Graph Neural Network Encoding for Community Detection in Attribute Networks

Jianyong Sun, *Senior Member, IEEE*, Wei Zheng, Qingfu Zhang, *Fellow, IEEE*, and Zongben Xu, *Member, IEEE*

***Abstract*—In this article, we first propose a graph neural network encoding method for the multiobjective evolutionary algorithm (MOEA) to handle the community detection problem in complex attribute networks. In the graph neural network encoding method, each edge in an attribute network is associated with a continuous variable. Through nonlinear transformation, a continuous valued vector (i.e., a concatenation of the continuous variables associated with the edges) is transferred to a discrete valued community grouping solution. Further, two objective functions for the single-attribute and multiattribute network are proposed to evaluate the attribute homogeneity of the nodes in communities, respectively. Based on the new encoding method and the two objectives, a MOEA based upon NSGA-II, called continuous encoding MOEA, is developed for the transformed community detection problem with continuous decision variables. Experimental results on single-attribute and multiattribute networks with different types show that the developed algorithm performs significantly better than some well-known evolutionary- and nonevolutionary-based algorithms. The fitness landscape analysis verifies that the transformed community detection problems have smoother landscapes than those of the original problems, which justifies the effectiveness of the proposed graph neural network encoding method.**

***Index Terms*—Community detection, complex attribute network, graph neural network encoding, multiobjective evolutionary algorithm (MOEA).**

I. Introduction

A GRAPH network can be represented as a set of nodes (vertices) and edges that connect these nodes. Complex networks have been used to model many real-world network systems, such as the World Wide Web [1], scientific collaboration networks [2], social and biological networks [3], and many others, since these networks all exhibit some community structures. Unveiling these structures, also called community detection, is thus of great importance to understand the behavior and organization of complex networks, and the relationships among generic entities.

The goal of community detection is to partition all nodes in a complex network into some clusters such that nodes within a cluster are densely connected to each other and sparsely to nodes in other clusters. This problem has been proved to be NP-hard [4]. Due to the importance of the complex network detection problem, research on this subject has become popular since the 1930s [5]. A large interdisciplinary community of scientists has been working on this problem and a large amount of methods have been proposed for different types of complex networks. Surveys of community detection in graphs and networks can be found every several years from 2005 until recently [6], [7]. In this article, we do not intend to review all literature but only on approaches based on the evolutionary algorithm (EA) which is closely related to our work.

Various metrics have been proposed to quantitatively measure the quality of a partition to a graph network [6], for example, the modularity (Q) [8], the community score (CS) [9], the sum of community fitness $\mathcal{P}(\mathcal{S})$ [10], and others. The community detection problem can then be formalized as a discrete optimization problem based on the optimization of one or several metrics. As a promising paradigm for discrete optimization, EAs or multiobjective EAs (MOEAs) have also been applied to this problem; please refer to a recent survey in [11].

The genetic algorithm (GA) was first adopted in [12] for maximizing Q. Since then, several GAs, including MIGA [13], MAGA-Net [14] and Meme-Net [15], and GDPSO [16], were also developed based on optimizing Q, while GA-Net [9] was proposed based on optimizing CS.

MOEAs have also been applied since it is not comprehensive to measure a partition solution by only a single objective. The first multiobjective GA, dubbed as MOGA-Net [17], [18], was proposed in 2009. It was built upon NSGA-II in which two objectives, including CS and $\mathcal{P}(\mathcal{S})$, are used. MOEA/D-Net [19] takes the negative ratio association (NRA) and ratio cut (RC) as two objectives, which is built upon the framework of MOEA based on decomposition (MOEA/D). MICD [20], MODBSA [21], and DIM-MOEAD [22] were all developed based on the two objectives, but under different

Manuscript received 28 April 2020; revised 16 August 2020, 1 November 2020, and 24 November 2020; accepted 3 January 2021. Date of publication 10 February 2021; date of current version 19 July 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0809800; in part by the National Natural Science Foundation of China under Grant 11991023 and Grant 62076197; in part by the Major Project of National Science Foundation of China under Grant U1811461; in part by Key Project of National Science Foundation of China under Grant 11690011; in part by the Project of National Science Foundation of China under Grant 61721002; and in part by the Fundamental Research Funds for the Central Universities under Grant xzy022019074. This article was recommended by Associate Editor H. Ishibuchi. *(Corresponding author: Wei Zheng.)*

Jianyong Sun, Wei Zheng, and Zongben Xu are with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: jy.sun@xjtu.edu.cn; weizheng@stu.xjtu.edu.cn; zbxu@xjtu.edu.cn).

Qingfu Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: qingfu.zhang@cityu.edu.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3051021.

Digital Object Identifier 10.1109/TCYB.2021.3051021

MOEA frameworks. In MOCD [23] and MMCD [24], two objectives obtained by decomposing the modularity Q, which are to measure the intracluster edge density and intercluster sparsity, were used. Besides these works, MOEAs have also been applied on large networks [25], signed networks [26], and dynamic networks [27], in which various metrics for specific networks are developed and optimized.

In this article, we focus on the community detection for complex attribute networks. In many real complex networks, besides the connecting edges among nodes, there are also attributes associated with each node which are describe the node's properties. For example, in a social network, each user may have attributes like age, sex, degree, hobby, and other tags. Such networks are often called attributed complex networks [7]. For such a network, community detection requires to reveal not only the distinct network topological structure but the homogeneity of attributes within clusters. For example, we may wish to find a group of users with similar hobbies. The extra homogeneity requirement makes the community detection problem for attribute network much more difficult.

For attribute complex networks, revealing network structure and node attributes' homogeneity are two desired goals. Through establishing appropriate objectives for network structure and node attributes, MOEAs could also be applied for the attribute complex network community detection problems.

To the best of our knowledge, only two promising papers based on MOEAs have been published for attribute network detection. The first one is MOEA-SA [28], in which a new objective S_A was proposed to measure the attribute similarity within clusters. Together with the modularity [8], MOEA-SA is developed upon NSGA-II [29] with a hybrid network encoding method and a multi-individual-based mutation operator. Besides, a neighborhood correction strategy is proposed to repair the improper solution. The other one is MOGA-@Net [30], which is also developed based on NSGA-II. In MOGA-@Net, three objectives (namely, modularity [8], community score [9], and conductance [31]) for evaluating the structural dimension and three objectives (namely, Jaccard, cosine, and Euclidean-based similarity) for measuring the attribute homogeneity are considered. A post processing local merge procedure is further introduced to merge the communities.

In the above MOEA-based community detection algorithms, an encoding process is required to initialize individuals and a decoding process to retrieve individuals to their corresponding partitions for evaluating their qualities. There are two widely used encodings, including the locus based [32] and label based [12]. It is argued in [28] that the locus-based encoding is able to initialize for good individuals, but is time-consuming when decoding. The label-based encoding, on the other hand, is easy for designing evolutionary operators, but is not good at initialization since the adjacency information is not involved.

In this article, we propose a novel encoding method. The new encoding method is implemented by first associating each edge in a graph network with a continuous variable, then transforming the concatenation of the continuous variables to a partition solution of the considered attribute network by a series of nonlinear functions. Based on this encoding, the original discrete-valued community detection problem is transformed into a continuous one. We then propose a continuous-coded MOEA built upon NSGA-II [29], in which each individual is a continuous valued vector as opposed to a discrete valued vector in the locus-based and label-based encodings.

There are mainly two reasons that motive us to develop such transformation. First, different continuous problems, there is no sufficient and useful neighborhood information to help searching in discrete problems [33]. Second, the local structure of a discrete problem usually has high ruggedness, which means that the fitness landscape is not smooth. That is, a small change of the genotype may result in a substantial change of the phenotype [34], [35]. As a result, this may cause oscillation of the search process [36]. These factors make it very difficult to search over the discrete search space. On the contrary, the proposed continuous encoding method can make the fitness landscape of the transformed problem smoother than the original landscape. This will not only make the search easier, but also remedy the shortcomings of the locus and label-based encoding methods. To verify these grounds, a fitness landscape analysis is carried out in this article. The analysis confirms that the transformed continuous problem indeed has a smoother landscape than that of the original community detection problem.

Further, to better measure the attribute homogeneity in the communities, we propose two objectives, similar to those proposed in [28], to handle the single-attribute and multiattribute similarity, respectively.

The remainder of this article is organized as follows. Section II introduces some preliminaries, including the definition of the complex attribute network, relevant concepts in multiobjective optimization, and MOEA. The proposed method, including the graph neural network encoding, the objectives for attributes, and the continuous encoding MOEA (CE-MOEA), is presented in Section III. Experiment studies on a variety of networks with different types are carried out in Section IV. The fitness landscape analysis is presented in Section V. Related work on non-EA-based community detection is reviewed in Section VI. Section VII concludes this article.

II. Preliminaries

A. Complex Attribute Network

An attribute network is a 3-tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{V_1, V_2, \ldots, V_r\}$ is the set of nodes, $\mathcal{E} = \{e_{ij} : 1 \leq i, j \leq r\}$ is the set of edges ($e_{ij} = 1$ means V_i links to V_j), and $\mathcal{A} = \{a_1, a_2, \ldots, a_r\}$ is the set of attributes for the nodes. Here, a_i, $1 \leq i \leq r$ can be discrete or continuous, and may be one or multiple dimensional.

Fig. 1 shows a simple attribute network example. The network has eight nodes and ten edges. Each node has four attributes (age, sex, degree, and major). According to the attributes of each node, it is seen that this network can be divided into two communities: 1) $\{V_1, V_2, V_3, V_4\}$ and 2) $\{V_5, V_6, V_7, V_8\}$. However, if considering only the "age" attribute, it is rather difficult to partition this network.

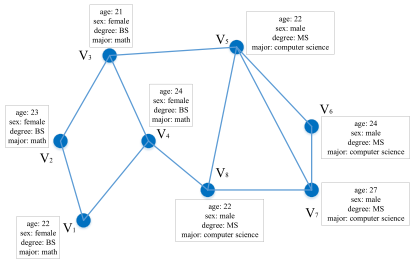


Fig. 1. Example attribute network with eight nodes, ten edges. There are four attributes for each node.

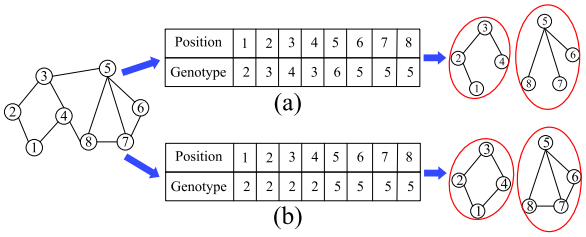


Fig. 2. Example of the locus-based encoding and label-based encodings. (a) Locus-based encoding. (b) Label-based encoding.

On the other hand, it is also not easy to partition the network based purely on its structure. However, considering both attributes and network structure, it might be easy to determine two communities: 1) $\{V_1, V_2, V_3, V_4\}$ and 2) $\{V_5, V_6, V_7, V_8\}$. This partition not only minimizes the similarity within communities, but maximizes the communities' attributes homogeneity.

B. Multiobjective Optimization

A multiobjective optimization problem (MOP) can be stated as follows:

$$\begin{aligned} &\text{minimize } F(\mathbf{w}) = (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))^{\mathsf{T}} \\ &\text{subject to } \mathbf{w} \in \mathbf{\Omega} \end{aligned} \tag{1}$$

where $\mathbf{\Omega}$ is the search space (could be continuous or discrete), $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbf{\Omega}$ is the decision variable. $F : \mathbf{\Omega} \to \mathbb{R}^m$ consists of m real-valued objective functions.

In the MOP taxonomy, a vector $\mathbf{x} = (x_1, \ldots, x_m)$ is said to dominate a vector $\mathbf{y} = (y_1, \ldots, y_m)$ denoted as $\mathbf{x} \prec \mathbf{y}$) if and only if there exists at least one k such that $x_j \leq y_j \forall j \in \{1, \ldots, m\}$ but $x_k < y_k$. If a solution $\mathbf{x}^* \in \mathbf{\Omega}$ is not dominated by any other solution, $\mathbf{x}^*$ is called a Pareto optimal solution. There exists many optimal solutions that are nondominated to each other. The set of all these optimal solutions is called the Pareto set (PS), while its image is called the Pareto front (PF).

The primal advantage of the MOEA paradigm is that an approximated PS can be reached in a single run. The study of MOEA is one of the most popular avenues in computational intelligence. There are main four categories of MOEAs, namely: 1) Pareto dominance relation-based (such as NSGA-II [29] and NSGA-II/CSDR [37]); 2) performance metric-based (such as HypE [38] and FV-MOEA [39]); 3) decomposition-based (such as MOEA/D [40], MOEA/D-IR [41], and MOEA/D-CMA [42]); and 4)learning-based MOEAs (such as OCEA [43], CA-MOEA [44], and GMOEA [45]). We do not intend to review the rich literature of MOEA in this article. Interested readers should refer to [46].

In this article, the purpose of the attribute complex network community detection problem is to find a partition of communities such that two requirements are satisfied, including: 1) the edges between communities are sparse and those within the community are dense and 2) the node attributes in the same community should be similar as much as possible while the similarity of node attributes in different communities should be dissimilar. Therefore, the community detection problem can be readily modeled as a two-objective optimization problem. Using MOEA to solve this problem is thus straightforward and maybe promising.

III. Method

In this section, the graph neural network encoding method is first presented, followed by two newly developed objectives for attribute homogeneity, and CE-MOEA.

A. Graph Neural Network Encoding

The locus-based [32] and label-based [12] encodings have been widely used in MOEAs for the network-related optimization problem. Fig. 2 shows an example of the two encodings for the network in Fig. 1. It is seen that both encodings have a coding length equivalent to the number of nodes in the network.

In the locus-based encoding, a node's genotype is taken as one of its linked nodes. For example, in the example network, node 1 links to nodes 2 and 4. The genotype of node 1 could thus be 2 or 4. The shown individual genotype $(2, 3, 4, 3, 6, 5, 5, 5)$ in Fig. 2(a) is obtained by associating each node with one of its linked nodes. This individual can be decoded into two communities, that is, $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$, by simply retrieving it to an induced graph to the original one.

In the label-based encoding, each node's genotype can be any integer in $\{1, 2, \ldots, r\}$. This integer indicates which cluster this node belongs to. As shown in Fig. 2(b), 2 and 5 are selected as the genotype of the nodes. The decoding process is to simply take the nodes with the same cluster index together. It is seen that the same partition of communities as the previous encoding are obtained after decoding. It should be noted that the resulting genotypes by the two encoding methods are all discrete vector.

As argued in [28], it is difficult to design evolutionary operators for the locus-based encoding, and difficult to initialize individuals with high quality for the label-based encoding since the adjacency information among nodes is not used.

In the following, we present the proposed graph neural network encoding method. We summarize its pseudo code in Algorithm 1. In Algorithm 1, a continuous valued vector $\mathbf{x} \in \mathbb{R}^d$, where $d = \sum_{i,j} e_{ij}$ is the number of edges, is taken as the input. $\mathbf{x}$ is a concatenation of r subvectors, where $\mathbf{x}_i$ represents the continuous vector associated with node V_i, $1 \leq i \leq r$. The length of $\mathbf{x}_i$ is $d_i = \sum_j e_{ij}$. That is, each link connecting V_i to the other nodes is assigned with one continuous value. We denote the set of nodes that links with V_i as D_i.

Algorithm 1: Graph Neural Network Encoding Method

Input: $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_r] \in \mathbb{R}^d$.
Output: A community partition $\mathcal{G}_\mathcal{S}$.
1 Set $\mathcal{S} = \emptyset$ and $\mathcal{E}_\mathcal{S} = \emptyset$;
2 **for** $i \leftarrow 1$ *to* r **do**
3 $\quad \mathbf{h}_i \leftarrow \sigma(\mathbf{x}_i)$;
4 $\quad \mathbf{p}_i \leftarrow \text{softmax}(\mathbf{h}_i)$;
5 $\quad s_i \leftarrow \text{argmax}(\mathbf{p}_i)$;
6 $\quad \mathcal{S} \leftarrow \mathcal{S} \bigcup s_i$;
7 $\quad \mathcal{E}_\mathcal{S} \leftarrow \mathcal{E}_\mathcal{S} \bigcup e_{V_i, V_{s_i}}$;
8 **end**
9 **return** $\mathcal{G}_\mathcal{S} \leftarrow \text{Decoding}(\mathcal{S}, \mathcal{E}_\mathcal{S})$.

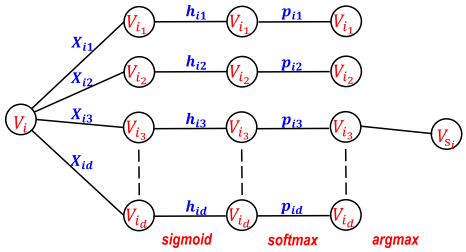


Fig. 3. Demo of the graph network encoding of single node V_i.

For node V_i, denote $\mathbf{x}_i = [x_{i1}, \ldots, x_{id_i}]$. We then first apply a sigmoid function σ, which is defined as follows:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{2}$$

over $\mathbf{x}_i$ element by element. This gives $\mathbf{h}_i \in (0, 1)^{d_i}$ (line 3). A softmax function is then applied on $\mathbf{h}_i$ to obtain $\mathbf{p}_i = [\mathbf{p}_{i1}, \ldots, \mathbf{p}_{id_i}]$ (line 4) where

$$\mathbf{p}_{ij} = \frac{\exp(\mathbf{h}_{ij})}{\sum_j \exp(\mathbf{h}_{ij})}, 1 \le j \le d_i. \tag{3}$$

Since $\mathbf{p}_{ij} \ge 0$ and $\sum_j \mathbf{p}_{ij} = 1$, this actually gives the probability of choosing a node from D_i. We propose to choose node s_i such that

$$s_i = \arg \max_{j=1,\ldots,d_i} \mathbf{p}_{ij}$$

that is, the argmax operation as seen in line 5. This means that node V_i is linked to V_{s_i} in the genotype. The above process is carried out for all nodes in the considered network to obtain the set of nodes (i.e., $\mathcal{S}$ in line 6) and the set of edges (i.e., $\mathcal{E}_\mathcal{S}$ in line 7). With the obtained $\mathcal{S}$ and $\mathcal{E}_\mathcal{S}$, a partition $\mathcal{G}_\mathcal{S}$ can be returned after decoding (line 9).

Fig. 3 shows the encoding process of a single node V_i. From the figure, it is seen that for each V_i, a continuous value is asscociated with each node in D_i. Through sigmoid, softmax, and argmax operation, node V_{s_i} is selected to be linked to V_i in the genotype.

Fig. 4 shows the full process of encoding and decoding, taking the network in Fig. 1 as an example. Given the network (denoted as $\mathcal{G}$), with continuous vector $\mathbf{x}$ associated with the edges, the sigmoid operation (which can be regarded as the sigmoid layer in the neural network) is applied to obtain $\mathbf{h}$. The softmax layer is then applied on $\mathbf{h}$ to obtain $\mathbf{p}$. Through argmax operation (layer), each node is linked to the node that is with the greatest probability entity in its corresponding $\mathbf{p}$ values. This concludes the encoding process, which lead to a locus-based representation. The decoding process can thus turn the representation into a partition $\mathcal{G}_\mathcal{S}$ of $\mathcal{G}$.

In the following, for the sake of simplicity, we use:

$$\mathcal{G}_\mathcal{S} = \text{GNN}(\mathbf{x}; \mathcal{G}) \tag{4}$$

to represent the encoding of a genotype to a network $\mathcal{G}$. That is, given $\mathbf{x}$, a network partition $\mathcal{G}_\mathcal{S}$ can be obtained by function GNN$(\cdot)$. With the obtained network partition, objectives, such as the modularity, can be calculated.

B. Objective Functions

1) Objective Regarding the Network Structure: The well-known modularity Q proposed in [8] is used as the first objective in our study for revealing the network structure. Given a network $\mathcal{G}$ and its partition $\mathcal{G}_\mathcal{S}$, let c be the number of obtained communities, l_k be the total number of edges that connect the nodes within the community k, d_k is the sum of degrees of nodes of community k, and L stands for the total number of edges, The modularity Q is defined as

$$Q = \sum_{k=1}^{c} \left[\frac{l_k}{L} - \left(\frac{d_k}{2L} \right)^2 \right] \triangleq f_Q(\mathcal{G}_\mathcal{S}; \mathcal{G}). \tag{5}$$

A higher Q value indicates a network with a more well-defined community structure.

Together with (4), given $\mathbf{x}$, the modularity can be computed by composing functions f_Q and GNN as follows:

$$Q = f_Q \circ \text{GNN}(\mathbf{x}; \mathcal{G}).$$

For the sake of simplicity, we denote $Q(\mathbf{x})$ as the function to compute the modularity, taking $\mathbf{x}$ as the decision variable.

2) Objective Evaluating Attribute Similarity: To measure the difference between two nodes' attributes, the following two objectives are used for single-attribute and multiattribute homogeneity, which is a modification to those objectives proposed in MOEA-SA [28], respectively.

For a single-attribute network with real-valued attributes, a similarity objective function f_s is proposed as follows:

$$f_s = \frac{S_O}{\sum_{k=1}^{c} r_k(r_k - 1)} \tag{6}$$

where

$$S_O = \sum_{k=1}^{c} \sum_{\substack{V_i, V_j \in C_k \\ i<j}} \sqrt{(a_i - a_j)^2} \tag{7}$$

where c is the number of obtained clusters, C_k is the cluster k, r_k stands for the number of nodes within cluster k, and a_i (respectively, a_j) is the attribute of V_i (respectively, V_j). S_O is the sum of the Euclidean distance between node attributes of each community. The denominator is the summation of all obtained clusters of values $r_k(r_k - 1)$. Note that in MOEA-SA, the numerator is $\sum_{k=1}^{c} \sum_{i,j \in k, i<j} 2s(i, j)$, where $s(i, j) = 0$ while $a_i = a_j$ and $s(i, j) = 1$ while $a_i \neq a_j$. It is to measure the distance between single attribute homogeneity within the detected communities.

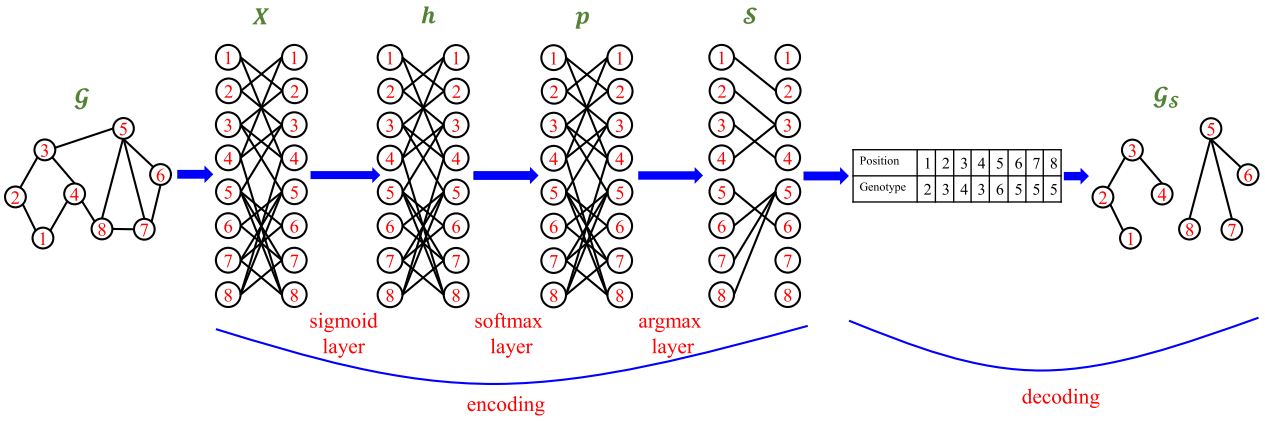


Fig. 4. Demo of the graph neural network encoding and decoding process.

For a multiattribute network with binary attribute values, a cosine-based similarity objective function f_m is proposed to measure the attribute similarity

$$f_m = \frac{M_O}{\sum_{k=1}^{c} r_k(r_k - 1)} \tag{8}$$

where

$$M_O = \sum_{k=1}^{c} \sum_{\substack{V_i, V_j \in C_k \\ i<j}} \frac{a_i \cdot a_j}{\|a_i\| \|a_j\|} \tag{9}$$

where $\|\cdot\|$ means the norm of a vector. The numerator is the cosine value of attributes of each node pair's attributes within a community k. The summation of all detected clusters is denoted as M_O. The denominator is the same as in f_s.

It can be found that in f_s (or f_m), the smaller the value of f_s (or f_m) is, the more homogeneous the node attributes are than in the obtained communities. Therefore, the node attribute clustering problem can be viewed as a problem of finding a division of a network such that the attribute similarity objective function f_s or f_m is minimized. Similar to the definition of $Q(\mathbf{x})$, we also define $f_s(\mathbf{x})$ and $f_m(\mathbf{x})$.

In summary, based on the proposed graph neural network encoding, given a continuous valued vector $\mathbf{x}$, the modularity and attribute similarity can be computed. The problem is thus to find an approximation set to PS with respect to $\mathbf{x}$ such that the objective vector $F = (-Q(\mathbf{x}), f_s(\mathbf{x}))$ [or $F = (-Q(\mathbf{x}), f_m(\mathbf{x}))$] is minimized. Formally, the community detection problem for the complex attribute network can be defined as follows:

$$\begin{aligned} &\text{minimize } F = (-Q(\mathbf{x}), f_s(\mathbf{x})) \\ &\text{or } F = (-Q(\mathbf{x}), f_m(\mathbf{x})) \\ &\text{s.t. } \mathbf{x} \in [0, 1]^d. \end{aligned} \tag{10}$$

Here, the reason to set $\mathbf{x} \in [0, 1]^d$ is to make the range of the sigmoid controllable and the softmax is scale invariant.

C. Algorithm

The developed algorithm is built upon the well-known nondominated sorting GA II (dubbed as NSGA-II) with differential evolution (DE) operators [47]. Its flowchart is shown in Fig. 5 and the pseudocode is summarized in Algorithm 3.

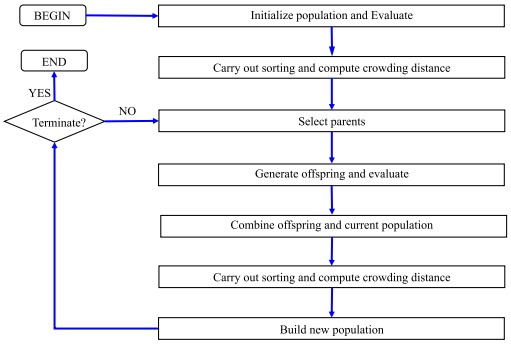


Fig. 5. Flowchart of CE-MOEA.

Algorithm 2: Nondominated Sorting Method

Input: Population P.
Output: L_s: some nondominated layers.
1 Set index $i = 1$;
2 **while** $P \neq \emptyset$ **do**
3 Find all solutions S who are not dominated by any solution in P;
4 $L_{s_i} \leftarrow S$ and $P \leftarrow P \setminus S$;
5 $i \leftarrow i + 1$;
6 **end**

The functions FNS(·), CWD(·), and BTS(·) in Algorithm 3 represent the fast nondominated sorting, crowding distance, and binary tournament selection, respectively. They are just standard operations used in NSGA-II. The FNS(·) function sorts solutions into several nondominated layers based on their dominance relationship. Its pseudocode is shown in Algorithm 2 (taken from [29]). The CWD(·) function is used to maintain the diversity of population. The DE(·) function stands for the DE operation, which will be described later. For more details of NSGA-II, please refer to [29].

In Algorithm 3, the first population P_1 is randomly initialized in $[0, 1]^d$ (line 1), and individuals are evaluated according to (10). Their objectives are organized in $\mathcal{F}$ (line 2). The nondominated layers and the crowding distances of $\mathcal{F}$ are then

Algorithm 3: CE-MOEA

Input: An attributed network $\mathcal{G}$, the population size N, the maximum number of generations: T; the parameters of the DE operator (F_{DE} and CR); and the PM operator parameters: p_m and η_m.
Output: an approximated PS and PF.

1 Set $g \leftarrow 1$ and randomly initialize $P_g \in [0,1]^{N\times d}$;
2 Evaluate $F_i \leftarrow F(P_g(i,:)), 1 \le i \le N$; Set $\mathcal{F}_1 = \{F_i\}$;
3 $L_s \leftarrow \text{FNS}(\mathcal{F})$ and $C_d \leftarrow \text{CWD}(\mathcal{F})$;
4 **while** $g < T$ **do**
5 $\quad \mathcal{P} \leftarrow \text{BTS}(P_g, \mathcal{F}_g, C_d)$;
6 $\quad$ Set $\mathcal{Y} = \emptyset$ and $\mathcal{F}_o = \emptyset$;
7 $\quad$ **for** $1 \le j \le |\mathcal{P}|$ **do**
8 $\quad\quad \mathbf{x}_1 \leftarrow \mathcal{P}(j,:)$;
9 $\quad\quad$ Randomly select $\mathbf{x}_2$ and $\mathbf{x}_3$ from $\{\mathcal{P} \setminus \mathbf{x}_1\}$;
10 $\quad\quad \mathbf{y} = \text{DE}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, F_{DE}, CR, p_m, \eta_m)$;
11 $\quad\quad \mathcal{Y} \leftarrow \mathcal{Y} \bigcup \mathbf{y}$;
12 $\quad\quad \mathcal{F}_o \leftarrow \mathcal{F}_o \bigcup F(\mathbf{y})$;
13 $\quad$ **end**
14 $\quad P_g \leftarrow P_g \cup \mathcal{Y}$ and $\mathcal{F}_g \leftarrow \mathcal{F}_g \cup \mathcal{F}_o$;
15 $\quad L_s \leftarrow \text{FNS}(\mathcal{F}_g)$ and $C_d \leftarrow \text{CWD}(\mathcal{F}_g)$;
16 $\quad$ Sort L_s based on C_d in the descending order;
17 $\quad$ Select nondominated solutions from the sorted L_s to fill P_{g+1} until its size equals to N;
18 $\quad g \leftarrow g + 1$;
19 **end**
20 **return** P_T as the approximated PS and $\mathcal{F}_T$ as the PF.

Algorithm 4: DE Operator

Input: individuals $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3 \in \mathbb{R}^d$ and recombination parameters F_{DE}, CR, p_m and η_m.
Output: An offspring $\mathbf{y}$.

1 **for** $1 \le i \le d$ **do**
2

$$y^i = \begin{cases} x_1^i + F_{DE} \times (x_2^i - x_3^i), & \text{if } rand() \le CR \\ x_1^i, & \text{otherwise} \end{cases}$$

3 **end**
4 For $i \in \{1, \ldots, d\}$, if $y^i < a_i$, $y^i = a_i$, otherwise, if $y^i > b_i$, $y^i = b_i$;
5 **for** $1 \le i \le d$ **do**
6

$$y^i = \begin{cases} y^i + \delta_i \times (b_i - a_i), & \text{if } rand() < p_m \\ y^i, & \text{otherwise} \end{cases}$$

and

$$\delta_i = \begin{cases} \left[2\widehat{r} + (1-2\widehat{r})\left(\frac{b_i - y^i}{b_i - a_i}\right)^{\eta_m}\right]^{\frac{1}{\eta_m}} - 1, & \text{if } \widehat{r} < 0.5 \\ 1 - \left[2 - 2\widehat{r} + (2\widehat{r} - 1)\left(\frac{y^i - a_i}{b_i - a_i}\right)^{\eta_m}\right]^{\frac{1}{\eta_m}}, & \text{otherwise} \end{cases}$$

where $\widehat{r}$ is an uniform random number in $[0, 1]$;
7 **end**
8 Repair $\mathbf{y}$ if necessary.
9 **return** $\mathbf{y}$.

computed in line 3. From lines 4–19, the NSGA-II operations are performed to optimize Problem (10). The binary tournament selection is carried out on P_g to obtain a parent set $\mathcal{P}$ based on $\mathcal{F}$ and the crowding distance C_d (line 5). By selecting parent individuals from $\mathcal{P}$, DE is applied to generate new offsprings (line 10). The newly generated individuals are evaluated in line 12 and combined with current population (line 14). The combined individual objectives are sorted to obtain the nondominated layers and the crowding distances (lines 15 and 16). Solutions are then selected from the sorted layers to obtain the next generation (lines 17 and 18). The algorithm continues until the maximum number of generations T has been reached. The final population is returned as the approximated PS and PF (line 20).

Algorithm 4 summarizes the DE$(\cdot)$ function used to generate offsprings. In Algorithm 4, $\mathbf{x}_1$ is first mutated by taking the difference of $\mathbf{x}_2$ and $\mathbf{x}_3$, while the mutation takes effect only when a random number in $(0, 1)$ [output by function *rand*()] is less than CR (line 2). The obtained $\mathbf{y}$ is repaired if any of its element is beyond the variable range (line 4). The PM operator [48] is the used to mutate $\mathbf{y}$ (line 6). The obtained individual $\mathbf{y}$ is repaired (line 8) and returned (line 9).

The advantages of the proposed approach can be summarized as follows.

1) The continuous encoding method makes full use of the adjacency information in a network by means of the softmax layer. This can increase the robustness of the search and ensure an MOEA to have a good performance.
2) The continuous encoding method can be applied to the attribute or nonattribute network, and to the undirected or directed network.
3) By transforming a discrete optimization problem into a continuous one, any promising MOEAs for continuous MOPs can be applied. As later described in the fitness landscape analysis, we find that the continuous encoding can result in a smoother landscape which are beneficial for problem solving.

D. Notes

Besides the proposed transformation method, the application of continuous approaches for solving discrete problems has also been studied based on different characterizations or reformulation. As pointed in [49], continuous approaches may be able to reveal some new properties to the original problem. This will allow the development of new promising approaches. Researchers have tried on some typical discrete problems based on equivalent continuous formulations or relaxations, such as 0-1 programming [50], maximum clique problem [51], discrete DC programming [33], nonlinear mixed programming [52], and others.

In these methods, to transform a discrete problem, it is required to represent its decision variables as binary. However, it is difficult to represent the community detection problem as a binary problem. To the best of our knowledge, no continuous approaches for community detection have been studied in literature.

E. Complexity Analysis

Let r be the number of nodes of a network $\mathcal{G}$, N be the population size, m be the number of objectives, L be the number of edges, and T be the maximum number of generations.

TABLE I
DETAILED CHARACTERISTICS OF THE BENCHMARK NETWORKS

Dataset	Network Type	Nodes	Edges	Attributes	with ground truth
Polbooks	Books co-purchasing	105	441	1	No
Polblogs	Blogs hyperlinks	1490	19025	1	No
Ego 0	Friendship	347	2519	224	No
Ego 107	Friendship	1016	25711	576	No
Ego 686	Friendship	170	1656	63	No
Ego 1684	Friendship	776	13826	319	No
Ego 1912	Friendship	748	29552	480	No
Ego 3437	Friendship	542	4749	262	No
Ego 3980	Friendship	58	143	42	No
Cora	Citation	2708	5429	1433	Yes
Citeseer	Citation	3312	4732	3703	Yes
Texas	A subset networks containing	187	328	1703	Yes
Cornell	web pages and hyperlink data	195	304	1703	Yes
Washington	of the four US universities	230	446	1703	Yes
Wisconsin	dataset from WebKB dataset	265	530	1703	Yes

Algorithm 1 requires a complexity of $\mathcal{O}(L)$ for the encoding process. The decoding process is the same as the locus-based decoding which requires a complexity of $\mathcal{O}(r)$ [53]. Thus, the total complexity of Algorithm 1 is $\mathcal{O}(L + r)$.

For CE-MOEA, its complexity at each generation is $\mathcal{O}(mN^2)$ which is the same as the complexity of NSGA-II. CE-MOEA needs $\mathcal{O}(L + r)$ for decoding each individual and $\mathcal{O}(L + r)$ for evaluating each individual. CE-MOEA also requires a complexity of $\mathcal{O}(LN)$ as the overhead for population initialization. Overall, the total complexity of CE-MOEA is $\mathcal{O}(mN^2T + (L + r)NT)$.

IV. EXPERIMENT RESULTS

In this section, experiments are carried out on a variety of networks with different types, with or without ground truth. CE-MOEA is implemented in MATLAB 2017b on a PC. The parameter settings of CE-MOEA is as follows: the number of population is $N = 100$, the maximum number of generations is $T = 200$, the DE parameters $F_{DE} = 0.7$, $CR = 0.5$, the mutation probability $p_m = 0.02$, and distribution index of mutation $\eta_m = 20$. CE-MOEA was run 31 times independently in all the study.

A. Benchmark Networks

A number of networks, including single-attribute and multiattribute networks have been used as the benchmark in our study.

Among these networks, Amazon U.S. Political Books [54] and Blogs [55] are with single attribute. They have no truth labels for the communities. The politics books dataset include all the books studying U.S. politics which were published for presidential election and sold by Amazon.com during 2004. It contains 105 nodes and 441 edges. An edge between two books means that the two books were both purchased by customers. Each book is associated with one attribute to demonstrate political complexion: 1) conservative; 2) liberal; and 3) neutrality. The political blogs dataset was compiled by Adamic and Glance in 2005 to show the political orientation of blogs. It contains 1490 nodes and 19 025 edges which connect blogs by hyperlinks. Each Web-blog has an attribute showing political complexion: 1) liberal or 2) conservative.

The rest of the networks, including the Ego facebook networks [56] (no ground truth available), the WebKB networks [57], the Cora citation network [58], and the Citeseer citation network [58] are with multiattributes. Ego facebook networks are a series of friendship networks. They are chosen from ten ego networks, consisting of 4039 users. The attribute dimension of all networks ranges from 42 to 576. A subset of WebKB dataset [57] consisting of four subnetworks from four U.S. universities: 1) Cornell; 2) Texas; 3) Washington; and 4) Wisconsin are used. The attribute dimension of all four networks: 1) Cornell; 2) Texas; 3) Washington; and 4) Wisconsin is 1703, which represent the Web pages and hyperlinks between them. The Cora dataset has 2708 nodes and 5429 edges, representing the scientific publications and their citation relationships. Each publication has been divided into seven categories: 1) neural network; 2) case-based reasoning; 3) GAs; 4) probabilistic methods; 5) reinforcement learning; 6) rule learning; and 7) theory. The attribute dimension of the Cora network is 1433. The Citeseer dataset has 3312 nodes and 4732 edges. Each publication has been classified into six categories: 1) artificial intelligence; 2) database; 3) information retrieval; 4) machine learning; 5) agents; and 6) human–computer interaction. The attribute dimension of Citeseer is 3703. Table I summarizes the detailed information of the benchmark networks.

B. Evaluation Metrics

To compare the performances of the compared algorithms, the following metrics, including density, entropy, and normalized mutual information (NMI), are used. The density and entropy metrics are applied to measure the detection performance on networks without ground truth, while NMI is for networks with true labels.

1) Density: The density D of a network is defined as

$$D = \sum_{k=1}^{c} \frac{l_k}{L} \tag{11}$$

where c is the number of communities, l_k is the number of edges within community k, and L represents the total number of edges in the network. The larger D, the more distinct the community structure in the network is.

2) Entropy: The entropy E of a network is defined as

$$E = \sum_{k=1}^{c} \frac{r_k}{r} \cdot H(k)$$
$$H(k) = -\sum_{a \in \mathcal{A}} p_{ak} \log(p_{ak}) \tag{12}$$

where p_{ak} is the percentage of nodes in a community C with attribute value a. r_k is the number of nodes in a cluster k, c is the total number detected clusters, r is the total number of nodes in the network. The smaller the E value, the more the homogeneous of nodes in the detected communities is.

3) Normalized Mutual Information: NMI [59] is proposed to measure the similarity between the true partitions and the detected communities. Given two partitions P and P^* of a network and M be the confusion matrix whose element M_{ij} is the number of nodes in community i of the partition P which are also in the community j of partition P^*. The NMI(P, P^*)

TABLE II
STATISTICS OF D AND E VALUES OBTAINED BY CE-MOEA AND THE COMPARED ALGORITHMS ON THE POLITICAL NETWORKS, WHERE *WR* MEANS WILCOXON'S RANKSUM TEST AT THE 5% SIGNIFICANCE LEVEL

Dataset	**Algorithms**	$D_{\max}$	$D_{\min}$	D_{avg}	WR	$E_{\max}$	$E_{\min}$	E_{avg}	WR	k
Polbooks	CE-MOEA	0.907	**0.889**	**0.899(0.007)**		0.200	**0.091**	**0.149(0.027)**		5-8
	MOEA-SA	0.864	0.849	0.859(0.005)	†	0.267	0.206	0.243(0.021)	†	4-8
	MOGA-@Net	**0.945**	0.789	0.869(0.049)	†	**0.198**	0.114	0.186(0.024)	†	4-10
Polblogs	CE-MOEA	**0.916**	**0.896**	**0.906(0.006)**		**0.042**	**0.030**	**0.036(0.004)**		14-32
	MOEA-SA	0.914	**0.896**	**0.906(0.007)**	≈	0.153	0.117	0.139(0.009)	†	4-16
	MOGA-@Net	0.902	0.891	0.898(0.003)	≈	0.095	0.040	0.061(0.013)	†	3-8

is defined as

$$\text{NMI}(P, P^*) = \frac{-2\sum_{i=1}^{c_P}\sum_{j=1}^{c_{P^*}} M_{ij}\log\left(\frac{r\cdot M_{ij}}{M_{i\cdot}M_{\cdot j}}\right)}{\sum_{i=1}^{c_P} M_{i\cdot}\log\left(\frac{M_{i\cdot}}{r}\right)+\sum_{j=1}^{c_{P^*}} M_{\cdot j}\log\left(\frac{M_{\cdot j}}{r}\right)} \tag{13}$$

where c_P (respectively, c_{P^*}) is the number of clusters in the partition P (respectively, P^*). $M_{i\cdot}$ (respectively, $M_{\cdot j}$) is the summation of elements of matrix M in row i (column j).

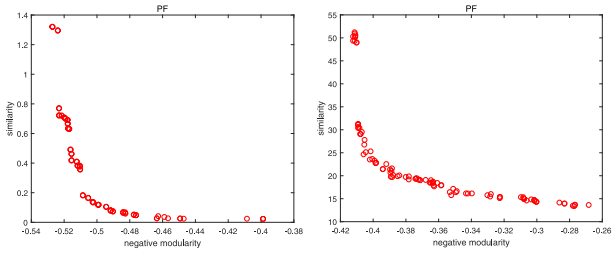
It is obvious that if $P = P^*$, then $\text{NMI}(P, P^*) = 1$. Otherwise, if P is entirely different from P^*, $\text{NMI}(P, P^*) = 0$. Therefore, a larger NMI indicates a better quality of the detected communities, and hence a better performance of the detection algorithm.

C. Results on Networks Without Ground Truth

Both MOEA-SA and MOGA-@Net are used for comparison. Their parameter settings are set as same in the literature. Note that in MOGA-@Net [30], three objectives regarding the network structure are employed. In our experiment, we choose to employ Q which is the same as used in CE-MOEA and MOEA-SA for a fair comparison.

1) Results on the Political Networks: Table II shows the detection results of CE-MOEA, MOEA-SA, and MOGA-@Net on the political books and blogs, in which the maximum, minimum, and average values of D and E are reported in columns. The standard deviations are shown in brackets. In the corresponding column, the best metric values are typeset in bold. In addition, Wilcoxon's rank sum test at a significance level of 5% is performed to test whether the results obtained by CE-MOEA and the compared algorithms are significantly different. In the tables, column *WR* shows the hypothesis test results, where †, ≈, and § means that the result obtained by CE-MOEA is better than, similar to, and worse than the result obtained by the compared algorithms, respectively.

From Table II, it is seen that CE-MOEA obtains the best results on the two political networks than MOEA-SA and MOGA-@Net, except on $D_{\max}$ for Polbooks. Wilcoxon's rank sum tests also suggest that CE-MOEA performs significantly better than MOEA-SA and MOGA-@Net, except on the D metrics for Polblogs where there is no significant difference between them. In the last column of Table II, k means the number of clusters obtained by the corresponding algorithm. We find that on the Polbooks network, CE-MOEA and MOEA-SA find similar number of clusters and MOGA-@Net obtains more. The communities found by CE-MOEA are with much larger number of clusters than those found by MOEA-SA and MOGA-@Net on the Polblogs.



(a) (b)

Fig. 6. PF plots of (a) Polbooks network and (b) Polblogs network obtained by CE-MOEA.

To further show the performance of CE-MOEA, the PFs obtained among the runs with the median value D are shown in Fig. 6 for the two political networks. In the figure, the x-axis is the negative modularity, and the y-axis shows the attribute similarity. From the two figures, we found that Q and f_s are indeed conflicting with each other. Further, it is seen that the PFs obtained by CE-MOEA are almost evenly distributed, which could reflect the good performance of CE-MOEA.

2) Results on the Facebook Ego Networks: Experimental results of the seven ego facebook networks with multiattribute and no ground truth are given in Table III. Again, in the columns, the best metric values are marked in bold and the rank sum test results at 5% significance level are shown.

It is seen from Table III that in terms of the average D and E, CE-MOEA always obtain better results than MOEA-SA and MOGA-@Net. The standard deviations of the obtained D values are all less than 0.03, expect for Ego 3980. This clearly shows that CE-MOEA performs well and quite stable on different networks. The hypothesis test suggests that CE-MOEA performs significantly better than MOEA-SA on 5 out of 7 networks in terms of D. On the remaining two networks, CE-MOEA and MOEA-SA perform similarly. Table III also shows that CE-MOEA performs better than MOGA-@Net on all the Ego facebook networks.

On the other hand, all the average E values obtained by CE-MOEA are less than 0.171, while the standard deviations are less than 0.03. According to the hypothesis test, we found that CE-MOEA performs significantly better than MOEA-SA and MOGA-@Net on all Ego facebook networks except Ego 686 for which MOGA-@Net performs similar. Further, we found that CE-MOEA has obtained generally smaller number

TABLE III
STATISTICS OF THE OBTAINED *D* AND *E* METRIC VALUES BY CE-MOEA AND THE COMPARED ALGORITHMS ON EGO FACEBOOK NETWORKS, WHERE *WR* MEANS WILCOXON'S RANKSUM TEST AT THE 5% SIGNIFICANCE LEVEL

Dataset	**Algorithms**	$D_{\max}$	$D_{\min}$	D_{avg}	WR	$E_{\max}$	$E_{\min}$	E_{avg}	WR	k
Ego 0	CE-MOEA	**0.964**	**0.859**	**0.934(0.030)**		**0.071**	**0.040**	**0.051(0.010)**		3-13
	MOEA-SA	0.815	0.632	0.707(0.049)	†	0.146	0.142	0.144(0.001)	†	6-17
	MOGA-@Net	0.944	0.642	0.742(0.132)	†	0.125	0.097	0.111(0.008)	†	32-34
Ego 107	CE-MOEA	**0.946**	**0.930**	**0.940(0.005)**		**0.036**	**0.024**	**0.030(0.003)**		5-17
	MOEA-SA	0.938	0.804	0.917(0.037)	†	0.078	0.076	0.077(0.001)	†	4-29
	MOGA-@Net	0.916	0.909	0.911(0.002)	†	0.065	0.057	0.061(0.002)	†	35-38
Ego 686	CE-MOEA	**0.758**	**0.687**	**0.723(0.021)**		**0.081**	0.067	**0.069(0.003)**		3-5
	MOEA-SA	0.648	0.578	0.621(0.021)	†	0.295	0.271	0.282(0.007)	†	4-11
	MOGA-@Net	0.643	0.435	0.540(0.061)	†	0.084	**0.064**	0.075(0.006)	≈	5-9
Ego 1684	CE-MOEA	0.900	**0.891**	**0.897(0.002)**		**0.031**	**0.023**	**0.026(0.002)**		5-10
	MOEA-SA	**0.926**	0.853	0.888(0.019)	≈	0.091	0.088	0.090(0.001)	†	6-24
	MOGA-@Net	0.864	0.786	0.813(0.028)	†	0.064	0.056	0.061(0.003)	†	26-28
Ego 1912	CE-MOEA	**0.976**	**0.960**	**0.966(0.005)**		**0.031**	**0.021**	**0.026(0.003)**		4-9
	MOEA-SA	0.918	0.785	0.849(0.040)	†	0.090	0.086	0.087(0.001)	†	3-15
	MOGA-@Net	0.952	0.912	0.921(0.016)	†	0.044	0.039	0.043(0.002)	†	18-20
Ego 3437	CE-MOEA	**0.916**	**0.866**	**0.875(0.009)**		**0.065**	**0.045**	**0.055(0.004)**		7-13
	MOEA-SA	0.898	0.806	0.861(0.029)	≈	0.106	0.101	0.103(0.001)	†	11-23
	MOGA-@Net	0.900	0.836	0.852(0.022)	†	0.106	0.094	0.103(0.003)	†	22-25
Ego 3980	CE-MOEA	**0.923**	**0.769**	**0.855(0.048)**		**0.171**	**0.070**	**0.118(0.021)**		3-6
	MOEA-SA	0.669	0.597	0.626(0.021)	†	0.310	0.267	0.284(0.012)	†	6-10
	MOGA-@Net	0.788	0.709	0.752(0.029)	†	0.207	0.185	0.194(0.009)	†	15-17

of communities than MOEA-SA and MOGA-@Net on all the networks, while MOGA-@Net obtains much larger number of communities generally.

The PFs of the ego networks obtained by CE-MOEA in the run with the median *D* value are shown in Fig. 7. Similar to Fig. 6, we find that the two objectives are conflicting with each other. Further, it is found that the PFs are mostly evenly distributed, which reflects a good performance of CE-MOEA.

In summary, we may conclude that CE-MOEA is able to achieve a good balance between network structure and attribute similarity when solving the detection problem of multiattribute networks, and performs better than MOEA-SA and MOGA-@Net.

D. Results on Networks With Ground Truth

In this section, six multiattribute networks (Cornell, Texas, Washington, Wisconsin, Cora, and Citeseer) with true labels are used as the benchmark. Ten state-of-the-art community detection algorithms, including distance-based methods (SA-Cluster [60] and Inc-Cluster [61]); model-based methods (vGraph [62], PCL [58], BAGC [63], SCI [64], TLSC [65], and CDE [66]); and MOEA-based methods (MOEA-SA [28] and MOGA-@Net [30]) are compared with CE-MOEA in terms of NMI. Note vGraph is not applicable for attribute network since it only considers network structure.

The statistics of the NMI metrics, including the mean and standard deviation, obtained by the compared algorithms are shown in Table IV. For each network, the best mean NMI values are typeset in bold. Further, the z-test was carried out to find out the differences between CE-MOEA and the compared algorithms. In Table IV, symbols "+" (respectively, "−" and "=") denotes that the performance of the compared algorithm is significantly better than (respectively, worse than and similar to) CE-MOEA at the 5% significant level.

From Table IV, the z-test suggests that CE-MOEA performs significantly better than vGraph, SA-Cluster, Inc-Cluster, BAGC, TLSC, MOEA-SA, and MOGA-@Net on all the networks. CE-MOEA performs worse than PCL on Cora, worse than SCI on Texas, worse than CDE on Cornell, Texas, Washington and Wisconsin. However, we find that CE-MOEA is ranked the second on Cornell, Washington, Wisconsin and Cora, ranked the third on Texas. CE-MOEA performs similarly to SCI on Wisconsin, to CDE on Cornell and Texas.

Note that except MOEA-SA, MOGA-@Net, and CE-MOEA, all the other compared algorithms require to know the true number of communities as *a priori*. Hence, CE-MOEA is more favorable for practical application.

E. More Experiments

To further investigate the performance of CE-MOEA, we generated three large synthesis networks with single attribute, called LFR3500, LFR3700, and LFR3900, by using the LFR benchmark generator [67]. The number of nodes of these networks are 3500, 3700, and 3900, and each node has an attribute value within [1, 83], [1, 86], and [1, 89], respectively. Further, a subnetwork (id. 629863) of the Twitter network [68] is used for comparison. It has 171 nodes indicating Tweets and 796 edges and its attribute dimension is 578. Since the ground truths of these networks are known, NMI is applied to evaluate the compared methods.

The following four algorithms, namely: 1) MOGA-@Net [30]; 2) MOEA-SA [28]; 3) CDE [66]; and 4) TLSC [65] are compared with CE-MOEA. The mean and standard deviations (in brackets) of the obtained NMI values in 31 runs are summarized in Table V. From the table, we

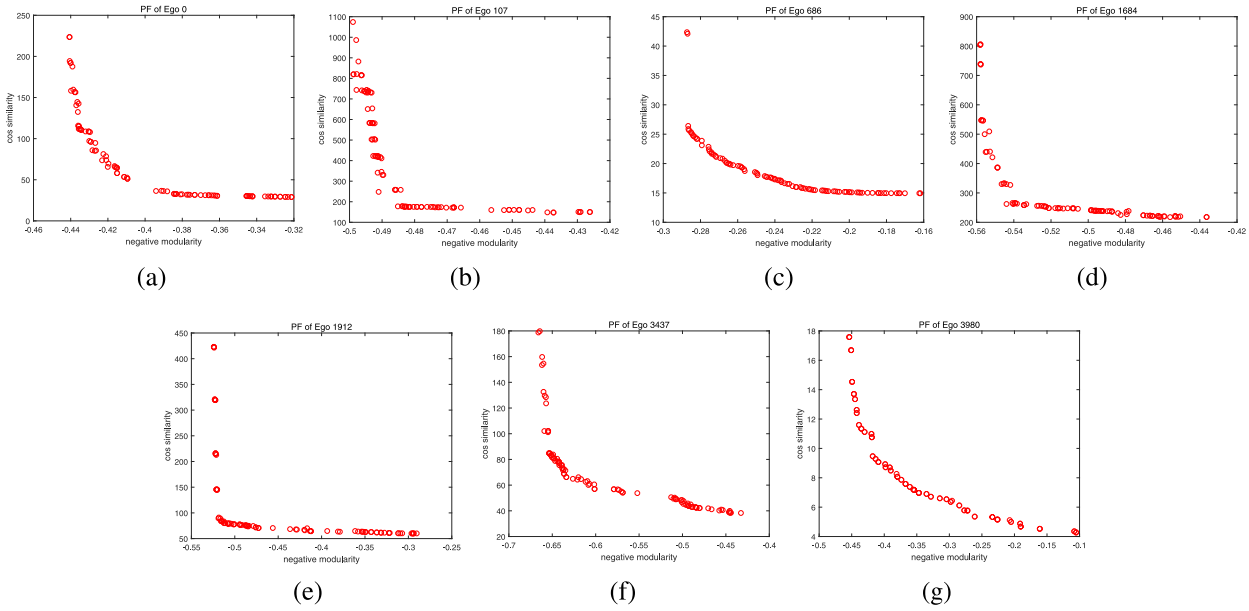


Fig. 7. PF plots of the ego networks obtained by CE-MOEA in the run with median D values. (a) Ego 0. (b) Ego 107. (c) Ego 686. (d) Ego 1684. (e) Ego 1912. (f) Ego 3437. (g) Ego 3980.

TABLE IV
MEAN AND STANDARD DEVIATION OF THE OBTAINED NMI VALUES BY THE COMPARED METHOD AND CE-MOEA ON MULTIATTRIBUTE NETWORKS WITH GROUND TRUTH

Dataset	Methods										
	Distance-based		Model-based						MOEA-based		
	SA-Cluster	Inc-Cluster	vGraph	PCL	BAGC	SCI	TLSC	CDE	MOEA-SA	MOGA-@Net	CE-MOEA
Cornell	0.064$^-$	0.038$^-$	0.039(0.012)$^-$	0.073(0.010)$^-$	0.040(0.006)$^-$	0.166(0.008)$^-$	0.092(0.032)$^-$	**0.203(0.043)**$^=$	0.133(0.015)$^-$	0.172(0.016)$^-$	0.199(0.007)
Texas	0.082$^-$	0.106$^-$	0.024(0.006)$^-$	0.061(0.011)$^-$	0.052(0.007)$^-$	**0.200(0.019)**$^+$	0.122(0.063)$^-$	0.181(0.082)$^=$	0.114(0.013)$^-$	0.091(0.020)$^-$	0.163(0.012)
Washington	0.077$^-$	0.063$^-$	0.020(0.007)$^-$	0.092(0.015)$^-$	0.053(0.006)$^-$	0.146(0.006)$^-$	0.166(0.056)$^-$	**0.276(0.072)**$^+$	0.140(0.013)$^-$	0.167(0.010)$^-$	0.244(0.006)
Wisconsin	0.101$^-$	0.089$^-$	0.043(0.008)$^-$	0.060(0.001)$^-$	0.034(0.015)$^-$	0.184(0.001)$^=$	0.081(0.022)$^-$	**0.264(0.082)**$^+$	0.158(0.015)$^-$	0.173(0.010)$^-$	0.191(0.006)
Cora	0.117$^-$	0.112$^-$	0.080(0.012)$^-$	**0.416(0.003)**$^+$	0.008(0.005)$^-$	0.205(0.008)$^-$	0.248(0.004)$^-$	0.325(0.084)$^-$	0.119(0.002)$^-$	0.379(0.005)$^-$	0.399(0.003)
Citeseer	0.047$^-$	0.043$^-$	0.052(0.012)$^-$	0.170(0.003)$^-$	0.017(0.001)$^-$	0.077(0.037)$^-$	0.150(0.019)$^-$	0.281(0.055)$^-$	0.143(0.003)$^-$	0.331(0.002)$^-$	**0.346(0.001)**

TABLE V
STATISTICS OF THE NMI RESULTS OBTAINED BY CE-MOEA AND THE COMPARED METHODS ON TWITTER AND THREE LARGE LFR NETWORKS

Dataset	CE-MOEA	MOGA-@Net	MOEA-SA	CDE	TLSC
Twitter	**0.473(0.020)**	0.441(0.028)	0.470(0.034)	0.352(0.090)	0.325(0.051)
LFR3500	**0.807(0.002)**	0.802(0.006)	0.798(0.006)	0.630(0.059)	0.541(0.019)
LFR3700	**0.803(0.002)**	0.797(0.008)	0.796(0.009)	0.642(0.057)	0.619(0.023)
LFR3900	**0.809(0.001)**	0.804(0.008)	0.798(0.005)	0.636(0.054)	0.579(0.007)

TABLE VI
BEST E OF THE SIX VERSIONS OF CE-MOEA ON THREE NETWORKS

Dataset	CE-MOEA	-v2	-v3	-v4	-v5	-v6
Polbooks	**0.091**	0.114	0.116	0.116	0.116	0.117
Ego 686	**0.067**	0.070	0.068	0.068	0.068	0.071
Ego 3980	**0.070**	0.113	0.113	**0.070**	0.113	0.182

find that CE-MOEA obtains higher NMI values than those obtained by the compared algorithms in general.

F. Effectiveness of the Proposed Objectives

To verify the difference between the proposed objective function f_s (or f_m) [see (6) and (8)] and S_A as proposed in [28], we replaced S_A with f_s (or f_m) in CE-MOEA, called CE-MOEA-v2.

Further, we investigated the influences of the denominator used in f_s (or f_m). Four variants of the denominator, namely $\sum_{k=1}^{c} r_k$, $\sum_{k=1}^{c} r_k^2$, $\sum_{k=1}^{c}(r_k-1)^2$, and no denominator, are incorporated within the objective, where c is the number of obtained communities and r_k stands for the number of nodes in the kth community. The corresponding CE-MOEAs are called CE-MOEA-v3, CE-MOEA-v4, CE-MOEA-v5, and CE-MOEA-v6, respectively. Three datasets (Polbooks, Ego 686, and Ego 3980) are used as the benchmark.

The best E values are used to measure the objectives' efficacy. Table VI summarizes the results. From the table, we find that CE-MOEA performs the best especially for Polbooks, which indicates that using f_s or f_m are more appropriate than the other objectives.

G. Comparison Between MOEA-SA and CE-MOEA in Terms of Running Time

We compare the average running time of one generation of CE-MOEA and MOEA-SA on three networks (including Polbooks, Ego 686 and Ego 3980). The average running times (in seconds) are summarized in Table VII. From the table, we find that MOEA-SA is faster than CE-MOEA. However, note that MOEA-SA is implemented in C++ and CE-MOEA in MATLAB. The difference in terms of the running time between the two algorithms is not as big as shown in the table. The running time of CE-MOEA should be acceptable for real applications.

TABLE VII
RUNNING TIME COMPARISON BETWEEN CE-MOEA AND MOEA-SA FOR ONE GENERATION (IN SECONDS)

Dataset	CE-MOEA	MOEA-SA
Polbooks	1.005s	0.033s
Ego 686	6.475s	1.348s
Ego 3980	1.425s	0.037s

V. FITNESS LANDSCAPE ANALYSIS

From the above experimental study, we may conclude that the proposed algorithm performs better than existing algorithms for both single-attribute and multiattribute networks with known or unknown ground truth.

Notice that MOEA-SA is also built upon NSGA-II. This makes us to think that maybe the proposed graph neural network encoding is the reason for the better performance of CE-MOEA. Since through graph neural network encoding, the original discrete optimization problem is transformed to a continuous one. We thus further conjecture that due to the continuous encoding, the fitness landscape of the original problem becomes smoother.

In this section, we resort to the fitness landscape analysis to confirm our conjecture. Six networks, including Polbooks, Ego 0, Ego 107, Ego 686, Ego 3437, and Ego 3980, are used as examples to conduct the analysis based on the modularity Q and the attribute similarity f_s or f_m. In our experiments, the ruggedness of the community detection problem landscape is measured by three metrics, including local optimum density (LOD), escaping rate (ER) and fitness distance correlation (FDC) [69]. All these metrics are obtained by applying the iterated local search (ILS) [70] heuristic. The metrics are defined as follows.

1) *LOD:* It is the number of local optima encountered by an ILS per 100 moves. Here, one move indicates that the local search moves from the current solution to a new solution within its neighborhoods.
2) *ER:* This refers to the success rate of the ILS to reach a new local optimum by perturbing the current local optimum.
3) *FDC:* To compute FDC, we have randomly selected 1000 local optima ($\mathbf{x}_{\mathrm{LO}}$) from the set obtained by the ILS and their function values are $f(\mathbf{x}_{\mathrm{LO}})$. Then, the distances of 1000 local optima to the nearest global optimum are calculated as d_{opt}. Overall, the FDC is defined as

$$\mathrm{FDC}(f(\mathbf{x}_{\mathrm{LO}}), d_{\mathrm{opt}}) = \frac{\mathtt{cov}(f(\mathbf{x}_{\mathrm{LO}}, d_{\mathrm{opt}}))}{\sigma(\mathbf{x}_{\mathrm{LO}})\sigma(d_{\mathrm{opt}})} \quad (14)$$

where $\mathtt{cov}(\cdot)$ means the covariance and $\sigma(\cdot)$ means the standard deviation.

It is generally acknowledged that a lower LOD (and ER) or a higher FDC means that a heuristic can find the global optimum easier, which means a smoother landscapes [71].

The ILS performs a local search process and a perturbation process iteratively until the stopping condition is met. In the local search process, it tries to find a better solution in current solution's neighborhood. If such a solution is found, it is used to replace the current solution. The process continues until there is no better solution in the neighborhood. A perturbation is performed once the local search process is stuck.

TABLE VIII
FITNESS LANDSCAPE METRICS OBTAINED FOR THE ORIGINAL PROBLEM AND TRANSFORMED PROBLEM ON THE SELECTED NETWORKS IN TERMS OF THE MODULARITY Q

Dataset	Metrics					
	$\mathtt{LOD}_o$	$\mathtt{LOD}_t$	$\mathtt{ER}_o$	$\mathtt{ER}_t$	$\mathtt{FDC}_o$	$\mathtt{FDC}_t$
Polbooks	4.562	**4.282**	0.519	**0.001**	0.048	**0.133**
Ego 0	3.821	**3.756**	0.561	**0.034**	0.168	**0.189**
Ego 107	4.211	**2.033**	0.590	**0.027**	0.291	**0.326**
Ego 686	4.220	**1.982**	0.535	**0.003**	0.172	**0.195**
Ego 3437	4.290	**2.861**	0.019	**0.009**	0.102	**0.128**
Ego 3980	3.421	**3.235**	0.510	**0.004**	0.023	**0.217**

TABLE IX
FITNESS LANDSCAPE METRICS OBTAINED FOR THE ORIGINAL PROBLEMS AND TRANSFORMED PROBLEMS ON THE SELECTED NETWORKS IN TERMS OF f_s OR f_m

Dataset	Metrics					
	$\mathtt{LOD}_o$	$\mathtt{LOD}_t$	$\mathtt{ER}_o$	$\mathtt{ER}_t$	$\mathtt{FDC}_o$	$\mathtt{FDC}_t$
Polbooks	5.359	**4.217**	0.048	**0.002**	0.067	**0.209**
Ego 0	3.951	**3.789**	0.050	**0.001**	0.147	**0.219**
Ego 107	4.208	**4.142**	0.517	**0.513**	0.136	**0.141**
Ego 686	4.214	**3.929**	0.003	**0.001**	0.141	**0.151**
Ego 3437	4.197	**3.979**	0.005	**0.004**	0.138	**0.158**
Ego 3980	4.206	**3.255**	0.004	**0.002**	0.213	**0.317**

In our study, the fitness landscape analysis is based on the locus-based encoding method for the original problem. The neighborhood is defined as follows. Given two genotypes $\mathbf{x} = (x^1, x^2, \ldots, x^r)$ and $\mathbf{y} = (y^1, y^2, \ldots, y^r)$, the distance between them is defined by

$$\mathrm{dist}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{r} |\mathrm{sgn}(x^i - y^i)| \quad (15)$$

where

$$\mathrm{sgn}(z) = \begin{cases} 1, & z \neq 0 \\ 0, & z = 0. \end{cases} \quad (16)$$

The neighborhood of a genotype $\mathbf{x}$ is thus defined as

$$\mathcal{N}_O(\mathbf{x}) = \{\mathbf{y} | \mathrm{dist}(\mathbf{x}, \mathbf{y}) = 1\}. \quad (17)$$

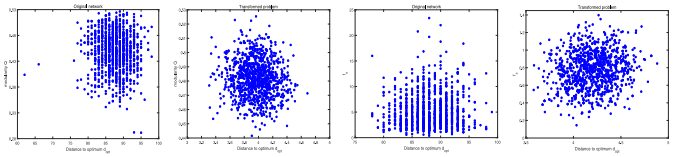
The perturbation process is implemented by replacing ten random edge of the current solution.

For the transformed problem, the neighborhood of a solution $\mathbf{x}$ is defined as

$$\mathcal{N}_T(\mathbf{x}; \epsilon) = \{\mathbf{x}' | \|\mathbf{x} - \mathbf{x}'\|_2 \leq \epsilon\} \quad (18)$$

where ϵ is a threshold. For a certain problem, ϵ is obtained as follows: first, we sample 100 000 solutions randomly. The maximum distance $d_{\max}$ and the minimum distance $d_{\min}$ among the solution pairs are used to compute $\epsilon = (d_{\max} + d_{\min})/2$. To apply the ILS, in the perturbation process, we randomly sample a solution such that its distance to the current solution is greater than ϵ.

To carry out fitness landscape analysis, we first obtain a set of 10 000 local optima by applying the ILS method. Based on the obtained local optima, the fitness landscape metrics are computed, which are shown in Tables VIII and IX for modularity and attribute similarity, respectively. In the tables, $\mathtt{LOD}_o$



(a) (b) (c) (d)

Fig. 8. FDC analysis of the original and transformed problems for the Polbooks network. (a) and (b) are for the modularity Q and (c) and (d) are for the objective f_s. The correlation coefficients between the x-axis and y-axis values are 0.0475, 0.1326, 0.0666, and 0.2091 for plots (a), (b), (c), and (d), respectively.

(respectively, $\mathtt{ER}_o$ and $\mathtt{FDC}_o$) means LOD (respectively, ER and FDC) metric for the original problem and $\mathtt{LOD}_t$ (respectively, $\mathtt{ER}_t$ and $\mathtt{FDC}_t$) is for the transformed problem. The better results of the corresponding metrics are typeset in bold.

From Tables VIII and IX, we find that all the three metrics obtained for the transformed problems are better than those for the original problems on the selected networks. Therefore, we may conclude that the graph network encoding method can smooth the landscape of the community detection problem, which is clearly beneficial to search-based algorithms.

To show the landscape differences, Fig. 8 shows the FDC plots of the original [in subplots (a) and (c)] and transformed problem [in subplots (b) and (d)] on the Polbooks networks. The x-axis is the distance to the optimum, while y-axis is the objective value. The correlation coefficients between the x-axis and y-axis for Q are (a) 0.0475 and (b) 0.1326, for f_s are (c) 0.0666 and (d) 0.2091. It is seen that the coefficients obtained for the transformed problems (b) and (d) are much smaller than those of the original problem (a) and (c), respectively. This indicates that the landscape of the transformed problem is smoother than that of the original problem.

VI. Related Work

This section reviews non-EA-based methods for community detection in attribute network. These methods can be roughly categorized into two groups, namely: 1) distance based and 2) model based.

In the distance-based methods, the distance between nodes considering both network structure and attribute homogeneity is key. In [60], a graph clustering algorithm, called SA-Cluster, was proposed in which a unified distance metric and a neighborhood random walk distance model was used to estimate the vertex closeness. An improved SA-Cluster, called Inc-Cluster, was proposed to incrementally update the random walk distance [61].

The model-based methods are constructed based on modeling the relationship between network structure and node attributes. One promising method, called BAGC, was proposed in [63], in which the community detection for attribute network is modeled under the Bayesian probabilistic framework. Its parameters are estimated by Bayesian inference. A popularity-based conditional link model, called PCL [58], was proposed to model the node's popularity while the model parameters are estimated by maximum-likelihood estimation. In [65] and [72], two probabilistic generative models, called TLSC and BTLSC, respectively, were proposed for topic-related social networks. The generative models are able to distinguish between general and specialized topics. The model parameters are obtained by variational expectation–maximization. In [73], a generative model, called Circles, was proposed using both network structure and node attribute information for ego social networks.

Non-negative matrix factorization (NMF) model was popular for network community detection. In [64], an NMF model, called SCI, was proposed by taking both the community membership and attribute matrices as decisive variables. Through embedding community structure, attribute network community detection was formulated as an NMF optimization problem in CDE [66]. In [74], a method called DCM was proposed by alternating between maximizing the community score and inducing a fitting concise description. In [75], a locally weighted K-means algorithm, called Adapt-SA, was proposed to learn a fusion weight for each node to balance network structure and node attributes. The fuzzy clustering algorithm was also applied to detect the attribute complex network, such as FCAN [76].

Note that all non-EA-based methods mentioned above require the number of communities as *a prior*. On the contrary, our method does not need such information which is of more practical use.

VII. Conclusion

In this article, we proposed a new graph neural network encoding method for complex attribute network community detection problem. Based on the encoding method, the search space of the problem is transformed from discrete to continuous. Our fitness landscape analysis verified that the encoding can smooth the landscape of the original problem for the search-based algorithm.

Based on the novel encoding method, combing with two newly developed objectives for single-attribute and multiattribute similarity, respectively, and the modularity objective for network structure, we developed a MOEA, called CE-MOEA, under the framework of NSGA-II. CE-MOEA was extensively compared against state-of-the-art MOEAs and some well-known non-EA-based detection algorithms on a set of real-life networks with different types and with or without true labels. The experimental results clearly showed that CE-MOEA performed significantly better than MOEA-SA, MOGA-@Net, and those non-EA algorithms in general. Particularly, since MOEA-SA, MOGA-@Net, and CE-MOEA were built upon NSGA-II, the superior performance of CE-MOEA implied that the developed graph neural network encoding is beneficial for the optimization.

In the future, we intend to 1) apply the graph neural network encoding method for overlapping complex attribute network community detection and 2) develop specific neural network encoding for other discrete optimization problems, such as traveling salesman problem and others.

ACKNOWLEDGMENT

The authors would like to thank Prof. J. Liu and Prof. C. Pizzuti for providing the codes of their MOEAs.

REFERENCES

[1] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.

[2] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, 2001.

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

[4] 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, Sep. 2007, Art. no. 036106.

[5] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3, pp. 75–174, 2010.

[6] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. Comput. Appl.*, vol. 108, pp. 87–111, Apr. 2018.

[7] P. Chunaev, "Community detection in node-attributed social networks: A survey," *Comput. Sci. Rev.*, vol. 37, Aug. 2020, Art. no. 100286.

[8] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, Feb. 2004, Art. no. 026113.

[9] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Parallel Problem Solving From Nature – PPSN X*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds. Berlin, Germany: Springer, 2008, pp. 1081–1090.

[10] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, Mar. 2009, Art. no. 033015.

[11] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 464–483, Jun. 2018.

[12] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms," 2007. [Online]. Available: arXiv:0711.0491.

[13] R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Physica A Stat. Mech. Appl.*, vol. 392, no. 5, pp. 1215–1231, 2013.

[14] Z. Li and J. Liu, "A multi-agent genetic algorithm for community detection in complex networks," *Physica A Stat. Mech. Appl.*, vol. 449, pp. 336–347, May 2016.

[15] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, Nov. 2011, Art. no. 056101.

[16] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, and L. Jiao, "Greedy discrete particle swarm optimization for large-scale social network clustering," *Inf. Sci.*, vol. 316, pp. 503–516, Sep. 2015.

[17] C. Pizzuti, "A multi-objective genetic algorithm for community detection in networks," in *Proc. 21st IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2009, pp. 379–386.

[18] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 418–430, Jun. 2012.

[19] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Physica A Stat. Mech. Appl.*, vol. 391, no. 15, pp. 4050–4060, 2012.

[20] M. Gong, X. Chen, L. Ma, Q. Zhang, and L. Jiao, "Identification of multi-resolution network structures with multi-objective immune algorithm," *Appl. Soft Comput.*, vol. 13, no. 4, pp. 1705–1717, 2013.

[21] F. Zou, D. Chen, S. Li, R. Lu, and M. Lin, "Community detection in complex networks: Multi-objective discrete backtracking search optimization algorithm with decomposition," *Appl. Soft Comput.*, vol. 53, pp. 285–295, Apr. 2017.

[22] F. Zou, D. Chen, D.-S. Huang, R. Lu, and X. Wang, "Inverse modelling-based multi-objective evolutionary algorithm with decomposition for community detection in complex networks," *Physica A Stat. Mech. Appl.*, vol. 513, pp. 662–674, Jan. 2019.

[23] C. Shi, Z. Yan, Y. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Appl. Soft Comput.*, vol. 12, no. 2, pp. 850–859, 2012.

[24] P. Wu and L. Pan, "Multi-objective community detection based on memetic algorithm," *PLoS ONE*, vol. 10, May 2015, Art. no. e0126845.

[25] X. Zhang, K. Zhou, H. Pan, L. Zhang, X. Zeng, and Y. Jin, "A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 703–716, Feb. 2020.

[26] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014.

[27] X. Zeng, W. Wang, C. Chen, and G. G. Yen, "A consensus community-based particle swarm optimization for dynamic community detection," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2502–2513, Jun. 2020.

[28] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 1963–1976, Jul. 2018.

[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[30] C. Pizzuti and A. Socievole, "Multiobjective optimization and local merge for clustering attributed graphs," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4997–5009, Dec. 2020.

[31] U. Brandes, M. Gaertler, and D. Wagner, "Engineering graph clustering: Models and experimental evaluation," *ACM J. Exp. Algorithmics*, vol. 12, p. 1, Jun. 2008.

[32] Y. Park and M. Song, "A genetic algorithm for clustering problems," in *Proc. 3rd Annu. Conf. Genet. Program.*, 1998, pp. 568–575.

[33] T. Maehara, N. Marumo, and K. Murota, "Continuous relaxation for discrete DC programming," *Math. Program.*, vol. 169, no. 1, pp. 199–219, 2018.

[34] F. Peng, X. Wang, and Y. Ouyang, "Approximation of discrete spatial data for continuous facility location design," *Integr. Comput.-Aided Eng.*, vol. 21, no. 4, pp. 311–320, 2014.

[35] F. Rothlauf, "Analysis and design of representations for trees," in *Representations for Genetic and Evolutionary Algorithms*. Heidelberg, Germany: Springer, 2006, pp. 141–215.

[36] G. E. Chatzarakis and T. Li, "Oscillation criteria for delay and advanced differential equations with nonmonotone arguments," *Complexity*, vol. 2018, Apr. 2018, Art. no. 8237634.

[37] J. Shen, P. Wang, and X. Wang, "A controlled strengthened dominance relation for evolutionary many-objective optimization," *IEEE Trans. Cybern.*, early access, Sep. 10, 2020, doi: 10.1109/TCYB.2020.3015998.

[38] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[39] S. Jiang, J. Zhang, Y. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.

[40] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[41] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[42] H. Li, Q. Zhang, and J. Deng, "Biased multiobjective optimization and decomposition algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 52–66, Jan. 2017.

[43] J. Sun *et al.*, "Learning from a stream of nonstationary and dependent data in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 541–555, Aug. 2019.

[44] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2758–2770, Jul. 2019.

[45] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Trans. Cybern.*, early access, Apr. 30, 2020, doi: 10.1109/TCYB.2020.2985081.

[46] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[47] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[48] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genet. Algorithms*, 1985, pp. 93–100.
[49] P. M. Pardalos, O. A. Prokopyev, and S. Busygin, "Continuous approaches for solving discrete optimization problems," in *Handbook on Modelling for Discrete Optimization*. Boston, MA, USA: Springer, 2006, pp. 39–60.
[50] Y. Li, T. Tan, and X. Li, "A gradient-based approach for discrete optimum design," *Struct. Multidiscipl. Optim.*, vol. 41, no. 6, pp. 881–892, 2010.
[51] J. Harant, A. Pruchnewski, and M. Voigt, "On dominating sets and independent sets of graphs," *Comb. Probab. Comput.*, vol. 8, no. 6, pp. 547–553, 1999.
[52] C. Yu, K. L. Teo, and Y. Bai, "An exact penalty function method for nonlinear mixed discrete programming problems," *Optim. Lett.*, vol. 7, no. 1, pp. 23–38, 2013.
[53] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, Feb. 2007.
[54] V. Krebs. (2004). *Books About U.S. Politics*. 2004. [Online]. Available: http://www.orgnet.com
[55] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd Int. Workshop Link Disc.*, 2005, pp. 36–43.
[56] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 539–547.
[57] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, p. 93, 2008.
[58] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: a discriminative approach," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2009, pp. 927–936.
[59] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech. Theory Exp.*, vol. 2005, no. 09, Sep. 2005, Art. no. P09008.
[60] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 718–729, Aug. 2009.
[61] Y. Zhou, H. Cheng, and J. X. Yu, "Clustering large attributed graphs: An efficient incremental approach," in *Proc. IEEE Int. Conf. Data Min.*, Dec. 2010, pp. 689–698.
[62] F.-Y. Sun, M. Qu, J. Hoffmann, C.-W. Huang, and J. Tang, "vGraph: A generative model for joint community detection and node representation learning," in *Proc. 32nd Adv. Neural Inf. Process. Syst.*, 2019, pp. 514–524.
[63] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2012, pp. 505–516.
[64] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic community identification in large attribute networks," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 265–271.
[65] G. Zhang *et al.*, "Finding communities with hierarchical semantics by distinguishing general and specialized topics," in *Proc. IJCAI*, Jul. 2018, pp. 3648–3654.
[66] Y. Li, C. Sha, X. Huang, and Y. Zhang, "Community detection in attributed graphs: An embedding approach," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 338–345.
[67] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, Oct. 2008, Art. no. 046110.
[68] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data
[69] P. Merz, "Advanced fitness landscape analysis and the performance of memetic algorithms," *Evol. Comput.*, vol. 12, no. 3, pp. 303–325, 2004.
[70] H. R. Lourenço, O. C. Martin, and T. Stützle, *Iterated Local Search: Framework and Applications*. Boston, MA, USA: Springer, 2010, pp. 363–397.
[71] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 337–352, Nov. 2000.
[72] D. Jin *et al.*, "Detecting communities with multiplex semantics by distinguishing background, general and specialized topics," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2144–2158, Nov. 2020.
[73] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 1, p. 4, Feb. 2014.
[74] S. Pool, F. Bonchi, and M. V. Leeuwen, "Description-driven community detection," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 1–28, Apr. 2014.
[75] Y. Li, C. Jia, X. Kong, L. Yang, and J. Yu, "Locally weighted fusion of structural and attribute information in graph clustering," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 247–260, Jan. 2019.
[76] L. Hu and K. C. C. Chan, "Fuzzy clustering in a complex network based on content relevance and link structures," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 2, pp. 456–470, Apr. 2016.

Jianyong Sun (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Xi'an Jiaotong University, Xi'an, China, in 1997 and 1999, respectively, and the Ph.D. degree in computer science from University of Essex, Colchester, U.K., in 2006.

He is a Full Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University. His research interests include evolutionary computation and optimization, statistical machine learning, and big data and their applications.

Wei Zheng received the M.S. degree from the School of Information Science and Engineering, Shandong Normal University, Jinan, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China.

His research interests include evolutionary algorithms and their applications.

Qingfu Zhang (Fellow, IEEE) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984, and the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi'an, China, in 1991 and 1994, respectively.

He is a Chair Professor of Computational Intelligence with the Department of Computer Science, City University of Hong Kong, Hong Kong. His main research interests include evolutionary computation, optimization, and machine learning.

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. He is a Web of Science Highly Cited Researcher in computer science for four consecutive years from 2016.

Zongben Xu (Member, IEEE) received the Ph.D. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 1987.

He serves as the Chief Scientist of the National Basic Research Program of China (973 Project), and the Director of the Institute for Information and System Sciences with Xi'an Jiaotong University. He served as a Vice President of Xi'an Jiaotong University, Xi'an, China, from 2003 to 2014. He delivered a 45 minutes talk on the International Congress of Mathematicians 2010. His current research interests include intelligent information processing and applied mathematics.

Dr. Xu is the owner of the Tan Kan Kee Science Award in science technology in 2018, the National Natural Science Award of China in 2007, the National Award on Scientific and Technological Advances of China in 2011CSIAM Su Buchin Applied Mathematics Prize in 2008, and the ITIQAM Richard Price Award. He was elected as a member of Chinese Academy of Science in 2011.