III, we introduce some notations and definitions on random walks, Markov chain and conductance, which are the basis of structural entropy. Then we give the formal definition of the two-dimensional structural entropy on directed graphs. In Section IV, we present the local algorithm for the flow-based clustering problem. In Section V, we present the experiments to verify the effectiveness of our algorithm and analyze the results. We conclude our study in Section VI.
denoted by \(d_{in}(v)\) (resp. \(d_{out}(v)\)), is defined to be the sum of weights of the edges that point to (resp. from) \(v\). That is, 
\[
d_{in}(v) = \sum_{(u,v)\in E} w(u,v) \quad \text{and} \quad d_{out}(v) = \sum_{(v,u)\in E} w(v,u).
\]

**A. RANDOM WALK AND MARKOV CHAIN**

Consider the random walk on \(G\) defined as follows. Starting from a fixed vertex, in each round, the walker who is located at vertex \(u\) takes a random step to a neighbor \(v\) with probability
\[
p(u,v) = \frac{w(u,v)}{\sum_{v'(u,v)\in E} w(u,v)}.
\]

It is well known that the random walk is a Markov process, and the probability transition matrix \(M\) is the one whose \((u,v)\)-th \((u,v \in V)\) entry is \(p(u,v)\). So the sum of each row of \(M\) is 1. A probability distribution vector \(\pi\) is said to be a stationary one if \(\pi M = \pi\), that is, the further step of random walk keeps the distribution unchanged. For undirected graphs, the stationary distribution is unique if and only if the graph is connected, and the probability on each vertex is proportional to its degree. But for directed graphs, the condition becomes a bit sophisticated. If the directed graph is strongly connected, that is, for each pair of vertices there is a path from any of them to the other, then the stationary distribution is unique. Otherwise, the graph has one or more absorbing sets, where each of them is a vertex set from which no edge points out. If the number of minimal absorbing sets is more than one, then the stationary distribution is not unique in the sense that the probability mass on each absorbing set can be arbitrary, and for other parts of the graph, the probability mass is zero. Therefore, sometimes we also require that for a stationary distribution \(\pi\), \(\pi = \lim_{n \to \infty} \mathbf{1}_v M^n\), for any vertex \(v\), where \(\mathbf{1}_v\) is the vector whose \(v\)'s entry is one while others are zero. This means that starting from any vertex \(v\), the random walk converges to the stationary distribution.

It is well-known that the Markov chain is ergodic if and only if the graph is strongly connected and aperiodic. An ergodic Markov Chain has the unique stationary distribution. So in practice, people often make the random walk take teleportations to make it ergodic. For example, in the Pagerank algorithm proposed Page et al. [32], a positive mass is allocated to the initial state in each round of transfer. If the initial state is a uniform distribution, then the stationary distribution is not unique in the sense that the probability mass on each absorbing set can be arbitrary, and for other parts of the graph, the probability mass is zero. Therefore, sometimes we also require that for a stationary distribution \(\pi\), \(\pi = \lim_{n \to \infty} \mathbf{1}_v M^n\), for any vertex \(v\), where \(\mathbf{1}_v\) is the vector whose \(v\)'s entry is one while others are zero. This means that starting from any vertex \(v\), the random walk converges to the stationary distribution.

**B. CONDUCTANCE OF DIRECTED GRAPHS**

In graph theory, conductance of a vertex set is usually used to evaluate its clustering quality. For an undirected graph \(G' = (V', E')\) and a vertex subset \(S' \subseteq V'\), the conductance of \(S'\) is defined as
\[
\Phi(S') = \frac{\sum_{j=1}^L \sum_{v \in V_j} d_v \log_2 \frac{d_v}{\text{vol}(V_j)}}{2m \log_2 \frac{\text{vol}(V_j)}{2m}},
\]
where \(\sum_{j=1}^L \sum_{v \in V_j} d_v \log_2 \frac{d_v}{\text{vol}(V_j)}\) is the number of edges between \(S'\) and its complement, and \(\text{vol}(S')\) is the volume of \(S'\), that is the sum of degrees of vertices in \(S'\). The conductance of \(G'\) is defined as
\[
\Phi(G') = \min_{\text{vol}(S') \leq \text{vol}(V')/2} \{\Phi(S')\}.
\]

The conductance of a vertex set on undirected graph is closely related to the random walk, since \(\Phi(S')\) is in fact the probability, under the stationary distribution, of random walk going out of \(S'\) conditional on \(S'\). Thus a good clustering of vertices traps the random walk easily and has low conductance.

This understanding of conductance can be generalized to directed graphs naturally. The following definition of conductance in digraphs has been formulated in [33]. Consider a random walk on a strongly connected digraph \(G = (V, E)\). Under the stationary distribution \(\pi\), for any vertex set \(S \subseteq V\), \(\pi(S) = \sum_{v \in S} \pi(v)\) is the probability of being in \(S\). For each edge \((u,v) \in E\), we define
\[
\pi(u,v) = \pi(u)p(u,v),
\]
which is the probability that edge \((u,v)\) is traversed. For an edge set \(A \subseteq E\), we also define \(\pi(A) = \sum_{(u,v) \in A} \pi(u,v)\). So, similar to the case of undirected graphs, the conductance of \(S\) is defined as
\[
\Phi(S) = \frac{\pi(\partial(S))}{\pi(S)},
\]
where \(\partial(S) = \{(u,v) \in E \mid u \in S, v \notin S\}\). By definition, \(\Phi(S)\) is the probability of random walk under the stationary distribution going out of \(S\) conditional on \(S\). The conductance of \(G\) is defined as
\[
\Phi(G) = \min_{\pi(S) \leq 1/2} \{\Phi(S)\}.
\]

**C. TWO-DIMENSIONAL STRUCTURAL ENTROPY OF GRAPHS**

The structural entropy of undirected graphs is raised by Li and Pan [29]. For an undirected graph \(G' = (V', E')\) and a partition \(\mathcal{P} = \{V'_1, \ldots, V'_L\}\) on \(V'\), the structural entropy on \(\mathcal{P}\) is defined as
\[
\mathcal{H}(\mathcal{P}) = - \sum_{j=1}^L \sum_{v \in V_j} \frac{d_v}{2m} \log_2 \frac{d_v}{\text{vol}(V_j)} - \sum_{j=1}^L \frac{g_j}{2m} \log_2 \frac{\text{vol}(V'_j)}{2m},
\]
where \(m = |E|\) and \(g_j = |(V'_j, V'_j)|\).

The idea of this definition is based on the coding of the random walk. A random walk can be encoded by the sequential vertices that it visits. By information theory, the average length of the binary codeword for each step in an infinite random walk is lower bounded by
\[
- \sum_{v \in V}(d_v/2m) \log_2 (d_v/2m),
\]
which is the famous Shannon’s entropy over the degree distribution of the graph. The structural entropy on a partition \(\mathcal{P}\) is obtained by the following way. For vertex subset \(V'_j\) in \(\mathcal{P}\), we give a codeword whose length is \(- \log_2 \frac{\text{vol}(V'_j)}{2m}\). This is the self-information of
on the probability distribution on $\mathcal{P}$, each entry of which is proportional to its volume. Meanwhile, there is a codeword for each vertex. But unlike the global encoding in the former method, each vertex is encoded within each vertex subset in $\mathcal{P}$. The self-information is given by the proportional degree $d_i/vol(V_j)$. For the random walk, the codeword of $V_j$ is not used unless it gets into $V_j$ from $\overline{V}_j$ (or equivalently, out of $V_j$ to $\overline{V}_j$), whose probability under the stationary distribution is $g_j/2m$. Putting the codeword for each vertex together (the first summation of Equation (1)), we get the lower bound of the average length of the binary codeword in the new way for each step in an infinite random walk. It is easy to see that, roughly, the smaller the conductance of each set is, the smaller $H^{\mathcal{P}}(G')$ is, which means that good clustering (in terms of a partition) leads to small two-dimensional structural entropy. The two-dimensional structural entropy of $G'$ is defined as the minimum one over all partitions. That is,

$$H(G') = \min_{\mathcal{P}} [H^{\mathcal{P}}(G')].$$

(2)

It is worth noting that the definition of structural entropy can be generalized to a weighted graph naturally. Therefore, $H(G)$ can be used to evaluate the clustering of weighted graphs in many practical fields [30], [31]. For more information and explanation on structural entropy, see [29].

For directed graphs, the same idea can be transplanted easily. Formally, we have

**Definition 1:** Let $G = (V, E)$ be a strongly connected directed graph, $\mathcal{P} = \{V_1, V_2, \ldots, V_\ell\}$ be a partition on $V$, and $\pi$ be the stationary distribution. The structure entropy of $G$ for partition $\mathcal{P}$ is defined as

$$H^{\mathcal{P}}(G) = -\sum_{j=1}^{\ell} \left( \sum_{v \in V_j} \pi(v) \log_2 \frac{\pi(v)}{\pi(V_j)} \right) - \sum_{j=1}^{\ell} \Phi(V_j) \cdot \pi(V_j) \log_2 \pi(V_j),$$

(3)

where $\Phi(V_j)$ is the conductance of $V_j$. Similar to the undirected case, the two-dimensional structural entropy of $G$ is defined as the minimum one over all partitions. That is,

$$H(G) = \min_{\mathcal{P}} [H^{\mathcal{P}}(G)].$$

In practice, it is possible that the given graph is not strongly connected. So usually teleportations are imposed to make the Markov Chain ergodic. Note that the stationary distribution $\pi$ cannot be explicitly expressed although it exists. However, there are a plenty of algorithms to compute it approximately, for example, the Arnoldi method [34].

Compared to the two-dimensional structural entropy for undirected graphs, for a vertex subset $V_j$ in $\mathcal{P}$, the self-information of $V_j$ becomes $-\log_2 \pi(V_j)$. Under the stationary distribution of the random walk, the probability of leaving $V_j$ is $\Phi(V_j) \cdot \pi(V_j)$, which is the counterpart of $g_j/2m$ in the undirected case. Thus, Equation (3) is a natural generalization of Equation (1) to directed graphs. Similarly, a small $H^{\mathcal{P}}(G)$ implies a good clustering on directed graphs.

**IV. ALGORITHMS**

The two-dimensional structural entropy on undirected graphs is computed approximately by an agglomerative process [29], [30]. The goal is to find a partition $\mathcal{P}$ such that $H^{\mathcal{P}}(G)$ is minimized. Suppose that we have already known an approximate stationary distribution $\pi$ (with teleportation) for $G$, which can be computed by any efficient algorithm for the Pagerank vector.

To minimize $H^{\mathcal{P}}(G)$, initially, each vertex is viewed as an individual set, and the two-dimensional structural entropy is essentially the Shannon’s entropy of $\pi$, which has the largest value over all partitions. This can be easily checked by definition. In the agglomerative process, a greedy strategy is used. Recursively, two sets are picked such that when merging them into a single set, the new partition makes the structural entropy decrease most, until no merge of two sets makes it decrease. That is, for the former partition $\mathcal{P}_1$ and the latter one $\mathcal{P}_2$, $H^{\mathcal{P}_2}(G) - H^{\mathcal{P}_1}(G)$ is positive and maximum in each step.

Above is a global algorithm for $H^{\mathcal{P}}(G)$ that outputs a partition on all vertices. In practice, the input graph is usually very large, the global algorithm is time-consuming since the time complexity is related to (although almost linear in) the graph size. In this paper, for the application that finds a small set of small conductance that containing a certain vertex, we develop a local algorithm.

The idea of the local algorithm is similar to the global one. The only difference is that, starting from the input vertex $v$, in each round, only a single vertex is added to the set containing $v$, called main set. The same criterion is used here with a greedy strategy. The high efficiency of the local algorithm is because not only that not all sets need to be considered, but also the difference of structural entropy $H^{\mathcal{P}_2}(G) - H^{\mathcal{P}_1}(G)$ can be calculated locally since only one vertex joins the present main set. In this case, we denote by $\mathcal{P}_S$ this kind of partition in which the main set is $S$. The end of this process can be determined by any reasonable condition. For example, an explicit external parameter on the set size or a structural entropy value, or an internal criterion for truncation this process can be determined by any reasonable condition.

Algorithm 1 gives a detailed explosion of the approach mentioned above. It has a directed model network $G = (V, E, w)$ as input and outputs a fund-raising set $\mathcal{S}$ from $G$. In the local structural entropy minimization (LSEM) algorithm, we start from a vertex $v$ and make use of two sets to perform the local clustering process. The set $S$ as the main set restores the currently selected vertices in $G$, while the set $N$ restore the neighboring vertices of current $S$. In each round of updating $S$, only the one from $N$ which has the maximum variation of structural entropy between previous and current stages is chosen as the new member of $S$. We extend set $S$ with
Algorithm 1 Local Structural Entropy Minimization (LSEM) Algorithm

Input: a directed graph $G$; a vertex $v$ in $G$.

Parameters: $s$ (the size of output set); $\alpha$ (the probability of teleportations in PageRank).

Output: a vertex subset $S$ with size no more than $s$ that contains $v$.

1: Using PageRank to calculate stable distribution with teleportation probability $\alpha$
2: $S = \{v\}$
3: $\delta(S) = 0$
4: Set $N$ to denote all neighbors of $S$
5: while $|S| < s$ and $\delta(S) \geq 0$ do
6: for each vertex $u \in N$ do
7: $\Delta S(u) = \mathcal{H}_{Ps}(G) - \mathcal{H}_{Ps\cup\{u\}}(G)$
8: end for
9: $\delta(S) = \max_{u \in N} \Delta S(u)$
10: $u = \arg \max_{u \in N} \Delta S(u)$
11: $S \leftarrow S \cup \{u\}$
12: Add neighbors of vertex $u$ not in $S$ to $N$
13: end while
14: Return $S$

the above strategies until the scale of $S$ reaches the predefined community size.

V. EXPERIMENTS

In this section, we conduct experiments that try to find fund-raising communities in financial networks, in which a vertex represents an account and a weighted edge represents a fund transfer. A fund-raising community, denoted by $S$, is a collection of accounts that satisfies two properties. The first is that the connections in $S$ (weights based) is relatively dense, and the second is that the funds flowing into $S$ is more than that getting out. These two properties imply that a fund-raising community has small conductance.

Both synthetic and real fund-raising communities are used in our experiments. We will describe the structure of the synthetic one in Section V-A, and explain how to adjust its fund-raising tendency via changing its parameters. The real one comes from a real illegal fund-raising case. Because of the privacy of commercial data, we will embed both of these two structures as ground-truth into synthetic underlying capital flow networks, which are generated from the directed power law graph model. We will use the LSEM algorithm to find them.

A. DESCRIPTION OF DATASETS

1) SIMULATED CAPITAL FLOW NETWORK

Before we establish a simulated capital flow network, let’s look at the degree distributions of a real capital flow network, which indicates that a directed power law graph model is suitable for synthetic underlying capital flow networks. A graph has a power law degree distribution if the (fractional) number of vertices of degree $d$, denoted by $n_d$, is proportional to $d^{-\beta}$ for some constant $\beta > 0$, which implies the log–log relationship between $n_d$ and $d$ is $\log n_d = -\beta \log d + \log \alpha$ for some positive $\alpha$. Many real networks have such degree distribution for $2 < \beta < 3$. This real capital flow network is constructed from the real transaction data of a city commercial bank in China which involves all transactions within 18 consecutive months. It involves 14,467,045 vertices and 24,011,707 directed edges that represent accounts and transaction relations, respectively. The in-degree and out-degree distributions of unweighted and weighted networks are illustrated in Figure 1. The weight means the total amount of funds transferred from one account to another within 18 months. Figure 1 shows that both in-degree and out-degree of unweighted and weighted versions have evidently power-law distribution.

Therefore, in our simulation, we establish the simulated capital flow network by using a classic directed power law network model, that is, the preferential attachment model for directed graphs introduced in [35]. This model is generalized from the classic preferential attachment model [36] for undirected graphs. For the completeness of our paper, we introduced it briefly.

A graph is generated with vertices added one by one. Preferential attachment means that in the process of graph growth, the vertices with a large degree tend to have more links than the ones with a small degree. Starting from any directed graph, three kinds of steps are chosen randomly in each round: source-vertex step, sink-vertex step and edge step. For the source-(resp. sink-)vertex step, a new vertex is added with $d$ directed edges pointing from (resp. to) it. The other endpoints of these edges are chosen randomly with probability proportional to the in-degree (resp. out-degree) of the vertices in the current graph. For the edge-step, a new edge is added with head (resp. tail) attaching to a vertex randomly with probability proportional to the in-degree (resp. out-degree) of vertices in the current graph. To avoid the drawback that zero in-degree (resp. out-degree) vertex will always have zero in-degree (resp. out-degree), whenever we mention “proportional to” some degree above, we mean “proportional to” such degree plus a positive parameter $\gamma$. Assume that in each round, the source-vertex step, sink-vertex step and the edge step are taken with probabilities $p_1$, $p_2$ and $1 - p_1 - p_2$, respectively. Then it can proved that with high probability, this model produces a power law graph for the in-degree distribution with $\beta_{in} = 2 + \frac{p_2 + (p_1 + p_2)\gamma}{1 - p_2}$, and the out-degree distribution with $\beta_{out} = 2 + \frac{p_1 + (p_1 + p_2)\gamma}{1 - p_1}$.

Definition 2: A preferential attachment process is any of a class of processes in which some quantity, typically some form of wealth or credit, is distributed among a number of
embedded in the directed network, denoted by \( G \). It is designed to be a ground-truth community. The synthetic pattern of this sub-network is shown in Figure 1.

2) SYNTHETIC FUND-RAISING COMMUNITY STRUCTURE

The synthetic pattern of this sub-network is shown in Figure 1. It is designed to be a ground-truth community embedded in the directed network, denoted by \( G' \), that is generated from the directed PA model. This community has two entrances and two exits for flows. When embedded into another network (that is \( G' \) in our experiments), \( k_1 \) vertices link to the two entrances and \( k_2 \) vertices are linked by the two exits. This kind of pattern matches the architecture of the real fund-raising community we use in this paper. They represent few accounts that absorb funds from the public and refund interests back. Within this community, there are \( L \) transfer layers, each of which has \( t \) vertices. The connections between any two neighboring layers are complete with orientation from entrances to exits. The weight for each edge pointing to the entrances are chosen uniformly and randomly from the interval \([1, a]\), denoted by \( U[1, a]\). There is an attenuation coefficient within \([0, 1]\), denoted by AC, such that for each vertex in the community, it holds \( 1 - AC \) fraction of funds that flow into it, and allocates the rest funds uniformly to the outgoing edges.

Our algorithm is applied to find the community with a collection of funds mixed in the capital flow network. In our experiments, \( G' \) is generated from \( PA(n = 10000, d=3, p_1 = 0.4, p_2 = 0.4, \gamma = 1, \bar{w} = 10) \). This is a synthetic environment for the fund-raising set \( S_0 \) to be embedded in, and the parameters do not influence the property of \( S_0 \) directly. So here, we simply pick a group of moderate parameters for \( G' \).

3) REAL FUND-RAISING COMMUNITY

Our experimental data are derived from desensitization data of a state-owned bank. The real fund-raising community comes from a real illegal fund-raising case which involves 149 accounts and 510476 transactions associated with them. Each transaction is recorded in the format of \(<primary account, counterparty account, transaction date, money amount, which represents the primary account transferred such amount of money to the counterparty account on that day. The time span of this case covers 714 active days within the period from Dec. 26, 2007 to Jul. 10, 2015. These parameters are summarized in Table 1. Our community is constructed by summing up all amounts of funds transferred from one account to another in this period as the weight of the corresponding edge, which results in the ground-truth community of 149 vertices and 99130 edges. Since the transactions of the 149 accounts are given intact, we map the other accounts that are in- or out-neighbors of them uniformly and randomly to the vertices of a network which are generated from the directed power law graph model \( PA(n = 399851, d = 3, p_1 = 0.4, p_2 = 0.4, \gamma = 1, \bar{w}) \). The whole simulated capital flow network has 400000 vertices and 1599035 weighted edges in all. The weights of edges in the generated power law network are drawn from the exponential distribution with mean \( \bar{w} \) being \( \rho = 1/10 \) of the average weight of the edges associated with the 149 ground-truth accounts.

We pick the threshold \( \rho = 1/10 \) for the following reason. In practice, the outflow of personal funds can be classified into two categories: personal consumption expenditure and investment expenditure. In order to validate the effectiveness of our algorithm in a real capital flow network, more vertices individuals or objects according to how much they already have, so that those who are already wealthy receive more than those who are not.

After the above construction, we endow a weight for each edge. Each weight is drawn from an exponential distribution with mean \( \bar{w} \). We denote by \( PA(n, d, p_1, p_2, \gamma, \bar{w}) \) the preferential attachment model with these parameters.

where \( d \) is degree of vertex in graph, \( n \) is number of vertices of degree \( d \) in the graph, \( p_1 \) is the probability of source-vertex step, \( p_2 \) is the probability of sink vertex step, \( \gamma \) is any positive parameter and \( \bar{w} \) is the mean of exponential distribution from which each weight is drawn.

Note that we choose the preferential attachment model to build the network with power law degree distribution. However, this model cannot produce community structure with non-negligible probability. In our experiments, we embed ground-truth communities in such an environment. If our algorithm works well, it should find them effectively and efficiently.

FIGURE 1. Illustration of the synthetic ground-truth fund-raising community.
connected in pair by edges which have the characteristic of personal consumption expenditure need to be added. Then, we set a parameter $\rho$ which represents the mean ratio of the amount between two kinds of capital flows. It is used to reflect the difference in the amount of money between personal consumption behavior and personal investment behavior of participating in a private equity fund. $\rho$ is set to be 1/10 in the real data experiment, which means that the amount of personal consumption funds divided by the amount of personal private investment funds equals 1/10. This value is based on the following two facts:

1. According to the annual data of the National Bureau of Statistics of China, as figure 2 shows, people’s annual consumption expenditure is increasing from 2014 to 2018, and the average in five years is 17,097.8 RMBs.

2. The drafting group of the Law of the People’s Republic of China on Securities Investment Funds pointed out that in order to ensure the interests of investors, the minimum amount of funds raised by individuals is generally 200,000 RMBs.

On the one hand, the per capita consumption expenditure in many cities exceeds the average. On the other hand, there are several kinds of personal consumption expenditure, which will divert capital flow. Therefore, we conservatively set $\rho$ to be 1/10.

In this illegal fund-raising case, there are two special accounts in the 149 ones. They have the highest unweighted in-degree (88,190 connectivities from accounts out of the ground-truth community) and out-degree (88,258 connectivities to accounts out of the ground-truth community), respectively. The ones with the second highest in-degree and out-degree have large differences from both of them. They are only 1,427 and 1,221, respectively. The highest degree vertices represent the two main accounts that absorb funds and release interests, respectively. This is a significant feature of fund-raising community from the public. In practice, these two accounts can be detected easily since both of them transfer money with the public. That is why it is reasonable to detect fund-raising communities from a specific account, since we can get its information easily. They are the analogs of the entrances and exits of the synthetic fund-raising community. The funds collected from the public would be split up and transferred for several times. Some of them will be used to refund interests and others are held by intermediate accounts, which can be simulated with flexible parameters in our synthetic fund-raising architecture.

From now on, we denote by $G$ the simulated capital flow network, and by $S_0$ the fund-raising account set for both synthetic and real cases without ambiguity by context.

### TABLE 1. Real dataset of the illegal fund-raising case.

| Real Dataset of the illegal fund-raising transaction data | Time Interval | Size | Available Files |
|---------------------------------------------------------|---------------|------|-----------------|
| 26/12/2007-10/07/2015 | 510476 useful records | primary account; counterparty account; transaction date; money amount |

![FIGURE 2. Per capita consumption of China.](image_url)

**B. RESULTS AND ANALYSIS**

1) SYNTHETIC FUND-RAISING COMMUNITY DETECTION

To demonstrate the effect of our algorithm for the synthetic fund-raising community, we change the pattern of $S_0$ by adjusting several parameters: the number of layers, the expected amount of funds that flow in, the number of vertices out of $S_0$ that $S_0$ contacts with, and the attenuation coefficient of capital transfer in $S_0$. Recall that $S_0$ simulates a fund-raising set with two entrances and two exits, and the intermediate 60 vertices are layered into $L$ levels. Here, we pick $L = 2, 4, 6, 10$ to simulate the capital transfer chains in $S_0$ changing from short to long. The amount of funds $w$ that flow into $S_0$ from each vertex is chosen randomly from $U[1, a]$, where $a = 10, 20, 30, 40, 50$. The number of vertices that contact to $S_0$, which is denoted by $k_1$ and $k_2$ in the description of $S_0$, are chosen simply to be the same, denoted by $c$ here. It varies as $c = 50, 100, 150, 200$. For the attenuation coefficient $AC$, it is set to be $0.6, 0.65, 0.7, 0.75$, respectively, to simulate the fund amount that is held during each transfer step in $S_0$. These three settings demonstrate the fund-raising ability of $S_0$ changing from weak to strong. The purpose of these four groups of experiments is to demonstrate the effectiveness of our algorithm since we can see that our algorithm obtains higher precision and recall rates as the fund-raising tendency gets stronger.

Two diagnostic tools that help in the interpretation of probabilistic forecast for binary (two-class) classification predictive modeling problems are ROC Curves and Precision-Recall curves. “A precision-recall curve is a plot of the precision (y-axis) and the recall (x-axis) for different thresholds, much like the receiver operating characteristic curve. The no-skill line is defined by the total number of positive cases divide by the total number of positive and negative cases”. The precision and recall can be calculated for thresholds using the precision recall curve function that takes the true output values and the probabilities for the positive class as output and returns the precision, recall and threshold values.
When plotting precision and recall for each threshold as a curve, it is important that recall is provided as the x-axis and precision is provided as the y-axis.

Suppose that starting from vertex $v_1$, our local algorithm involves $v_2, \ldots, v_s$ sequentially into the main set. We demonstrate our results by plotting the Precision-Recall (P-R) curves as $s$ varies from 1 to 100. Since the ground-truth fund-raising set is $S_0$, the precision rate of set $V_s := \{v_1, \ldots, v_s\}$ is

$$p = \frac{|S_0 \cap V_s|}{s},$$

while the recall rate is

$$r = \frac{|S_0 \cap V_s|}{|S_0|}.$$

The point at which the precision and recall rates are equal is called the break-even point.

Figure 3, 4 and 5 are depicted for our algorithm in which the starting vertex is one of the two entrances. Figure 3 demonstrates the P-R curves when $a = 10, c = 200, AC = 0.7$ while $L$ varies. It shows that as the layer number grows up, the precision and recall rates improves. For $L = 10$, our algorithm even finds the first 64 vertices totally correct and shows a perfect P-R curve! For other values of $L$, the recall rate can also get 100% before $s$ gets 100. The reason why our algorithm performs better on larger layer structure is that $S_0$ of larger layer holds more funds within itself, and intuitively has lower conductance.

Figure 4 demonstrates the P-R curves when $L = 6, c = 200, AC = 0.7$ while $a$ varies. Recall that $w$ is chosen uniformly from the interval $[1, a]$. It can be observed that as $w$ gets heavier, the P-R curve gets better. Our algorithm can also give a perfect recall rate within the first one hundred (even earlier) vertices. Larger $a$ means stronger fund-raising tendency, for which our algorithm performs better.

Figure 5 demonstrates the P-R curves when $L = 6, a = 10, AC = 0.7$ while $c$ varies. Larger $c$ means a wider range of fund-raising objects and stronger fund-raising tendency also. Our algorithm performs better on graphs of larger c, and also gets perfect recall rate within the first 100 vertices.

Figure 6 demonstrates the P-R curves when $L = 6, a = 10, c = 200$ while $AC$ varies. Our algorithm has better performance for lower attenuation coefficients. The reason for this is similar to the case that the layer number varies, that is, low attenuation coefficients indicate a strong fund-raising tendency and low conductance.

For all the figures mentioned above, the precision rate for most cases drops on the first several vertices, which means that there are some wrong ones are added to the main set at the beginning of our algorithm. This is because the starting vertex is one of the entrances of $S_0$. But the connections of
2) REAL FUND-RAISING COMMUNITY DETECTION

Before we show the results of our experiments, we look at some features about the ground-truth illegal fund-raising community $S_0$. For a set of vertices $S$, denote by $f_{in}(S)$ and $f_{out}(S)$ the total amount of funds that flow into and out of $S$, respectively. Recall that in the capital flow network, the vertices represent accounts and the weighted edges associated with such vertices represent the total amount of transmitted funds related to these accounts. We depict the curve of $f_{in}(S_0) - f_{out}(S_0)$ for the whole fund-raising period in Figure 7, in which the value of $f_{in}(S_0) - f_{out}(S_0)$ at time $t$ means the accumulative fund amount that $S_0$ raised up to time $t$.

It can be observed in Figure 7 that the accumulative fund amount during almost all time is negative! So it seems that no money collecting behavior exists for the ground-truth community! But it is certainly not true, since in practice, an illegal fund-raising community has various means of evading investigation, for example, fixed assets purchase, which cannot be observed via the $f_{in}(S_0) - f_{out}(S_0)$ curve. However, our algorithm performs well for detecting capital transfer behavior by various means. Suppose a vertex $v$ represents an account that $S_0$ bought something with. Although it is possible that a large amount of funds flows into $v$, our algorithm tends to involve $v$ into the main set and view $v$ as a part of the suspected community. Although an innocent account is wrongly attributed, the fund-raising property is able to be detected and preserved in the process of algorithm execution. For the same reason, the accounts for intentional transfer and hiding of funds can be involved easily. This is the reason why our algorithm is suitable to detect even illegal fund-raising sets.

Moreover, after a careful investigation to $S_0$, we find that the negative accumulation is caused by only a few accounts. Concretely, there are only five accounts that have a throughput, which is the minimum of in-degree and out-degree of a single account, larger than one billion. They dominate most of out-flows of $S_0$. We also depict in Figure 9 the accumulative fund amount curve of $S_0$ without these five vertices. The accumulative fund amount becomes positive. This implies that a large amount of funds deposited originally on these accounts. These funds are believed unrelated to fund-raising behavior since not too much money is transferred to their accounts during the whole fund-raising period, but these accounts are involved in this actual case.

In the experiment, we start from the vertex with the highest in-degree in $S_0$, just like that we start from an entrance in the synthetic fund-raising community detection. Note that this account is usually public for the deposit, and so we know its information. The size of the output community is set to be 200, at which our algorithm halts. The P-R curve has been depicted in Figure 8. It can be observed that after a short fluctuation at the very beginning, our algorithm involves constantly the vertices in $S_0$ into its main set. The precision rate $p$ does not drop sharply until the recall rate $r$ gets around half. For the 200 vertices that our algorithm outputs, 94 of them are correctly from $S_0$, and thus the final precision and recall rates at the end of the curve are 0.47 and 0.63, respectively. However, for the intermediate part, the highest $F_1$ score which is the harmonic mean of $p$ and $r$ is 0.673 when $p = 0.92$ and $r = 0.53$, and the break-even point at which the precision and recall rates are equal is $p = r = 0.62$.

For a further investigation, we have a look at the \"$\pi$-weighted\" precision and recall rates of the output community. The (approximate) stationary distribution $\pi$ reflects the importance of vertices. So it seems more reasonable to use the mass of $\pi$ than simply the cardinality in considering precision and recall rates. Using the aforementioned notations, the $\pi$-weighted precision rate of set $V_s := \{v_1, \ldots, v_s\}$ is defined to be

$$p_\pi = \frac{\pi(S_0 \cap V_s)}{\pi(V_s)} ,$$

and the $\pi$-weighted recall rate is

$$r_\pi = \frac{\pi(S_0 \cap V_s)}{\pi(S_0)} .$$
The $\pi$-weighted P-R curve has been depicted in Figure 9. The $\pi$-weighted precision and recall rates of the 200 output vertices are 0.95 and 0.70, respectively. Both of them are higher than the common P-R rates of it. Compared to the common P-R curve in Figure 8, the $\pi$-weighted one does not drop down, and the lowest precision during the whole procedure of output is as high as 0.86, even the unweighted one drops to 0.47. The reason is that the $\pi$ values of the latter part vertices are very small, which implies that they are not so important as the former part vertices. The $\pi$-weighted recall rate 0.70 which is higher than the unweighted one 0.63. This also implies that the output vertices in the ground-truth set $S_0$ is relatively more important than the rest part of it.

For the community $S$ of size 200 that our algorithm explores, we also depict the accumulative fund amount $f_{\text{in}}(S) - f_{\text{out}}(S)$ as shown in Figure 10. Compared to the accumulative fund curve of $S_0$ in Figure 7, two things should be noticed. The first is that although there are only 94 common vertices both in $S$ and $S_0$, the essential starting time of fund raising is almost the same. That is, the time points that the two curves deviate evidently from zero are just before the 300th day both. The second thing we have to emphasize is that the accumulative fund of $S$ is always non-negative and $S$ is indeed a fund-raising community. This means that fund-raising behavior has been detected by our algorithm. Moreover, compared to Figure 9, we can find that the accumulative curve of $S$ is closely resembling to that of $S_0$ after excluding the five vertices with the highest throughput. This verifies also the effectiveness of our algorithm. Fund transfer can be easily detected by our algorithm at the cost of involving possibly some innocent accounts in the community. In practice, we need further semantic analysis for the suspected accounts to determine the legality of them. But it is beyond the scope of this paper.

VI. CONCLUSION

In this paper, we proposed the definition of two-dimension structural entropy for weighted directed graphs and developed a local algorithm LSEM for the flow-based clustering problem. To demonstrate the effectiveness of our algorithm, we utilize it to solve the fund-raising problem on synthetic and real datasets. It can be observed from the experimental results that the LSEM algorithm is able to find the synthetic fund-raising community with high precision and recall rates for all patterns of the ground-truth set. Moreover, when the parameters are adjusted, the performance of LSEM changes according to the parameters varies. In general, a fund-raising set with a stronger fund-raising tendency is explored by our algorithm with higher precision and recall rates. For real datasets, the LSEM algorithm is able to find the real fund-raising community with $F_1$ score 0.673 and break-even point 0.62. The $\pi$-weighted P-R curve evaluate the importance of the output vertices for the ground-truth set. The important ones are also outputted early by our algorithm.

The LSEM algorithm is a new method for flow-based clustering on the directed graphs. For the future study, how to develop this structural entropy based algorithm is an open question.
important problem. Strategies other than simply agglomerative process can be considered. More applications of our algorithm in other fields is also worth study. More significantly, we are looking forward other insights and algorithms for the flow-based clustering problem. The intuitions behind classic clustering algorithms might be helpful.

REFERENCES

[1] M. E. J. Newman, “The structure and function of complex networks,” SIAM Rev., vol. 45, no. 2, pp. 167–256, 2003.
[2] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” Rev. Mod. Phys., vol. 74, no. 1, p. 47, 2002.
[3] M. Zhang, J. Deng, C. V. Fang, X. Zhang, and L. J. Lu, “Molecular network analysis and applications,” in Knowledge-Based Bioinformatics, 2010, pp. 251–287, doi: 10.1002/9780470669716.ch11.
[4] H. Weishaupt, P. Johansson, C. Engrström, S. Neland, S. Silvestrov, and F. J. Stawrting, “Loss of conservation of graph centralities in reverse-engineered transcriptional regulatory networks,” Methodol. Comput. Appl. Probab., vol. 19, no. 4, pp. 1089–1105, 2017.
[5] S. Zhang, G. Jin, X. Zhang, and L. Chen, “Discovering functions and revealing mechanisms at molecular level from biological networks,” Pro- teomics, vol. 7, no. 16, pp. 2856–2869, 2007.
[6] S. Fortunato, “Community detection in graphs,” Phys. Rep., vol. 486, nos. 3–5, pp. 75–174, 2010.
[7] M. E. J. Newman, “Detecting community structure in networks,” Eur. Phys. J. B, vol. 38, no. 2, pp. 321–340, 2004.
[8] J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and comparative study,” ACM Comput. Surv., vol. 45, no. 4, p. 43, 2013.
[9] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 69, no. 6, 2004, Art. no. 066133.
[10] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 76, no. 3, Sep. 2007, Art. no. 036106.
[11] F. D. Malliaros and M. Vazirgiannis, “Clustering and community detection in directed networks: A survey,” Phys. Rep., vol. 533, no. 4, pp. 95–142, 2013.
[12] S. Harenberg, G. Bello, L. Gjesdotter, S. Ranshous, J. Harlalka, R. Seay, F. J. Swartling, “Loss of conservation of graph centralities in reverse-engineered transcriptional regulatory networks,” in Proc. 1st ACM Int. Workshop Complex Netw., 2009, pp. 11–18.
[13] A. Fortunato and M. Barthélemy, “Resolution limit in community detection,” Proc. Nat. Acad. Sci. USA, vol. 104, no. 1, pp. 36–41, 2007.
[14] S. Xiao, “A hybrid fuzzy soft sets decision making method for assessing health-care waste treatment technologies based on D numbers,” Eng. Appl. Artif. Intell., vol. 71, pp. 216–225, May 2018.
[15] S. Xiao, “An improved method for combining conflicting evidences based on the similarity measure and belief function entropy,” Int. J. Fuzzy Syst., vol. 20, no. 4, pp. 1256–1266, 2018.
[16] S. Xiao, “Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy,” Inf. Fusion, vol. 46, pp. 23–32, Mar. 2019.
[17] S. Xiao, “A novel multi-criteria decision making method for assessing health-care waste treatment technologies based on D numbers,” Proc. Nat. Acad. Sci. USA, vol. 101, no. 1, pp. 2501–2506, 2004.
[18] S. Xiao, “A novel evidence theory and fuzzy preference approach-based multi-sensor data fusion technique for fault diagnosis,” Sensors, vol. 17, no. 11, p. 2504, 2017.
[19] S. Xiao, “Information integration fuzzy approximation reasoning method for high quality decision making,” Eng. Appl. Artif. Intell., vol. 30, pp. 190–198, 2014.
[20] S. Xiao, “A new multi-criteria decision making model for health-care waste treatment technologies based on D numbers,” Eng. Appl. Artif. Intell., vol. 30, pp. 190–198, 2014.
[21] S. Xiao, “A novel multi-criteria decision making model for assessing health-care waste treatment technologies based on D numbers,” Eng. Appl. Artif. Intell., vol. 30, pp. 190–198, 2014.
[22] Y. Yao, “Resolution limit in community detection,” Proc. Nat. Acad. Sci. USA, vol. 104, no. 1, pp. 36–41, 2007.
[23] F. Xiao, “EFMCDM: Evidence-based fuzzy multicriteria decision making based on belief entropy,” IEEE Trans. Fuzzy Syst., to be published.
[24] F. Xiao, “Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy,” Inf. Fusion, vol. 46, pp. 23–32, Mar. 2019.
[25] F. Xiao, “A novel multi-criteria decision making method for assessing health-care waste treatment technologies based on D numbers,” Eng. Appl. Artif. Intell., vol. 71, pp. 216–225, May 2018.
[26] F. Xiao, “An improved method for combining conflicting evidences based on the similarity measure and belief function entropy,” Int. J. Fuzzy Syst., vol. 20, no. 4, pp. 1256–1266, 2018.

YUAN YAO was born in 1985. He received the B.S. degree in computer science and the M.S. degree in software engineering from Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in electronic and information from the Institute of Software, Chinese Academy of Sciences, Beijing, China. He is currently with the University of Chinese Academy of Sciences, Beijing, and the Beijing Key Laboratory of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing. Since 2014, he has been engaged in the research of intelligent information processing.

YICHENG PAN was born in 1983. He received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2012. In 2015, he completed his Postdoctoral study at the Institute of Information Engineering, Chinese Academy of Sciences. He is currently with the State Key Laboratory of Software Development Environment, Beihang University, Beijing. His research interests include computation complexity, local algorithms, and network theory.
SHAOJIANG WANG received the Ph.D. degree from the Institute of Software, Chinese Academy of Science, Beijing, China, in 2019. Recently, he starts his Postdoctoral study at Tsinghua University with the direction of network theory and local algorithms.

HONAN WANG received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1999. Since 2002, he has been a Professor with the Institute of Software, Chinese Academy of Sciences. He is currently the Director of the Intelligence Engineering Laboratory. His research interests include human–computer interaction, real-time intelligence, and real-time active database.

WAQAS NAZEER received the Ph.D. degree in mathematics from the Abdus Salam School of Mathematical Sciences, Government College University, Lahore, Pakistan. He is currently an Assistant Professor with Government College University, Lahore. He is also a Mathematician from Pakistan. During his studies, he was funded by the Higher Education Commission of Pakistan. He has published over 100 research articles in different international journals. His research interests are analysis and graph theory. He received the Outstanding Performance Award for the Ph.D. degree.

---

**VOLUME 8, 2020**

152591