Continuous Encoding for Overlapping Community Detection in Attributed Network

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

***Abstract*—Detecting overlapping communities of an attribute network is a ubiquitous yet very difficult task, which can be modeled as a discrete optimization problem. Besides the topological structure of the network, node attributes and node overlapping aggravate the difficulty of community detection significantly. In this article, we propose a novel continuous encoding method to convert the discrete-natured detection problem to a continuous one by associating each edge and node attribute in the network with a continuous variable. Based on the encoding, we propose to solve the converted continuous problem by a multiobjective evolutionary algorithm (MOEA) based on decomposition. To find the overlapping nodes, a heuristic based on double-decoding is proposed, which is only with linear complexity. Furthermore, a postprocess community merging method in consideration of node attributes is developed to enhance the homogeneity of nodes in the detected communities. Various synthetic and real-world networks are used to verify the effectiveness of the proposed approach. The experimental results show that the proposed approach performs significantly better than a variety of evolutionary and nonevolutionary methods on most of the benchmark networks.**

***Index Terms*—Attribute network, continuous encoding method, multiobjective evolutionary algorithm (MOEA), overlapping communities.**

I. Introduction

REAL-WORLD systems such as World Wide Web are usually studied by modeling parts of it as networks [1]. Such study can be dated back to 1930s [2]. Generally speaking, a network is a 2-tuple graph consisting of a set of nodes and edges. As one of the most important research goals, community detection is to find a partition of nodes such that nodes have dense intraconnections within communities (clusters) and sparse interconnections among communities [3]. A variety of techniques from interdisciplinary research areas has been proposed, such as random walks [4], modularity maximization [5], spectral clustering [6], and others. Interested readers can see in a number of recent surveys [2], [7], [8].

Recent studies have been focusing on complex networks, which are of important practical uses, for example, attribute and overlapping networks. For example, in a social network, a user may belong to more than one communities, such as family, friend, colleagues, team, etc., while he/she has a number of attributes, such as age, gender, height, and others. The node attributes and multiple-community memberships (i.e., node overlapping) can make the detection of communities much more challenging than disjoint networks (i.e., networks with no overlapping nodes) [7].

To realize community detection, the primary question is on how to measure the partition quality, that is, on what sense a partition of the nodes is better than another. A variety of metrics has been proposed, including the widely used modularity (denoted as Q) [9] (which is developed in consideration of the network topological structure) and its many extensions [10]. With such metrics, the community detection problem can be modeled as an optimization problem.

A number of optimization methods have been developed and applied for network community detection [11]. In the early days, methods, such as greedy search [5], simulated annealing [12], spectral optimization [13], and genetic algorithms [14], were developed based on the optimization of Q. However, for an attribute and/or overlapping network, considering only its topological structure is not enough as the detected communities could be meaningless w.r.t. the node attribute and overlapping [15].

To address this problem, some metrics have been developed for measuring the partition quality in terms of the node attributes and overlapping. For example, Q is modified to Q_{OV} [16] in consideration of the overlapping community structure. For the node attributes, S_A is developed for measuring the homogeneity of continuous and discrete attribute [17]; $\mathrm{sim}_{\mathrm{COS}}$ (cosine-based similarity) for binary attribute; and $\mathrm{sim}_{\mathrm{ED}}$ (similarity based on Euclidean distance) for single attribute value [18].

Considering both the network topology and node attribute's homogeneity leads to a multiobjective optimization problem (MOP). To this end, as a promising paradigm, a multiobjective evolutionary algorithm (MOEA) has been applied since last decade and has shown its effectiveness for community

Manuscript received 24 May 2021; revised 5 September 2021 and 4 December 2021; accepted 26 February 2022. Date of publication 14 March 2022; date of current version 18 August 2023. This work was supported 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 the Key Project of National Science Foundation of China under Grant 11690011; and in part by the Project of National Science Foundation of China under Grant 61721002. This article was recommended by Associate Editor S. Das. *(Corresponding author: Jianyong Sun.)*

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

Qingfu Zhang is with the Department of Computer Science, The 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.2022.3155646.

Digital Object Identifier 10.1109/TCYB.2022.3155646

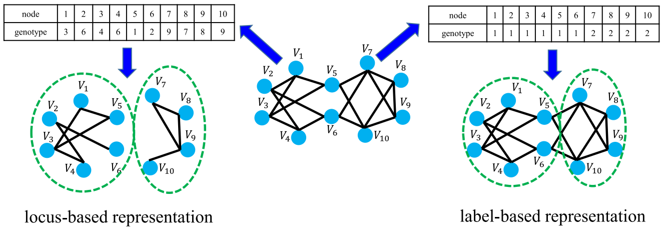

node 1 2 3 4 5 6 7 8 9 10
genotype 3 6 4 6 1 2 9 7 8 9
node 1 2 3 4 5 6 7 8 9 10
genotype 1 1 1 1 1 1 2 2 2 2
V_1
V_2
V_3
V_4
V_5
V_6
V_7
V_8
V_9
V_{10}
locus-based representation
label-based representation

Fig. 1. Locus- and label-based encoding methods for disjoint complex network.

detection in complex networks, such as attribute network with or without overlapping [19].

To apply EA or MOEA-based methods for community detection in complex networks, it is a must to develop an individual encoding (or solution representation) method besides the metrics. Locus-based [20] and label-based [14] representations are two mostly used encoding methods for disjoint network. For locus-based encoding, a node's genotype is one of the connected node number. For label-based encoding, a node's genotype can be an integer from $\{1, \ldots, n\}$, where n is the number of nodes in a network. A simple example of these two representations is shown in Fig. 1, in which an individual is represented as a set of node and genotype pairs.

However, both locus-based and label-based encoding cannot represent the overlapping communities directly. Therefore, for EA-based methods, a suitable encoding method for the overlapping community detection is imperative. Moreover, metrics to measure the network topology and homogeneity regarding node attributes are also very important.

There are some encoding methods developed specifically for the overlapping problem, and techniques for specifying the overlapping nodes. These methods and techniques are always coupled. In the following, we briefly review these methods and techniques.

The first EA developed for the overlapping community detection problem is GA-NET+ [21], in which the individual is encoded based on the conversion of the original network to a line graph where edges represent the adjacent relationships among the original edges. This method has the advantage of maintaining the structure and inherent overlapping characteristic of the original network. By optimizing the community score metric, GA-NET+ obtained a set of communities composed by edges. The node communities can be acquired through conversion, but this acquiring process is time consuming [22]. IMOQPSO [23] was proposed based on this encoding, while two objectives are obtained by separating the modularity Q [9]. In [24], a modified weighted line graph was proposed and an extended version of the modularity density [25] was optimized by a memetic algorithm.

Some EAs use edges to encode individuals. That is, a genotype of an individual component is one of its adjacent edge and the length of the individual is equal to the number of edges. Based on the edge encoding method, an EA can be applied and returns edge communities (i.e., communities composed of edges). The desired communities can then be established through decoding. During decoding, the overlapping nodes can be determined naturally. GaoCD [26] is developed based on the edge encoding method in which the partition density proposed in [27] is optimized. In [28], MOEA-OCD uses the framework of NSGA-II [29] to optimize two objectives, namely: 1) the negative fitness sum and 2) the unfitness sum, based on the edge encoding.

Some EAs mix the label-based encoding. For example, in MR-MOEA [30], a mixed representation scheme and an overlapping node determination method are developed for the overlapping problem. Nodes in a network are considered to be either candidate overlapping nodes or nonoverlapping nodes. For candidate overlapping nodes, a binary variable is used to specify nodes' status; while for nonoverlapping nodes, the label-based encoding method is used. Both in MR-MOEA [30] and MC-MOEA [31], two objectives, that is: 1) kernel k-means (KKM) and 2) ratio cut (RC), are employed. OCD_MOGA [32] employs the same encoding method as in [30] but the objective functions are defined based on the fuzzy memberships of nodes. In MC-MOEA [31], a clique-based representation approach is proposed in which the considered network needs to be transformed to a maximal-clique graph. In an individual, a constructed "clique node" of the maximal-clique graph, which contains some original graph nodes, is assigned to one community label. The label-based encoding method is then applied to find the disjoint communities. Since some original graph nodes might belong to multiple constructed clique node, they are assumed to be the overlapping nodes.

Some EAs embed prior knowledge about the overlapping nodes in the encoding. For example, in MEA_CDPs [33], a solution is encoded as a permutation. When decoding, nodes are added into communities in the permutation order. If a node can increase the community fitness function [34] of one community, it will be added into that community. Since some nodes might increase the fitness function of more than two communities, these nodes are considered to be overlapping.

For attribute complex networks, MOEA-based algorithms have been recently developed and applied. To the best of our knowledge, only three promising MOEAs, that is: 1) MOEA-SA [17]; 2) MOGA-@Net [18]; and 3) CE-MOEA [35], were reported recently, but without considering the overlapping problem. Only one peer-reviewed work, called MOEA-SA_{OV} [36], was proposed for the overlapping community detection in attribute networks.

Among these algorithms, MOEA-SA [17] uses a hybrid network encoding method in which the locus-based encoding method is used to initialize population and the initialization population are then decoded as label-based encoding individuals, which is used for the evolution. MOGA-@Net [18] applies the locus-based encoding method directly. In CE-MOEA [35], a continuous encoding method was proposed. The suitable connective node for each node is selected by performing some nonlinear operations on continuous-encoded individuals.

In MOEA-SA_{OV} [36], a two-part encoding and decoding method was proposed. Specifically, the locus-based encoding is used to initialize population, while the label-based encoding is applied during evolution. To determine the overlapping nodes, an indirect decoding process was proposed in which all nodes in the considered network are greedily searched to optimize the community fitness function [34].

In this article, we propose a novel continuous encoding method for the overlapping community detection problem in attribute networks. Compared with the continuous encoding method developed in CE-MOEA [35], except for the node structure information, the new continuous encoding method contains also the node attribute information.

Based on the new continuous encoding, we build our algorithm upon the framework of MOEA/D [37], which is called as continuous encoding MOEA for overlapping community detection (dubbed as CEMOV). In CEMOV, a double-decoding procedure is proposed to find the overlapping nodes. The proposed procedure is with linear complexity, which is much less than the procedure used in MOEA-SA$_{\mathrm{OV}}$. In addition, based on the features of the developed encoding method, a postprocess community merging method is proposed to enhance the quality of the found communities after evolution.

The remainder of this article is organized as follows. Section II introduces related works about non-EA-based methods for only overlapping, only attribute, and both overlapping and attribute networks. Section III presents the continuous encoding method for attribute network, the objective functions, and the developed algorithm in detail. The experimental results obtained by the developed algorithm and the comparison with some well-known algorithms on synthetic and real-world networks are presented in Section IV. The ablation study of the algorithmic components are summarized in Section V. Finally, Section VI concludes this article.

II. Related Work

Besides aforementioned EA-based methods, there are some related works in multidisciplinary areas for community detection in attribute and/or overlapping networks. In the following, we briefly review these non-EA-based methods.

A. Community Detection in Attribute Networks

Existing non-EA methods for community detection in attribute networks can be categorized into two groups, either distance-based or model-based.

For distance-based methods, they always need to transform the original network to a new network with semantic information incorporated. With various distances defined for the transformed network, many algorithms have been proposed. SA-Cluster [38] is one of the most influential ones, in which a neighborhood random walk distance model is proposed to measure the distance between the nodes in the transformed network. To reduce the complexity of the random walk distance in SA-Cluster, Inc-Cluster was proposed [39].

In model-based methods, models are proposed to establish the relationship between network structure and node attribute. For example, in [40], a Bayesian probabilistic model was established by distinguishing the background, general, and specialized topics of words. Similarly, the Bayesian probabilistic model proposed in [41] differentiates generalized and topical communities, but explores their latent correlations simultaneously to make both two aspects mutually reinforcing.

In [42], a statistical approach is first proposed to measure the associations among node attribute values. Based on this measure, finding interesting subgraphs in the attribute network is modeled as a constrained optimization problem. A weighted K-means algorithm was proposed to cluster the attributed graph in [43]. Other representative methods include a nonnegative matrix factorization (NMF)-based method called semantic community identification (SCI) [44], an embedding-based method [45], and fuzzy method [46].

Recently, methods based on graph convolutional networks (GCNs) [47] have been developed for the community detection problem. A semisupervised community detection algorithm based on GCN was developed in [48], in which GCN is used to learn from known belongings of some nodes and to predict the belongings for the rest nodes. By combining the autoencoder technique [49], GUCD [50] was proposed for unsupervised community detection in attribute networks. Note that both references cannot be used for community detection in overlapping networks. For more about deep learning methods for community detection in attribute complex network, refer to a very recent survey in [51].

B. Overlapping Community Detection in Complex Networks With and Without Node Attributes

Existing non-EA based methods for overlapping network without attributes can be roughly categorized as clique percolation, label propagation, fuzzy detection, and local expansion algorithms. The basic assumption of the clique percolation methods (CPMs) is that some cliques in a network will share nodes naturally. These shared nodes can be seen as the overlapping nodes. CPM [52] is one of such methods. In the label propagation method, the nodes with the same labels will form a community and nodes can spread labels based on some rules. With different rules, several methods were proposed. Among them, the community overlap propagation algorithm (COPRA) [53] and the speaker–listener label propagation algorithm (SLPA) [54] are two mostly promising ones.

Fuzzy detection methods require to compute a belonging factor for each node to decide whether the node belongs to one or more communities. The fuzzy c-means clustering method [55] is usually used. The fuzziness of overlapping nodes has been investigated in [56]. The local expansion algorithms are developed roughly based on growing an inherent or topical community in complex networks. Two representative methods in this category are LFM [34] and MONC [57].

Some probabilistic-based and NMF-based methods have also been developed for community detection in attribute networks. The communities from edge structure and node attributes (CESNA) [58] is the first probabilistic-based method, which builds the links model based on affiliation network models [59] and node attribute model based on a separate logistic model, respectively. The SCI [44] method can be extended to solve the overlapping community detection problems.

III. Method

A. Problem

An attribute complex network can be defined as a 3-tuple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, where $\mathcal{V} = \{V_1, \ldots, V_n\}$ is the node (vertex) set, $\mathcal{E} = \{e_{ij} : 1 \leq i, j \leq n\}$ is the set of edges ($e_{ij} = 1$ means

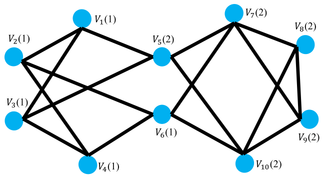


Fig. 2. Example attribute network, where $\mathcal{V} = \{V_1, \ldots, V_{10}\}$. Attribute value "1" means "mathematics" and "2" means "computer science", which are shown in brackets. In this graph, nodes represent scholars, edges represent scholars have cooperative relationship, and the numbers in bracket represent his/her attributes.

an edge links V_i and V_j), and $\mathcal{A} = \{a_{V_1}, a_{V_2}, \ldots, a_{V_n}\}$ is the attribute set, where $a_{V_i}, 1 \leq i \leq n$ is the attribute value of node V_i and can be one or multiple dimensional taking continuous or discrete value, respectively.

The community detection problem is to find a set of c communities $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$ such that: 1) $C_k \neq \varnothing, 1 \leq k \leq c$; 2) any two communities C_{k_1} and C_{k_2} are not identical if $k_1 \neq k_2$; and 3) $\bigcup_{k=1,\ldots,c} C_k = \mathcal{V}$. $C_{k_1} \bigcap C_{k_2} = \varnothing$ for any $k_1 \neq k_2$ refers to a disjoint detection problem; otherwise, it refers to the overlapping community detection problem.

In this article, the goal is to find a set of overlapping communities of an attribute network such that: 1) the edges between communities are as sparse as possible, and edges within communities are as dense as possible; 2) the node attributes in the same community are as similar as possible, while in different communities they are as different as possible; and 3) nodes can belong to different communities.

Fig. 2 shows an example attribute network, which represents a small social network of interdisciplinary scholars. In the attribute network, if only network structure is considered, we might obtain the detected communities as shown in Fig. 1. On the other hand, if only node attribute is considered, the detected communities could be $\{V_1, V_2, V_3, V_4, V_6\}$ and $\{V_5, V_7, V_8, V_9, V_{10}\}$. However, if both network structure and node attributes are considered, the network might be divided as $\{V_1, V_2, V_3, V_4, V_5, V_6\}$ and $\{V_5, V_6, V_7, V_8, V_9, V_{10}\}$, where V_5 and V_6 can be viewed as two overlapping nodes. By taking the node attributes into consideration, the detected communities are more persuasive.

B. Encoding and Decoding Method

In this section, the continuous encoding and double-decoding method are presented.

The pseudocode of the encoding and decoding methods is summarized in Algorithm 1. Its input is a continuous vector $\mathbf{x} \in \mathbb{R}^d$ where d is the number of edges in the network $\mathcal{G}$ plus the number of nodes, that is, $d = \sum_{i,j} e_{ij} + n$. The vector $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ is actually a concatenation of n subvectors. Each subvector $\mathbf{x}_i = (x_{i1}, \ldots, x_{i\tilde{d}_i}, x_{a_i})$ is the continuous vector associated with node V_i, where x_{a_i} is the continuous variable associated with V_i's attribute. The length (dimension) of $\mathbf{x}_i$ is equal to $d_i = \tilde{d}_i + 1$ where $\tilde{d}_i = \sum_j e_{ij}$ is the degree of node V_i. In the sequel, the set of nodes, which links with V_i, is denoted as D_i.

Algorithm 1: Attribute Network Continuous Encoding Method (ANCE)

```
   Input: $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^d$.
   Output: A partition $\mathcal{G}_u$ (and overlapping nodes $\mathcal{P}_O$).
1  Set $\mathcal{L}_1 = \mathcal{L}_2 = \emptyset$;
2  for $i = 1 \to n$ do
3      $\mathbf{h}_i \leftarrow \sigma(\mathbf{x}_i)$;
4      $\mathbf{p}_i \leftarrow \mathrm{softmax}(\mathbf{h}_i)$;
5      $s_{1,i} \leftarrow \arg\max_{1 \leq j \leq \tilde{d}_i}(\mathbf{p}_i)$;
6      $\mathcal{L}_1 \leftarrow \mathcal{L}_1 \bigcup (V_i, V_{s_{1,i}})$;
7      $s_{2,i} \leftarrow \arg\max_{1 \leq j \leq \tilde{d}_i, j \neq s_{1,i}}(\mathbf{p}_i)$;
8      $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \bigcup (V_i, V_{s_{2,i}})$;
9  end
10 $\mathcal{G}_{\mathcal{L}_1} = \{(V_i, \ell_{1,i}), 1 \leq i \leq n\} \leftarrow \mathrm{Decoding}(\mathcal{L}_1)$;
11 $\mathcal{G}_{\mathcal{L}_2} = \{(V_i, \ell_{2,i}), 1 \leq i \leq n\} \leftarrow \mathrm{Decoding}(\mathcal{L}_2)$;
12 Set $\mathcal{G}_u = \emptyset, \mathcal{P}_O = \emptyset$;
13 for $i = 1 \to n$ do
14     if $\ell_{1,i} == \ell_{2,i}$ then
15         $\mathcal{G}_u \leftarrow \mathcal{G}_u \bigcup \{(V_i, \ell_{1,i})\}$;
16     else
17         $\mathcal{P}_O \leftarrow \mathcal{P}_O \bigcup \{V_i\}$;
18         $\mathcal{G}_u \leftarrow \mathcal{G}_u \bigcup \{(V_i, \ell_{1,i}, \ell_{2,i})\}$;
19     end
20 end
21 return $\mathcal{G}_u$ (and $\mathcal{P}_O$).
```

For each node V_i, in case $a_{V_i} \in \mathbb{R}$, its associated x_{a_i} is calculated as follows:

$$x_{a_i} = \frac{a_{V_i} - a_{\min}}{a_{\max} - a_{\min}} \tag{1}$$

where $a_{\min}$ and $a_{\max}$ are the minimum and maximum attribute value of $\mathcal{A}$, respectively.

In case $a_{V_i} \in \mathbb{R}^b, b > 1$ (that is, each node V_i has b attributes), its associated x_{a_i} is computed as follows. We first denote $\mathbf{A} = [a_{ir}]_{n \times b}$ as the attribute matrix. The associated x_{a_i} for each V_i is then defined as

$$x_{a_i} = \frac{\|a_{V_i}\|}{\|\mathbf{A}\|_F} \tag{2}$$

where $\|\cdot\|$ and $\|\cdot\|_F$ mean the ℓ_2-norm and the Frobenius-norm, respectively.

In Algorithm 1, first we initialize $\mathcal{L}_1$ and $\mathcal{L}_2$ to be empty set (line 1). For each node V_i, a sigmoid function σ is applied on its associated continuous encoding vector $\mathbf{x}_i$ element by element (line 3). The σ function is defined as follows:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{3}$$

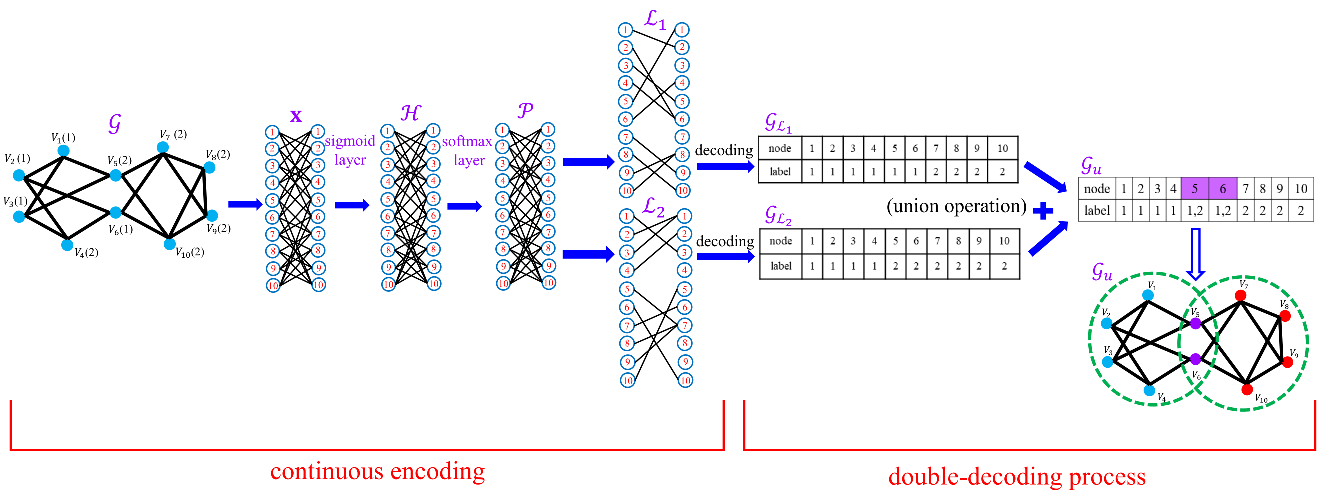
Assume the sigmoid operation returns $\mathbf{h}_i$ for each V_i. A softmax function as shown in (4) is applied on $\mathbf{h}_i$ to obtain a probability $\mathbf{p}_i$ (line 4) where

$$\mathbf{p}_{ij} = \frac{\exp(\mathbf{h}_{ij})}{\sum_j \exp(\mathbf{h}_{ij})}. \tag{4}$$

Based on $\mathbf{p}_i$, we choose index $s_{1,i}$ among D_i such that

$$s_{1,i} = \arg\max_{1 \leq j \leq \tilde{d}_i} \mathbf{p}_{ij} \tag{5}$$

as seen in line 5. $V_{s_{1,i}}$ is then regarded as the decoded genotype of V_i. The node and genotype pair $(V_i, V_{s_{1,i}})$ is saved in $\mathcal{L}_1$



$\mathcal{G}_{\mathcal{L}_1}$

node	1	2	3	4	5	6	7	8	9	10
label	1	1	1	1	1	1	2	2	2	2

$\mathcal{G}_{\mathcal{L}_2}$

node	1	2	3	4	5	6	7	8	9	10
label	1	1	1	1	2	2	2	2	2	2

$\mathcal{G}_u$

node	1	2	3	4	5	6	7	8	9	10
label	1	1	1	1	1,2	1,2	2	2	2	2

Fig. 3. Demo of the continuous encoding and double-decoding method for the overlapping network with attribute.

(line 6). Next, we propose to choose index $s_{2,i}$ within D_i such that

$$s_{2,i} = \arg \max_{1 \le j \le \tilde{d}_i, j \ne s_{1,i}} \mathbf{p}_{ij}. \tag{6}$$

That is, the index of the second largest value of $\mathbf{p}_{ij}$ is chosen. We consider $V_{s_{2,i}}$ to be the decoded genotype to node V_i as well. So the pair $(V_i, V_{s_{2,i}})$ is saved in $\mathcal{L}_2$ (line 8). Note that during the choosing processes [i.e., (5) and (6)], the index of attribute value of $\mathbf{p}_{ij}$ is not allowed to select as the linking node.

By carrying out these procedures for all nodes, we end up with two locus-based solutions

$$\mathcal{L}_1 = \{(V_1, V_{s_{1,1}}), (V_2, V_{s_{1,2}}), \ldots, (V_n, V_{s_{1,n}})\} \tag{7}$$
$$\mathcal{L}_2 = \{(V_1, V_{s_{2,1}}), (V_2, V_{s_{2,2}}), \ldots, (V_n, V_{s_{2,n}})\}. \tag{8}$$

They are decoded to two partitions $\mathcal{G}_{\mathcal{L}_1}$ and $\mathcal{G}_{\mathcal{L}_2}$, respectively (line 10 and 11). Here, we denote $\mathcal{G}_{\mathcal{L}_1} = \{(V_i, \ell_{1,i}), 1 \le i \le n\}$ where V_i is the node, and $\ell_{1,i}$ represents the cluster it belongs to.

From lines 13 to 20, a union operation is carried out over $\mathcal{G}_{\mathcal{L}_1}$ and $\mathcal{G}_{\mathcal{L}_2}$. That is, for each node V_i, we first check whether its associated cluster indices in $\mathcal{G}_{\mathcal{L}_1}$ (i.e., $\ell_{1,i}$) and $\mathcal{G}_{\mathcal{L}_2}$ (i.e., $\ell_{2,i}$) are the same. If they are not the same, the node belongs to different clusters, which means it is an overlapping node. It is stored in $\mathcal{P}_O$ (line 17). Then, for each node, its associated cluster indices in both partitions are combined to form a new pair, either $\{(V_i, \ell_{1,i})\}$ (line 15) or $\{(V_i, \ell_{1,i}, \ell_{2,i})\}$ (line 18). At the end of the encoding–decoding process, the partition $\mathcal{G}_u$ is returned. As a byproduct, the overlapping nodes $\mathcal{P}_O$ can also be returned if necessary.

For each continuous encoding-based individual, the proposed "double-decoding" method requires only two times of the locus-based decoding process. In comparison with existing methods for finding the overlapping nodes, the complexity of the proposed method is much smaller.

A demo of the proposed encoding and decoding method is shown in Fig. 3. From the figure, it can be seen that given an attribute network $\mathcal{G}$ and a continuous vector $\mathbf{x}$, through sigmoid and softmax operations, the connection probabilities between each pair of nodes are obtained. The double-decoding process is performed to obtain two partitions. By combining the two disjoint communities, the overlapping communities is finally secured.

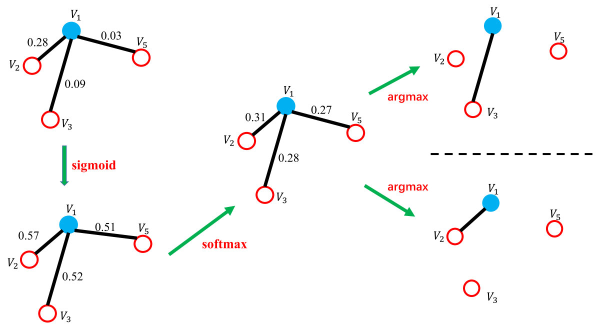


Fig. 4. Demo of the encoding and double-decoding processes for node V_1 of $\mathcal{G}$ of Fig. 3.

To better demonstrate the coding process, we take node V_1 as an example, showing how to decide its decoded genotype in Fig. 4. It is seen that V_1 connects with V_2, V_3, and V_5 in the original graph, and each link is associated with a continuous value. By using the sigmoid and softmax operations, the continuous value for each link is transformed as seen in the middle plot. The argmax functions [i.e., (5) and (6)] are then applied, from which the decoded genotypes of node V_1 are appeared to be (V_1, V_3) and (V_1, V_2). Note that we omit the attribute information associated with V_1 since we do not use it to decide the connecting nodes.

In the sequel, we use the following compact formula to represent the entire decoding process as follows:

$$\mathcal{G}_u = \text{ANCE}(\mathbf{x}, \mathcal{G}). \tag{9}$$

It means that given a continuous encoding-based vector $\mathbf{x}$, associated with the attribute network $\mathcal{G}$, a partition $\mathcal{G}_u$, which is a set of overlapping communities of $\mathcal{G}$, is obtained.

We need to point out that there are two differences between the ANCE and the continuous encoding method developed in [35]. On the one hand, there is an extra component in the continuous vector for each node, which is to account for the attribute of the node. On the other hand, the double-decoding method is used to locate the overlapping nodes.

C. Objectives

In this section, we present the objective functions w.r.t. network topology and node attribute, respectively.

1) Objective for Network Structure: One of the most popular metrics to evaluate the complex network structure is the modularity Q [9]. To measure the overlapping communities better, Shen *et al.* extended Q to Q_{ov} [16], which is defined as follows:

$$Q_{\text{ov}} = \frac{1}{2L}\sum_{k=1}^{c}\sum_{v\in C_k, w\in C_k}\frac{1}{O_v O_w}\left[A_{vw} - \frac{d_v d_w}{2L}\right] \triangleq f_{Q_{\text{ov}}}(\mathcal{G}'; \mathcal{G}) \tag{10}$$

where L is the total number edges of $\mathcal{E}$, c is the number of communities, C_k is the kth community, O_v (resp. O_w) is the number of communities to which node V_v (resp. V_w) belongs to, d_v (resp. d_w) is the degree of node V_v (resp. V_w), and $\mathcal{G}'$ is a partition of $\mathcal{G}$.

In this article, given a continuous encoding based individual $\mathbf{x}$, together with (9), the network structure objective function can be written as follows:

$$f_S = -Q_{\text{ov}} = -f_{Q_{\text{ov}}} \circ \text{ANCE}(\mathbf{x}, \mathcal{G}). \tag{11}$$

2) Objective for Node Similarity: For a single-attribute network with real-valued attributes, the objective proposed in [35] in consideration of the node attribute is adopted in this article, which is defined as follows:

$$f_{\mathcal{A}} = \frac{\mathcal{S}_{\mathcal{A}}}{\sum_{k=1}^{c} n_k(n_k - 1)} \triangleq f_{\mathcal{A}}(\mathcal{G}'; \mathcal{G}) \tag{12}$$

where

$$\mathcal{S}_{\mathcal{A}} = \sum_{k=1}^{c}\sum_{\substack{V_i, V_j \in C_k\\ i<j}}\sqrt{\left(a_{V_i} - a_{V_j}\right)^2}$$

where C_k is the kth community, n_k is the number of nodes within it, and $\mathcal{S}_{\mathcal{A}}$ is the summation of the Euclidean distance of node attributes between all pairs of nodes in each community.

For the multiattribute network with binary attribute values, its objective function is defined as follows [35]:

$$f_{\mathcal{M}} = \frac{\mathcal{S}_{\mathcal{M}}}{\sum_{k=1}^{c} n_k(n_k - 1)} \triangleq f_{\mathcal{M}}(\mathcal{G}'; \mathcal{G}) \tag{13}$$

where

$$\mathcal{S}_{\mathcal{M}} = \sum_{k=1}^{c}\sum_{\substack{V_i, V_j \in C_k\\ i<j}}\frac{a_{V_i} \cdot a_{V_j}}{\left\|a_{V_i}\right\|\left\|a_{V_j}\right\|}.$$

Here, $\mathcal{S}_{\mathcal{M}}$ is the summation of the cosine value between each node pair's attributes within a community. The denominator is the same as in $f_{\mathcal{A}}$.

Correspondingly, in this article, given a continuous vector $\mathbf{x}$, together with (9), the objective function in terms of the node attribute similarity can be defined as $f_A = f_{\mathcal{A}} \circ \text{ANCE}(\mathbf{x}, \mathcal{G})$ or $f_M = f_{\mathcal{M}} \circ \text{ANCE}(\mathbf{x}, \mathcal{G})$ for single-attribute or multiattribute networks, respectively.

In summary, given the continuous encoding, we propose to optimize the following multiobjective problem:

$$\begin{aligned} \text{minimize } & F = (f_S(\mathbf{x}), f_A(\mathbf{x})) \\ \text{or } & F = (f_S(\mathbf{x}), f_M(\mathbf{x})) \\ \text{s.t. } & \mathbf{x} \in \Omega = [-10, 10]^d. \end{aligned} \tag{14}$$

Here, the range of $\mathbf{x}$ is set to $[-10, 10]^d$ for which the nonlinear functions (sigmoid and softmax) are controllable.

D. Algorithm

The overlapping community detection problem is modeled as a continuous multiobjective minimization problem. It is seen that the two objectives in terms of network structure and node attribute similarity are conflicting with each other. Furthermore, the derivatives of the objectives are obviously not available. This motives us to resort MOEA to solving this problem.

MOEA has been considered to be a promising paradigm for MOPs. The primary advantage is that they can find a set of solutions during a single run. Existing MOEAs can be categorized as Pareto dominance based [29], decomposition based [37], [60], performance metric based [61], and learning based [62], [63]. Interested readers can refer to the comprehensive survey in [64].

We first briefly introduce some concepts. For an MOP, the attainable objective set is defined as $\{F(\mathbf{x})|\mathbf{x} \in \Omega\}$, where Ω is decision space (could be continuous or discrete). Given two feasible solutions $\mathbf{x}_A, \mathbf{x}_B \in \Omega$, it is assumed that $\mathbf{x}_A$ dominates $\mathbf{x}_B$ if and only if $\forall i \in \{1, \ldots, m\}, f_i(x_A) \leq f_i(x_B)$ as well as $\exists j \in \{1, \ldots, m\}, f_j(x_A) < f_j(x_B)$, denoted by $\mathbf{x}_A \prec \mathbf{x}_B$. If there is no solution dominates $\mathbf{x}^* \in \Omega$, $\mathbf{x}^*$ is called as a Pareto optimal solution. All Pareto optimal solutions compose the Pareto set (PS) and the image of PS is called as Pareto front (PF). MOEAs aim to find an approximation solution set to the PS.

In this article, we propose to apply the decomposition-based MOEA (MOEA/D) framework for the overlapping community detection problem. MOEA/D [37] was first proposed by Zhang and Li in 2007. Since then, it has been well studied from the perspectives of theoretical analysis and practical applications [65]. In MOEA/D, an MOP is decomposed into a set of scalar single-objective optimization problems. Any EA can be combined within to search for the PS. For example, MOEA/D with differential evolution (DE) [66], terms as MOEA/D-DE [67], has shown its ability for a number of MOPs with complicated PSs.

Our algorithm, CEMOV, is built upon the framework of MOEA/D. In CEMOV, the Tchebycheff approach is applied to decompose the constructed MOP [cf. (14)]. Based on an

```
Algorithm 2: CEMOV
   Input: Maximum generation: $T$; population size: $N$;
          neighborhood size: $t$; a set of uniform weight vectors:
          $\{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N\}$; the DE parameters: $F$ and $CR$; PM
          parameters: $p_m$ and $\eta_m$; and the control parameters: $\delta$
          and $n_r$.
1  Set $g \leftarrow 1$ and randomly initialize $P_g \in [-10, 10]^{N \times d}$;
2  for $i = 1 \rightarrow N$ do
3      Set neighborhood index $B^i \leftarrow \{i_1, \ldots, i_t\}$ for each weight
       vector;
4  end
5  Initialize reference point $\mathbf{z}^*$;
6  for $g = 1 \rightarrow T$ do
7      for $j = 1 \rightarrow N$ do
8          $\mathbf{x^1} \leftarrow P_g(j, :)$;
9          Set mating pool for each individual
10             $$\mathrm{MP}_j \leftarrow \begin{cases} \mathcal{B}^j, & \text{if } rand() \leq \delta, \\ \{1, 2, \ldots, N\}, & \text{otherwise} \end{cases}$$
11         Randomly select $\mathbf{x^2}$ and $\mathbf{x^3}$ from $\mathrm{MP}_j$;
12         $\mathbf{y} \leftarrow \mathrm{DE}(\mathbf{x^1}, \mathbf{x^2}, \mathbf{x^3}, F, CR)$;
13         $\mathbf{y} \leftarrow \mathrm{PM}(\mathbf{y})$;
14         Evaluate $\mathbf{y}$ and update $\mathbf{z}^*$;
15         Set $c_r \leftarrow 0$;
16         for $l = 1 \rightarrow |MP_j|$ do
17             if $g^{tch}(\mathbf{y}|\boldsymbol{\lambda}, z^*) < g^{tch}(\mathbf{x}^l|\boldsymbol{\lambda}, z^*)$ then
18                 $\mathbf{x}^l \leftarrow \mathbf{y}$;
19                 $c_r \leftarrow c_r + 1$;
20                 $\mathrm{MP}_j \leftarrow \mathrm{MP}_j \setminus l$;
21             end
22             if $c_r == n_r \| MP_j == \varnothing$ then
23                 Break;
24             end
25         end
26     end
27     $g = g + 1$;
28 end
```

evenly spread set of weight vectors $\{\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N\}$ where N is the population size, $\boldsymbol{\lambda}^i = (\lambda_1^i, \lambda_2^i)$ such that $\lambda_1^i + \lambda_2^i = 1$ for $i = 1, \ldots, N$, the transformed scalar optimization problem can be expressed as follows:

$$\begin{aligned} \text{minimize } & g^{\text{tch}}\big(\mathbf{x}|\boldsymbol{\lambda}^i, \mathbf{z}^*\big) \\ &= \max \left\{\lambda_1^i |f_S(\mathbf{x}) - z_1^*|, \lambda_2^i |f_A(\mathbf{x}) - z_2^*|\right\} \end{aligned} \tag{15}$$

where $\mathbf{z}^* = (z_1^*, z_2^*)$ is the reference point.

The pseudocode of CEMOV is shown in Algorithm 2. An initial population P_1 is first generated randomly (line 1). From lines 2 to 4, a neighborhood of each individual i is computed through the Euclidean distances of any two pairs of weight vectors. The reference point $\mathbf{z}^*$ is set as the minimum of each objective (line 5).

During the evolution procedure, a mating pool MP_j is first constructed for each individual (lines 9 and 10). Two parent individuals are selected from the pool randomly (line 11), and used by a DE operator to generate new offsprings (line 12). Here, DE varies the current solution $\mathbf{x}_1$ by adding the component difference between two randomly solutions $\mathbf{x}_2$ and $\mathbf{x}_3$, which is represented as

$$\mathbf{y} = \begin{cases} \mathbf{x}_1 + F \times (\mathbf{x}_2 - \mathbf{x}_3), & \text{if rand()} \leq \text{CR} \\ \mathbf{x}_1, & \text{otherwise} \end{cases} \tag{16}$$

where rand() returns a random number between (0, 1), and F and CR are two parameters of the DE operator. The polynomial mutation (PM) [68] operator is then applied on $\mathbf{y}$ (line 13). The reference point $\mathbf{z}^*$ is updated (line 14) right after observing the new offsprings. Individuals are then updated within MP_j according to the Tchebycheff decomposition approach (lines 15–25). The algorithm terminates when the maximum number of generations T has been reached.

```
Algorithm 3: Community Merging Method
   Input: a solution $\mathbf{x}$ and node attribute set $\mathcal{A}$.
1  if $a_{V_i} \in \mathbb{R}$ then
2      $\eta \leftarrow 1 - 1/\sqrt{AM(\mathcal{G})}$;
3  else
4      $\eta \leftarrow \sqrt{AM(\mathcal{G})}$;
5  end
6  $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_c\} \leftarrow \mathrm{ANCE}(\mathbf{x})$;
7  Set $\mathcal{C}_f = \emptyset$;
8  for $i = 1 \rightarrow c - 1$ do
9      if $\mathcal{C}_i == \emptyset$ then
10         Break;
11     end
12     for $j = i + 1 \rightarrow c$ do
13         if $\mathcal{C}_j == \emptyset$ then
14             Break;
15         end
16         if $|AM(\mathcal{C}_i) - AM(\mathcal{C}_j)| < \eta$ then
17             $\mathcal{C}_i \leftarrow \mathcal{C}_i \bigcup \mathcal{C}_j$;
18             $\mathcal{C}_j \leftarrow \varnothing$;
19             $\mathcal{C}_f \leftarrow \mathcal{C}_f \bigcup \mathcal{C}_i$;
20         end
21     end
22 end
23 return $\mathcal{C}_f$.
```

When the algorithm has reached termination, for each obtained solution $\mathbf{x}$, a community merging method (called CMM) is applied to further improve its quality. CMM is summarized in Algorithm 3. Before presenting CMM, in case $a_{V_i} \in \mathbb{R}$, we define the average attribute value over a set of nodes $\mathcal{C} = \{V_1, \ldots, V_{|\mathcal{C}|}\}$ as $\mathrm{AM}(\mathcal{C}) = (1/|\mathcal{C}|) \sum_{i=1}^{|\mathcal{C}|} a_{V_i}$, where $|\mathcal{C}|$ is the cardinality of set $\mathcal{C}$. In case $a_{V_i} \in \mathbb{R}^b$, we first compute an attribute similarity matrix $\mathbf{S^a} = [s_{ij}^a]_{n \times n}$ where

$$s_{ij}^a = 1 - \frac{\langle \mathbf{A}(i, :), \mathbf{A}(j, :) \rangle}{\|\mathbf{A}(i, :)\| \|\mathbf{A}(j, :)\|} \tag{17}$$

and $\langle \cdot, \cdot \rangle$ is the inner product. It is seen that if $i \neq j$, $s_{ij}^a = s_{ji}^a$; otherwise, $s_{ii}^a = 0$. The value of AM for a set of nodes $\mathcal{C}$ is defined as $\mathrm{AM}(\mathcal{C}) = (1/2|\mathcal{C}|) \sum_i^{|\mathcal{C}|} \sum_j^{|\mathcal{C}|} s_{ij}^a$.

In Algorithm 3, a control threshold η is calculated from lines 1 to 5 for single-attribute or multiattribute networks, respectively. By applying Algorithm 1 over $\mathbf{x}$, a set of communities is obtained (line 6). Among these communities, any two communities will be merged if the difference between their AM values is smaller than η (lines 8–23). The algorithm returns a new set of communities. After the CMM operation, the most similar communities in terms of node attributes shall be merged. As a result, the returned set of communities $\mathcal{C}_f$ will be more dense. We will justify the effectiveness of the CMM operator in Section V-B.

E. Complexity Analysis

Let n be the number of nodes, m be the number of objectives, L be the number of edges, N be the population size, t be the neighborhood size, c be the maximum number of communities, and T be the maximum number of generation.

In Algorithm 1, the encoding process needs a time complexity of $\mathcal{O}(L+n)$; the double-decoding process needs $\mathcal{O}(n)$, which is the same as the locus-based decoding process [69]. The total complexity of Algorithm 1 is thus $\mathcal{O}(L + n)$. For Algorithm 2, the population initialization process needs $\mathcal{O}((L+n)N)$. The complexity of CEMOV at each generation is the same as the complexity of MOEA/D [37], which is $\mathcal{O}(mtN)$. In addition, CEMOV needs $\mathcal{O}(nN)$ for the decoding processes and $\mathcal{O}(L + n)$ for calculating the objective functions. For Algorithm 3, the worst case complexity is $\mathcal{O}(c^2)$ when all the communities do not merge at all; while the best case complexity is $\mathcal{O}(c)$ when all the communities are merged into one community in a single run. Note that $c \ll n \ll L$ in general. Therefore, the overall complexity of CEMOV is $\mathcal{O}(mtNT + (L+n)NT)$.

IV. Experiments

In this section, experimental results obtained by applying the proposed algorithm on synthetic benchmark networks and real-world networks are presented.

A. Benchmark Networks

In the considered networks, ten artificial networks are generated by the Lancichinetti–Fortunato–Radicchi (LFR) benchmark generator [70]. To generate a network, the LFR requires ten essential parameters, including: 1) the number of nodes n; 2) the mixing parameter for specifying the network topology μ; 3) the number of memberships of the overlapping nodes O_m; 4) the average degree k; 5) the maximum degree max$k$; 6) the minimum community size min$c$; 7) the maximum community size max$c$; 8) the number of overlapping nodes O_n; 9) the negative exponent for the degree sequence t_1; and 10) the negative exponent for the community size distribution t_2.

To generate the benchmark networks, we keep nine of the parameters the same, including $n = 1000$, $k = 5$, max$k = 25$, min$c = 20$, max$c = 80$, $O_n = 30$, $O_m = 2$, $t_1 = 2$, and $t_2 = 1$, while the mixing parameter μ ranges from 0.1 to 1.0 with step size 0.1. We name the generated networks as LFR1 to LFR10 in the sequel, respectively.

Furthermore, 15 widely used real-world networks, including Polbooks [71], Polblogs [72], American College football [1] (called "Football" for the sake of simplicity in the sequel), Twitter [73], Texas, Cornell, Washington, Wisconsin [74], PubMed [75], Ego 0, Ego 686, Ego 1684, Ego 1912, Ego 3437, and Ego 3980 are employed. The basic information of the 15 networks is summarized in Table I.

TABLE I
CHARACTERISTICS OF THE 15 REAL-WORLD NETWORKS

Network	n	L	No. of Attributes	with ground truth	Network	n	L	No. of Attributes	with ground truth
Polbooks	105	441	1	Yes	PubMed	19717	44338	1	No
Polblogs	1490	19025	1	Yes	Ego 0	347	2519	224	No
Football	115	613	1	Yes	Ego 686	170	1656	63	No
Twitter	171	796	1	No	Ego 1684	776	13826	319	No
Texas	187	328	1	No	Ego 1912	748	29552	480	No
Cornell	195	304	1	No	Ego 3437	542	4749	262	No
Washington	230	446	1	No	Ego 3980	58	143	42	No
Wisconsin	187	328	1	No					

Polbooks was organized and published in 2004. Its nodes represent books sold by Amazon.com during 2004 and edges represent both two books were bought by the same consumer. Each node has an attribute value of three optional values: 1) conservative; 2) liberal; and 3) neutrality. In Polblogs, the nodes are blogs and edges represent the hyperlinks between two blogs. Each node owns an attribute value of two alternatives: 1) liberal and 2) conservative. Football is a complex social network created based on the American college football league. A node represents a football team and an edge means two teams have a game between them. There are 12 attribute values representing which league each team belongs to. The Twitter network is a subnetwork (id. 629863) of Twitter. Nodes are the tweets and edges mean they are connected. Each tweet has an attribute value of seven values. Texas, Cornell, Washington, and Wisconsin are selected from the WebKB networks. They are Webpages and hyperlinks data chosen from four corresponding American universities. Each Webpage has an attribute value of five values. PubMed is a citation network on literatures regarding diabetes. Node (resp. edge) refers to as publication (resp. reference relationship). Each node has an attribute, which is used to represent their publication type (either "Diabetes Mellitus Experimental," "Diabetes Mellitus Type 1," or "Diabetes Mellitus Type 2." Ego 0, Ego 686, Ego 1684, Ego 1912, Ego 3437, and Ego 3980 are six friendship networks chosen from ten Ego facebook networks [76], consisting of 4039 users. The attribute dimension of these networks ranges from 42 to 480.

B. Compared Algorithms and Experimental Settings

For non-EA-based methods, three well-known methods, including CFinder [77], SLPA [54], and CESNA [58], are used to compare with the proposed algorithm. These methods are briefly summarized as follows.

CFinder: It is the implementation of a k-clique-community finding algorithm [77], which belongs to the CPM [52]. It is found that k taken value from 3 to 6 can achieve good results empirically. In our experiments, all $k \in \{3, 4, 5, 6\}$ are tried and the best results obtained are reported.

SLPA: It is a representative label propagation algorithm. SLPA has a postprocess in which a threshold $r \in [0, 1]$ is set to control the probability selection of each node. For all the networks, r varies from 0.01 to 0.1 in the step of 0.01 and the maximum number of iterations is set as 100 according to the original reference [54].

CESNA: It is a non-EA-based method for overlapping community detection in attribute networks by modeling the network structure and node attributes in probability. In our experiments, as suggested in [58], the locally minimal neighborhoods method is used to initialize community memberships F. The two hyperparameters of CESNA are set as

$\alpha = 0.5$ and $\lambda = 1$, which is the same as in the original reference.

Five EA-based methods, including GaoCD [26], IMOQPSO [23], MC-MOEA [31], MR-MOEA [30], and MOEA-SA$_{\mathrm{OV}}$ [36], are compared with CEMOV in this article. Among these methods, only MOEA-SA$_{\mathrm{OV}}$ was proposed mainly for detecting overlapping networks with *single-attributed* node.

In the comparison study, the parameter settings of the compared algorithms are set the same as in the original references. The parameter settings of CEMOV are set as follows: $N = 100$, $T = 100$, $\delta = 0.9$, $t = 10$, $n_r = 2$, $F = 0.7$, $\mathrm{CR} = 0.4$, $p_m = 0.01$, and $\eta_m = 20$. CEMOV is implemented in MATLAB and run on a personal computer for 31 times. In the following, the experimental results are summarized in tables. In the tables, the best results are typeset in bold and Wilconxon's rank sum test at a significance level of 5% is applied to reveal the statistical significance between the compared algorithms. Furthermore, we use symbols ‡, §, and † to indicate that CEMOV performs better than, equal to, and worse than the compared methods, respectively. In the tables, "NA" means that the algorithm cannot return the results for the corresponding network in an acceptable time.

C. Performance Metrics

Three popular metrics, including density (denoted as D), entropy (denoted as E) and normalized mutual information (denoted as NMI) [34], are used to measure the performances of the compared algorithms.

Density D: It is the percentage of the number of edges in the obtained communities to the total number of edges of the network $\mathcal{G}$. It is defined as follows:

$$D = \sum_{k=1}^{c} \frac{L_k}{L} \tag{18}$$

where c is the number of communities, L_k is the number of edges within the community C_k, and L is the total number of edges of network $\mathcal{G}$. A larger value of D means that the detected communities have a more dense community structure.

Entropy E: It is defined as follows:

$$E = \sum_{k=1}^{c} \frac{n_k}{n} \cdot H(k)$$
$$H(k) = -\sum_{a \in \mathcal{A}} p_{ak} \log(p_{ak}) \tag{19}$$

where n_k is the number of nodes in community C_k, and n is the number of nodes in network $\mathcal{G}$. $H(k)$ is the entropy for each community C_k, in which p_{ak} is the percentage of nodes who has attribute value a in C_k. A smaller value of E means that attributes in the detected communities are more homogeneous.

NMI: The NMI metric measures the deviation of the detected communities (denoted as B) and the true communities (denoted as A). It is defined as follows:

$$\mathrm{NMI}(A, B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} I_{ij}\log\big(n \cdot I_{ij}/\big(I_{i\cdot}I_{\cdot j}\big)\big)}{\sum_{i=1}^{c_A} I_{i\cdot}\log(I_{i\cdot}/n) + \sum_{j=1}^{c_B} I_{\cdot j}\log\big(I_{\cdot j}/n\big)} \tag{20}$$

where n is the number of nodes in the network, c_A (resp. c_B) is the number of communities in partition A (resp. B), and I is a confusion matrix of which element I_{ij} means the number of nodes in community C_i of partition A that are also in community C_j of partition B and $I_{i\cdot}$ (resp. $I_{\cdot j}$) means the sum of elements of row i (resp. column j) of I. A larger value of NMI means that the detected community structure is more similar to the true community structure.

TABLE II
EXPERIMENTAL RESULTS OBTAINED BY THE COMPARED ALGORITHMS ON TEN SYNTHETIC NETWORKS IN TERMS OF NMI

Networks	Methods				
	SLPA	CFinder	CESNA	MOEA-SA$_{\mathrm{OV}}$	CEMOV
LFR1	0.568(0.027)‡	0.305‡	0.352‡	0.484(0.015)‡	**0.650(0.008)**
LFR2	0.405(0.026)‡	0.239‡	0.364‡	0.321(0.031)‡	**0.545(0.006)**
LFR3	0.198(0.026)‡	0.116‡	0.211‡	0.247(0.009)‡	**0.495(0.005)**
LFR4	0.044(0.005)‡	0.027‡	0.160‡	0.235(0.003)§	**0.237(0.011)**
LFR5	0.003(0.002)‡	0‡	0.332‡	0.233(0.002)‡	**0.423(0.011)**
LFR6	0.003(0.001)‡	0‡	0.213‡	0.237(0.001)‡	**0.381(0.008)**
LFR7	0.002(0.001)‡	0‡	0.159‡	0.188(0.001)‡	**0.334(0.011)**
LFR8	0.001(0.001)‡	0‡	0.130‡	0.197(0.002)‡	**0.312(0.018)**
LFR9	0.001(0.001)‡	0‡	0.141‡	0.222(0.003)‡	**0.323(0.013)**
LFR10	0‡	0‡	0.135‡	0.210(0.001)‡	**0.321(0.015)**
‡/§/†	10/0/0	10/0/0	10/0/0	9/1/0	

D. Experiment Results on Synthetic Networks

For the ten LFR synthetic networks, the NMI metric is used as the performance metric since the ground truth is available. The compared algorithms are SLPA, CFinder, CESNA, and MOEA-SA$_{\mathrm{OV}}$. The final results are summarized in Table II, in which the mean NMI values with standard deviations obtained over 31 runs for each network are reported. Notice that for CFinder, we report the best results obtained among its different hyperparameter k settings. Hence, the results have no standard deviations.

For the LFR benchmark networks, a larger μ value means that it is more difficult to detect. We can find that along the increasing of μ, all the compared methods show declined performances. The deteriorating tendency is clearly consistent to the difficulty level of the problems. Particularly, it is seen that the performances of SLPA and CFinder deteriorate dramatically from LFR1 to LFR10.

It is seen that the performance of CFinder is the worst among the compared algorithms. SLPA performs better than CFinder on above all the problems, but worse than CESNA on all the problems except LFR1 and LFR2.

Regarding the performances of the rest compared algorithms, MOEA-SA$_{\mathrm{OV}}$ achieves better results than SLPA, CFinder and CESNA in general. This indicates that the EA-based methods are very promising for the overlapping community detection problem in attribute networks. Compared with other methods, CEMOV obtains the best results on all networks correspondingly, which indicates that CEMOV has a better performance in total. Furthermore, Wilconxon's rank sum test results also suggest that CEMOV performs significantly better than SLPA, CFinder, CESNA on all networks, better than MOEA-SA$_{\mathrm{OV}}$ on nine out of ten networks.

TABLE III
EXPERIMENTAL RESULTS OBTAINED BY THE COMPARED ALGORITHMS ON THE NINE REAL-WORLD NETWORKS IN TERMS OF D AND E

Networks	D					E				
	SLPA	CFinder	CESNA	MOEA-SA$_{\mathrm{OV}}$	CEMOV	SLPA	CFinder	CESNA	MOEA-SA$_{\mathrm{OV}}$	CEMOV
Polbooks	1.232(0.111)‡	0.984‡	0.712‡	1.427(0.158)§	**1.446(0.057)**	0.803(0.056)‡	0.449‡	0.659‡	0.583(0.065)‡	**0.314(0.059)**
Football	**1.004(0.070)**†	0.584‡	0.684‡	0.933(0.021)§	0.948(0.065)	2.145(0.053)‡	3.093‡	2.424‡	2.315(0.243)§	**2.099(0.314)**
Polblogs	0.942(0.016)‡	0.883‡	0.918‡	NA‡	**1.089(0.092)**	0.058(0.001)§	**0.027**†	0.061§	NA‡	0.062(0.011)
Texas	0.939(0.017)‡	0.438‡	0.605‡	0.884(0.068)‡	**1.156(0.024)**	0.636(0.029)‡	0.607‡	0.640‡	0.834(0.017)‡	**0.554(0.064)**
Cornell	0.887(0.013)‡	0.399‡	0.565‡	0.797(0.067)‡	**1.290(0.062)**	0.621(0.018)‡	0.614‡	0.617‡	2.208(0.065)‡	**0.582(0.076)**
Washington	0.885(0.017)‡	0.503‡	0.806‡	0.946(0.115)‡	**1.137(0.075)**	0.709(0.039)†	**0.551**†	0.816†	2.141(0.416)‡	0.872(0.097)
Wisconsin	1.256(0.120)§	0.453‡	0.688‡	0.642(0.063)‡	**1.270(0.032)**	0.893(0.070)‡	0.901‡	0.920‡	1.093(0.144)‡	**0.829(0.066)**
Twitter	0.922(0.053)‡	0.952‡	0.783‡	0.981(0.008)§	**0.995(0.048)**	1.654(0.059)‡	1.422‡	1.508‡	1.399(0.046)‡	**1.218(0.243)**
PubMed	0.851(0.066)‡	0.409‡	0.945‡	NA‡	**1.083(0.018)**	0.468(0.017)‡	**0.122**†	0.313‡	NA‡	0.283(0.006)
‡/§/†	7/1/1	9/0/0	9/0/0	6/3/0		7/1/1	6/0/3	7/1/1	8/1/0	

TABLE IV
NUMBER OF CLUSTERS OBTAINED BY MOEA-SA$_{\mathrm{OV}}$ AND CEMOV

Network	MOEA-SA$_{\mathrm{OV}}$	CEMOV
Polbooks	3-8	5-11
Football	2-11	5-13
Polblogs	NA	13-29
Texas	2-16	12-21
Cornell	8-26	16-22
Washington	2-34	17-30
Wisconsin	5-21	19-35
Twitter	2-8	7-14
PubMed	NA	794-861

E. Experiment Results on Single-Attribute Real-World Networks

For the nine single-attribute real-world attributed networks, SLPA, CFinder, CESNA, and MOEA-SA$_{\mathrm{OV}}$ are used to compare with CEMOV in terms of D and E. SLPA, CFinder, CESNA, IMOQPSO, GaoCD, MC-MOEA, MR-MOEA, and MOEA-SA$_{\mathrm{OV}}$ are the compared algorithms on networks with ground truth in terms of NMI.

1) Experimental Results in Terms of D and E: The experimental results are summarized in Table III, in which the mean values (with standard deviations) of D and E for each network are reported. Since CEMOV and MOEA-SA$_{\mathrm{OV}}$ can also give the number of the detected communities, we list the obtained numbers of communities in Table IV.

From the two tables, we find that CEMOV performs better than the compared algorithms on almost all networks. Particularly, in terms of D, it is seen from Table III that CEMOV obtains most of the best mean values except on Football for which SLPA performs better. Furthermore, CEMOV performs significantly better than CFinder and CESNA on all the nine networks, better than MOEA-SA$_{\mathrm{OV}}$ on six networks and SLPA on seven networks according to Wilconxon's rank sum test.

In terms of E, CEMOV obtains six best mean values on the nine networks, respectively. CEMOV performs significantly better than MOEA-SA$_{\mathrm{OV}}$ on eight networks, better than SLPA and CESNA on seven networks, and better than CFinder on six networks according to Wilconxon's rank sum test, respectively. In addition, it is seen from Table IV that compared with MOEA-SA$_{\mathrm{OV}}$, CEMOV always achieves a larger number of communities but the range of the achieved communities is smaller.

TABLE V
EXPERIMENTAL RESULTS OBTAINED BY THE COMPARED METHODS ON THREE REAL-WORLD NETWORKS WITH GROUND TRUTH IN TERMS OF NMI

Networks	Methods								
	SLPA	CFinder	CESNA	IMOQPSO	GaoCD	MC-MOEA	MR-MOEA	MOEA-SA$_{\mathrm{OV}}$	CEMOV
Polbooks	0.194(0.011)‡	0.323‡	0.235‡	0.444(0.023)‡	0.152(0.012)‡	0.271(0.073)‡	0.344(0.003)‡	0.453(0.017)‡	**0.576(0.019)**
Football	0.430(0.016)‡	0.571‡	0.624‡	0.358(0.015)‡	0.619(0.058)§	0.536(0.012)‡	0.667(0.009)§	0.586(0.063)‡	**0.689(0.016)**
Polblogs	0.287(0.017)‡	0.047‡	0.147‡	0.046(0.016)‡	NA‡	0.121(0.005)‡	0.172(0.001)‡	NA‡	**0.320(0.026)**
‡/§/†	3/0/0	3/0/0	3/0/0	3/0/0	2/1/0	3/0/0	2/1/0	3/0/0	

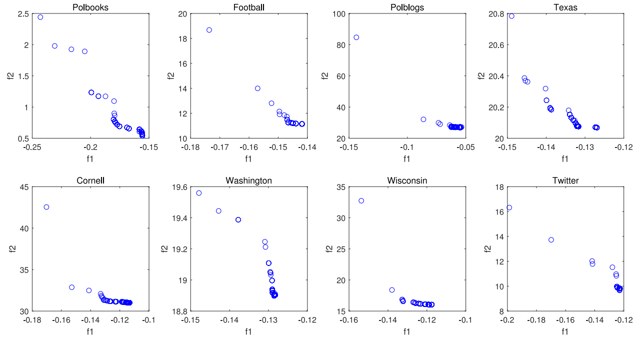


Fig. 5. PF plots of eight real-world networks.

2) Experimental Results in Terms of NMI: Three attributed networks with ground truth, namely, Polbooks, Football, and Polblogs, are used in the comparison. The mean NMI values with standard deviations obtained by the compared algorithms are summarized in Table V. From the table, we can find that CEMOV obtains the best means values over the eight compared algorithms. Especially, CEMOV performs significantly better than MOEA-SA$_{\mathrm{OV}}$ on all the networks. Note that CEMOV and MOEA-SA$_{\mathrm{OV}}$ differ mainly on the use of encoding. This implies that the continuous encoding method is beneficial for improving the performance of MOEA.

CEMOV is a MOEA-based method, which implies that its performance can be visualized by the found approximated PFs. The representative PFs of CEMOV for eight networks are shown in Fig. 5, in which f_1 (resp. f_2) means negative modularity (resp. attribute similarity). From these figures, we find that the PFs obtained by CEMOV are mostly evenly distributed, which means that the performance of CEMOV on these networks is generally good.

TABLE VI
EXPERIMENTAL RESULTS OBTAINED BY CEMOV AND MOEA-SA$_{\text{OV}}$ ON SIX EGO FACEBOOK NETWORKS IN TERMS OF D AND E

Networks	D				E			
	CEMOV		MOEA-SA$_{\text{OV}}$		CEMOV		MOEA-SA$_{\text{OV}}$	
	best	mean(std)	best	mean(std)	best	mean(std)	best	mean(std)
Ego 0	**1.688**	**1.133(0.242)**	0.933	0.919(0.019)	**0.066**	**0.084(0.014)**	0.132	0.134(0.001)
Ego 686	**1.551**	**1.071(0.163)**	0.153	0.152(0.001)	**0.030**	**0.043(0.014)**	0.277	0.278(0.001)
Ego 1684	**1.141**	**1.039(0.056)**	0.181	0.174(0.004)	**0.027**	**0.040(0.010)**	0.083	0.084(0.001)
Ego 1912	**1.184**	**1.029(0.067)**	0.931	0.874(0.032)	**0.025**	**0.033(0.005)**	0.081	0.087(0.002)
Ego 3437	**1.347**	**1.231(0.091)**	0.956	0.865(0.021)	**0.076**	**0.084(0.006)**	0.097	0.100(0.001)
Ego 3980	**1.301**	**1.191(0.073)**	0.986	0.665(0.133)	**0.108**	**0.112(0.010)**	0.281	0.325(0.018)

TABLE VII
EXPERIMENTAL RESULTS OBTAINED BY THE COMPARED ALGORITHMS ON THREE LARGE SYNTHETIC NETWORKS IN TERMS OF NMI

Networks	Methods				
	SLPA	CFinder	CESNA	MOEA-SA$_{\text{OV}}$	CEMOV
LFR3K	0.646(0.011)	0.354	0.693	NA	**0.719(0.003)**
LFR4K	0.603(0.007)	0.308	0.665	NA	**0.724(0.002)**
LFR5K	0.574(0.015)	0.277	0.642	NA	**0.727(0.003)**

F. Experiment Results on Multiattribute Real-World Networks

For the six Ego facebook networks, MOEA-SA$_{\text{OV}}$ is used to compare CEMOV. Note that when employing MOEA-SA$_{\text{OV}}$ for multiattribute networks, the objectives used in [36] are applied. Again D and E are used as the comparison metrics.

Table VI shows the final comparison results where the best and average (with standard deviation) values are reported. From the results, we can observe that CEMOV has better performance than MOEA-SA$_{\text{OV}}$ on all tested multiattribute networks in both metrics D and E significantly.

G. More Experiments

Three LFR networks with 3000, 4000, and 5000 nodes are generated and tested. We name these networks as LFR3K, LFR4K, and LFR5K, respectively. The essential parameters of these networks are the same as the benchmarks LFR1–LFR10 with the mixing parameter $\mu = 0.1$. Four algorithms, including SLPA, CFinder, CESNA, and MOEA-SA$_{\text{OV}}$, are compared with CEMOV. All the compared methods are executed for 31 times. The obtained NMI results are summarized in Table VII. From the table, we find that CEMOV achieves better results than those obtained by the compared algorithms.

The aforementioned LFR networks are essentially with no attributes when generated. We further generate ten benchmark networks with attributes (called LNet1-LNet10) using the method proposed in [78]. The parameter setting for these networks are the same as those for LFR1–LFR10, except the number of nodes is 500.

The performance of CEMOV on these networks is compared with MOEA-SA$_{\text{OV}}$. Table VIII summarizes the comparison results obtained by the compared algorithms over 31 runs. From the results, it can be found that CEMOV performs significantly better than MOEA-SA$_{\text{OV}}$ on all the networks.

Furthermore, two representative networks: 1) Polbooks and 2) Football, are used to investigate how long the running time required for the compared methods. The running times in seconds are shown in Table IX. From the table, we see that the running time of MOEA-SA$_{\text{OV}}$ requires the longest time, while CEMOV requires the third longest time among the compared algorithms. Though CEMOV requires quite long time, it has significantly better performance than the compared algorithms. Especially, CEMOV takes only one-tenth time less than MOEA-SA$_{\text{OV}}$ but achieves better detection results.

TABLE VIII
EXPERIMENTAL RESULTS OBTAINED BY CEMOV AND MOEA-SA$_{\text{OV}}$ ON LNET1-LNET10 IN TERMS OF NMI

Methods	LNet1	LNet2	LNet3	LNet4	LNet5
MOEA-SA$_{\text{OV}}$	0.642(0.001)	0.469(0.011)	0.260(0.007)	0.334(0.008)	0.273(0.010)
CEMOV	**0.649(0.013)**	**0.474(0.012)**	**0.438(0.016)**	**0.377(0.010)**	**0.335(0.013)**
Methods	LNet6	LNet7	LNet8	LNet9	LNet10
MOEA-SA$_{\text{OV}}$	0.232(0.012)	0.165(0.001)	0.204(0.009)	0	0.141(0.017)
CEMOV	**0.239(0.012)**	**0.227(0.007)**	**0.246(0.007)**	**0.208(0.015)**	**0.177(0.007)**

TABLE IX
AVERAGE RUNNING TIME OBTAINED BY COMPARED METHODS ON POLBOOKS AND FOOTBALL (IN SECONDS)

Networks	SLPA	CFinder	CESNA	IMOQPSO	GaoCD	MC-MOEA	MRMOEA	MOEA-SA$_{\text{OV}}$	CEMOV
Polbooks	<1s	<1s	5s	48s	1462s	6s	56s	19643s	339s
Football	<1s	<1s	10s	63s	2724s	7s	47s	5407s	383s

TABLE X
EXPERIMENTAL RESULTS OBTAINED BY CEMOV AND MOEA-SA$_{\text{OV}}$ ON TWO DIRECTED NETWORKS IN TERMS OF D AND E

Networks	D				E			
	CEMOV		MOEA-SA$_{\text{OV}}$		CEMOV		MOEA-SA$_{\text{OV}}$	
	best	mean(std)	best	mean(std)	best	mean(std)	best	mean(std)
Cornell	**1.172**	**1.130(0.057)**	1.096	0.912(0.097)	**0.135**	**0.171(0.030)**	3.201	4.170(0.594)
Washington	1.223	1.155(0.042)	**2.386**	**1.693(0.463)**	**0.133**	**0.219(0.059)**	4.083	5.484(0.865)

H. Comparison Between MOEA-SA$_{\text{OV}}$ and CEMOV on Directed Networks

CEMOV and MOEA-SA$_{\text{OV}}$ [36] can both be applied to directed network. We hereby compare their performances on two real-world directed networks, including Cornell and Washington. Note that the two directed networks have been used in [36].

To make a fair comparison, we apply the same objective function EQ_{OV} as used in [36] to evaluate the network structure, which is defined as

$$EQ_{\text{OV}} = \frac{1}{2L} \sum_{k=1}^{c} \sum_{v \in C_k, w \in C_k} \frac{1}{O_v O_w} \left[A_{vw} - \frac{d'_v d'_w}{2L} \right] \quad (21)$$

where

$$d'_v = \left| \left\{ V_w | \overrightarrow{(V_w, V_v)} \in \mathcal{E} \text{ and } v \neq w \right\} \right|$$

that is, d'_v (resp. d'_w) stands for the in-degree of node V_v (resp. V_w), $\overrightarrow{(V_w, V_v)}$ means directed edge between V_w and V_v. The rest components of CEMOV remain unchanged.

The obtained best, mean, and standard deviation of the D and E values over 31 runs for the two networks are shown in Table X. From the table, we can find that CEMOV outperforms MOEA-SA$_{\text{OV}}$ in terms of D and E on Cornell. On Washington, MOEA-SA$_{\text{OV}}$ obtains better D, but CEMOV obtains a much smaller E value than MOEA-SA$_{\text{OV}}$. From the experiments, we may conclude that the performance of CEMOV on overlapping community detection in the directed attribute network is very promising.

TABLE XI
EXPERIMENTAL RESULTS OBTAINED BY CEMOV AND CEMOV/WA ON FOUR NETWORKS IN TERMS OF D AND E

Networks	D				E			
	CEMOV		CEMOV/wA		CEMOV		CEMOV/wA	
	best	mean(std)	best	mean(std)	best	mean(std)	best	mean(std)
Polbooks	**1.549**	**1.446(0.057)**	1.215	1.099(0.077)	**0.228**	**0.314(0.059)**	0.407	0.501(0.061)
Texas	**1.196**	**1.156(0.024)**	1.192	1.140(0.029)	**0.465**	**0.554(0.064)**	0.466	0.596(0.082)
Cornell	**1.413**	**1.290(0.062)**	1.353	1.275(0.043)	**0.491**	**0.582(0.076)**	0.519	0.583(0.062)
Wisconsin	**1.327**	**1.270(0.032)**	1.301	1.267(0.036)	**0.722**	**0.829(0.066)**	0.746	0.841(0.048)

TABLE XII
EXPERIMENTAL RESULTS OBTAINED BY CEMOV AND CEMOV/WC ON FOUR NETWORKS IN TERMS OF D AND E

Networks	D				E			
	CEMOV		CEMOV/wC		CEMOV		CEMOV/wC	
	best	mean(std)	best	mean(std)	best	mean(std)	best	mean(std)
Texas	**1.196**	**1.156(0.024)**	1.079	1.026(0.033)	**0.465**	**0.554(0.064)**	1.147	1.341(0.116)
Cornell	**1.413**	**1.290(0.062)**	1.335	1.254(0.043)	**0.491**	**0.582(0.076)**	1.079	1.182(0.062)
Washington	**1.276**	**1.137(0.075)**	1.093	1.068(0.027)	**0.759**	**0.872(0.097)**	1.324	1.508(0.077)
Wisconsin	**1.327**	**1.270(0.032)**	1.198	1.152(0.022)	**0.722**	**0.829(0.066)**	1.204	1.333(0.115)

V. SENSITIVITY ANALYSIS

In this section, we analyze the performance of CEMOV w.r.t. its main components, including the continuous encoding, the community merging operator, and the control parameters (i.e., F and CR).

A. Attribute Network Encoding Method

In this section, we study how the incorporation of the attribute information in the encoding affects the performance of the proposed algorithm. We name the algorithm in which no attribute information is used as CEMOV/wA.

In the encoding and decoding process of CEMOV/wA, the continuous vector $\mathbf{x}$ is composed of all the $\mathbf{x}_i$'s for each node, but without the attribute information x_{a_i}. The rest components of CEMOV/wA are the same as CEMOV. In the experiments, both CEMOV and CEMOV/wA have been run 31 times to obtain the best and average (with standard deviation) values of the performance metrics on four attributed networks, including Polbooks, Cornell, Texas, and Wisconsin.

The obtained D and E values for these networks are summarized in Table XI, where the best values are typeset in bold. From this table, we find that all results obtained by CEMOV are better than CEMOV/wA in terms of D and E, which indicates that incorporating the attribute information is indeed beneficial for the overlapping community detection problems in attribute networks.

B. Merging Operator

This section studies the CMM operator. Four WebKB networks, including Texas, Cornell, Washington, and Wisconsin, are used as benchmarks. We name CEMOV without the CMM operator as CEMOV/wC. The obtained results are shown in Table XII, where the best results are also typeset in bold. From Table XII, we observe that all the results obtained by CEMOV are clearly better than CEMOV/wC in terms of D and E. This indicates that CMM can certainly enhance the performance of CEMOV.

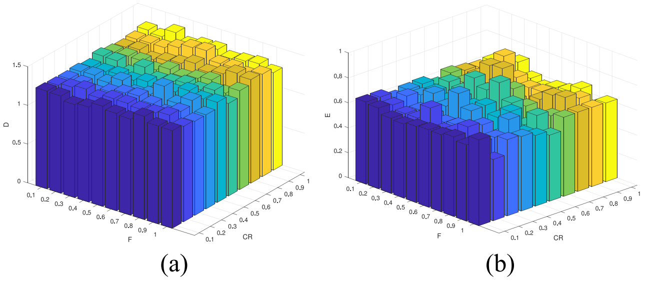


Fig. 6. Performance of CEMOV w.r.t. the values of F and CR in terms of D (a) and E (b).

C. F and CR

In this section, we investigate the effects of the DE control parameters, that is, F and CR, on CEMOV's algorithmic performance. We discretize $F \in [0.1, 1.0]$ and CR $\in [0.1, 1.0]$ with interval 0.1. By performing CEMOV on the Cornell network with each pair of F and CR, we obtained 100 D and E values. Fig. 6 summarizes the results. From the figures, in terms of D, we find that F and CR have no major influences; in terms of E, F and CR can be of some importance to some extent but with no significant difference. We may thus conclude that CEMOV is not sensitive to the settings of F and CR.

VI. CONCLUSION

In this article, a novel continuous encoding method was proposed for solving the overlapping community detection problem for attribute networks. Based on this method, we developed a MOEA upon the MOEA/D framework, called CEMOV. The encoding method embeds not only the network structure but the attribute information in a continuous vector. With the new encoding method, the discrete-natured overlapping community detection problem was transformed to a continuous one. To determine the overlapping nodes, a double-decoding method was proposed. With such a decoding method, the overlapping nodes can be quickly determined though not optimally. Furthermore, a community merging method was proposed after evolution to improve the quality of the found communities.

In the experiments, various networks have been used to test CEMOV against some well-known algorithms in the literature, including EA-based or non-EA-based methods for the overlapping community detection problem with or without node attributes. Particularly, we not only compared CEMOV with MOEA-SA$_{\text{OV}}$ on solving the community detection problem for both single-attribute and multiattribute networks but also for undirected and directed attribute networks. The experimental results showed that CEMOV achieves the state-of-the-art performance. We also investigated the effectiveness of the encoding method and the merging operator. The analysis showed that the proposed encoding and merging operator indeed is beneficial for problem solving.

In the future, we intend to develop new encoding methods for other types of complex networks, and continuous approaches for other $\mathcal{NP}$-hard discrete optimization problems.

Acknowledgment

The authors would like to thank Prof. J. Liu and Dr. X. Wen for providing the code of MOEA-SA$_{\mathrm{OV}}$ and MC-MOEA.

References

[1] 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, 2002.

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

[3] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[4] H. Zhou, "Distance, dissimilarity index, and network community structure," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 67, Jun. 2003, Art. no. 061901.

[5] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, Jun. 2004, Art. no. 066133.

[6] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 3, pp. 430–452, 1990.

[7] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, p. 43, Aug. 2013.

[8] 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.

[9] 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.

[10] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Comput. Surveys*, vol. 50, no. 4, p. 54, Aug. 2018.

[11] D. A. Abduljabbar, S. Z. M. Hashim, and R. Sallehuddin, "Nature-inspired optimization algorithms for community detection in complex networks: A review and future trends," *Telecommun. Syst.*, vol. 74, pp. 225–752, Jan. 2020.

[12] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, "Modularity from fluctuations in random graphs and complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, Aug. 2004, Art. no. 025101.

[13] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, Sep. 2006, Art. no. 036104.

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

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

[16] H. Shen, X. Cheng, K. Cai, and M. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A Stat. Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.

[17] 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.

[18] 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.

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

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

[21] C. Pizzuti, "Overlapped community detection in complex networks," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Montreal, QC, Canada, 2009, pp. 859–866.

[22] G. Bello-Orgaz, S. Salcedo-Sanz, and D. Camacho, "A multi-objective genetic algorithm for overlapping community detection based on edge encoding," *Inf. Sci.*, vol. 462, pp. 290–314, Sep. 2018.

[23] Y. Li, Y. Wang, J. Chen, L. Jiao, and R. Shang, "Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization," *J. Heuristics*, vol. 21, no. 4, pp. 549–575, Aug. 2015.

[24] M. Li and J. Liu, "A link clustering based memetic algorithm for overlapping community detection," *Physica A Stat. Mech. Appl.*, vol. 503, pp. 410–423, Aug. 2018.

[25] Z. Li, S. Zhang, R. Wang, X. Zhang, and L. Chen, "Quantitative function for community detection," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 77, Mar. 2008, Art. no. 036109.

[26] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, "A link clustering based overlapping community detection algorithm," *Data Knowl. Eng.*, vol. 87, pp. 394–404, Sep. 2013.

[27] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, Aug. 2010.

[28] Y. Zhao, S. Li, and F. Jin, "Overlapping community detection in complex networks using multi-objective evolutionary algorithm," *Comput. Appl. Math.*, vol. 36, no. 1, pp. 749–768, Mar. 2017.

[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] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2703–2716, Sep. 2017.

[31] X. Wen *et al.*, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, Jun. 2017.

[32] A. Kumar, D. Barman, R. Sarkar, and N. Chowdhury, "Overlapping community detection using multiobjective genetic algorithm," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 3, pp. 802–817, Jun. 2020.

[33] J. Liu, W. Zhong, H. A. Abbass, and D. G. Green, "Separated and overlapping community detection in complex networks using multiobjective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Barcelona, Spain, 2010, pp. 1–7.

[34] 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.

[35] J. Sun, W. Zheng, Q. Zhang, and Z. Xu, "Graph neural network encoding for community detection in attribute networks," *IEEE Trans. Cybern.*, early access, Feb. 10, 2021, doi: 10.1109/TCYB.2021.3051021.

[36] X. Teng, J. Liu, and M. Li, "Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 138–150, Jan. 2021.

[37] 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.

[38] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 718–729, Aug. 2009.

[39] 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.

[40] 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.

[41] D. Jin, X. Wang, M. Liu, J. Wei, W. Lu, and F. Fogelman-Souli, "Identification of generalized semantic communities in large social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2966–2979, Oct.–Dec. 2020.

[42] T. He and K. C. C. Chan, "MISAGA: An algorithm for mining interesting subgraphs in attributed graphs," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1369–1382, May 2018.

[43] 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.

[44] 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.

[45] 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.

[46] T. He and K. C. C. Chan, "Discovering fuzzy structural patterns for graph analytics," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2785–2796, Oct. 2018.

[47] M. Welling and Thomas N. Kipf, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–14.

[48] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet markov random fields: Semi-supervised community detection in attribute networks," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 152–159.

[49] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[50] D. He *et al.*, "Community-centric graph convolutional network for unsupervised community detection," in *Proc. IJCAI*, Jul. 2020, pp. 3515–3521.

[51] D. Jin *et al.*, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Trans. Knowl. Data. Eng.*, early access, Aug. 11, 2021, doi: 10.1109/TKDE.2021.3104155.

[52] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[53] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, Oct. 2010, Art. no. 103018.

[54] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proc. IEEE 11th Int. Conf. Data Min. Workshops (ICDMW)*, 2011, pp. 344–349.

[55] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A Stat. Mech. Appl.*, vol. 374, no. 1, pp. 483–490, Jan. 2007.

[56] S. Gregory, "Fuzzy overlapping communities in networks," *J. Stat. Mech. Theory Exp.*, vol. 2011, Feb. 2011, Art. no. P02017.

[57] F. Havemann, M. Heinz, A. Struck, and J. Glaser, "Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels," *J. Stat. Mech. Theory Exp.*, vol. 2011, Jan. 2011, Art. no. P01023.

[58] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. 13th IEEE Int. Conf. Data Min.*, Dallas, TX, USA, 2013, pp. 1151–1156.

[59] J. Yang and J. Leskovec, "Structure and overlaps of ground-truth communities in networks," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, p. 26, Apr. 2014.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[66] 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.

[67] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[68] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genet. Algorithms*, 1985, pp. 93–100.

[69] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, Feb. 2007.

[70] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, Jul. 2009, Art. no. 016118.

[71] V. Krebs. "Books About U.S. Politics." 2004. [Online]. Available: http://www.orgnet.com

[72] 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.

[73] J. Leskovec and A. Krevl. "SNAP Datasets: Stanford Large Network Dataset Collection." Jun. 2014. [Online]. Available: http://snap.stanford.edu/data

[74] 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.

[75] G. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification," in *Proc. 10th Int. Workshop Min. Learn. Graphs*, pp. 1–8, 2012.

[76] 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.

[77] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, "CFinder: Locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, Feb. 2006.

[78] H. Elhadi and G. Agam, "Structure and attributes community detection: Comparative analysis of composite, ensemble and selection methods," in *Proc. ACM 7th Workshop Soc. Netw. Min. Anal.*, 2013, p. 10.

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.

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 the 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, big data, 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 IEEE Transactions on Cybernetics. He is a Web of Science Highly Cited Researcher of Computer Science for five 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 served 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, Xi'an Jiaotong University. His current research interests include intelligent information processing and applied mathematics.

Dr. Xu is the owner of 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 2011, the CSIAM Su Buchin Applied Mathematics Prize in 2008, and the ITIQAM Richard Price Award. He delivered a 45-min talk on the International Congress of Mathematicians 2010. He was elected as a member of Chinese Academy of Science in 2011.