Graph Neural Network Encoding for Community Detection in Attribute Networks

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

Abstract—In this paper, we first propose a graph neural network encoding method for multiobjective evolutionary algorithm 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 non-linear transformation, a continuous valued vector (i.e. a concatenation of the continuous variables associated with all edges) is transferred to a discrete valued community grouping solution. Further, two new objective functions for single- and multi-attribute network are proposed to evaluate the attribute homogeneity of the nodes in communities, respectively. Based on the new encoding method and the two new objectives, a multiobjective evolutionary algorithm (MOEA) based upon NSGA-II, termed as continuous encoding MOEA, is developed for the transformed community detection problem with continuous decision variables. Experimental results on single- and multi-attribute real-life networks with different types show that the developed algorithm performs significantly better than some well-known evolutionary and non-evolutionary 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—Complex attribute network, community detection, graph neural network encoding, multiobjective evolutionary algorithm

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 and difficulties of the complex network detection problem, research on this subject has become popular since 1930s [5]. A large interdisciplinary community of scientists have 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 in every several years from 2005 until recently [6]. In this paper, we do not intend to review all network literatures but only on approaches based on 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], e.g. the modularity ($Q$), the community score ($CS$), the sum of community fitness $P(S)$, 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 multi-objective EAs (MOEAs) have also been applied on this problem.

Genetic algorithm was firstly adopted in [10] for maximizing $Q$. Since then, several genetic algorithms (GA), including MIGA [11], MAGA-Net [12] and Meme-Net [13], and a discrete particle swarm optimization algorithm (GDPSO) [14], were also developed based on optimizing $Q$. On the other hand, GA-Net [8] 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 genetic algorithm, dubbed as MOGA-Net [15] [16], was proposed in 2009. It was built upon NSGA-II in which two objectives including $CS$ and $P(S)$ are used. MOEA/D-Net [17] takes the Negative Ratio Association (NRA) and Ratio Cut (RC) as two objectives, which is built upon the framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D). MICD [18], MODBSA [19], and DIMMDE [20], were all developed based on the two objectives, but under different MOEA frameworks. In MOCO [21] and MMCD [22], two objectives obtained by decomposing the modularity $Q$, which are to measure the intra-cluster edge density and inter-cluster sparsity, were used. In existing work, the research focus is on the development of new objectives for measuring the grouping quality of communities.

In this paper, 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 to 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 [23]. 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 complex networks much more difficult.

The study of community detection for attribute networks only emerged since 2003 [24]. It is still in its infancy. Existing research work on the attribute complex network community detection roughly fall into three categories, namely distance-based, model-based, and evolutionary algorithm-based.

In distance-based methods, the distance definition between nodes considering both network structure and attribute homogeneity is key to community detection. In [25], a graph clustering algorithm, named SA-Cluster, was proposed in which a unified distance measure 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 [26].

Model-based methods are constructed based on modeling the relationship between network structure and node attributes. In [27] [28], the clustering problem for attribute network is modeled under the Bayesian probabilistic framework, and solved by Bayesian inference. A popularity-based conditional link model [29] was proposed to model the node’s popularity while the model parameters are estimated by maximum likelihood estimation. A non-negative matrix factorization model was proposed in [30] where the community membership and community attribute matrices were viewed as decisive parameters for community detection. The interaction between network structure and node attributes was modeled statistically in [31].

In [32], a method named descriptive community mining was proposed to solve attribute network clustering by alternating between maximizing the community score and inducing a fitting concise description. Recently, fuzzy clustering algorithm was applied to detect the attribute complex network [33].

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 used to solve the attribute complex network community detection problems.

To the best of our knowledge, only two papers based on MOEAs have been published for attribute network detection. The first one is MOEA-SA [34], in which a new objective $S_A$ was proposed to measure the attribute similarity within clusters. Together with the modularity [7], MOEA-SA is developed upon NSGA-II [35] with a hybrid network encoding method and a multi-individual-based mutation operator. Besides, a neighborhood correction strategy is proposed to repair improper solution. The other one is MOGA-@Net [36], which is also developed based on NSGA-II. In MOGA-@Net, three objectives (namely, modularity [7], community score [8] and conductance [37]) 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 an individual and a decoding process to retrieve an individual to a partition for evaluating its quality. There are two widely used encodings, including the locus-based [38] and label-based [10].

It is argued in [34] that the locus-based encoding is easy to initialize for relatively good individuals, but is time-consuming when decoding. The label-based encoding, on the other hand, is easy to design evolutionary operators, but is not good at initialization since the adjacency information is not involved. MOEA-SA uses a hybrid encoding method by combining these two encodings.

In this paper, 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 non-linear functions. Based on this encoding, the original discrete-valued community detection problem is transformed into a continuous optimization problem. We then propose a continuous coded MOEA built upon NSGA-II [35], in which each individual is a continuous valued vector as opposed to a discrete valued vector in the locus-based and label-based encodings.

Such a continuous encoding method can make the fitness landscape of the transformed community detection 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 better measure the attribute homogeneity in the communities, we propose two new objectives to handle the single- and multi-attribute similarity, respectively. Further, the developed algorithm, named as continuous encoding MOEAs (CE-MOEA) is able to determine the number of communities automatically due to MOEA’s population-based search mechanism.

The rest of this paper is organized as follows. Section II introduces some preliminaries, including the definition of attribute complex network, relevant concepts in multiobjective optimization and MOEA. The proposed method, including the new graph neural network network encoding, the proposed objectives for attributes and the 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 to justify the effectiveness of the encoding method is given in Section V. Section VI concludes this paper.

II. Preliminaries

A. Complex Attribute Network

An attribute network is a 3-tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{V_1, V_2, \cdots, V_r\}$ 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 $V_j$), $\mathcal{A} = \{a_1, a_2, \cdots, a_r\}$ 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 8 nodes and 10 edges. Each node has 4 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: $\{V_1, V_2, V_3, V_4\}$ and $\{V_5, V_6, V_7, V_8\}$. However,
that two requirements are satisfied, including 1) the density between communities is sparse and the density within the community is 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. THE METHOD

In this section, the graph neural network encoding method is first presented, followed by two newly developed approaches in consideration of attribute homogeneity. The CE-MOEA is then presented.

A. Graph Neural Network Encoding

The locus-based [38] and label-based [10] encodings have been widely used in MOEAs for network-related optimization problems. 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 node 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, i.e. \{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 is obtained after decoding. It should be noted that the resultant genotypes by the two encoding methods are all discrete vectors.

As argued in [34], 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

Fig. 1. An example attribute network with 8 nodes, 10 edges. There are 4 attributes for each node.

if consider only the “age” attribute, it is rather difficult to partition this network.

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: \{V_1, V_2, V_3, V_4\} and \{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 multi-objective optimization problem (MOP) can be stated as follows:

\[
\text{minimize } F(w) = (f_1(w), f_2(w), \ldots, f_m(w))^{\top} \\
\text{subject to } w \in \Omega
\]

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

In the MOP taxonomy, a vector \(x = (x_1, \ldots, x_m)\) is said to dominate a vector \(y = (y_1, \ldots, y_m)\) (denoted as \(x \succ y\)) if and only if there exists at least one \(k\) such that \(x_j \geq y_j, \forall j \in \{1, \ldots, m\}\) but \(x_k > y_k\). If a solution \(x^* \in \Omega\) is not dominated by any other solution, \(x^*\) is called a Pareto optimal solution. There exists many optimal solutions that are non-dominated 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 Pareto dominance relation based (such as NSGA-II [35]), performance metric based (such as HypE [39]), decomposition based (such as MOEA/D [40]) and learning based MOEAs (such as OCEA [41]). We do not intend to review the rich literature of MOEA in this paper. Interested readers please refer to [42].

In this paper, the purpose of attribute complex network detection problem is to find a partition of communities such as

\[
\text{age} = 22, \text{sex} = \text{female}, \text{degree} = \text{BS}, \text{major} = \text{math}; \\
\text{age} = 23, \text{sex} = \text{female}, \text{degree} = \text{BS}, \text{major} = \text{... science}; \\
\text{age} = 27, \text{sex} = \text{male}, \text{degree} = \text{MS}, \text{major} = \text{computer science}; \\
\text{age} = 22, \text{sex} = \text{male}, \text{degree} = \text{MS}, \text{major} = \text{computer science}
\]
In Alg. 1 a continuous valued vector $\mathbf{x} \in \mathbb{R}^d$ where $d = \sum_{i,j} c_{ij}$ is the number of edges, is taken as the input. $\mathbf{x}$ is a concatenation of $r$ sub-vectors, 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 $d_i = \sum_j c_{ij}$. 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 $D_i$.

For node $V_i$, denote $\mathbf{x}_i = [x_{i,1}, \cdots, x_{i,d_i}]$, we first apply a sigmoid function $\sigma$ which is defined as follows:

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

over $x_i$ element by element. This gives $h_i = (0, 1)^d_i$ (line 3). A softmax function is then applied on $h_i$ to obtain $p_i = [p_{i,1}, \cdots, p_{i,d_i}]$ (line 4) where

$$p_{ij} = \frac{\exp(h_{ij})}{\sum_j \exp(h_{ij})}, \quad 1 \leq j \leq d_i.$$  

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

$$s_i = \arg\max_{j=1, \cdots, d_i} p_{ij}$$

i.e. 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 $s$ (line 6). With the obtained $s$, a partition $G_s$ can be returned after decoding (line 7).

**Algorithm 1: The Graph Network Encoding Method**

**Input:** $\mathbf{x} = [x_1, \cdots, x_r] \in \mathbb{R}^d$

**Output:** A community partition $s$.

1. Set $s = \emptyset$;
2. for $i \leftarrow 1$ to $r$ do
   3. $h_i \leftarrow \sigma(\mathbf{x}_i)$;
   4. $p_i \leftarrow \text{softmax}(h_i)$;
   5. $s_i \leftarrow \text{argmax}(p_i)$;
   6. $s \leftarrow s \cup s_i$;
3. end
4. return $G_s \leftarrow \text{Decoding}(s)$.

Fig. 3 shows the encoding process of a single node $V_i$. From the figure, it is seen that for each $V_i$, there associates a continuous value for each node in $D_i$. Through sigmoid, softmax and argmax operation, node $V_j$ is selected to be linked to $V_{s_i}$ 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 $G$), with continuous vector $\mathbf{x}$ associated with the edges, the sigmoid operation (which can be regarded as the sigmoid layer in neural network) is applied to obtain $h$. The softmax layer is then applied on $h$ to obtain $p$. Through argmax operation (layer), each node is linked to the node that is with the greatest probability entity in its corresponding $p$ values. The concludes the encoding process, which lead to a locus-based representation. The decoding process can thus turn the representation into a partition $G_s$ of $G$.

In the following, for the sake of simplicity, we use $G_s = \text{GNN}(\mathbf{x}; G)$ to represent the encoding of a genotype to a network $G$. That is, given $\mathbf{x}$, a network partition $G_s$ can be obtained by function $\text{GNN}(\cdot)$. With the obtained network partition, objectives such as the modularity can be calculated.

**B. The New Objectives**

1) **Objective Regarding the Network Structure:** The well-known modularity $Q$ proposed in [2] is used as the first objective in our study for revealing the network structure. Given a network $G$ and its partition $G_s$, 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 in the network. The modularity $Q$ is defined as:

$$Q = \sum_{k=1}^c \left[ \frac{l_k}{L} - \left( \frac{d_k}{2L} \right)^2 \right] \triangleq f_Q(G_s; G)$$

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

Together with Eq. 4 we see that given $\mathbf{x}$, the modularity can be computed by composing functions $f_Q$ and GNN as follows:

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

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, we suggest to use the following two objectives for measuring single- and multi-attribute homogeneity among the detected communities, respectively.

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

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

where

$$S_O = \sum_{k=1}^c \sum_{V_i \in V_k} \sum_{i<j} \sqrt{(a_i - a_j)^2}$$

and $c$ is the number of obtained clusters, $C_k$ is the cluster $k$, $r_k$ stands for the number of nodes within cluster $k$, $a_i$ (resp.
\(a_j\) is the attribute of \(V_i\) (resp. \(V_j\)). \(S_G\) is the sum of the Euclidean distance between node attributes of each community. The denominator is the summation of all obtained clusters of values \(r_k(r_k - 1)\). It is to measure the distance between single attribute homogeneity within the detected communities.

For a multi-attribute network with binary attribute values, a cosine-based similarity objective function \(f_m\) is proposed to measure the attribute similarity:

\[
f_m = \frac{M_O}{\sum_{k=1}^{c} r_k(r_k - 1)}
\]

where

\[
M_O = \sum_{k=1}^{c} \sum_{i<j}^{V_i, V_j \in C_k} \frac{a_i \cdot a_j}{\|a_i\| \cdot \|a_j\|}
\]

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 of the node attributes in the obtained communities is. 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 \(Q(x)\), we also define \(f_s(x)\) and \(f_m(x)\).

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

\[
\text{minimize } F = (-Q(x), f_s(x)) \\
\quad \text{or } F = (-Q(x), f_m(x))
\]

s.t. \(x \in [0, 1]^d\) (10)

Here the reason to set \(x \in [0, 1]^d\) is to make the range of the sigmoid function controllable and the softmax function is scale-invariant.

### C. The Algorithm

The developed algorithm is built upon the well-known Non-dominated Sorting Genetic Algorithm II (dubbed as NSGA-II) with differential evolution (DE) operators [43]. It is summarized in Alg. 2.

The functions FNS(·), CWD(·) and BNS(·) in Alg. 2 represent the fast non-dominated sorting, crowding distance, and binary tournament selection, respectively. They are just standard operations used in NSGA-II. The DE(·) function standards for the differential evolution operation which will be described later.

In Alg. 2, the first population \(P_1\) is randomly initialized in \([0, 1]^d\) (line 1), and individuals are evaluated according to Eq. 10. Their objectives are organized in \(F\) (line 2). The non-dominated layers and the crowding distances of \(F\) are then computed in line 3. From line 4 to 19, the NSGA-II operations are performed to optimize Problem 10. The binary tournament selection is carried out on \(P_t\) to obtain a parent set \(P\) based on the objectives \(F\) and the crowding distance \(C_d\) (line 5). By selecting parent individuals from \(P\), DE is applied to generate new offsprings taking respective control parameters (line 10). The newly generated individuals are evaluated in line 12 and combined with current population (line 15). The combined individual objectives are sorted to obtain the non-dominated layers and the crowding distances (line 16). Solutions are then selected from the sorted layers to obtain the next generation (lines 17-18). The algorithm continues until the maximum number of generations \(T\) has been reached. The final population is returned as the approximated Pareto set and front (line 20).

Alg. 3 summarizes the DE(·) function used to generate offsprings. In Alg. 3, \(x_1\) is first mutated by taking the difference of \(x_2\) and \(x_3\), 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 gene \(y\) is then repaired if any of its element is beyond the variable range (line 4). The PM operator [44] is the used to mutate the offspring \(y\) (line 6). The obtained individual \(y\) is repaired (line 8) and returned (line 9).

It should be pointed out that there are several advantages of the proposed approach for the community detection problem.
for complex attribute networks:

- The graph neural network encoding method makes fully 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.
- The graph neural network encoding method can be applied to attribute or non-attribute network, and to undirected or directed network.
- 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 solving the problem.

D. Complexity Analysis

Let \( r \) be the number of nodes in network \( G \), \( N \) the population size, \( m \) the number of objectives, \( L \) the number of edges and \( T \) the maximum of generations. Alg. 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 \( \mathcal{O}(r) \) [45]. Thus, the total complexity of Alg. 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. The only difference is that CE-MOEA requires a complexity of \( \mathcal{O}(LN) \) as the overhead for population initialization, and the complexity of decoding the population which is \( \mathcal{O}(rN) \) at each generation. Overall, the total complexity of CE-MOEA is \( \mathcal{O}(mN^2T + rNT) \), plus an overhead \( \mathcal{O}(LN) \).

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 \( F = 0.7, CR = 0.5 \), the mutation probability \( p_m = 0.02 \) and distribution index of mutation \( \eta_m = 20 \). CE-MOEA was run eleven times independently for the considered networks.

A. The Benchmark Networks

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

Among these networks, Amazon U.S. Politics Books [46] and Political Blogs [47] are with single attribute. They have no ground truth labels for the communities. The politics books dataset include all the books studying U.S. politics which were purchased by customers. Each book is associated with one political blog in which the books were both studied. This dataset has been used as the benchmark in our study.

Algorithm 2: CE-MOEA

**Input:** An attributed network \( G \), the population size \( N \), the maximum number of generations: \( T \); the parameters of the DE operator (\( F \) and \( CR \)); and the PM operator parameter: \( p_m \) and \( \eta_m \).

**Output:** an approximated PS and PF.

```plaintext
1 Set \( g \leftarrow 1 \) and randomly initialize \( P_g \in [0, 1]\) \( N \times d \);
2 Evaluate \( F_i \leftarrow F(P_g(i,:)), 1 \leq i \leq N \); Set \( F_1 = \{ F_i \} \);
3 \( L_s \leftarrow \text{FNS}(F) \) and \( C_d \leftarrow \text{CWD}(F) \);
4 while \( g < T \) do
5 \( \mathcal{P} \leftarrow \text{BTS}(P_g, F_g, C_d) \);
6 Set \( \mathcal{Y} = \emptyset \) and \( F_o = \emptyset \);
7 for \( 1 \leq j \leq |\mathcal{P}| \) do
8 \( x_3 \leftarrow \mathcal{P}(j,:) \);
9 \( y = \text{DE}(x_1, x_2, x_3, F, CR, p_m, \eta_m) \);
10 \( \mathcal{Y} \leftarrow \mathcal{Y} \cup y \);
11 \( F_o \leftarrow F_o \cup F(y) \);
12 end
13 \( P_g \leftarrow P_g \cup \mathcal{Y} \) and \( F_g \leftarrow F_g \cup F_o \);
14 \( L_s \leftarrow \text{FNS}(F_g) \) and \( C_d \leftarrow \text{CWD}(F_g) \);
15 Sort \( L_s \) based on \( C_d \) in the descending order;
16 Select non-dominated solutions from the sorted \( L_s \) to fill \( P_{g+1} \) until its size equals to \( N \);
17 \( g \leftarrow g + 1 \);
18 end
19 return \( P_T \) as the approximated PS and \( F_T \) as the PF.
```

Algorithm 3: The Differential Evolution Operator (the DE(·) function)

**Input:** individuals \( x_1, x_2 \) and \( x_3 \in \mathbb{R}^d \) and recombination parameters \( F, CR, p_m \) and \( \eta_m \).

**Output:** An offspring \( y \).

```plaintext
1 for \( 1 \leq i \leq d \) do
2 \( y_i = \begin{cases} x_1^i + F \times (x_2^i - x_3^i), & \text{if } \text{rand}(\cdot) \leq CR, \\ x_1^i, & \text{otherwise} \end{cases} \)
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 \( 1 \leq i \leq d \) do
6 \( y_i = \begin{cases} y_i + \delta_i \times (b_i - a_i), & \text{if } \text{rand}(\cdot) < p_m, \\ y_i, & \text{otherwise} \end{cases} \)
7 end
8 Repair \( y \) if necessary.
9 return \( y \).
```
compiled by Adamic and Glance in 2005 to show the political orientation of blogs. It contains 1490 nodes and 19025 edges which connect blogs by hyperlinks. Each web-blog has an attribute demonstrating political complexion: 1) liberal or 2) conservative.

The rest of the networks, including the Ego facebook networks [48] (no ground truth available), the WebKB networks [49], the Cora citation network [29] and the Citeseer citation network [29] are with multi-attributes. 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 [49] consisting of four subnetworks from four U.S. universities: Cornell, Texas, Washington and Wisconsin are used. The attribute dimension of all four networks: Texas, Cornell, Washington and 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 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 Corise is 3703.

Table I summarizes the detailed information of the benchmark networks mentioned above.

| Dataset          | Network Type          | Nodes | Edges | Attributes with ground truth |
|------------------|-----------------------|-------|-------|------------------------------|
| Polblogs Books co-purchasing | 185 | 441 | 1 | No |
| Polblogs Blogs hyperlinks | 1490 | 19029 | 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 | 3429 | 1453 | Yes |
| Citeese Citation  | 3132 | 4972 | 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 | 187 | 328 | 1703 | Yes |

Table I: Detailed characteristics 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

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

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 the $D$ value, the more distinct the community structure in the network is.

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

$$E = \sum_{k=1}^{c} \frac{r_k}{r} \cdot H(k)$$

$$H(k) = -\sum_{a \in A} p_{ak} \log(p_{ak})$$

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 homogeneous of nodes in the detected communities is.

3) **Normalized mutual information (NMI):** NMI [50] 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($P, P^*$) is defined as

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

where $c_P$ (resp. $c_{P^*}$) is the number of clusters in the partition $P$ (resp. $P^*$), $M_i$ (resp. $M_{ij}$) is the summation of elements of matrix $M$ in row $i$ (column $j$), $r$ is the total number of nodes in network.

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**

MOEA-SA is used as the compared algorithm. It was also run eleven times, while the parameter configurations described in [34] were applied.

1) **Results on the Political Networks:** Table II shows the detection results of CE-MOEA and MOEA-SA on the Political Books and Political Blogs networks, in which the maximum, minimum, average values of the metrics $D$ 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, the Wilcoxon’s rank sum test at a significance level of 5% is performed to test whether the results obtained by CE-MOEA and MOEA-SA are significantly different. The column WR shows the hypothesis test results, where $\dagger$, $\approx$ and $\|\|$ means that the result obtained by CE-MOEA is better than, similar to, and worse than the result obtained by MOEA-SA, respectively.

From Table III it is seen that CE-MOEA can always obtain better results on the two political networks than MOEA-SA.
except on the $D$ metrics for Polblogs. The Wilcoxon’s rank sum tests also suggest that CE-MOEA performs significantly better than MOEA-SA except on the $D$ metrics for Polblogs where there is no significant different between them. In the last column of Table II, $k$ means the number of clusters obtained by the corresponding algorithm. We found that for Political Books network, both CE-MOEA and MOEA-SA find similar number of clusters. However, the communities found by CE-MOEA are with much larger number of clusters than those found by MOEA-SA on the Polblogs.

To further show the performance of CE-MOEA, the PFs obtained among the runs with the median value $D$ are shown in Fig. 5 for the two political networks. In the figure, the $x$-axis is the negative modularity, the $y$-axis shows the attribute similarity. From the two figures, we found 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 multi-attribute and no ground truth are given in Table III. Again, in the columns, the best metric values are marked in bold and the Wilcoxon’s rank sum test results at a significance level of 5% are shown.

It is seen from Table III that in terms of average $D$ and $E$, CE-MOEA always obtain better results than MOEA-SA. Particularly, we found that all average values of $D$ are more than 0.85 except for Ego 686. The standard deviations of all values of $D$ are less than 0.02, expect for Ego 0 and 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.

On the other hand, all the average $E$ values obtained by CE-MOEA are less than 0.13, 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 on all Ego facebook networks. Finally, in the last column, we found that CE-MOEA has obtained generally smaller ranges of number of communities than MOEA-SA on all networks.

The PFs of the ego networks obtained by CE-MOEA in the run with the median $D$ value are shown in Fig. 5. Similar to Fig. 5, we found 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 multi-attribute networks, and performs better than MOEA-SA.

D. Results on Networks with Ground Truth

In this section, the six multi-attribute networks (Cornell, Texas, Washington, Wisconsin, Cora, and Citeseer) with ground truth are used as the benchmark. Six state-of-the-art community detection algorithms for attribute complex network are compared with CE-MOEA, including SA-Cluster [25], Inc-Cluster [26], PCL [29], BAGC [27], SCI [30] and MOEA-SA [34] in terms of the NMI metric.

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 type set in bold. Further, the $z$-test was carried out to find out the significant differences between CE-MOEA and the compared algorithms. In Table IV symbols ‘+’ (resp. ‘−’ and ‘=’) denotes that the performance of compared algorithm is significantly better than (resp. worse than and similar to) CE-MOEA at the 5% significant level.

From Table IV, it is observed that CE-MOEA performs better than SA-Cluster, Inc-Cluster, BAGC and MOEA-SA in terms of the NMI metric on all the networks. CE-MOEA only performs worse than PCL on Cora, worse than SCI on Texas. However, we found that CE-MOEA is ranked the second on the two networks. The $z$-test also suggests that CE-MOEA performs significantly better than SA-Cluster, Inc-Cluster and BAGC on all the networks. CE-MOEA performs similarly to SCI on Cornell and Wisconsin, and similarly to MOEA-SA on Wisconsin. In summary, we may conclude that CE-MOEA performs better than these compared algorithms in general. Notice that these algorithms are with different types. This clearly shows the effectiveness of the developed approach.

V. Fitness Landscape Analysis

From the above experimental study, we may conclude that the proposed algorithm performs better than existing algorithms for both single- and multi-attribute 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 $f_x$ or $f_m$. In our experiments, the ruggedness of the community detection problem landscape
TABLE II
Statistics of the obtained $D$ and $E$ metric values by CE-MOEA and the compared algorithms on the political networks, where $WR$ means the 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    | 0.891    | 0.896(0.007) | 0.201 | 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 |
| Polblogs | CE-MOEA    | 0.916    | 0.896    | 0.906(0.005) | 0.042 | 0.030 | 0.037(0.003) |    |    | 14-32 |
|          | MOEA-SA    | 0.914    | 0.896    | 0.906(0.007) | ≈   | 0.153 | 0.117 | 0.139(0.009) | † | 4-16 |

TABLE III
Statistics of the obtained $D$ and $E$ values obtained by CE-MOEA and the compared algorithms on the political networks, where $WR$ means the 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    | 0.933(0.032) | 0.071 | 0.039 | 0.051(0.009) |    |    | 3-13 |
|          | MOEA-SA    | 0.815    | 0.632    | 0.707(0.049) | †   | 0.146 | 0.142 | 0.144(0.001) | † | 6-17 |
| Ego 107 | CE-MOEA    | 0.946    | 0.930    | 0.940(0.004) | 0.036 | 0.025 | 0.031(0.003) |    |    | 5-17 |
|          | MOEA-SA    | 0.938    | 0.804    | 0.917(0.037) | †   | 0.078 | 0.076 | 0.077(0.001) | † | 4-29 |
| Ego 686 | CE-MOEA    | 0.758    | 0.687    | 0.717(0.018) | 0.081 | 0.067 | 0.069(0.004) |    |    | 3-5  |
|          | MOEA-SA    | 0.648    | 0.578    | 0.621(0.021) | †   | 0.295 | 0.271 | 0.282(0.007) | † | 4-11 |
| Ego 1684| CE-MOEA    | 0.900    | 0.892    | 0.897(0.003) | 0.031 | 0.024 | 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 |
| Ego 1912| CE-MOEA    | 0.976    | 0.961    | 0.965(0.004) | 0.031 | 0.021 | 0.027(0.002) |    |    | 4-9  |
|          | MOEA-SA    | 0.918    | 0.785    | 0.849(0.040) | †   | 0.090 | 0.086 | 0.087(0.001) | † | 3-15 |
| Ego 3437| CE-MOEA    | 0.916    | 0.860    | 0.876(0.014) | 0.066 | 0.046 | 0.055(0.005) |    |    | 7-13 |
|          | MOEA-SA    | 0.898    | 0.806    | 0.861(0.029) | ≈   | 0.106 | 0.101 | 0.103(0.001) | † | 11-23 |
| Ego 3980| CE-MOEA    | 0.923    | 0.769    | 0.851(0.055) | 0.170 | 0.113 | 0.128(0.023) |    |    | 3-6  |
|          | MOEA-SA    | 0.669    | 0.597    | 0.626(0.021) | †   | 0.310 | 0.267 | 0.284(0.012) | † | 6-10 |

Fig. 6. The PF plots of the Ego networks obtained by CE-MOEA in the run with median $D$ values.
The neighborhood of a genotype $x$ is thus defined as

$$\mathcal{N}_O(x) = \{y | \text{dist}(x, y) = 1\}$$

(17)

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

For the transformed problem, the neighborhood of a solution $x$ is defined as

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

(18)

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

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 V and VI for modularity and attribute similarity, respectively. In the tables, LOD$_o$ (resp. ER$_o$ and FDC$_o$) means LOD (resp. ER and FDC) metric for the original problem and LOD$_t$ (resp. ER$_t$ and FDC$_t$) is for the transformed problem. The better results of the corresponding metrics are typeset in bold.

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

To further show the landscape differences, Fig. 7 shows the FDC plots of the original (in subplots (a) and (c)) and transformed problem (in subplots (b) and (d)) on the Polbooks network. 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 values 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 clearly indicates that the landscape of the transformed problem is smoother than that of the original problem.

### Table IV

The mean and standard deviation of the obtained NMI values by the six compared method and CE-MOEA on multi-attribute networks with ground truth.

| Dataset       | SA-Cluster | Inc-Cluster | PCL | BAGC | SCI | MOEA-SA | CE-MOEA |
|---------------|------------|-------------|-----|-----|-----|---------|---------|
| Cornell       | 0.064$^-$  | 0.038$^-$   | 0.073(0.010)$^-$ | 0.040(0.006)$^-$ | 0.166(0.008)$^a$ | 0.132(0.015)$^-$ | 0.176(0.019) |
| Texas         | 0.082$^-\text{c}$ | 0.106$^-\text{c}$ | 0.061(0.011)$^-\text{c}$ | 0.052(0.007)$^-\text{c}$ | 0.202(0.019)$^+$ | 0.111(0.013)$^-\text{c}$ | 0.155(0.018) |
| Washington    | 0.077$^-\text{c}$ | 0.063$^-\text{c}$ | 0.092(0.015)$^-\text{c}$ | 0.053(0.006)$^-\text{c}$ | 0.146(0.006)$^-\text{c}$ | 0.139(0.014)$^-\text{c}$ | 0.186(0.019) |
| Wisconsin     | 0.101$^-\text{c}$ | 0.089$^-\text{c}$ | 0.060(0.001)$^-\text{c}$ | 0.034(0.015)$^-\text{c}$ | 0.184(0.001)$^-\text{c}$ | 0.156(0.016)$^-\text{c}$ | 0.190(0.022) |
| Cora          | 0.117$^-\text{c}$ | 0.112$^-\text{c}$ | 0.416(0.003)$^+$ | 0.008(0.005)$^-\text{c}$ | 0.204(0.008)$^-\text{c}$ | 0.118(0.002)$^-\text{c}$ | 0.398(0.002) |
| Citeseer      | 0.047$^-\text{c}$ | 0.043$^-\text{c}$ | 0.170(0.003)$^-\text{c}$ | 0.017(0.001)$^-\text{c}$ | 0.075(0.036)$^-\text{c}$ | 0.142(0.003)$^-\text{c}$ | 0.345(0.001) |

is measured by three metrics, including local optimum density (LOD), escaping rate (ER) and fitness distance correlation (FDC) [51]. All these metrics are obtained by applying the Iterated Local Search (ILS) [52] heuristic. The metrics are defined as follows:

- 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.

- ER: This refers to the success rate of the ILS to reach a new local optimum by perturbing the current local optimum.

- FDC: To compute FDC, we have randomly selected 1000 local optima (x$_{LO}$) from the set obtained by the ILS and their function values are $f(x_{LO})$. Then, the distances of 1000 local optima to the nearest global optimum are calculated as $d_{\text{opt}}$. Overall, the FDC is defined as

$$\text{FDC}(f(x_{LO}), d_{\text{opt}}) = \frac{\text{cov}(f(x_{LO}), d_{\text{opt}})}{\sigma(x_{LO})\sigma(d_{\text{opt}})}$$

(14)

where $\text{cov}(\cdot)$ 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 [33].

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.

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 $x = (x_1, x_2, \ldots, x_r)$ and $y = (y_1, y_2, \ldots, y_r)$, the distance between them is defined by

$$\text{dist}(x, y) = \sum_{i=1}^{r} |\text{sgn}(x_i - y_i)|$$

(15)

where

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

(16)
and others. Optimization problems such as traveling salesman problem, 2) develop specific neural network encoding for other discrete overlapping complex attribute network community detection; 1) apply the graph neural network encoding method for beneficial for the optimization. In the future, we intend to applied that the developed graph neural network encoding is significantly better than MOEA-SA and those non-EA algorithms on a set of real-life networks EA based detection algorithms on a newly developed objectives for single- and multi-attribute based algorithm.

Based on the novel encoding method, combing with two newly developed objectives for single- and multi-attribute similarity respectively and the modularity objective for network structure, we developed a multi-objective evolutionary algorithm, named as CE-MOEA, under the framework of NSGA-II. CE-MOEA was extensively compared against state-of-the-art MOEA (MOEA-SA) 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 performed showed that CE-MOEA performed significantly better than MOEA-SA and those non-EA algorithms in general.

Particularly, since MOEA-SA and CE-MOEA both 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; 2) develop specific neural network encoding for other discrete optimization problems such as traveling salesman problem, and others.

### Table V

**The fitness landscape metrics obtained for the original problem and transformed problem on the selected networks in terms of the modularity Q.**

| Dataset   | LOD_0 | LOD_1 | ER_0 | ER_1 | FDC_0 | FDC_1 |
|-----------|-------|-------|------|------|-------|-------|
| 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 VI

**The fitness landscape metrics obtained for the original problems and transformed problems on the selected networks in terms of f_x or f_y.**

| Dataset   | LOD_0 | LOD_1 | ER_0 | ER_1 | FDC_0 | FDC_1 |
|-----------|-------|-------|------|------|-------|-------|
| 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 |

Fig. 7. The FDC analysis of the original and transformed problems for the Polbooks network. (a) and (b) are for the modularity Q, (c) and (d) are for the objective f_x. The correlation coefficients between the x-axis and y-axis values are 0.0475, 0.1326, 0.0666 and 0.2091 for plot (a), (b), (c) and (d), respectively.

### VI. Conclusion

In this paper, 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 search-based algorithm.

ACKNOWLEDGEMENT

The authors would like to thank Prof. J. Liu for providing the code of their MOEA-SA.

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,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.
[3] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences*, 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*, vol. 76, p. 036106, Sep 2007.
[5] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, p. 75–174, 2010.
[6] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, “Community detection in networks: A multidisciplinary review,” *Journal of Network and Computer Applications*, vol. 108, pp. 87–111, 2018.
[7] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, p. 026113, Feb 2004.
[8] 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, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1081–1090.
[9] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, no. 3, p. 033015, May 2009.
[10] M. Tasgin, A. Herdagdelen, and H. Bingol, “Community detection in complex networks using genetic algorithms,” *arXiv preprint arXiv:0711.0491*, 2007.
[11] R. Shang, J. Bai, L. Jiao, and C. Jin, “Community detection based on modularity and an improved genetic algorithm,” *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 5, pp. 1215–1231, 2013.
[12] Z. Li and J. Liu, “A multi-agent genetic algorithm for community detection in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 449, pp. 336–347, 2016.
[13] M. Gong, B. Fu, L. Jiao, and H. Du, “Memetic algorithm for community detection in networks,” Phys. Rev. E, vol. 84, p. 056101, Nov 2011.

[14] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, and L. Jiao, “Greedy discrete particle swarm optimization for large-scale social network clustering,” Information Sciences, vol. 316, pp. 503–516, 2015, nature-Inspired Algorithms for Large Scale Global Optimization.

[15] F. Zou, M. Adler, and D. Jiang, “A multiobjective genetic algorithm for community detection in networks,” in 2009 21st IEEE International Conference on Tools with Artificial Intelligence, Nov 2009, pp. 379–386.

[16] ——, “A multiobjective genetic algorithm to find communities in complex networks,” IEEE Transactions on Evolutionary Computation, vol. 16, no. 3, pp. 418–430, June 2012.

[17] M. Gong, L. Ma, Q. Zhang, and L. Jiao, “Community detection in networks by using multiobjective evolutionary algorithm with decomposition,” Physica A: Statistical Mechanics and its Applications, vol. 391, no. 15, pp. 4050–4060, 2012.

[18] M. Gong, X. Chen, L. Ma, Q. Zhang, and L. Jiao, “Identification of multi-resolution network structures with multi-objective immune algorithm,” Applied Soft Computing, vol. 13, no. 4, pp. 1705–1717, 2013.

[19] 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,” Applied Soft Computing, vol. 53, pp. 285–297, 2015.

[20] 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: Statistical Mechanics and its Applications, vol. 513, pp. 4050–4060, 2018.

[21] C. Shi, Z. Yan, Y. Cai, and B. Wu, “Multi-objective community detection in complex networks,” Applied Soft Computing, vol. 12, no. 2, pp. 850–859, 2012.

[22] P. Wu and L. Pan, “Multi-objective community detection based on memetic algorithm,” PLOS ONE, vol. 10, p. 0126845, 05 2015.

[23] C. Bothorel, J. D. cruz, M. Magnani, and B. Micenkova, “Clustering attributed graphs: Models, measures and methods,” Network Science, vol. 3, no. 3, pp. 408–444, 2015.

[24] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks by using multiobjective evolutionary algorithm with decomposition,” in Proceedings of the 3rd International Workshop on Link Analysis Workshop, 18th International Conference on Artificial Intelligence, 2003, pp. 9–15.

[25] 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.

[26] Y. Zhou, H. Cheng, and J. X. Yu, “Clustering large attributed graphs: An efficient incremental approach,” in 2010 IEEE International Conference on Data Mining, Dec 2010, pp. 689–698.

[27] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, “A model-based approach to attributed graph clustering,” in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. New York, NY, USA: ACM, 2012, pp. 505–516.

[28] Y. Zhou, H. Cheng, and J. X. Yu, “Clustering attributed graphs,” in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 927–936.

[29] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, “Semantic community identification in large attribute networks,” in AAAI Conference on Artificial Intelligence, 2016.

[30] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” in 2013 IEEE 13th International Conference on Data Mining, Dec 2013, pp. 1151–1156.

[31] S. Pool, F. Bonchi, and M. V. Leeuwen, “Description-driven community detection,” ACM Trans. Intell. Syst. Technol., vol. 5, no. 2, pp. 28:1–28:24, Apr. 2014.

[32] L. Hu and K. C. C. Chan, “Fuzzy clustering in a complex network based on content relevance and link structures,” IEEE Transactions on Fuzzy Systems, vol. 24, no. 2, pp. 456–470, April 2016.

[33] Z. Li, J. Liu, and K. Wu, “A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks,” IEEE Transactions on Cybernetics, vol. 48, no. 7, pp. 1963–1976, July 2018.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, Apr 2002.

[35] C. Pizzuti and A. Socciowole, “Multiobjective optimization and local merge for clustering attributed graphs,” IEEE Transactions on Cybernetics, pp. 1–13, 2019.

[36] U. Brandes, M. Gaertler, and D. Wagner, “Engineering graph clustering: Models and experimental evaluation,” ACM Journal of Experimental Algorithmics, vol. 12, 2007.

[37] Y. Park and M. Song, “A genetic algorithm for clustering problems,” in Proceedings of the third annual conference on genetic programming, 1998, pp. 568–575.

[38] J. Bader and E. Zitzler, “HypE: An algorithm for fast hypervolume-based many-objective optimization,” Evolutionary Computation, vol. 19, no. 1, pp. 45–76, 2011.

[39] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, Dec 2007.

[40] J. Sun, H. Zhang, A. Zhou, Q. Zhang, K. Zhang, Z. Tu, and K. Ye, “Learning from a stream of nonstationary and dependent data in multiobjective evolutionary optimization,” IEEE Transactions on Evolutionary Computation, vol. 23, no. 4, pp. 541–555, Aug 2019.

[41] 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 and Evolutionary Computation, vol. 1, no. 1, pp. 32–49, 2011.

[42] R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[43] J. D. Schaffer, “Multiple objective optimization with vector evaluated genetic algorithms,” in Proceedings of the 1st International Conference on Genetic Algorithms. USA: L. Erlbaum Associates Inc., 1985, pp. 93–100.

[44] J. Handl and J. Knowles, “An evolutionary approach to multiobjective clustering,” IEEE Transactions on Evolutionary Computation, vol. 11, no. 1, pp. 56–76, Feb 2007.

[45] V. Krebs, “Books about US politics,” unpublished, http://www.orgnet.com, 2004.

[46] L. A. Adamic and N. Glance, “The political blogosphere and the 2004 U.S. election: Divided they blog,” in Proceedings of the 3rd International Workshop on Link Discovery. New York, NY, USA: ACM, 2005, pp. 36–43.

[47] J. Leskovec and J. J. McAuley, “Learning to discover social circles in ego networks,” in Advances in neural information processing systems, 2012, pp. 539–547.

[48] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, “Collective classification in network data,” AI magazine, vol. 29, no. 3, pp. 93–93, 2008.

[49] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, “Comparing community structure identification,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2005, no. 09, p. P09008, Sep 2005.

[50] P. Merz, “Advanced fitness landscape analysis and the performance of memetic algorithms: Evolutionary Computation, vol. 12, no. 3, pp. 303–325, 2004.

[51] H. R. Lourenço, O. C. Martin, and T. Stützle, Iterated Local Search: Framework and Applications. Boston, MA: Springer US, 2010, pp. 363–397.

[52] P. Merz and B. Freisleben, “Fitness landscape analysis and memetic algorithms for the quadratic assignment problem,” IEEE Transactions on Evolutionary Computation, vol. 4, no. 4, pp. 337–352, 2000.