Reconstruction of Tree Network via Evolutionary Game Data Analysis

Xiaoping Zheng, Wenhan Wu, Wenfeng Deng, Chunhua Yang, *Senior Member, IEEE*, and Keke Huang, *Member, IEEE*

***Abstract*—As one of the most effective technologies for network reconstruction, compressive sensing can recover signals from a small amount of observed data through sparse search or greedy algorithms in the assumption that the unknown signal is sufficiently sparse on a specific basis. However, there often occurs loss of precision even failure in the process of reconstruction without enough prior information. Therefore, the purpose of this article is to solve the problem of low reconstruction accuracy by mining implicit structural information in the network. Specifically, we propose a novel and efficient algorithm (MCM_TRA) for reconstructing the structure of the *K*-forked tree network. Based on evolutionary game dynamics, the modified clustering method (MCM) classifies all nodes into two sets, then a two-stage reconstruction algorithm (TRA) is illustrated to recover the node signals in different sets. The experimental results demonstrate that the MCM_TRA enhances the reconstruction accuracy prominently than previous algorithms. Moreover, extensive sensitivity analysis shows that the reconstruction effect can be promoted for a broad range of parameters, which further indicates the superiority of the proposed method.**

***Index Terms*—Compressive sensing, evolutionary game, network reconstruction, tree network.**

I. Introduction

COMPLEX networks have long been a question of great interest in a wide range of fields [1]. Since considerable complex systems [2] can be modeled into complex networks for analysis, the topology [3], evolution [4], propagation [5], and control [6] of the network have attracted increasing attention. Among them, the topology of a network is a crucial property, as the structure will affect its function. For instance, the structure of the power network [7] influences power transmission stability, also the structure of the traffic network [8] has an impact on transportation efficiency. Nonetheless, the existing body of research suggests that the structure of complex networks, such as biological networks, sensor networks, and social networks [9]–[11], is hard to observe in practical applications. Thus, reconstructing the network structure is a significant but challenging task [12]–[14]. Here, we mention a kind of network with apparent structural information, called *K*-forked tree network, where the nodes within each layer are connected with K subtrees. The structure of the *K*-forked tree network is widely used to form the information flow in biological networks [15], summarize and visualize the relationship between samples [16], assess the operation security of power system [17], etc. Such a significant network is worth exploring and reconstructing; however, there has been little consideration about the combination of latent structural information and reconstruction methods up to now.

Recent years have witnessed great progress in the exploration of network reconstruction, where the reconstruction methods are mainly divided into two categories: 1) one is the model-based method [18] and 2) the other is the data-based method [19], where the data-based reconstruction method is challenging for several reasons. On the one hand, the network reconstruction problem is essentially an underdetermined problem as the observed data are frequently less [20]. On the other hand, a large number of nodes in a complex network may be connected by some hidden contents, which is difficult to observe from the measurement data [21].

With the arrival of the big data era, compressive sensing, a data-based method has been of wide concern and achieved great results [22], [23]. It is generally assumed that the prior information of the unknown signal is sparse enough on a specific basis [24]. Nevertheless, the potential structure information in the network is not well utilized, resulting in poor reconstruction accuracy. In fact, making full use of the prior information of the network structure to reconstruct is effective. Baraniuk *et al.* [18] proposed a model-based compressive sensing framework, adding structural information to the recovery algorithm. Khajehnejad *et al.* [25] established the weighted l_1 minimization for sparse recovery with prior information, which successfully enhanced the reconstruction accuracy. Yu *et al.* [26] reconstructed sparse signals using a hierarchical Bayesian model based on the sparse prior and cluster prior. Recently, Huang *et al.* [27] excavated the symmetry characteristics of the network structure and integrated this constraint into the problem of network structure reconstruction, which notably improved the reconstruction accuracy.

Manuscript received 29 November 2019; revised 23 May 2020 and 22 August 2020; accepted 30 November 2020. Date of publication 31 December 2020; date of current version 1 July 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62073340 and Grant 61927804; in part by the Innovation-Driven Plan in Central South University under Grant 2019CX020; and in part by the Natural Science Foundation of Hunan Province under Grant 2019JJ50777. *(Xiaoping Zheng and Wenhan Wu contributed equally to this work.)* This article was recommended by Associate Editor W. X. Zheng. *(Corresponding author: Keke Huang.)*

Xiaoping Zheng and Wenhan Wu are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: asean@mail.tsinghua.edu.cn; wwh19@mails.tsinghua.edu.cn).

Wenfeng Deng, Chunhua Yang, and Keke Huang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: wfdeng@csu.edu.cn; ychh@csu.edu.cn; huangkeke@csu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.3043227.

Digital Object Identifier 10.1109/TCYB.2020.3043227

Inspired by the aforementioned research, we propose a novel method (MCM_TRA) to reconstruct the K-forked tree network by incorporating the implicit structural information. Our main contributions are three-fold. First, given the prior information of the degree characteristics in a K-forked tree network, the modified clustering method (MCM) is proposed to classify all nodes. Second, a two-stage reconstruction algorithm (TRA) is employed to reconstruct the sparse signal vectors corresponding to the nodes in different sets, and sparse signal vectors are recombined and transformed into the reconstructed matrix. Last but not least, the sensitivity analysis of MCM_TRA is conducted to reveal the reconstruction effect can be promoted for a broad range of parameters, further indicating the superiority of our method. Compared with existing compressive sensing algorithms for reconstructing tree networks, such as orthogonal matching pursuit (OMP), compressive sampling matching pursuit (CoSaMP), weighted OMP (WOMP), and modified clustered OMP (MCOMP) [28]–[31], our proposed method substantially improves the reconstruction performance, and inspires the idea of mining potential structure information in similar networks.

The remainder of this article is organized as follows. In Section II, we review the compressive sensing theory and formulate the problem of network structure identification based on evolutionary game data. In Section III, our method (MCM_TRA) is proposed to reconstruct K-forked tree networks. Section IV provides corresponding numerical simulation experiments for MCM_TRA. Finally, conclusions and future research topics are described in Section V.

II. Preliminaries

A. Compressive Sensing Theory

Compressive sensing theory has proved that it is possible to obtain the original signal information from the measured data with the sampling rate much lower than that of Nyquist in the framework of sparse signal recovery. To be concrete, one can recover the raw signal $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T \in R^N$, which is a κ-sparse vector (i.e., the l_0-norm of the vector $\mathbf{x}$ satisfies $\|\mathbf{x}\|_0 \leq \kappa$ and $\kappa \ll N$) with a low-dimensional measurement vector $\mathbf{y} = [y_1, y_2, \ldots, y_M]^T \in R^M$ by the equation as follows:

$$\mathbf{y} = \Phi\mathbf{x} \tag{1}$$

where $\Phi \in R^{M\times N}$ is a measurement matrix and $M \ll N$. However, even though the vector $\mathbf{y}$ and matrix Φ are known, it is still difficult to reconstruct the high-dimensional signal by solving (1) since it is an underdetermined equation with infinite solutions for $M \ll N$. Therefore, l_0-norm minimization is employed to resolve the sparse signal recovery problem as follows:

$$\begin{aligned} &\min \|\mathbf{x}\|_0 \\ &\text{s.t.} \quad \mathbf{y} = \Phi\mathbf{x}. \end{aligned} \tag{2}$$

In fact, the l_0-norm minimization problem aims to calculate the number of nonzero elements in the vector. As the search space is excessively enormous, the method is an NP-hard problem. Hence, the restricted isometry property (RIP) [32] is introduced

$$(1-\delta_\kappa)\|\mathbf{x}\|_2^2 \leq \|\Phi\mathbf{x}\|_2^2 \leq (1+\delta_\kappa)\|\mathbf{x}\|_2^2 \tag{3}$$

where $\delta_\kappa (0 \leq \delta_\kappa \leq 1)$ is a constant related to sparsity κ. Equation (3) ensures the matrix Φ satisfies the κ-RIP condition, which guarantees all submatrixes are nearly equidistant. Hence, the l_0-norm minimization can be transformed into the l_1-norm minimization with convex relaxation as follows:

$$\begin{aligned} &\min \|\mathbf{x}\|_1 \\ &\text{s.t.} \quad \mathbf{y} = \Phi\mathbf{x}. \end{aligned} \tag{4}$$

So far, there have been numerous algorithms for the solution of (4), among which the most commonly used is the l_1-norm minimization algorithm for sparse matrix equation [33], such as the matching pursuit (MP) [34], and the following extended OMP, CoSaMP, StOMP [35], etc.

B. Compressive Sensing Method for Network Reconstruction Based on Evolutionary Game Data

The evolutionary game model is a typical model reflecting the interaction of agents in complex systems. In a network with known structure, agents tend to adopt different strategies to play games for gaining the maximum benefit under certain interaction modes. These agent-to-agent interactions, which can be used to generate time-series data, are dominated by the dynamics of evolutionary games.

In the networked game dynamics, each node is regarded as an agent, which will play a pair game with each neighbor in the network structure. Specifically, an agent can select one of two strategies: 1) cooperation (C) or 2) defection (D), which can be expressed in mathematical form as $\mathbf{s}(C) = [1, 0]^T$ and $\mathbf{s}(D) = [0, 1]^T$. The payoff acquired by an agent in the process of game is determined by the payoff matrix $\mathbf{P}$, which can be defined as follows:

$$\mathbf{P} = \begin{bmatrix} R & S \\ T & P \end{bmatrix}. \tag{5}$$

Here, R represents the reward for mutual cooperation, recorded as (C, C). Conversely, P denotes the punishment for mutual defection, recorded as (D, D). If one chooses to cooperate and the other chooses to betray, recorded as (C, D) or (D, C), the cooperator will obtain the sucker's payoff S, while the defector will gain the temptation to defect T.

Prisoner's dilemma game (PDG) [36] is a representative example of game theory, explaining the cooperation and competition among individuals in society. We adopt it as our game model, which satisfies the following conditions: $T > R > P > S$, and $2R > T + S$. Thus, the realistic payoff matrix is given by

$$\mathbf{P}_{\text{PDG}} = \begin{bmatrix} 1 & -0.04 \\ b & 0.15 \end{bmatrix} \tag{6}$$

where $1 < b < 2$, tallying with the condition, we hold $b = 1.2$ in this article.

Next, we generate the relevant data of the evolutionary game. With regard to agent i, $\mathbf{s}_i(t)$ represents its strategy at time step t. The payoff of agent i is $s_i^T(t)Ps_j(t)$, which is

obtained from playing with agent j. At time step t, all agents play games with their neighbors in a specific network, the payoff of each agent i reads

$$u_i(t) = \sum_{j \in \Gamma_i} \mathbf{s}_i^T(t) \mathbf{P} \mathbf{s}_j(t) \tag{7}$$

in which Γ_i represents all neighbors connected to agent i. After each round of the game, the agent updates its strategy in the light of the profits of its neighbors. The difference in payoffs between agent i and j can be described by: $\Delta u(t) = u_i(t) - u_j(t)$. If $\Delta u(t) \geq 0$, agent i will maintain its strategy in the next time step; otherwise, agent i will adopt the strategy of agent j at time step $(t+1)$ based on the probability calculated by the Fermi rule [37]

$$p\big(s_i(t+1) \leftarrow s_j(t)\big) = \frac{|\Delta u(t)|}{\langle k \rangle b} \tag{8}$$

where $\langle k \rangle$ is the maximum degree between agent i and j, and b denotes the largest payoff difference, holding $b = T - S$ in the prisoner's dilemma. After the above process, the time-series data of payoff and strategy can be obtained for the network reconstruction process.

Assume the number of agents in the network is N, and the adjacency matrix $\mathbf{X} \in R^{N \times N}$ corresponding to the network structure is given by

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{N1} \\ x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1N} & x_{2N} & \cdots & x_{NN} \end{bmatrix}. \tag{9}$$

If agents i and j are interconnected, $x_{ij} = x_{ji} = 1$; otherwise, $x_{ij} = x_{ji} = 0$. The relation between payoff and strategy of agent i at time step t can be deduced as follows:

$$U_i(t) = \sum_{j=1}^{N} x_{ij} F_{ij}(t) \tag{10}$$

where $F_{ij}(t) = \mathbf{s}_i^T(t) \mathbf{P} \mathbf{s}_j(t)$ is the visual payoff, which depends on the strategies of agent i and j at time step t, transforming into a valid term in case $x_{ij} = 1$. Next, we obtain the payoff vector $\mathbf{y}_i = [U_i(t_1), U_i(t_2), \ldots, U_i(t_M)]^T \in R^M$ by using the measurement data from the time step t_1 to t_M. For agent i, the ith column of $\mathbf{X}$ contains its structure information, written as $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]^T \in R^N$. Here, the strategy matrix of agent i is defined as $\mathbf{A}_i \in R^{M \times N}$, then the following equation can be obtained:

$$\mathbf{y}_i = \mathbf{A}_i \mathbf{x}_i \tag{11}$$

where

$$\mathbf{y}_i = [U_i(t_1), U_i(t_2), \ldots, U_i(t_M)]^T \tag{12}$$

$$\mathbf{A}_i = \begin{bmatrix} F_{i1}(t_1) & F_{i2}(t_1) & \cdots & F_{iN}(t_1) \\ F_{i1}(t_2) & F_{i2}(t_2) & \cdots & F_{iN}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ F_{i1}(t_M) & F_{i2}(t_M) & \cdots & F_{iN}(t_M) \end{bmatrix} \tag{13}$$

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]^T. \tag{14}$$

Since the network consists of N nodes, the equations of all nodes can be stacked and combined as follows:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & \mathbf{A}_2 & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & \mathbf{A}_N \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} \tag{15}$$

where O stands for zero matrix. For simplicity, (15) is rewritten in the form as

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{16}$$

where

$$\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T \tag{17}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & \mathbf{A}_2 & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & \mathbf{A}_N \end{bmatrix} \tag{18}$$

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T = \mathrm{vec}(\mathbf{X}). \tag{19}$$

In view of the derivation and discussion of the aforementioned equations, the problem of reconstructing $\mathbf{X}$ by columns has been successfully transformed into recovering vector $\mathbf{x}$ directly. Since the vector $\mathbf{x} = \mathrm{vec}(\mathbf{X})$ is sparse for most of the networks, the compressive sensing method based on evolutionary game data becomes a general framework for network reconstruction

$$\begin{aligned} &\hat{\mathbf{x}} = \arg\min \|\mathbf{x}\|_1 \\ &\text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \end{aligned} \tag{20}$$

where $\hat{\mathbf{x}}$ is the reconstructed vector of $\mathbf{x}$. Finally, we transformed the vector $\hat{\mathbf{x}}$ into the matrix $\hat{\mathbf{X}}$ by using the matrixing method, which represents the reconstructed network structure. The matrixing method can be expressed by

$$\hat{\mathbf{X}} = \mathrm{unvec}_{N,N}(\hat{\mathbf{x}}) = \begin{bmatrix} \hat{x}_{11} & \hat{x}_{21} & \cdots & \hat{x}_{N1} \\ \hat{x}_{12} & \hat{x}_{22} & \cdots & \hat{x}_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{1N} & \hat{x}_{2N} & \cdots & \hat{x}_{NN} \end{bmatrix}. \tag{21}$$

Under the general framework, network structure reconstruction is merely driven by the measurement data, corresponding to the strategies and payoffs of agents at different times. However, many networks possess specific structural characteristics in reality. For instance, assume the network with 40 nodes, Fig. 1(a) and (b) shows the visual adjacency matrix of two specific network structures, where the nonzero elements gather near the diagonal and pile up into blocks, respectively. Traditional compressive sensing methods based on evolutionary game data are incapable of utilizing the structure information of these networks, resulting in poor reconstruction accuracy. Therefore, it is significant to adequately mine the potential structural characteristics and design the reconstruction algorithm for these networks with a specific structure.

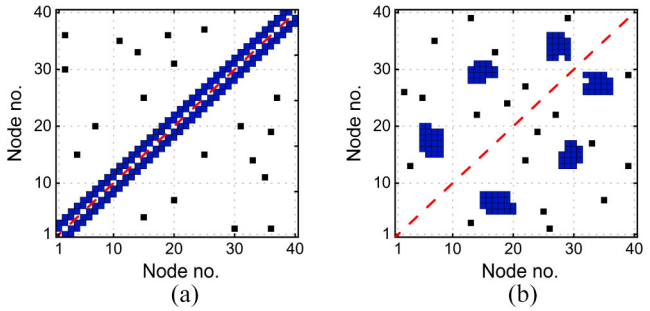

Fig. 1. Visual adjacency matrix of networks with specific structure. (a) Adjacency matrix where nonzero elements gather near the diagonal. (b) Adjacency matrix where nonzero elements are stacked into block. The nonzero elements reflecting special structures are marked in blue, while others are marked in black.

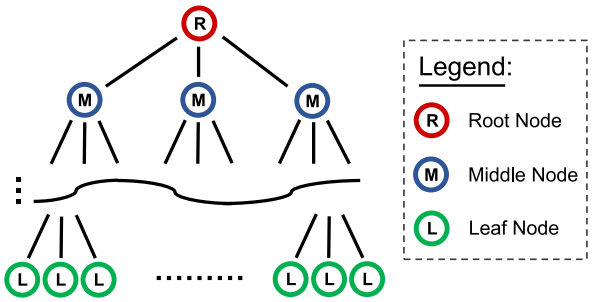

Fig. 2. Structure of K-forked tree network. The red, blue, and green circles represent the root node, middle nodes, and leaf nodes, respectively.

III. Proposed Approach

A. Structure Characteristics of K-Forked Tree Network

Tree networks, generally connected as the open-loop structure, are common in reality. The development of tree networks has spawned many research fields, such as the tree search algorithm in data structure [38], the decision tree in machine learning [39], the gene inference in biological sciences [40], etc. The K-forked tree network is a typical tree network with special structure, as shown in Fig. 2, each node of which is provided with K child nodes. In order to number all nodes of the K-forked tree network, assume that the depth of K-forked tree is L, the sum of nodes can be calculated according to the following formula:

$$N_s = \frac{K^L - 1}{K - 1} \tag{22}$$

where $K \geq 2$ and K is an integer. When numbering the nodes in a K-forked tree network, the root node number is defined as 1, increasing from top to bottom and left to right. The way of numbering, widely applied in power system networks and social networks, is regarded as a paradigm.

The structure of the K-forked tree network presents obvious structural characteristics and there are merely three degree values for all nodes, in which root node, middle node, and leaf node respectively, correspond to K, $K+1$, and 1. The degree value reveals latent structural features between the node and its neighbors. From a mathematical perspective, the information reflected by these features may appear in the adjacency matrix. However, present reconstruction algorithms barely consider the characteristics of degree as prior information.

In this case, the TRA based on the MCM (MCM_TRA) is proposed to reconstruct K-forked tree networks and improve the accuracy. Fig. 3 illustrates the overall diagram of the K-forked tree network reconstruction method. The details of this approach are described in the following section.

B. Modified Clustering Method

In the previous section, the latent structural information of the K-forked tree network is mainly reflected in the degree values. Hence, the MCM is introduced as a classifier to distinguish the nodes with different degree values. Before conducting the classification algorithm, feature selection is a critical step.

Because the measurement data of the K-forked tree network are included in payoff vector $\mathbf{y}$, how to mine valid features deserves in-depth consideration. For intuitive analysis, the payoff vector $\mathbf{y}$ is converted to the matrix $\mathbf{Y} \in R^{M\times N}$ by

$$\mathbf{Y} = \text{unvec}_{M,N}(\mathbf{y}) = \begin{bmatrix} U_1(t_1) & \cdots & U_N(t_1) \\ \vdots & \ddots & \vdots \\ U_1(t_M) & \cdots & U_N(t_M) \end{bmatrix} \tag{23}$$

where the ith column of $\mathbf{Y}$ represents the measurement data of node i from the time step t_1 to t_M.

Fig. 4 describes the analysis process of feature selection. As shown in Fig. 4(a), the 5-forked tree network with 31 nodes is taken as an example, where nodes 1–6 in the shaded area represent big degree nodes and nodes 7–31 are small degree nodes. On the one hand, a big degree node owns more opportunities to play games with its neighbors, generally resulting in higher payoff. That is to say, the mean payoff of each node can be regarded as a feature. Nevertheless, in some special cases, the node is a cooperator while most of its neighbors are defectors. It violates the widespread phenomenon that a node with bigger degree gains more, which is illustrated in Fig. 4(b) that nodes in the red box are potential misclassified nodes. On the other hand, the standard deviation of each node's payoff is considered as another feature since the payoff of a big degree node might significantly vary in different moments. However, there exists a certain probability that the standard deviation corresponds to a small degree node is large, as is shown in Fig. 4(c), also increasing potential classification errors.

To complement the shortcomings of each feature, the mean and standard deviation of payoff are considered together to promote the accuracy of classification. Specifically, the mean payoff of node i within the sampling time is given by

$$\mu_i = \frac{1}{M}\sum_{k=1}^{M} U_i(t_k). \tag{24}$$

Analogously, the standard deviation of payoff corresponding to node i reads

$$\sigma_i = \sqrt{\frac{1}{M}\sum_{k=1}^{M}[U_i(t_k) - \mu_i]^2}. \tag{25}$$

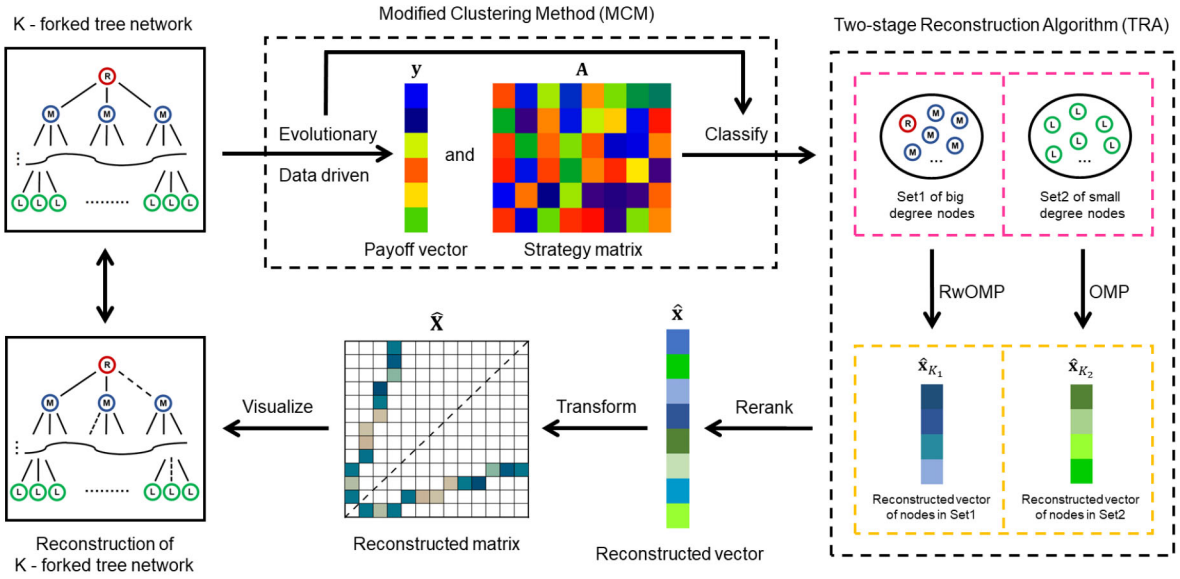

Fig. 3. Overall diagram of the K-forked tree network reconstruction method. For a K-forked tree network, the evolutionary game is used to drive the data to obtain the payoff vector and strategy matrix. Next, the MCM is adopted to classify all nodes into Set1 and Set2. Then, the TRA is proposed to, respectively, reconstruct the vectors of nodes in two sets, which are recombined into the reconstructed vector $\hat{\mathbf{x}}$. Finally, we transform the vector $\hat{\mathbf{x}}$ into the matrix $\hat{\mathbf{X}}$ to realize the reconstruction of the K-forked tree network.

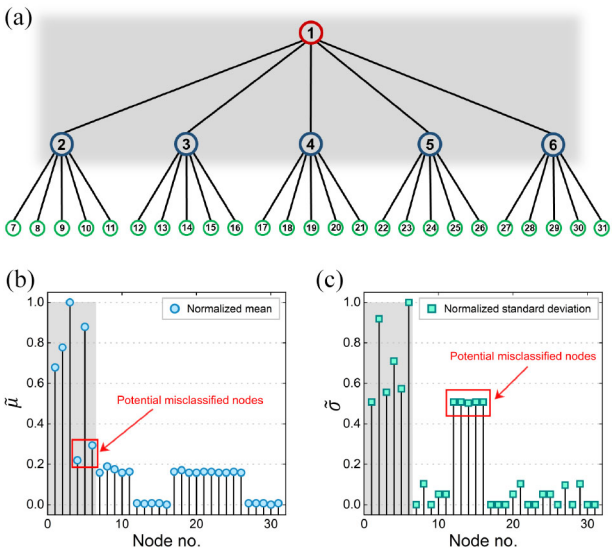

Fig. 4. Analysis process of feature selection. (a) Suppose a 5-forked tree network with 31 nodes, nodes 1-6 in shaded area represent big degree nodes and nodes 7-31 are small degree nodes. (b) Mean payoff of each node. (c) Standard deviation of each node's payoff. The red box marks potential misclassified nodes in (b) and (c), respectively.

We map the above data to the interval $[0, 1]$ using the normalization method. Thus, the normalized mean vector and standard deviation vector are, respectively, expressed by $\tilde{\mu} = [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_N]^T$ and $\tilde{\sigma} = [\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_N]^T$, which can be used as the input of classification algorithm. In the output, based on feature vectors, nodes are separated into two sets by the classification algorithm. The index of big degree nodes in Set1 is defined as $\Omega_1 \subset \{1, 2, \ldots, N\}$, and the index of small degree nodes in Set2 is $\Omega_2 \subset \{1, 2, \ldots, N\} \backslash \Omega_2$.

Next, selecting an appropriate classification algorithm is critical since it directly determines whether nodes can be divided into the correct set. As one of the unsupervised learning methods, clustering is widely used in designing classifiers. Here, K-means (partition based), BIRCH (hierarchy based), DBSCAN (density based), and GMM (model based) algorithms [41]–[44] are applied to classify these nodes. We use several indicators to evaluate the classification effect: the error rate represents the probability of misclassification, higher Rand index (external index) [45] corresponds to higher similarity between predicted and true values, while lower Davies–Bouldin index (internal index) [46] indicates the algorithm has higher intracluster cohesion. Moreover, if the average silhouette [47] is closer to 1, it means the clustering configuration is more appropriate. Based on the above indicators, the performance comparison of clustering algorithms is shown in Table I. It is obvious the error rates of K-means and DBSCAN are approximate and lower than other algorithms. However, the Rand index, Davies–Bouldin index, and average silhouette of k-means are better than DBSCAN. Therefore, K-means can be regarded as a more suitable classification algorithm for K-forked tree networks.

TABLE I
COMPARISON OF CLUSTERING ALGORITHM

Alogorithm	Error Rate	Rand Index	Davies–Bouldin Index	Average Silhouette
K-means	**0.050**	**0.922**	**0.519**	**0.870**
BIRCH	0.088	0.841	0.564	0.852
DBSCAN	0.052	0.919	0.631	0.806
GMM	0.121	0.902	0.574	0.849

Although K-means performs better than other clustering algorithms in the above analysis, there still exists a 5% error rate in classification results. Thus, an idea appears that the results can be modified by excavating latent information in the payoff matrix. Because a small degree node is merely connected with its parent node, the payoff in each round is limited

to four values corresponding to the game matrix. Assume that node $i \in \Omega_2$, for $\forall t_k$, if $U_i(t_k) \notin \{R, T, S, P\}$ exists, node i can be regarded as a big degree node and divided into Set1. In contrast, a big degree node plays game with multiple nodes, the cases of its payoff will exceed this range. Thus, with regard to node $i \in \Omega_1$, if $U_i(t_k) \in \{R, T, S, P\}$ is satisfied for all t_k, we transfer it into set2. After this modification process, the results are quite consistent with the actual situation.

The modification of clustering results leads to the change of elements in the index sets. Hence, Ω_1 and Ω_2, which include wrong classification indexes, are replaced by the correct index sets K_1 and K_2. Assume the amount of elements are $|K_1| = N_1$ and $|K_2| = N_2 = N - N_1$. Then, according to the index sets K_1 and K_2, the payoff vector $\mathbf{y}$ is reorganized by $\mathbf{y}_{K_1} = [\mathbf{y}_{K_1(1)}, \mathbf{y}_{K_1(2)}, \ldots, \mathbf{y}_{K_1(N_1)}]^T$ and $\mathbf{y}_{K_2} = [\mathbf{y}_{K_2(1)}, \mathbf{y}_{K_2(2)}, \ldots, \mathbf{y}_{K_2(N_2)}]^T$. Likewise, the strategy matrix $\mathbf{A}$ is reorganized into the following form:

$$\mathbf{A}_{K_1} = \begin{bmatrix} \mathbf{A}_{K_1(1)} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{K_1(2)} & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{A}_{K_1(N_1)} \end{bmatrix} \tag{26}$$

and

$$\mathbf{A}_{K_2} = \begin{bmatrix} \mathbf{A}_{K_2(1)} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{K_2(2)} & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{A}_{K_2(N_2)} \end{bmatrix}. \tag{27}$$

After adjusting the sequence of nodes according to the classification results, (15) is rewritten by

$$\begin{bmatrix} \mathbf{y}_{K_1} \\ \mathbf{y}_{K_2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{K_1} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{K_2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{K_1} \\ \mathbf{x}_{K_2} \end{bmatrix} \tag{28}$$

where the upper part and the lower part, respectively, correspond to Set1 and Set2. Consequently, an idea occurs that node signals in different sets can be reconstructed by

$$\begin{aligned} &\hat{\mathbf{x}}_{K_p} = \arg\min \left\| \mathbf{x}_{K_p} \right\|_1 \\ &\text{s.t. } \mathbf{y}_{K_p} = \mathbf{A}_{K_p} \mathbf{x}_{K_p} \end{aligned} \tag{29}$$

where $p = 1, 2$. In this case, the vectors $\hat{\mathbf{x}}_{K1}$ and $\hat{\mathbf{x}}_{K2}$ can be recombined into the reconstructed vector $\hat{\mathbf{x}}$ according to the index sets K_1 and K_2.

In the follow-up phase of the method, estimating the degree values of nodes in different sets is significant, because it may be used in the reconstruction algorithm. On the one hand, the degree value of a node in Set2 can be easily inferred as a small degree node is only linked to one neighbor. It means the degree value of each node in Set2 is $\langle k \rangle_2 = 1$. On the other hand, the situation is complicated for nodes in Set1. For simplicity, the degree value of each node in Set1 is uniformly expressed by $\langle k \rangle_1 = K + 1$, including the root node. Next, to estimate the specific degree value $\langle k \rangle_1$, an adaptive method is illustrated as follows.

For a node with multiple neighbors, the maximum payoff corresponds to an ideal case where it is a defector and all neighbors are cooperators. In this case, the ratio of $T_{\max} = \max\{\mathbf{y}_{K_1}\}$ to T accurately represents the number of neighbors, which is the definition of degree value. Thus, the degree value of each node in Set1 can be expressed by

$$\widetilde{\langle k \rangle}_1 = \frac{T_{\max}}{T}. \tag{30}$$

However, the probability of this ideal situation is minor when K is large. Normally, the maximum payoff $T_{\max}$ is obtained from the circumstances where the node is a defector and most of its neighbors are cooperators, rather than all of them. Consequently, the degree $\widetilde{\langle k \rangle}_1$ calculated by (30) is lower than $\langle k \rangle_1$ in actuality.

To improve the accuracy of estimated degree value, some laws are revealed by comparing it with the real situation. The degree $\widetilde{\langle k \rangle}_1$ is calibrated by the following formula:

$$\langle k \rangle_1 = \begin{cases} \left\lceil \widetilde{\langle k \rangle}_1 \right\rceil, & 2 < \widetilde{\langle k \rangle}_1 \le 3 \\ \left[\alpha \widetilde{\langle k \rangle}_1 + 0.5 \right], & \widetilde{\langle k \rangle}_1 > 3 \end{cases} \tag{31}$$

where α is a calibration coefficient. In Fig. 5, we explore the estimated degree $\langle k \rangle_1$ as a function of width K to find an appropriate calibration coefficient α, the points on baseline denote correct degree values. Obviously, the curve corresponding to $\alpha = 1.2$ is quite consistent with the baseline. Thus, we hold it in this article to estimate $\langle k \rangle_1$ more accurate.

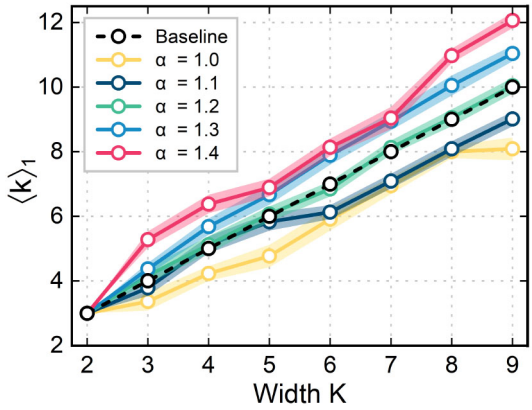

Fig. 5. Estimated degree $\langle k \rangle_1$ as a function of width K on different calibration coefficients α. The shape points and error bands represent the mean and SD based on 100 trials.

C. Two-Stage Reconstruction Algorithm

1) Stage 1 of TRA: In Set1, each big degree node is connected to its unique parent node in the upper layer and consecutive K child nodes in the lower layer. Naturally, the sequence numbers of neighbors of each big degree node can be deduced as the prior information. Suppose the cth element in K_1 corresponds to node i in Set1, which means $i = K_1(c)$. The sequence numbers of its neighbors are given by

$$i_q = \begin{cases} (i + K - 2)/K, & q = 1 \\ K(i-1) + q, & q = 2, \ldots, K+1. \end{cases} \tag{32}$$

Here, $q = 1$ and $q = 2, \ldots, K+1$, respectively, correspond to its unique parent node and K child nodes.

The sequence numbers of neighbors are exactly the positions of nonzero elements in the adjacency matrix X. Hence,

Algorithm 1 Stage 1 of TRA

Input:

The strategy matrix $\mathbf{A}_{K_1}$ and the payoff vector $\mathbf{y}_{K_1}$

The number of iterations $\langle k\rangle_1 N_1$

Output:

The reconstructed vector $\hat{\mathbf{x}}_{K_1}$

Initialize: $\mathbf{r}^0 = \mathbf{y}_{K_1}, \hat{\mathbf{x}}^0_{K_1} = 0, \mathbf{SUP}^0 = \emptyset$
for $i \leftarrow 1, \cdots, \langle k\rangle_1 N_1$ **do**
  Step 1: Get the reweighted matrix: $\mathbf{W}_{K_1}$
  Step 2: Match:
    $\mathbf{H}^i = \mathbf{A}^T_{K_1}\mathbf{r}^{i-1}$
  Step 3: Identify support indicator:
    $\sup^i = \left\{\arg\max_j \left|\mathbf{H}^i(j) \times w^j_{K_1}\right|\right\}$
  Step 4: Update the support:
    $\mathbf{SUP}^i = \mathbf{SUP}^{i-1} + \sup^i$
  Step 5: Update signal estimate:
    $\hat{\mathbf{x}}^i_{K_1} = \arg\min_{z:\mathrm{supp}(z)\subseteq \mathbf{SUP}^i} \left\|\mathbf{y}_{K_1} - \mathbf{A}_{K_1} z\right\|_2$
    $\mathbf{r}^i = \mathbf{y}_{K_1} - \mathbf{A}_{K_1}\hat{\mathbf{x}}^i_{K_1}$
end for
Return: $\hat{\mathbf{x}}_{K_1} \leftarrow \hat{\mathbf{x}}^i_{K_1}$

Algorithm 2 Stage 2 of TRA

Input:

The strategy matrix $\mathbf{A}_{K_2}$ and the payoff vector $\mathbf{y}_{K_2}$

The number of iterations N_2

Output:

The reconstructed vector $\hat{\mathbf{x}}_{K_2}$

Initialize: $\mathbf{r}^0 = \mathbf{y}_{K_2}, \hat{\mathbf{x}}^0_{K_2} = 0, \mathbf{SUP}^0 = \emptyset$
for $i \leftarrow 1, \cdots, N_2$ **do**
  Step 1: Match:
    $\mathbf{H}^i = \mathbf{A}^T_{K_2}\mathbf{r}^{i-1}$
  Step 2: Identify support indicator:
    $\sup^i = \left\{\arg\max_j \left|\mathbf{H}^i(j)\right|\right\}$
  Step 3: Update the support:
    $\mathbf{SUP}^i = \mathbf{SUP}^{i-1} + \sup^i$
  Step 4: Update signal estimate:
    $\hat{\mathbf{x}}^i_{K_2} = \arg\min_{z:\mathrm{supp}(z)\subseteq \mathbf{SUP}^i} \left\|\mathbf{y}_{K_2} - \mathbf{A}_{K_2} z\right\|_2$
    $\mathbf{r}^i = \mathbf{y}_{K_2} - \mathbf{A}_{K_2}\hat{\mathbf{x}}^i_{K_2}$
end for
Return: $\hat{\mathbf{x}}_{K_2} \leftarrow \hat{\mathbf{x}}^i_{K_2}$

for the purpose of making these nonzero elements more distinct identified, the weight constraint is considered to be added in prior knowledge. Here, reweighted OMP (RwOMP) algorithm inspired by [48] is proposed to reconstruct the signal vector in Set1, which may boost the reconstruction performance by putting the weight into the identify support indicator in step 2 of Algorithm 1. The criterion of selecting optimal term is related to the weight values, resulting in the positions of elements with large weight values are easier to be picked. Considering the initialization, for node i, the weights assigned to the locations of its neighbors can be expressed by

$$w^{i_q}_{K_1(c)} = \hat{w} \tag{33}$$

where $q = 1, 2, \ldots, K+1$ and $\hat{w} > 1$, while the weights of other positions are assigned as 1. Thus, the reweighted matrix of node i is organized by

$$\mathbf{W}_{K_1(c)} = \begin{bmatrix} w^1_{K_1(c)} & 0 & \cdots & 0 \\ 0 & w^2_{K_1(c)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w^N_{K_1(c)} \end{bmatrix}. \tag{34}$$

Similarly, for other nodes in Set1, the reweighted matrices can be obtained in the same way. Consequently, the reweighted matrix of each node in Set1 is stacked, and we obtain the reweighted matrix $\mathbf{W}_{K_1}$ corresponds to all nodes in Set1

$$\mathbf{W}_{K_1} = \begin{bmatrix} \mathbf{W}_{K_1(1)} & \mathrm{O} & \cdots & \mathrm{O} \\ \mathrm{O} & \mathbf{W}_{K_1(2)} & \cdots & \mathrm{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{O} & \mathrm{O} & \cdots & \mathbf{W}_{K_1(N_1)} \end{bmatrix}. \tag{35}$$

According to the above analysis, the latent sequence numbers of neighbors are incorporated into the framework of RwOMP as the prior information. Thus, the detailed steps about the stage 1 of TRA are summarized in Algorithm 1. The number of iterations is equal to the quantity of nonzero elements in vector $\mathbf{x}_{K_1}$, which is approximately $\langle k\rangle_1 N_1$. The sparse approximation vector $\hat{\mathbf{x}}_{K_1}$ will be recovered by the payoff vector $\mathbf{y}_{K_1}$ and the strategy matrix $\mathbf{A}_{K_1}$.

2) Stage 2 of TRA: In Set2, a small degree node is merely linked to its parent node. Thus, the true signal $\mathbf{x}_{K_2}$ is a N_2-sparse vector in that $|K_2| = N_2$. As there exists no obvious structure information, a recovery algorithm that is different from RwOMP in stage 1 with less complexity in the pursuit of shorter operation time is considered to be chosen. OMP is a greedy distribution least square method for fitting the sparse model, which guarantees the residual vector after each iteration is orthogonal to all previously selected column vectors. We take advantage of OMP to ensure the optimization of iteration and make the performance more robust.

The number of nonzero elements in vector $\mathbf{x}_{K_2}$ is exactly N_2, which is regarded as the number of iterations in OMP algorithm. The stage 2 of TRA is generalized in Algorithm 2, from which the sparse approximation vector $\hat{\mathbf{x}}_{K_2}$ will be restored in a similar way.

In summary, the TRA is adopted to recover the sparse approximation vectors $\hat{\mathbf{x}}_{K_1}$ and $\hat{\mathbf{x}}_{K_2}$, which can be reorganized into the reconstructed vector $\hat{\mathbf{x}}$. To further theoretically explain the improvement of TRA on reconstruction effect, the comparison between traditional algorithms (such as OMP or CoSaMP) and the TRA is shown in Fig. 6. Traditional algorithms reconstruct the network as a whole without using the potential structural information effectively. There may be fair a lot of errors between the reconstructed vector $\hat{\mathbf{x}}$ and the real signal vector $\mathbf{x}$. In contrast, the TRA makes full use of the prior information of nodes classified by the MCM, adopting appropriate methods to reconstruct the node signals. Therefore, the TRA may perform better than traditional algorithms in theory.

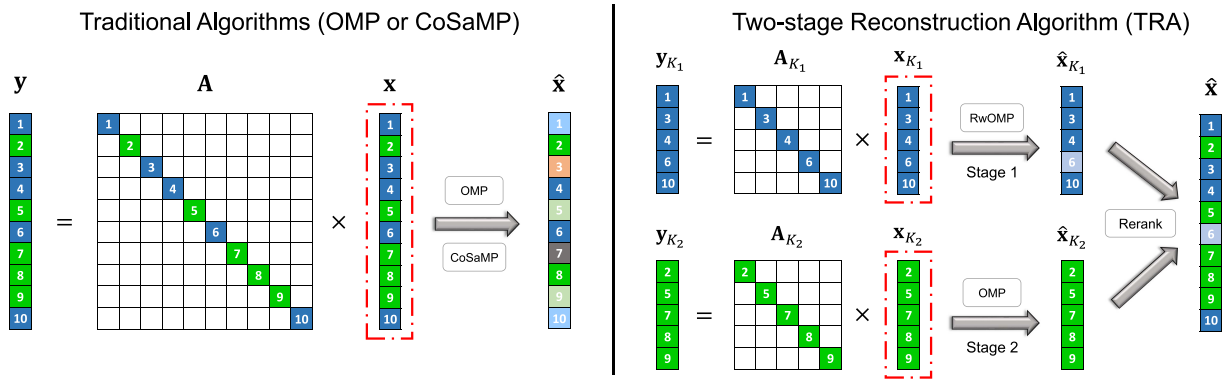

Fig. 6. Theoretical comparison of reconstruction effect between traditional algorithms (such as OMP or CoSaMP) and the TRA. Assume that ten nodes in the network: five nodes (1, 3, 4, 6, 10) with big degree, others (2, 5, 7, 8, 9) with small degree. The blue and green blocks correspond to big degree nodes and small degree nodes, respectively. The vectors in red dashed boxes represent completely accurate reconstruction results.

TABLE II
CONFUSION MATRIX OF THE BINARY CLASSIFICATION PROBLEM

		Actual Label	
		Target Class	Negative Class
Predicted Label	Target Class	TP	FP
	Negative Class	FN	TN

IV. NUMERICAL RESULTS

A. Reconstruction Evaluation

Before analyzing and evaluating the consistency between the reconstructed network and the original network, we introduce the notions of some evaluation indicators. For the binary classification, the confusion matrix is generally used to generate the evaluation criteria, where corresponding elements are included in Table II. Based on the confusion matrix, the reconstruction effect is quantified by the area under receiver operating characteristic curve (AUROC) and the area under precision–recall curve (AUPR).

The ROC curve illustrates the performance of a binary classification by presenting the tradeoff between the true positive rate (TPR) and the false positive rate (FPR). In our case, it can be explained as: given a nonzero element in the reconstructed matrix reflecting a latent edge in the network. If the latent edge corresponds to a real edge in the original network, the TPR increases. Otherwise, the FPR grows up. The score is typically between 0.5 (random level) and 1 (perfect reconstructed level). Therefore, the reconstruction performance can be evaluated by the AUROC in a nonparametric way.

Analogously, the PR curve indicates the relationship between the precision (P) and the recall (R), which has been regarded as an alternative to ROC curve for assignments with a large skew in the binary classification [49]. Here, the disparity between positive and negative samples cannot be effectively guaranteed. Thus, the AUPR is employed as another evaluation indicator to reveal the reconstruction effect with highly skewed samples. In the following experiments, we mainly adopt AUROC and AUPR to tradeoff the performance of network reconstruction for different algorithms.

B. Comparison of Different Reconstruction Algorithms

In this section, we focus on the performance of our algorithm MCM_TRA for the task of recovering K-forked tree networks. For comparison, we run two traditional algorithms for standard κ-sparse: 1) OMP and 2) CoSaMP. Moreover, two state-of-the-art reconstruction algorithms considering latent structural information, WOMP and MCOMP, are compared with our results. We set the weight $\hat{w} = 10$ in MCM_TRA, and default parameter settings are used in other algorithms.

For a standard reconstruction procedure, the time-series data in each round are recorded, consisting of the strategies and payoffs of nodes in the evolutionary game. The data ratio can be defined by $R_D = l/N_s$, where l is given as the length of data and N_s denotes the number of nodes in the network. Then, we discuss the form of K-forked tree networks used in numerical experiments. For a K-forked tree network, the depth L represents the number of layers, while the width K corresponds to the number of branches. Inspired by the depth-first search and width-first search of the branching tree in data structure, K-forked tree networks with different depths and widths are designed for conducting our experiments.

1) Reconstruction of K-Forked Tree Networks With Different Depths: First, K-forked tree networks with different depths $L = 3, 4, 5$ are established as the width $K = 3$ is fixed. Thus, the total number of nodes is $N_s = 13, 40, 121$ for specific width but different depths. Fig. 7 demonstrates that the reconstruction effect of K-forked tree networks with different depths. For each algorithm, without loss of generality, the reconstruction performance grows up with the increase of the data ratio R_D. Obviously, the effect of traditional algorithms is not well, since OMP and CoSaMP reconstruct the network as a whole involving no prior information, which is consistent with the previous theoretical analysis. The WOMP that incorporates weight information performs well in AUPR but poor in AUROC. The MCOMP, which identifies clustered elements more accurately, is opposite to WOMP. However, the higher AUROC and AUPR of MCM_TRA indicates that it is more accurate and efficient than other algorithms in reconstruction. Therefore, it preliminarily proves our algorithm successfully improves the reconstruction performance.

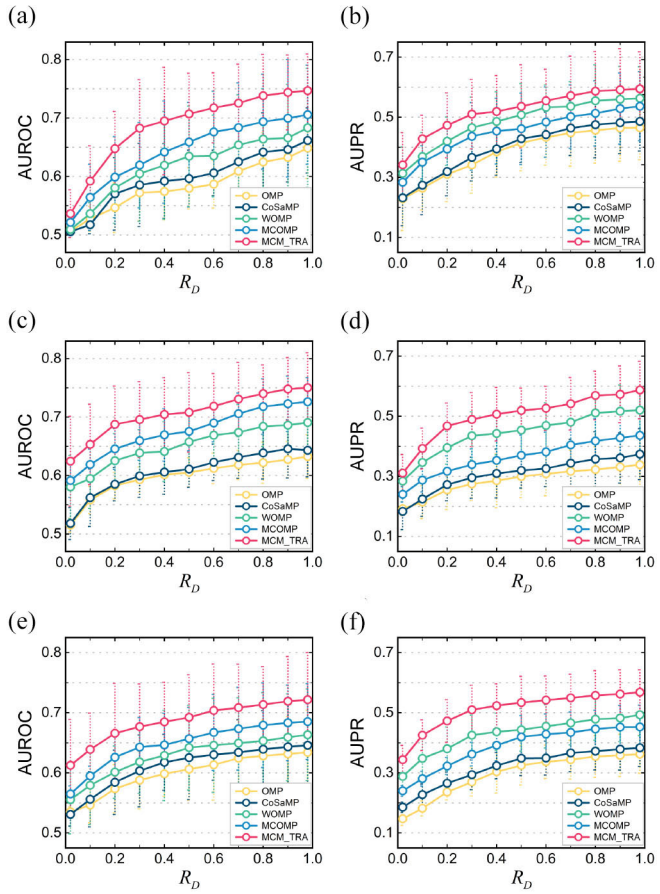

Fig. 7. Reconstruction effect of K-forked tree networks with different depths by OMP, CoSaMP, WOMP, MCOMP, and MCM_TRA. (a) and (b) $K = 3$ and $L = 3$. (c) and (d) $K = 3$ and $L = 4$. (e) and (f) $K = 3$ and $L = 5$. The shape points and error bars represent the mean and SD based on 100 trials, respectively. The left three are AUROC curves, and the right three are AUPR curves.

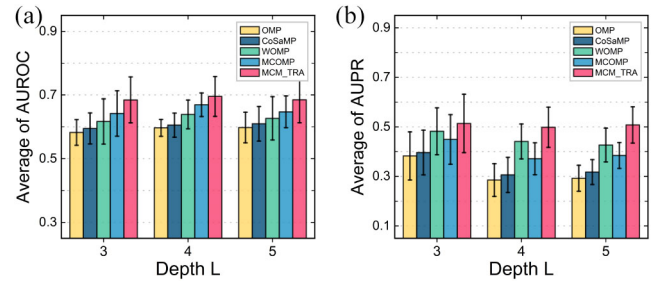

Fig. 8. Comparison of reconstruction performance of K-forked tree networks with different depths $L = 3, 4, 5$. (a) Average of AUROC. (b) Average of AUPR. Different colors correspond to different reconstruction algorithms. The error bars represent the SD, based on different data ratios R_D.

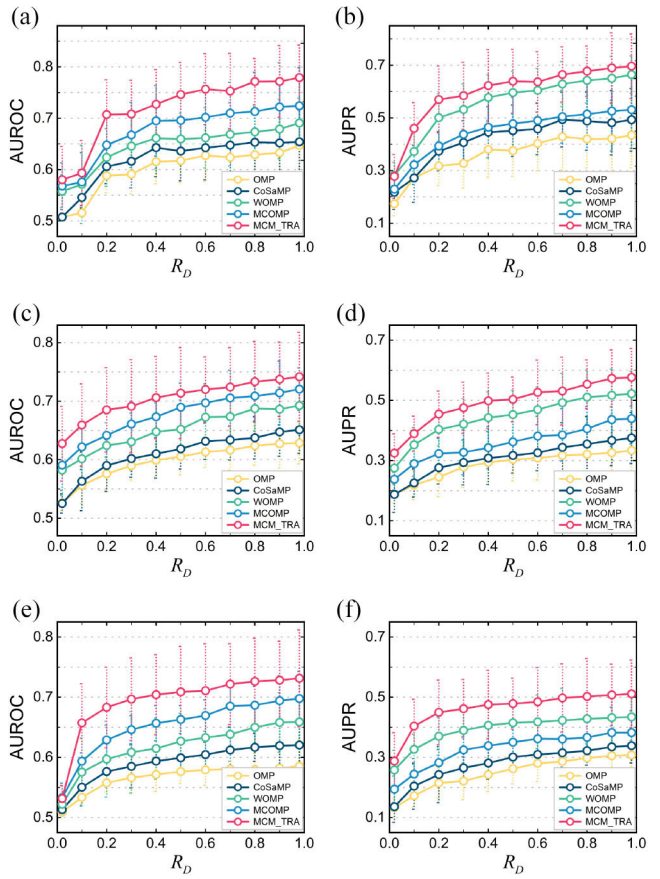

Fig. 9. Reconstruction effect of K-forked tree networks with different widths by OMP, CoSaMP, WOMP, MCOMP, and MCM_TRA. (a) and (b) $K = 2$ and $L = 4$. (c) and (d) $K = 3$ and $L = 4$. (e) and (f) $K = 4$ and $L = 4$. The shape points and error bars represent the mean and SD based on 100 trials, respectively. The left three are AUROC curves, and the right three are AUPR curves.

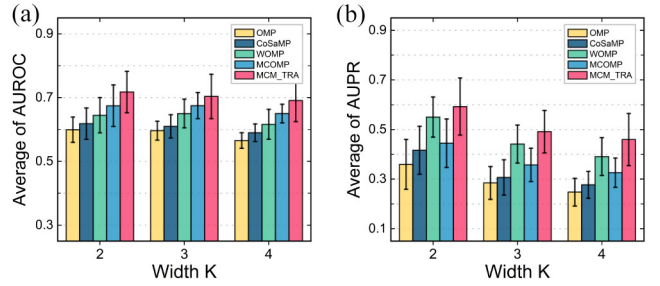

Fig. 10. Comparison of reconstruction performance of K-forked tree networks with different widths $K = 2, 3, 4$. (a) Average of AUROC. (b) Average of AUPR. Different colors correspond to different reconstruction algorithms. The error bars represent the SD, based on different data ratios R_D.

To further analyze the effect of depth on the reconstruction effect, from the perspective of statistics, the AUROC and AUPR under different data ratios R_D are averaged. As shown in Fig. 8, the average of AUROC seems to be hardly intervened by the depth. However, as the depth increases, the averages of AUPR of previous algorithms decrease, while the MCM_TRA still maintains at the higher level. This illustrates the depth has little influence on our algorithms. The underlying reason is that MCM_TRA makes full use of the degree characteristic, which remains constant when the depth changes. Hence, for deeper K-forked tree networks, our algorithm has a more significant reconstruction effect than other algorithms.

2) Reconstruction of K-Forked Tree Networks With Different Widths: The previous experiment shows that MCM_TRA exhibits reconstruction advantages for K-forked tree networks with different depths. Next, these algorithms are tested on K-forked tree networks with widths $K = 2, 3, 4$ when the depth $L = 4$. Accordingly, the total number of nodes is $N_s = 15, 40, 85$ for identical depth but different widths.

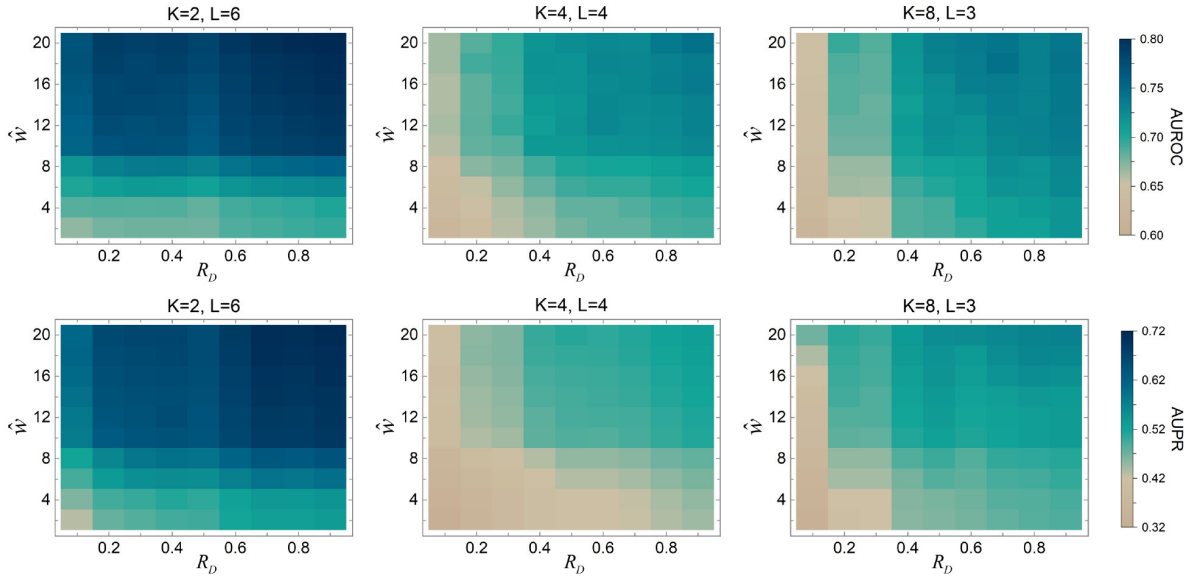

Fig. 11. Reconstruction evaluation indicators AUROC and AUPR as a function of the weight $\hat{w}$ and the data ratio R_D. The weight $\hat{w}$ increases from 2 to 20 and the data ratio R_D changes from 0.1 to 0.9.

Fig. 9 illustrates the reconstruction performance of K-forked tree networks with different widths. Similar to the results in the previous section, the MCM_TRA performs much better than other algorithms in terms of AUROC and AUPR. Moreover, we intuitively discover the AUROC and AUPR curves emerge a slightly reducing trend as the width increases, which may be related to the change of network structure.

Likewise, the effect of width on the reconstruction effect is shown in Fig. 10. With the extension of width, the reconstruction performance corresponding to each algorithm decreases, especially the average of AUPR decreases more prominently. This shows the width has a greater impact on the reconstruction effect. It should be emphasized that more branches of the K-forked tree network represent a more complex structure. That is to say, the neighbors of big degree nodes are hard to be identified as the width expands, resulting in poor reconstruction effect. Nevertheless, the MCM_TRA still shows the advantages over other algorithms, which is critical for the reconstruction of more complex K-forked tree networks.

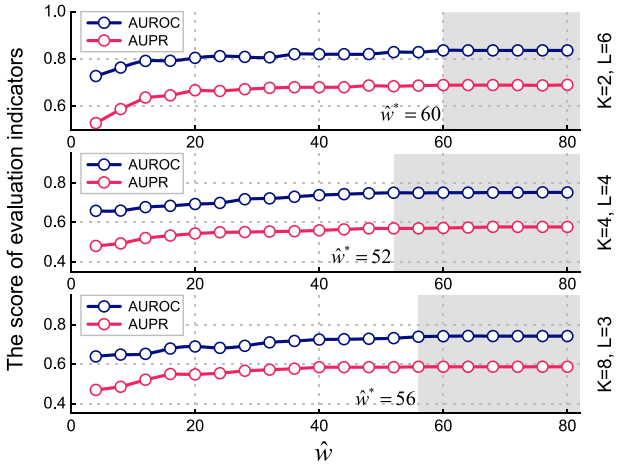

Fig. 12. Score of evaluation indicators as a function of the weight $\hat{w}$ for different K-forked tree networks. The blue and red lines represent the scores of AUROC and AUPR, respectively. The shaded areas mark the stable state intervals.

C. Parameter Sensitivity Analysis

The above experiments assume the weight $\hat{w} = 10$ is constant, however, its variation may have an influence on the reconstruction effect. Thus, it is necessary to conduct the parameter sensitivity analysis in this section.

Three kinds of K-forked tree networks with different widths and depths are designed: 1) 2-forked tree with 63 nodes ($K = 2$ and $L = 6$); 2) 4-forked tree with 85 nodes ($K = 4$ and $L = 4$); and 3) 8-forked tree with 73 nodes ($K = 8$ and $L = 3$). The reconstruction effect based on these networks reflects the influence of weight $\hat{w}$ more generally. Fig. 11 illustrates the reconstruction evaluation indicators AUROC and AUPR as a function of the weight $\hat{w}$ and data ratio R_D. In the horizontal direction, when $\hat{w}$ is fixed, both AUROC and AUPR grow up as R_D increases, which has been confirmed. In the vertical direction, it can be perceived the colors of blocks in these density maps change evidently with respect to $\hat{w}$ when R_D is fixed, which reflects the influence of weight on reconstruction performance.

To quantify the weight interval that further optimizes the reconstruction performance. Taking the data ratio is fixed at $R_D = 0.4$ as an example, the score of evaluation indicators as a function of the weight $\hat{w}$ is shown in Fig. 12. There seems to be a threshold $\hat{w}^*$ which determines the score of evaluation indicators. When the weight is below a threshold ($\hat{w} < \hat{w}^*$), the AUROC and AUPR enhance as $\hat{w}$ increases. Otherwise ($\hat{w} > \hat{w}^*$), the AUROC and AUPR tend to be in a stable state (shaded area).

In this case, the threshold is a crucial factor in weight setting, a feasible method to determine the value of $\hat{w}^*$ is given by

$$\hat{w}^* = \max\left\{\hat{w}_1^*, \hat{w}_2^*\right\} \tag{36}$$

where $\hat{w}_1^*$ and $\hat{w}_2^*$ represent the thresholds that stabilize the scores of AUROC and AUPR, respectively. Assume the scores of AUROC and AUPR are $\varphi(\hat{w})$ and $\gamma(\hat{w})$ that correspond to the weight $\hat{w}$, they are considered to be stable when the following conditions are satisfied:

$$\begin{cases} \left|\varphi(\hat{w}) - \varphi(\hat{w}_1^*)\right| \le \Delta\varphi(\hat{w}_1^*), & \hat{w} > \hat{w}_1^* \\ \left|\gamma(\hat{w}) - \gamma(\hat{w}_2^*)\right| \le \Delta\varphi(\hat{w}_2^*), & \hat{w} > \hat{w}_2^*. \end{cases} \tag{37}$$

Here, the fluctuation interval in a stable state is defined as $\Delta = 1\%$ by analogy with the stability theory in cybernetics.

By using this method, the values of $\hat{w}^*$ for these three networks are marked in Fig. 12. Note that the threshold obtained under different data ratios R_D could be heterogeneous. In summary, this method provides a reasonable explanation for the weight setting, which is beneficial to effectively promote the reconstruction performance.

V. Conclusion

This article presents the MCM_TRA to reconstruct K-forked tree networks from a new perspective. Different from traditional algorithms in compressive sensing, the proposed algorithm provides an explicit reconstruction framework and involves latent structural information of K-forked tree networks, which is of great significance for the reconstruction of networks with similar structural characteristics. By conducting numerical experiments and extensive experimental analysis, several prime conclusions are summarized as follows.

1) Given that the classification errors caused by directly adopting the clustering algorithm, the MCM takes advantage of the potential information in evolutionary game data to modify the clustering result, from which the correct classification outcome can be obtained.
2) In light of the classification result by the MCM, another valid method TRA is proposed to reconstruct the node signals in different sets. Compared with traditional methods, the TRA promotes the reconstruction effect as it contrapuntally incorporates the structural features.
3) The MCM_TRA can be regarded as a specific paradigm incorporating network structure information, which effectively improves the reconstruction performance for K-forked tree networks with different depths and widths. Furthermore, the reconstruction effect could be promoted for a broad range of parameters.

Similar to the sequential reconstruction algorithm used in frequency-difference electrical impedance tomography [50] and the decentralized Bayesian reconstruction algorithm for networked sensing systems [51]. The MCM_TRA proposed in this article may be applied in some potential application scenarios, such as reconstructing the topology of information flow in biological networks, tree-based structure in the optimization scheme of power networks, etc. Therefore, understanding latent structural information is a crucial step toward a more accurate network reconstruction.

Although the MCM_TRA performs well than previous methods for K-forked tree networks and other networks with similar structure characteristics, it still has certain limitations. In the future, further enhancing the generality and accuracy of the algorithm is our target. Furthermore, we also expect this method can stimulate the generation of more effective reconstruction algorithms, which will provide new insights for the field of network reconstruction.

References

[1] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, Mar. 2001.

[2] R. Foote, "Mathematics and complex systems," *Science*, vol. 318, no. 5849, pp. 410–412, Oct. 2007.

[3] S. Wang *et al.*, "Inferring dynamic topology for decoding spatiotemporal structures in complex heterogeneous networks," *Proc. Nat. Acad. Sci.*, vol. 115, no. 37, pp. 9300–9305, Aug. 2018.

[4] R. Eletreby, Y. Zhuang, K. M. Carley, O. Yağan, and H. V. Poor, "The effects of evolutionary adaptations on spreading processes in complex networks," *Proc. Nat. Acad. Sci.*, vol. 117, no. 11, pp. 5664–5670, Mar. 2020.

[5] D. Brockmann and D. Helbing, "The hidden geometry of complex, network-driven contagion phenomena," *Science*, vol. 342, no. 6164, pp. 1337–1342, Dec. 2013.

[6] J. Lu and G. Chen, "A time-varying complex dynamical network model and its controlled synchronization criteria," *IEEE Trans. Autom. Control*, vol. 50, no. 6, pp. 841–846, Jun. 2005.

[7] R. Espejo, S. Lumbreras, and A. Ramos, "A complex-network approach to the generation of synthetic power transmission networks," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3050–3058, Sep. 2019.

[8] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[9] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[10] R. Ma, F. Hu, and Q. Hao, "Active compressive sensing via pyroelectric infrared sensor for human situation recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 12, pp. 3340–3350, Dec. 2017.

[11] C. L. Apicella, F. W. Marlowe, J. H. Fowler, and N. A. Christakis, "Social networks and cooperation in hunter-gatherers," *Nature*, vol. 481, no. 7382, pp. 497–501, Jan. 2012.

[12] X. Han, Z. Shen, W.-X. Wang, and Z. Di, "Robust reconstruction of complex networks from sparse data," *Phys. Rev. Lett.*, vol. 114, no. 2, Jan. 2015, Art. no. 028701.

[13] G. Mei, X. Wu, Y. Wang, M. Hu, J.-A. Lu, and G. Chen, "Compressive-sensing-based structure identification for multilayer networks," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 754–764, Feb. 2018.

[14] Y. Zhang, C. Yang, K. Huang, M. Jusup, Z. Wang, and X. Li, "Reconstructing heterogeneous networks via compressive sensing and clustering," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, Jun. 8, 2020, doi: 10.1109/TETCI.2020.2997011.

[15] F. Morone, I. Leifer, and H. A. Makse, "Fibration symmetries uncover the building blocks of biological networks," *Proc. Nat. Acad. Sci.*, vol. 117, no. 15, pp. 8306–8314, Mar. 2020.

[16] M. Behr, M. A. Ansari, A. Munk, and C. Holmes, "Testing for dependence on tree structures," *Proc. Nat. Acad. Sci.*, vol. 117, Apr. 2020, Art. no. 201912957.

[17] W. D. Oliveira, J. P. Vieira, U. H. Bezerra, D. A. Martins, and B. das G. Rodrigues, "Power system security assessment for multiple contingencies using multiway decision tree," *Elect. Power Syst. Res.*, vol. 148, pp. 264–272, Jul. 2017.

[18] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[19] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J. Ye, "Network reconstruction based on evolutionary-game data via compressive sensing," *Phys. Rev. X*, vol. 1, no. 2, Dec. 2011, Art. no. 021021.

[20] C. Siegenthaler and R. Gunawan, "Assessment of network inference methods: How to cope with an underdetermined problem," *PLoS ONE*, vol. 9, no. 3, Mar. 2014, Art. no. e90481.

[21] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, "Reconstructing propagation networks with natural diversity and identifying hidden sources," *Nat. Commun.*, vol. 5, no. 1, p. 4323, Jul. 2014.

[22] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[23] F. Liu *et al.*, "Nonconvex compressed sensing by nature-inspired optimization algorithms," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1042–1053, May 2015.
[24] R. Jansen, "A Bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449–453, Oct. 2003.
[25] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi, "Weighted l1 minimization for sparse recovery with prior information," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, South Korea, Jun 2009, pp. 483–487.
[26] L. Yu, H. Sun, J. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Process.*, vol. 92, no. 1, pp. 259–269, Jan. 2012.
[27] K. Huang, Z. Wang, and M. Jusup, "Incorporating latent constraints to enhance inference of network structure," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 466–475, Jan.–Mar. 2020.
[28] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals Syst. Comput.*, 1993, pp. 40–44.
[29] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harm. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.
[30] L. Zhou and H. Man, "Cooperative compressive spectrum sensing in cognitive radio based on w-OMP," in *Proc. Military Commun. Conf. (MILCOM)*, Nov. 2013, pp. 1187–1192.
[31] M. Babakmehr, M. G. Simoes, M. B. Wakin, and F. Harirchi, "Compressive sensing-based topology identification for smart grids," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 532–543, Apr. 2016.
[32] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[33] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
[34] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
[35] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.
[36] J. M. McNamara, Z. Barta, and A. I. Houston, "Variation in behaviour promotes cooperation in the prisoners dilemma game," *Nature*, vol. 428, no. 6984, pp. 745–748, Apr. 2004.
[37] P. Buesser, M. Tomassini, and A. Antonioni, "Opportunistic migration in spatial evolutionary games," *Phys. Rev. E*, vol. 88, no. 4, Oct. 2013, Art. no. 042806.
[38] M. H. S. Segler, M. Preuss, and M. P. Waller, "Planning chemical syntheses with deep neural networks and symbolic AI," *Nature*, vol. 555, no. 7698, pp. 604–610, Mar. 2018.
[39] D. Denisko and M. M. Hoffman, "Classification and interaction in random forests," *Proc. Nat. Acad. Sci.*, vol. 115, no. 8, pp. 1690–1692, Feb. 2018.
[40] Y. Yuan and Z. Bar-Joseph, "Deep learning for inferring gene relationships from single-cell expression data," *Proc. Nat. Acad. Sci.*, vol. 116, no. 52, pp. 27151–27158, Dec. 2019.
[41] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, 1967, pp. 281–297.
[42] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996.
[43] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, 1996, pp. 226–231.
[44] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA, USA: MIT Press, 2000.
[45] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.
[46] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[47] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[48] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ 1 minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, Oct. 2008.
[49] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.
[50] S. Liu, Y. Huang, H. Wu, C. Tan, and J. Jia, "Efficient multitask structure-aware sparse Bayesian learning for frequency-difference electrical impedance tomography," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 463–472, Jan. 2021.
[51] W. Chen and I. J. Wassell, "A decentralized Bayesian algorithm for distributed compressive sensing in networked sensing systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1282–1292, Feb. 2016.

Xiaoping Zheng received the B.S. degree from the Chengdu University of TCM, Chengdu, China, in 1995, and the Ph.D. degree from Sichuan University, Chengdu, in 2003.

From 2006 to 2013, he was a Professor with the Institute of Safety Management, Beijing University of Chemical Technology, Beijing, China. He is currently a Professor with the Department of Automation, Tsinghua University, Beijing. His current research interests include large-scale crowd evacuation and evolutionary game theory.

Prof. Zheng was a 973 Chief Scientist in 2011, and recipient of the National Science Fund for Distinguished Young Scholars in 2012.

Wenhan Wu received the B.S. degree from the School of Automation, Central South University, Changsha, China, in 2019. He is currently pursuing the Ph.D. degree in control science and engineering with the Tsinghua University, Beijing, China.

His current research interests include crowd evacuation, complex networks, crowd dynamics, and robot collective control.

Wenfeng Deng received the B.S. degree from the School of Automation, Central South University, Changsha, China, in 2019, where he is currently pursuing the master's degree in control science and engineering.

His current research interests include complex networks, evolutionary game theory, and sparse Bayesian learning.

Chunhua Yang (Senior Member, IEEE) received the M.S. degree in automatic control engineering and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 1988 and 2002, respectively.

From 1999 to 2001, she was a Visiting Professor with the University of Leuven, Leuven, Belgium. Since 1999, she has been a Full Professor with the School of Information Science and Engineering, Central South University. From 2009 to 2010, she was a Senior Visiting Scholar with the University of Western Ontario, London, ON, Canada. Her current research interests include modeling and optimal control of complex industrial processes, fault diagnosis, and intelligent control systems.

Keke Huang (Member, IEEE) received the B.A. degree in automatic control from Northeastern University, Shenyang, China, in 2012, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2017.

He is currently an Associate Professor with Central South University, Changsha, China. His research interests include network science, big data, and process monitoring.