A Node Classification-Based Multiobjective Evolutionary Algorithm for Community Detection in Complex Networks

Haipeng Yang, Bin Li, Fan Cheng, Peng Zhou, *Member, IEEE*, Renzhi Cao, and Lei Zhang, *Member, IEEE*

***Abstract*— Multiobjective evolutionary algorithms (MOEAs) have been widely used in community detection in recent years. However, most of the existing MOEA-based ones adopted the same search strategies for all nodes and ignored the differences between the nodes. In fact, the nodes in a complex network have different structural characteristics and are of different importance during the search process of the community detection problem. To this end, in this article, a node classification-based search scheme is first proposed, where different kinds of nodes are searched in different ways. To be specific, the nodes in the network are classified into two types of nodes, candidate central (CC) nodes and noncentral (NC) nodes, by mapping the nodes into a structural similarity-based embedding space. The CC nodes are likely to be the centers of communities, and the rough structure can be searched quickly through activating the CC nodes. Then, the NC nodes are assigned to the communities with the activated central nodes. Based on the proposed scheme, a node classification-based MOEA named NCMOEA is then proposed. In NCMOEA, a mixed representation is designed to effectively encode the two different kinds of nodes. In addition, corresponding genetic operators are then suggested to search the two categories of nodes in different ways. Furthermore, an initialization strategy is also designed for initializing the population with high quality and good diversity. The experimental results on 15 real-world networks and several synthetic networks demonstrate the superiority of the proposed NCMOEA over nine representative algorithms for community detection.**

***Index Terms*— Community detection, complex networks, evolutionary algorithm (EA), multiobjective optimization.**

NOMENCLATURE

G	Complex network, $G = (V, E)$.
V	Node set of the network.
E	Edge set of the network.
n	Number of nodes in the network and $\|V\| = n$.
m	Number of edges in the network and $\|E\| = m$.
A	Adjacent matrix of the network.
$\mathcal{C}$	Set of communities in the network, which is a partitioning of V.
C_i	ith community in the network.
c	Number of communities, $\|C\| = c$.
CC	Set of candidate central nodes.
NC	Set of noncentral nodes.
$deg(v)$	Degree of node v.
$\Gamma(v)$	Neighbor nodes of node v.
SM	Similarity matrix of nodes.
pop	Size of the population.
$maxgen$	Maximum number of generations.
pc	Probability of crossover.
iv	Number of interval generations between the two consecutive local searches.

Manuscript received 20 May 2022; revised 9 October 2022 and 10 November 2022; accepted 15 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61976001, Grant 61876184, Grant 62076001, and Grant 62176001; in part by the Key Projects of University Excellent Talents Support Plan of Anhui Provincial Department of Education under Grant gxyqZD2021089; and in part by the Natural Science Foundation of Anhui Province under Grant 2008085QF309. *(Corresponding author: Lei Zhang.)*

Haipeng Yang, Bin Li, Fan Cheng, Peng Zhou, and Lei Zhang are with the Information Materials and Intelligent Sensing Laboratory of Anhui Province and the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: haipengyang@126.com; ahulibin@163.com; chengfan@mail.ustc.edu.cn; zhoupeng@ahu.edu.cn; zl@ahu.edu.cn).

Renzhi Cao is with the School of Computer Science, Pacific Lutheran University, Tacoma, WA 98447 USA (e-mail: caora@plu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCSS.2022.3223159, provided by the authors.

Digital Object Identifier 10.1109/TCSS.2022.3223159

I. INTRODUCTION

COMMUNITY detection has received increasing attention from a large number of researchers because of its important applications in varied complex networks [1], including social networks [2] and biological networks [3]. To be specific, the task of community detection is to detect communities with dense intraconnections and sparse interconnections in the network [4]. Many algorithms were proposed for community detection by researchers, such as hierarchical clustering algorithms [5], modularity optimization-based algorithms [6], community structure enhancement algorithms [7], spectral clustering algorithms [8], label propagation-based algorithms [9], random walk-based algorithms [10], and evolutionary algorithm (EA)-based methods [11].

Among them, EA-based algorithms achieve good performance. Some of the EA-based methods are single-objective optimization algorithms [12], [13], [14]. Besides these algorithms, the multiobjective EA (MOEA)-based methods are attractive because the two aspects (i.e., intraconnections should be dense, while interconnections should be sparse) of the community detection problem can be seen as two conflicting objectives and MOEAs naturally meet the optimization of the two objectives. MOEAs have been demonstrated to be widely used for community detection in recent years [11]. There are several advantages of adopting MOEAs for community detection: 1) MOEAs can determine the number of communities automatically; 2) MOEAs can provide a set of

Pareto optimal solutions (i.e., a set of network divisions at different hierarchical levels) instead of one optimal solution; and 3) MOEAs can overcome the resolution limitation of modularity [15] through optimizing multiple conflicting objective functions.

The first MOEA for community detection, called multiobjective genetic algorithms for networks (MOGA-Net), was proposed by Pizzuti [16]. In MOGA-Net, the community score and community fitness were used as the two objective functions, which achieved better performance than the single-objective algorithms because it could avoid the problem of the resolution limit [15] in the modularity optimization. MOGA-Net has shown the competitiveness of multiobjective optimization for community detection. After that, a variety of MOEA-based algorithms with different frameworks were proposed, such as the MOEA with decomposition for community detection (MOEA/D-Net) [17], the decomposition-based multiobjective discrete particle swarm optimization (MODPSO)algorithm [18], the multiobjective discrete algorithm based on teaching-learning-based optimization (MODTLBO/D) [19], the multiobjective backtracking search optimization algorithm with decomposition (MODBSA/D) [20], the multiobjective particle swarm optimization based on network embedding (NE-PSO) [21], and a network reduction-based MOEA (RMOEA) [22]. These MOEA-based community detection algorithms have achieved good performance. However, most of the existing MOEA-based community detection algorithms searched the nodes in the network equally during the searching process, while the structural characteristics of nodes were ignored, which could deteriorate the performance of the algorithm.

To this end, unlike these existing works, a node classification-based MOEA is proposed in this article, where different kinds of nodes with different structural characteristics are fully considered and searched in different ways. To be specific, the nodes in the network are divided into two categories, that is, candidate central (CC) nodes and noncentral (NC) nodes. Then, different search strategies are suggested for the two kinds of nodes. Specifically, the CC nodes are likely to be the central nodes of the communities, some of them will be activated in the search process, and a general structure of communities can be determined by the activated central nodes quickly. The NC nodes do not have obvious central node structural characteristics. Thus, during the searching process, the algorithm just focuses on which community these nodes should belong to. The main contributions of this article can be summarized as follows.

1) We propose a node classification-based search scheme, where different kinds of nodes are searched in different ways. Specifically, nodes in the network are classified into CC nodes and NC nodes according to their structural characteristics. During the searching process, the scheme optimizes which CC nodes should be activated and which communities should the remaining NC nodes belong to. With this scheme, the rough structure of communities can be determined quickly by the activated central nodes, and the algorithm can focus on the community assignment for the NC nodes.
2) We propose a node classification-based MOEA named NCMOEA for community detection based on the proposed search scheme. In NCMOEA, a novel mixed representation is suggested to effectively encode community centers and structure at the same time. Based on the mixed representation, tailored crossover and mutation operators are designed for searching the different kinds of nodes in different ways. Furthermore, a novel initialization strategy is also designed for producing better initial solutions.
3) To validate the effectiveness of NCMOEA, we compare the results of NCMOEA with one state-of-the-art non-EA-based baseline algorithm and eight representative EA-based baseline algorithms on 15 real-world networks and several Lancichinetti–Fortunato–Radicchi (LFR) synthetic networks [23]. We also implement six variants of NCMOEA to verify the effectiveness of the proposed node classification strategy, the mixed representation, the crossover and mutation operators, and the initialization strategy. According to the results of the experiments, the proposed NCMOEA is very competitive and superior over the baseline algorithms on most of the networks.

The rest of this article is organized as follows. In Section II, the definition of the community detection and the existing related work are reviewed. In Section III, the details of the proposed node classification-based multiobjective algorithm NCMOEA are presented. The experimental results of NCMOEA compared with nine baseline algorithms and six variants of NCMOEA are given in Section IV. The conclusions and future work are introduced in Section V.

II. Preliminaries and Related Work

In this section, we will give some preliminaries about the community detection problem and introduce some related works on EA-based community detection algorithms for community detection.

A. Community Detection Problem

The goal of the community detection task is to divide nodes in the network into different groups (called communities), where the nodes in the same community have dense connections and those in different communities have sparse connections [4]. In this article, we just consider the nonoverlapping community detection problem (i.e., each node belongs to one and only one community). The problem can be formally defined as follows: $G = (V, E)$ is a given network, where $V = \{v_1, v_2, \ldots, v_N\}$ is the node set and $E = \{(i, j)|v_i, v_j \in V, i \neq j\}$ is the edge set. The network can be divided into communities $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$, where $C_i \subset V$, $C_i \neq \emptyset$, $\cup_{i=1}^{c} C_i = V$, $C_i \cap C_j = \emptyset, \forall i \neq j, i, j \in \{1, 2, \ldots, c\}$, and c is the number of the detected communities. The notations used in this article are summarized in the Nomenclature.

B. Related Work

In the EA-based algorithms, many single-objective optimization algorithms performed well in a community detection

task. For example, Chang et al. [13] suggested a chemical reaction optimization-based community detection algorithm with dual representation, which designed two operators based on the two representations and get a good balance between exploration and exploitation. Xiao et al. [14] proposed a symbiotic organisms search community detection algorithm, which used neighborhood information of nodes to guide community optimization and assist the global search. Since this article focuses on MOEAs for community detection, several representative MOEA-based algorithms are reviewed in the following.

In 2009, a multiobjective genetic algorithm was proposed by Pizzuti [24], called MOGA-Net, in which community score and community fitness were used as objective functions. MOGA could determine the number of communities automatically and get a set of network divisions at different hierarchical levels. In 2012, Gong et al. [17] proposed an MOEA/D-Net, which selected two objective functions: negative ratio association (NRA) and ratio cut (RC). In the same year, Shi et al. [25] developed a multiobjective community detection algorithm based on PESA-II [26], called MOCD, which set intra and inter as two objectives to optimize the two aspects of the community quality. In 2016, Attea et al. [27] also suggested an MOEA with a heuristic operator for community detection. Note that these algorithms all used the locus-based representation, which treated the nodes in the network equally, and thus, they ignored the difference of the importance of different nodes in the process of searching.

Gong et al. [18] suggested MODPSO in 2014, where the kernel k-means (KKM) and RC are used as objective functions. KKM can measure the links in the same community and RC can measure the links between different communities. To make the links in intracommunity dense and the links in intercommunities sparse, the optimization algorithm minimizes KKM and RC. In 2016, Chen et al. [19] proposed MODTLBO/D, where a discrete teaching-learning-based optimization algorithm was adopted for community detection. After that, an MODBSA/D was proposed by Zou et al. [20]. In MODPSO, MODTLBO/D, and MODBSA/D, the community detection was optimized as a discrete problem and the label-based representation was adopted. In the three algorithms, the community labels were treated as discrete variables and there was no different treatment for different nodes during the search process.

In 2020, a network RMOEA [22] was proposed by ours for large-scale community detection, which reduced the search space with the proposed network reduction strategies and performed well on large-scale networks. RMOEA just considered the local community reduction, while it did not design different strategies for searching nodes with different structural characteristics. Recently, Liu et al. [21] proposed a multiobjective NE-PSO, where an arbitrary-order proximity-preserved embedding (AROPE) [28] was adopted to map the nodes into a low-dimensional space. The embedding focused on the proximity of nodes, and however, the structural difference of different nodes was still not considered.

The related works discussed above were designed for nonoverlapping community detection. There are also some MOEA-based methods proposed for overlapping community detection [29], [30], [31] and some algorithms proposed for different kinds of networks, such as attributed networks [32] and dynamic networks [33].

In summary, these MOEA-based community detection algorithms above have shown good performance for community detection, and however, most of the existing MOEA-based algorithms searched the nodes in the network equally and ignored the structural characteristics of nodes in the network during the searching process. Unlike these works, in this article, a node classification-based MOEA named NCMOEA is proposed, where different kinds of nodes with different structural characteristics are fully considered and thus can improve the detection performance of MOEAs. Note that there are some existing works in non-EAs, which harnessed the importance of central nodes to design some effective strategies for detecting community structure [34], [35], [36], [37]. Different from them, in this article, we focus on designing novel MOEA by considering the nodes with different structural characteristics in different ways in the process of optimization.

III. Proposed Algorithm NCMOEA

In this section, we will present the proposed NCMOEA method for community detection, including the proposed node classification-based search scheme, the mixed representation scheme, genetic operators, the population initialization, and the overall framework.

A. Node Classification-Based Search Scheme

1) Main Idea: The main idea of the proposed algorithm is that the nodes are classified into different categories based on the structural characteristics of nodes, and then, they are optimized in different ways during the evolutionary process. In the proposed classification method shown in Section III-B, the nodes are classified into two categories: the first category is CC nodes, this kind of nodes is important because they have many neighbors with dense links, and they are likely to be the central nodes of communities. In the process of search, some of the CC nodes are chosen and activated. Then, the rough structure of a community can be identified quickly. The second category is NC nodes. These nodes are assigned to the communities determined by the activated central nodes. In the proposed search scheme, after the node classification, the central nodes of communities with high-quality structure are detected from the CC nodes; for the NC nodes, the scheme focuses on which community they should belong to.

Fig. 1 shows the main idea of the proposed search scheme. Fig. 1(a) shows an illustrative network and the nodes in the network are classified into two classes, CC nodes and NC nodes, with the proposed classification strategy shown in Algorithm 1. In Fig. 1(b), the two kinds of nodes are marked in the network with different shapes, where triangles denote the CC nodes and squares denote the NC nodes. For a network $G = \langle V, E \rangle$, $\mathrm{CC} \cup \mathrm{NC} = V$ and $\mathrm{CC} \cap \mathrm{NC} = \emptyset$. After the searching process, a possible community structure as shown in Fig. 1(c) can be found. The nodes in different colors are in different communities and three triangles in the figure denote the community centers of the three found communities. The three activated centers are chosen from the CC nodes.

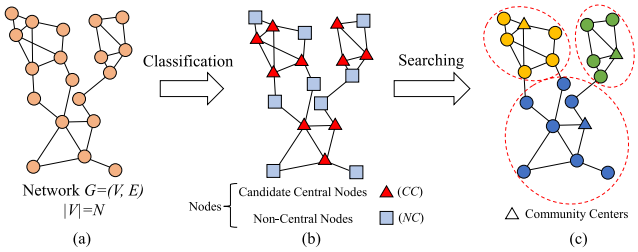

Fig. 1. Main idea of the proposed node classification search scheme. (a) Illustrative network. (b) Nodes in the network are divided into two categories: CC nodes (triangles) and NC nodes (squares). (c) After searching, three community centers marked with triangles are activated and the community structure is determined, where different colors denote different communities.

2) Challenges: From the proposed search scheme given above, it can be found that there are three main challenges that need to be solved: 1) how to classify the nodes into different types according to the structural characteristics; 2) how to design a suitable representation for encoding the different kinds of nodes; and 3) how to design genetic operators for searching different kinds of nodes in the evolution. To solve the first challenge, we suggest a node classification strategy, which can classify the nodes into two categories (see Section III-B). To solve the second challenge, we propose a mixed representation scheme for encoding the two kinds of nodes, which can represent the activate states of the central nodes and the community assignment of the NC nodes at the same time (see Section III-C). To solve the third challenge, the genetic operators corresponding to the proposed mixed representation are designed, which includes a two-phase crossover operator and a two-phase mutation operator (see Section III-E).

B. Node Classification Strategy

In this work, we classify the nodes into different categories according to their structural characteristics. However, the nodes' structural characteristics are usually not significant. Although there are many metrics for measuring some aspects of the node structural differences, it is still difficult to choose metrics and thresholds for classifying because a fixed setting cannot fit various networks.

In this article, we propose a node classification strategy based on the network embedding method, which can preserve the similarity of local node structures in the embedding result. To be specific, in the node classification strategy, struc2vec [38] is adopted to get the embedding of the network, which focuses on the structural similarity of the nodes in the network. Two nodes, which are structurally similar, will be close in the embedding space, independently of their position in the network and node labels in their vicinity. The embedding result of struc2vec is used for measuring the structural similarity and the nodes are classified according to the obtained embedding vectors. In this article, the nodes are divided into two groups (i.e., the central nodes and the NC nodes) according to the structural similarity with the struc2vec embedding of the network.

In order to distinguish the central nodes and the NC nodes, the centrality in [39] is adopted. The centrality is defined as follows:

$$\text{centrality}(v) = \deg(v) + \sum_{u \in \Gamma(v)} \left(\deg(u) + \sum_{w \in \Gamma(u)} \deg(w) \right) \tag{1}$$

where $\Gamma(v)$ is the set of neighbors of node v and $\deg(v)$ is the degree of node v. The centrality of a node is calculated by considering the degrees of neighborhood composed of two-layer neighbors. A large centrality value indicates that the node is important and it is more likely to be a central node of community. In the node classification strategy, we just use the maximum value of centrality to find the most representative node of the CC nodes.

Algorithm 1 NodeClassification

Input: A: the adjacent matrix of network;
Output: CC: the set of candidate central nodes; NC: the set of non-central nodes;
1: $cen \leftarrow$ calculate the $centrality$ for all nodes according to Eq. (1);
2: $Emb \leftarrow$ adopt struc2vec on A;
3: $rnode1 \leftarrow \arg\max_i cen(i)$;
4: $rnode2 \leftarrow$ the node with furthest distance to $rnode2$ in Emb;
5: $k \leftarrow 2$;
6: $initial\_centers \leftarrow \{rnode1, rnode2\}$;
7: $categories \leftarrow kmeans(Emb, k, initial\_centers)$;
// $categories$ contains of 2 sets of nodes
8: $CC \leftarrow$ the set containing $rnode1$ in $categories$;
9: $NC \leftarrow$ the set containing $rnode2$ in $categories$;

Algorithm 1 shows the process of the node classification strategy. First, the centralities of nodes in the network are calculated (Line 1) and the embedding method struc2vec is adopted (Line 2). Then, two representative nodes of the two categories are chosen (Lines 3 and 4). Specifically, the node with maximum value of centrality, denoted as rnode1, is selected as the representative node of the central nodes. In the struc2vec embedding space, the nodes that are structurally similar will be close, so the furthest node from rnode1 is the node with the most different structural characteristics, denoted as rnode2, and it is selected as the representative node of NC nodes. After that, the k-means [40] algorithm is adopted (Lines 5 and 6), where k is set to 2 and the initial centers are fixed at rnode1 and rnode2. Finally, we can obtain the set of CC nodes and the set of NC nodes (Lines 7 and 8).

Fig. 2 shows an example of the classification strategy. The illustrative network in Fig. 2(a) consists of 16 nodes and 22 edges. Then, the embedding result obtained with struc2vec is shown in Fig. 2(b). For display convenience, the embedding dimension is set to 2. In the embedding space, close nodes have similar structural characteristics. The centrality of all nodes in the network has been calculated and node 7 is the node with the highest centrality. Thus, node 7 and the node furthest from it in the embedding, node 6, are chosen as the two representative nodes. After that, a k-means algorithm is adopted, where k is set to 2 and the two nodes are used as initial points. The classification result is shown in

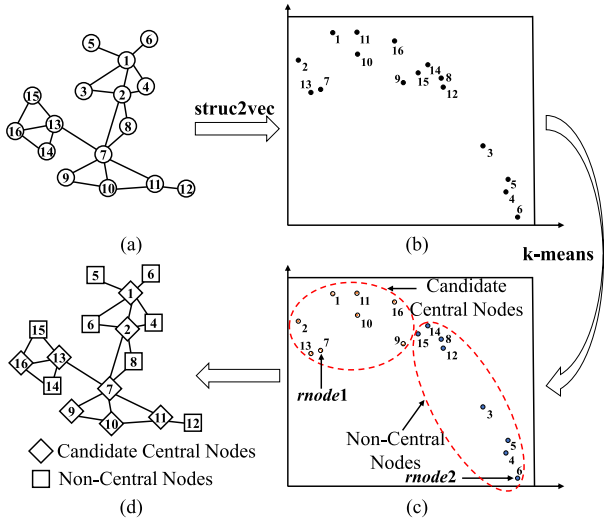

Fig. 2. Example of the network node classification strategy. (a) Illustrative network. (b) Embedding result of struc2vec and the embedding dimension is set to 2. (c) Two respective nodes are chosen and the nodes are classified into two categories. (d) Nodes in the network are classified into two categories of nodes in the network.

Fig. 2(c) and (d), where the nodes with similar structural features to node 7 are CC nodes ({1,2,7,9,10,11,13,16}) and the nodes different from it are NC nodes ({3,4,5,6,8,12,14,15}).

C. Mixed Representation Scheme

In order to encode different categories of nodes, a mixed representation scheme is designed. In the representation scheme, CC nodes and NC nodes use different representations. The CC nodes can be activated, each activated central node corresponds to a community, and each community just has one activated central node. In other words, there is a one-to-one correspondence between the community and the activated central node. Thus, the central node of the community can be used as the label of the community. The NC nodes and the inactivated CC nodes are assigned to the communities obtained by the activated central nodes.

In the mixed representation scheme, the nodes are divided into two parts: the CC nodes part and the NC nodes part. For the CC nodes, the representation needs to be able to indicate whether each node is activated or not and which community each inactive node belongs to. For the NC nodes, they cannot be activated and only need to focus on which community they belong to. In the mixed representation scheme, the gene positions corresponding to the CC nodes can be set to "-1," which means that the CC nodes are activated. The positions corresponding to inactivated central nodes and the NC nodes are set to the labels of communities they belong to. Formally, the mixed representation is denoted as

$$I\langle \mathrm{CC}, \mathrm{NC}\rangle = \left[l_{v_1'}, l_{v_2'}, \ldots, l_{v_N'}\right] \tag{2}$$

where N is the number of nodes in the network, CC is the set of CC nodes, and NC is the set of NC nodes. If node $v_i \in$ CC, $l_{v_i} \in \{-1, 1, 2, \ldots, N\}$; if node $v_j \in$ NC, $l_{v_j} \in \{1, 2, \ldots, N\}$. A CC node can be represented by "-1" or community labels, where "-1" means activated. The NC nodes are represented by

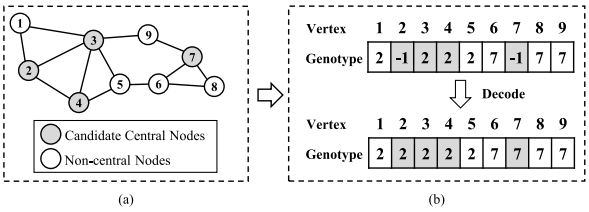

Fig. 3. Illustrative example of the mixed representation scheme. (a) Illustrative network, where the nodes with deep color are CC nodes and the others are NC nodes. (b) Individual with mixed representation and the individual decoded from it through replacing "-1" with the index of the activated central nodes.

community labels. In this representation, the community label is the index of its center, so the labels are positive integers.

Fig. 3 shows an illustrative example for the mixed representation scheme, where the network contains nine nodes and 13 edges. Suppose that there is an individual denoted as $I = [2, -1, 2, 2, 2, 7, -1, 7, 7]$. In the figure, the dark positions indicate that the corresponding nodes are CC nodes, and the genotype of these positions can be "-1," which means that the central node is activated. The genotypes of inactive nodes and NC nodes are community labels, where the activated center node of the community is used as the label of the community. Each activated central node corresponds to a community and each community just has one activated node. It is easy and fast to decode the individual: for each activated CC node, replace "-1" with the label index of the node, and the community partition can be obtained. For example, $I(2)$ is set to 2 and $I(7)$ is set to 7; the decoded individual is [2, 2, 2, 2, 2, 7, 7, 7, 7]. Like the label-based representation for decoding, the nodes with the same label are in the same community, and thus, the network is divided into two communities: $C_1 = \{v_1, v_2, v_3, v_4, v_5\}$ and $C_2 = \{v_6, v_7, v_8, v_9\}$.

D. Population Initialization Strategy

For the mixed representation, the CC nodes part and the NC nodes part are related to each other. The choosing of initial central nodes could influence the community structure of the solutions. If we only select active nodes randomly and assign NC nodes, the quality of the initial population will be low. Therefore, we propose a novel population initialization strategy.

In the initialization strategy, two ways to generate population individuals are considered: the first way is to initialize the CC nodes first and then determine the community structure according to the activated central nodes; the other way is to generate a community structure with the neighborhood of nodes and then activate central nodes for communities.

The detailed procedure of the initialization strategy is shown in Algorithm 2. For half of the population individuals, we generate a random number between 2 and $\sqrt{N}$ for each individual as the number of active central nodes, where N is the number of nodes in the network, and then activate central nodes randomly. The other nodes are assigned to the communities of the active central nodes according to the similarity matrix of nodes (SM). Each unlabeled node is assigned to the community of the active central node with the

Algorithm 2 Initialization

Input: A: the adjacent matrix of network; SM: the similarity matrix; N: the number of the nodes in the network; CC: the set of candidate central nodes; NC: the set of non-central nodes; pop: the size of population;
Output: P: the initial population;

$P \leftarrow \emptyset$;
for $i = 1$ to $pop/2$ **do**
  $indi \leftarrow$ zeros(1, N);
  $n \leftarrow$ A random integer between 2 to $\sqrt{N}$;
  $AC \leftarrow$ Select n nodes from CC randomly;
  $indi(v_q) \leftarrow -1, v_q \in \{v_l | v_l \in AC\}$;
  $AN \leftarrow \{v | indi(v) == 0\}$;
  $indi \leftarrow AssignNodes(AC, AN, SM)$;
  $P \leftarrow P \cup indi$;
end for
for $i = pop/2 + 1$ to pop **do**
  $x \leftarrow$ zeros(1, N);
  $x \leftarrow InitilizeLocusIndividual(A)$;
  $x \leftarrow DecodeLocusIndividual(x)$;
  for each community $comm$ in x **do**
    if $comm \cap CC! = \emptyset$ **then**
      $ac \leftarrow$ Randomly select a node in $comm \cap CC$;
      $indi(ac) \leftarrow -1$;
      $indi(v_q) \leftarrow ac, v_q \in \{v_l | v_l \in comm\}$;
    end if
  end for
  $indi(AC) \leftarrow -1$;
  $AN \leftarrow \{v | indi_v == 0\}$;
  $indi \leftarrow AssignNodes(indi, AC, AN, SM)$;
  $P \leftarrow P \cup indi$;
end for

highest similarity to it, and then, the node is labeled with the index of the active central node of the community.

For the other half of the population individuals, first, we initialize a solution with locus-based representation. Generate a temporary vector x of length N, choose a node i randomly, then choose a node j in its neighbors randomly, and set x_i to j. Repeated the operator until each position of x has been set to a node. After that, we can get a solution with locus-based representation. Then, the decoding method [24] of locus-based representation is adopted: the value j of x_i is seen as a link between nodes i and j, and the nodes in the same component are in the same community. In each community, a CC node is chosen randomly, then set the position in the individual to "-1," and the community label is the node index. If there is no CC node in a community, the community will be broken up and the nodes in community will be assigned to other communities.

E. Genetic Operators

In the mixed representation scheme, the two kinds of nodes (CC nodes and NC nodes) are two different parts and the genotypes corresponding to NC nodes depend on the central nodes because the central node of the community is used as the label of the community and the genotype of the NC nodes is these labels. The relationship of the two parts causes that traditional genetic operators cannot be used for the mixed representation because they could generate illegal and meaningless solutions. In this article, a crossover operator and a mutation operator are proposed and they are both two-phase operators to deal with the two parts of the mixed representation.

Algorithm 3 Crossover

Input: P: the parent individuals; CC: the set of candidate central nodes; NC: the set of non-central nodes; SM: the similarity matrix of the nodes; pc: the crossover probability; pcc: the probability of central nodes crossover; pop: the size of population;
Output: Off: the offspring individuals;

1: $Off \leftarrow \emptyset$;
2: **for** $i = 1$ to $pop/2$ **do**
3: $p1 \leftarrow P_{2*i-1}$, $p2 \leftarrow P_{2*i}$;
4: $c1 \leftarrow p1$, $c2 \leftarrow p2$;
5: $ra \leftarrow$ a random decimal in the interval (0,1);
6: **if** $ra < pc$ **then**
  Phase 1:
7: $pv \leftarrow$ Random $1 * N$ decimal vector in the interval (0, 1);
8: $cm \leftarrow pv < pcc$; // cm is a binary mask
9: $c1(v_q) \leftarrow p2(v_q)$, $v_q \in \{v_l | v_l \in CC$ and $cm(v_l) == 1\}$;
10: $c2(v_q) \leftarrow p1(v_q)$, $v_q \in \{v_l | v_l \in CC$ and $cm(v_l) == 1\}$;
  Phase 2:
11: **for** each changed position in $child1$ **do**
12: **if** the position j is changed to -1 **then**
13: $CM \leftarrow \{v | p2(v) == j$ and $c1(v)! = -1\}$;
14: $c1(v_k) \leftarrow j$, $v_k \in \{v_l | v_l \in CC\}$;
15: **else**
16: $CM \leftarrow j \cup \{v | c1(v) == j\}$;
17: $c1(v_k) \leftarrow 0$, $v_k \in \{v_l | v_l \in CC\}$;
18: **end if**
19: **end for**
  // Assign the unlabeled nodes
20: $AC1 \leftarrow \{v_q | c1(v_q) == -1\}$;
21: $NC1 \leftarrow \{v_q | c1(v_q) == 0\}$;
22: $c1 \leftarrow AssignNodes(c1, AC1, NC1, SM)$;
23: Do the same operations (Lines 11-22) for $c2$ and $p1$;
24: **end if**
25: $Off \leftarrow Off \cup c1 \cup c2$;
26: **end for**

1) Crossover: In the crossover operator, the CC nodes are considered first as the first phase, and the NC nodes' crossover is the second phase. The crossover operator is shown in Algorithm 3.

In the first phase of the crossover operator (Lines 7–10), a crossover operator such as uniform crossover is proposed for the CC nodes part of the individual. A binary mask cm is generated to control whether the value of a CC node position is exchanged (Lines 7 and 8). Each binary value controls the crossover of each CC node, and the probability of setting the value to 1 is defined as follows:

$$\text{pcc} = 0.5 * (1 - \text{gen/maxgen}) \tag{3}$$

where gen is the current number of generations during the evolution process and maxgen is the maximal number of the generations of the algorithm. The greater the value of pcc, the greater the possibility of the central nodes crossover. It is worth noting that pcc is different from the probability of the crossover operator (pc), which controls whether the crossover operator performs or not, and pcc just controls the number of the intensity of the crossover. The value of pcc is high when gen is small because we want the activated states of more

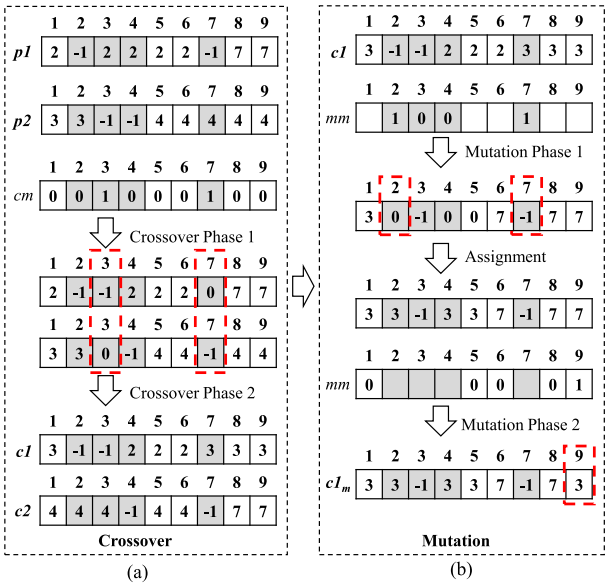

Fig. 4. Illustrative example of the genetic operators. (a) Using crossover operator, the two parents $p1$ and $p2$ generate two children $c1$ and $c2$. (b) Using mutation operator, $c1$ and $c2$ are changed to $c1_m$ and $c2_m$, respectively.

nodes to change in the early stages of the evolution so as to find a good rough structure quickly.

If the corresponding mask value is 1 and the activate states of the CC node in two parents are different, the values on the two positions exchange in the two offspring individuals. If an activated central node changes to inactive, the genotype will be changed to 0, where "0" means that the node does not belong to any community currently and it will be assigned in the next steps.

After the crossover of central nodes, in the second phase (Lines 11–23), some inactive central nodes and NC nodes also perform crossover. If a central node is activated in the offspring individual by crossover, the NC nodes in this community will also be put into the community in the offspring individual (Lines 13 and 14). If an activated central node is changed into inactive, the genotypes of nodes in its community will be set to 0 (Lines 16–18).

Then, the nodes whose genotypes are 0 will be assigned to communities with similarity matrix SM. Specifically, the diffusion kernel similarity [41] is adopted to measure the similarity between nodes, which is defined as

$$\mathrm{SM} = e^{-\beta \times L} \tag{4}$$

$$L = D - A \tag{5}$$

where β is suggested to be 1 and L is the Laplace matrix. D is a diagonal matrix and each element of the principal diagonal is the degree of a corresponding node, and A is the adjacent matrix of the network. The NC nodes and the inactive central nodes are assigned to the community with the most similar active central node.

As shown in Fig. 4(a), the two individuals parent1 and parent2 are parents, and the corresponding genotypes of nodes v_3 and v_7 are randomly selected for crossover. In the first phase, the active states of the CC nodes v_3 and v_7 are changed. Then, in the second phase, for the first child, v_3 is a new active central node, and the member nodes of the community in parent2 (v_1) are also labeled as 3 in child1. Node v_3 in child2 is changed to inactive, and the genotypes of v_1 and v_2 are 3, so their genotypes will be set to 0; then, the nodes are assigned; and finally, they are assigned to the community of v_4. The same things are done on the changed central node position v_7 in the two children, and the final results are child1 and child2 shown in Fig. 4(a).

Algorithm 4 Mutation

Input: A: the adjacent matrix of the network; Off: the offspring after crossover; N: the number of nodes in the network; CC: the set of candidate central nodes; NC: the set of non-central nodes; SM: the similarity matrix of the nodes; pop: the size of population;
Output: Off: the offspring after mutation;

1:
2: **for** each individual $indi$ in Off **do**
   Phase 1:
3: $pv \leftarrow$ Random $1 * N$ decimal vector in the interval (0,1);
4: $mm \leftarrow pv < 1/N$; // mm is a binary vector
5: **for** v_{cc} in $\{v | v \in CC$ and $mm(v) == 1\}$ **do**
6: **if** $indi(v_{cc}) == -1$ **then**
7: $indi(v_{cc}) \leftarrow 0$;
8: $indi(v_q) \leftarrow 0$, $v_q \in \{v_l | indi(v_l) == v_{cc}\}$;
9: **else**
10: $indi(v_{cc}) \leftarrow -1$;
11: $indi(v_q) \leftarrow v_{cc}$, $v_q \in \{v_l | indi(v_l)! = -1$ and $v_l \in \Gamma(v_{cc})\}$;
12: **end if**
13: **end for**
   // Assign the unlabeled nodes
14: $ACi \leftarrow \{v_q | indi(v_q) == -1\}$;
15: $NCi \leftarrow \{v_q | indi(v_q) == 0\}$;
16: $indi \leftarrow AssignNodes(indi, ACi, NCi, SM)$;
   Phase 2:
17: **for** v_{nc} in $\{v | v \in NC$ and $mm(v) == 1\}$ **do**
18: Randomly choose a node $v_i \in \Gamma(v_{nc})$;
19: **if** $indi(v_i) == -1$ **then**
20: $indi(v_{nc}) \leftarrow v_i$;
21: **else**
22: $indi(v_{nc}) \leftarrow indi(v_i)$;
23: **end if**
24: **end for**
25: **end for**

2) Mutation: The mutation is also divided into two phases, where the first phase is the mutation of the CC nodes and the second phase is the mutation of the NC nodes. Since the length of the individual is the number of nodes in the network (i.e., N), the probability of each node's position is set to $1/N$.

In the first phase (Lines 3–16), the CC nodes mutate. If the central node is active, it will change to inactive. The value of its position is set to 0 in the individual, and the positions whose community label is the central node are also set to 0. If the central node is inactive, it will be activated. The value of its position is set to -1, then the NC nodes and inactive central nodes in its direct neighbors are assigned to its community. After that, the nodes with genotype "0" are assigned to the changed communities.

In the second phase (Lines 17–24), the NC nodes mutate. The nodes whose genotypes are 0 are assigned to communities according to the similarity matrix SM (Line 14). When an

NC node mutates, a random neighbor in different community is chosen and the label of the mutated node will change to the neighbors. If the neighbor is an active central node, the label is set to the index of the neighbor node. Note that if an individual does not have more than one active central node, in order to avoid the illegal solution and meaningless solution, two CC nodes will be activated randomly and the other nodes will be assigned.

For example, in Fig. 4(b), the mutation operator is performed on child1. In phase 1, the CC nodes v_2 and v_7 are changed. Node v_7 is a new active central node and its neighbors' genotypes are set to 7; node v_2 is changed to inactive and the genotypes of the nodes in the community labeled as 2 are set to 0. Then, the nodes with genotypes "0" are assigned to the changed communities. In phase 2, suppose that the node v_9 mutates, its genotype is changed to the genotype of a random neighbor. After the mutation, child1 is changed to child1_m and the same operator is performed on other offspring individuals.

F. Overall Procedure of NCMOEA

In the proposed NCMOEA, the two objective functions of [18], KKM and RC, are adopted to evaluate the individuals during the evolutionary optimization process. They can be calculated as follows:

$$\text{minimize} \begin{cases} \text{KKM} = 2(n-c) - \sum_{i=1}^{c} \dfrac{L(C_i, C_i)}{|C_i|} \\ \text{RC} = \sum_{i=1}^{c} \dfrac{L\left(C_i, \overline{C_i}\right)}{|C_i|} \end{cases} \tag{6}$$

where c denotes the number of the detected communities, $L(C_i, C_i) = \sum_{i \in C_i, j \in C_i} A_{i,j}$ denotes the number of intralinks in community C_i, and $L(C_i, \overline{C_i}) = \sum_{i \in C_i, j \in \overline{C_i}} A_{i,j}$ denotes the number of interlinks between C_i and other communities (A is the adjacent matrix of the network and $A_{i,j} \in \{0, 1\}$ is the number of edges between nodes i and j).

The overall procedure of the proposed NCMOEA is presented in Algorithm 5. The inputs of the algorithm are defined as follows: A is the adjacent matrix of the network, popsize is the number of individuals in the population, maxgen is the maximum number of evolutionary iterations, and interval is the number of interval generations between local searches. NCMOEA consists of two steps: node classification (Line 1) and evolutionary optimization (Lines 2–15).

In step 1, the node classification strategy (see Algorithm 1) (Lines 1) divides the nodes in the network into CC nodes and NC nodes. In step 2, the evolutionary optimization (Lines 2–15) starts. The evolutionary framework of NCMOEA is similar to the framework of NSGA-II [42]. TournamentSelect, NonDominatedSort, and EnvironmentSelect mean binary tournament selection, nondominated sort, and environment selection, respectively. These are basic routines of EA and they are also adopted from [42]. First, the algorithm calculates the SM (Line 2). Then, the initial population is generated with initialization strategy Initilization (see Algorithm 2) (Line 3), CC nodes and NC nodes are different in the mixed representation scheme, and the genetic operators, Crossover (see Algorithm 3) and Mutation (see Algorithm 4), are used for generating offspring individuals. KKM and RC are two objective functions [see (6)] and they are used for evaluating the individuals (Line 9); then, environment selection selects individuals for the next-generation population (Line 10). To further improve the quality of the solutions, we adopt the local search suggested in [13] on the Pareto optimal solutions. After the evolution, the nondominated solution set (Line 15) in the final population is the obtained result.

Algorithm 5 NCMOEA

Input: A: the adjacent matrix of the network; $popsize$: the size of population; $maxgen$: the max number of generation of evolutionary; iv: the interval between two local searches; pc: the probability of crossover;
Output: PF: the final non-dominated solutions;
Step 1: Node Classification
1: $NC, CC \leftarrow NodeClassification(A)$;
Step 2: Evolutionary Optimization
2: $SM \leftarrow$ Calculate the similarity as Eq. (4);
3: $Pop \leftarrow Initialization(A, SM, CC, NC, popsize)$;
4: **for** $gen = 1$ to $maxgen$ **do**
5: $P \leftarrow TournamentSelect(Pop)$;
6: Calculate pcc according to Eq. (3);
7: $Off \leftarrow Crossover(Parents, CC, NC, SM, pc, pcc, popsize)$;
8: $Off \leftarrow Mutation(A, Off, N, CC, NC, SM, popsize)$;
9: Evaluate Off with KKM and RC;
10: $Pop \leftarrow EnvironmentSelect(Pop \cup Off)$;
11: **if** $gen\%iv == 0$ **then**
12: $Pop \leftarrow LocalSearch(Pop)$;
13: **end if**
14: **end for**
15: $PF \leftarrow NondominatedSort(Pop)$;

IV. Experimental Results

In this section, we will verify the performance of the proposed NCMOEA through experiments on 15 real networks and two groups of LFR synthetic networks in comparison with nine representative baselines and six variants of NCMOEA.

A. Experimental Settings

1) Comparison Algorithms: The performance of NCMOEA is compared with one state-of-the-art non-EA-based algorithm (i.e., community structure enhancement (CSE)-based community detection algorithm [7]) and six representative EA-based community detection algorithms, namely, cooperative coevolutionary modularity identification (CoCoMi) [12], dual-representation chemical reaction optimization (DCRO) [13], elite symbiotic organism search for fuzzy community detection (SOSFCD) [43], MOGA-Net [24], MODPSO [18], MODTLBO/D [19], network RMOEA [22], and a nondominated sorting genetic algorithm for community detection (NSGA-III-KRM) [44]. In these EA-based baseline algorithms, CoCoMi, DCRO, and SOSFCD are three single-objective EA-based algorithms and the remaining five ones are MOEA-based algorithms.

There are also six variants of NCMOEA, namely, NCMOEA with no classification (NCMOEA-NC), NCMOEA with

TABLE I
CHARACTERISTICS OF 15 REAL-WORLD NETWORKS. NOTE THAT "−" MEANS THAT THE REAL COMMUNITY STRUCTURE IS UNKNOWN

Real Networks	Nodes	Links	AD	RC	CC	AC	density
Karate [45]	34	78	4.59	2	0.57	-0.48	0.1390
Dolphin [46]	62	159	5.13	2	0.26	-0.04	0.0841
Polbook [47]	105	441	8.4	3	0.49	-0.13	0.0808
Football [47]	115	613	10.66	12	0.40	0.16	0.0935
Email [48]	1,133	5,451	4.81	-	0.22	0.08	0.0085
Netscience [18]	1,589	2,742	3.45	-	0.69	0.46	0.0026
Hamsterster [49]	2,426	16,630	13.71	-	0.54	0.05	0.0057
Fb-tvshow [49]	3,892	17,262	8.87	-	0.37	0.56	0.0023
Blogs [31]	3,982	6,803	3.42	-	0.28	-0.13	0.0009
Ca-GrQc [50]	5,242	14,496	5.53	-	0.53	0.66	0.0011
Erdos992 [12]	6100	9939	3.26	-	0.12	-0.24	0.0005
Ca-HepTh1 [50]	9,877	25,998	5.26	-	0.47	0.27	0.0005
PGP [51]	10,680	24,316	4.55	-	0.27	0.24	0.0004
Ca-HepTh2 [50]	12,008	118,521	19.74	-	0.61	0.63	0.0016
Ca-AstroPh [50]	18,772	198,080	21.10	-	0.63	0.21	0.0011

randomly initialization (NCMOEA-RI), NCMOEA used the average value of centrality to classification (NCMOEA-avg), NCMOEA used the median value of centrality to classification (NCMOEA-med), NCMOEA with no labels for NC nodes (NCMOEA-NL), and NCMOEA with normal operators (NCMOEA-NG), are also compared to verify the effectiveness of the proposed components in NCMOEA.

CSE is a state-of-the-art non-EA-based algorithm, and we use the recommended parament setting in [7]. For the sake of fairness, we set the population size to 100 and the number of maximum generation to 100 for all EAs except for DCRO because DCRO adopts a special optimization framework called chemical reaction optimization and we set parameters according to the reference. The other parameters (e.g., the probabilities of crossover and mutation) are set as the values recommended in their references. The MATLAB source codes of the baseline algorithms are obtained from their authors, with the four exceptions of DCRO, SOSFCD, MODPSO, and NSGA-III-KRM, which are implemented with MATLAB.

In the proposed NCMOEA, the interval between two local searches interval is set to 10 and the crossover probability pc is set to 0.9. We use the Python source code of struc2vec provided by the author, and the main part of NCMOEA is written with MATLAB. In struc2vec, the embedding dimension is set to 16 and the rest of the parameters used the default value given in the source code. The experimental results for all comparison algorithms on all networks are obtained by averaging over 15 independent runs. The struc2vec runs on Python 2.7, and the rest parts of NCMOEA and all baseline algorithms run on MATLAB R2020a. All the experiments run on a PC with Intel Core i7-8700K 3.70-GHz CPU, 32-GB RAM, and Windows 10 operating system.

2) Experimental Networks: In this article, 15 real-world networks with various characteristics are adopted to evaluate the performance of the proposed algorithm. These networks are Zachary's kareate club network [45], the dolphin social network [46], the books network about U.S. politics [47], the American college football network [47], the email network [48], the netscience network [18], Hamsterster network [49], fb-tvshow network [49], blogs network [31], Erdos992 network [12], PGP network [51], and five collaboration networks from [50] (Ca-GrQc, Ca-HepTh1, Ca-HepTh2, and Ca-AstroPh). Table I presents the characteristics of the 15 real-world networks in detail. In this table, "AD" means the average degree of the nodes in the network, "RC" means the number of real communities in the network ("−" means that the real community structure is unknown), "CC" means the average clustering coefficient, and "AC" means the assortativity coefficient. Note that karate, dolphin, polbooks, and football are networks with ground-truth community structure, while the remaining 11 network's true community structures are unknown.

To verify the effectiveness of the proposed NCMOEA comprehensively, two groups of LFR networks are also used as experimental networks. The first group of LFR networks consists of networks whose mixing parameter μ ranges from 0.1 to 0.7 with interval 0.1, while the size of networks (i.e., the number of nodes) n is fixed as 1000. The second group of LFR networks consists of networks whose size ranges from 500 to 2500 with an interval of 500, while the mixing parameter μ is fixed as 0.6. The remaining parameters are set as follows. The exponents of the power-law distribution of node degrees τ_1 are set to 2 and the community size τ_2 is set to 1. The average degree d_{ave} is set to 20, the maximum degree $d_{\max}$ is set to 50, and the range of community size is from 20 to 100.

3) Evaluation Metrics: In this article, two popular metrics are adopted to evaluate the quality of solutions: modularity (Q) [5] and normalized mutual information (NMI) [52]. For the MOEAs, we calculate the metric value (Q and NMI) for each solution in the final obtained nondominated solution set the algorithms get and then choose one with the best value for comparison as the result of the algorithm on this metric. This way is also widely adopted in existing MOEA-based community detection algorithms [18], [53].

The first metric, modularity Q, is used for measuring the quality of communities detected without the ground-truth community labels of the network. Specifically, Q is calculated as

$$Q = \sum_{s=1}^{c} \left[\frac{l_s}{M} - \left(\frac{d_s}{2M} \right)^2 \right]$$

where c is the number of communities, M is the number of edges in the network, l_s is the total number of edges connecting nodes in community s, and d_s is the sum of degrees of nodes in community s. a larger Q value indicates a higher quality of the communities detected.

The second metric, NMI, is used for measuring the similarity between the detected communities and the ground-truth community labels. NMI is calculated as

$$\text{NMI}(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{i,j} \log \left(C_{i,j} \dot{n} / C_{i.} C_{.j} \right)}{\sum_{i=1}^{C_A} C_{i.} \log \left(C_{i.}/n \right) + \sum_{j=1}^{C_B} C_{.j} \log \left(C_{.j}/n \right)}$$

where C_A (C_B) is the number of communities in the partition A (B), C is the confusion matrix, and $C_{i,j}$ denotes the number of shared nodes in the community i of partition A and community j of partition B. $C_{i.}$ is the sum of elements of C in row i, $C_{j.}$ is the sum of elements of C in column j, and n is the number of nodes in the network. A larger NMI value indicates that the detected community partition is more

TABLE II

COMPARISON RESULTS OF Q ON THE 15 REAL-WORLD NETWORKS, WHERE SYMBOLS "+," "−," AND "≈" INDICATE THAT THE PERFORMANCE IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OF NCMOEA, RESPECTIVELY. NOTE THAT "/" MEANS THAT Q VALUES ARE NOT PROVIDED HERE SINCE THESE RESULTS CANNOT BE OBTAINED WITHIN 10 h FOR ONE RUN

Network	Measure	CSE	CoCoMi	DCRO	SOSFCD	MOGA-Net	MODPSO	MODTLBO/D	RMOEA	NSGA-III-KRM	NCMOEA
Karate	Q_{Max}	0.3826	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**	0.3875	**0.4198**	**0.4198**
	Q_{Avg}	0.3826$^-$	**0.4198**$^\approx$	**0.4198**$^\approx$	0.4165$^-$	0.4161$^-$	0.4121$^-$	**0.4198**$^\approx$	0.3830$^-$	0.4194$^\approx$	**0.4198**
	std	0.0000	0.0000	0.0000	0.0060	0.0033	0.0083	0.0000	0.0068	0.0012	0.0000
	runtime(s)	0.11	5.15	4.89	2.51	1.06	5.50	30.73	8.62	2.58	22.79 (20.62)
Dolphin	Q_{Max}	0.3476	**0.5285**	**0.5285**	0.5268	0.5215	0.5268	0.5220	0.5268	0.5277	0.5268
	Q_{Avg}	0.3476$^-$	0.5272$^+$	**0.5279**$^+$	0.5239$^+$	0.4975$^-$	0.5187$^\approx$	0.5205$^-$	0.5262$^+$	0.5246$^\approx$	0.5223
	std	0.0000	0.0006	0.0006	0.0033	0.0191	0.0064	0.0013	0.0008	0.0031	0.0012
	runtime(s)	0.18	14.11	5.34	4.50	1.68	5.83	53.61	10.63	3.55	22.59 (19.33)
Polbooks	Q_{Max}	0.5000	0.5270	**0.5272**	0.5267	0.5262	0.5260	0.5270	0.5268	0.5269	0.5269
	Q_{Avg}	0.5001$^-$	**0.5272**$^+$	**0.5272**$^+$	0.5168$^-$	0.5162$^-$	0.5252$^-$	0.5268$^-$	0.5239$^-$	0.5258$^-$	0.5269
	std	0.0000	0.0001	0.0001	0.0060	0.0085	0.0014	0.0001	0.0023	0.0016	0.0000
	runtime(s)	0.92	27.86	10.02	9.40	3.05	8.98	91.82	10.68	6.06	25.87 (20.00)
Football	Q_{Max}	0.5989	**0.6046**	**0.6046**	0.5904	0.4805	**0.6046**	**0.6046**	**0.6046**	0.5681	**0.6046**
	Q_{Avg}	0.5989$^-$	**0.6045**$^\approx$	0.5991$^-$	0.5768$^-$	0.4426$^-$	0.6011$^-$	0.6044$^\approx$	0.6044$^\approx$	0.5462$^-$	0.6044
	std	0.0000	0.0001	0.0052	0.0091	0.0258	0.0030	0.0003	0.0000	0.0131	0.0001
	runtime(s)	0.94	15.73	17.91	11.75	3.26	9.26	99.50	11.12	6.26	24.68 (20.93)
Email	Q_{Max}	0.4429	0.5625	0.5546	0.5082	0.2783	0.5299	0.3329	0.4704	0.3440	**0.5702**
	Q_{Avg}	0.4368$^-$	0.5522$^-$	0.5424$^-$	0.4892$^-$	0.2496$^-$	0.4580$^-$	0.3158$^-$	0.4562$^-$	0.3344$^-$	**0.5603**
	std	0.0058	0.0058	0.0048	0.0126	0.0216	0.0424	0.0114	0.0088	0.0053	0.0061
	runtime(s)	20.67	818.19	245.56	4545.36	109.88	281.09	3518.60	134.26	75.63	199.70 (77.05)
Net-science	Q_{Max}	0.9152	0.9087	0.9237	0.8754	0.9460	0.9436	0.9229	**0.9548**	0.9050	0.9133
	Q_{Avg}	0.9148$^+$	0.9022$^-$	0.9214$^+$	0.8713$^-$	0.9436$^+$	0.9313$^+$	0.9129$^+$	**0.9512**$^+$	0.9006$^-$	0.9048
	std	0.0002	0.0043	0.0015	0.0025	0.0017	0.0063	0.0044	0.0021	0.0024	0.0059
	runtime(s)	2.33	453.39	252.53	1990.01	56.05	201.22	5904.68	143.53	131.68	120.23 (36.83)
Hamsterster	Q_{Max}	0.4064	0.5316	0.5351	/	0.2870	0.5185	0.3384	0.4889	0.3538	**0.5369**
	Q_{Avg}	0.4007$^-$	0.5228$^-$	**0.5273**$^\approx$	/	0.2709$^-$	0.4343$^-$	0.3262$^-$	0.4747$^-$	0.3396$^-$	0.5273
	std	0.0033	0.0041	0.0051	/	0.0110	0.0440	0.0090	0.0058	0.0066	0.0044
	runtime(s)	119.38	1065.34	870.51	/	263.09	581.86	22726.54	279.88	253.71	482.11 (213.49)
Fb-tvshow	Q_{Max}	0.7823	0.8351	0.8376	/	0.7361	0.8425	/	0.8261	0.7031	**0.8505**
	Q_{Avg}	0.7786$^-$	0.8325$^-$	0.8335$^-$	/	0.7259$^-$	0.8324$^-$	/	0.8151$^-$	0.6936$^-$	**0.8456**
	std	0.0011	0.0015	0.0020	/	0.0059	0.0080	/	0.0040	0.0065	0.0032
	runtime(s)	139.56	4505.04	1047.02	/	243.56	1496.57	/	655.73	546.54	1027.16 (411.53)
Blogs	Q_{Max}	0.7730	0.7672	0.7853	/	0.7265	0.8057	/	0.7908	0.7308	**0.8349**
	Q_{Avg}	0.7692$^-$	0.7594$^-$	0.7772$^-$	/	0.7218$^-$	0.7932$^-$	/	0.7848$^-$	0.7234$^-$	**0.8272**
	std	0.0014	0.0031	0.0037	/	0.0038	0.0068	/	0.0034	0.0037	0.0040
	runtime(s)	30.39	21436.80	1201.55	/	308.90	2156.58	/	857.67	504.58	969.66 (256.06)
Ca-GrQc	Q_{Max}	0.7564	/	0.7868	/	0.7391	0.8090	/	0.7933	0.7105	**0.8214**
	Q_{Avg}	0.7547$^-$	/	0.7821$^-$	/	0.7371$^-$	0.7913$^-$	/	0.7872$^-$	0.7041$^-$	**0.8139**
	std	0.0010	/	0.0020	/	0.0011	0.0110	/	0.0032	0.0034	0.0045
	runtime(s)	186.10	/	2945.41	/	504.16	3818.99	/	1377.05	1019.55	1899.54 (330.26)
Erdos992	Q_{Max}	0.6098	0.6310	0.6490	/	0.6355	0.6746	/	0.6908	0.6411	**0.6978**
	Q_{Avg}	0.6051$^-$	0.6180$^-$	0.6432$^-$	/	0.6265$^-$	0.6490$^-$	/	0.6831$^\approx$	0.6377$^-$	**0.6861**
	std	0.0026	0.0089	0.0036	/	0.0052	0.0129	/	0.0031	0.0023	0.0091
	runtime(s)	180.81	18367.00	3734.67	/	840.18	3412.53	/	2258.62	708.95	2176.52 (499.38)
Ca-HepTh1	Q_{Max}	0.6148	/	0.6672	/	0.5664	0.7012	/	0.6500	0.5403	**0.7274**
	Q_{Avg}	0.6124$^-$	/	0.6637$^-$	/	0.5632$^-$	0.6755$^-$	/	0.6429$^-$	0.5345$^-$	**0.7131**
	std	0.0012	/	0.0016	/	0.0013	0.0123	/	0.0041	0.0029	0.0063
	runtime(s)	240.90	/	7905.99	/	1572.43	11937.90	/	4690.43	2687.23	5978.60 (741.74)
PGP	Q_{Max}	0.7261	/	0.7882	/	0.6975	0.8198	/	0.8012	0.6718	**0.8488**
	Q_{Avg}	0.7208$^-$	/	0.7827$^-$	/	0.6901$^-$	0.8047$^-$	/	0.7960$^-$	0.6595$^-$	**0.8437**
	std	0.0031	/	0.0023	/	0.0049	0.0112	/	0.0023	0.0049	0.0038
	runtime(s)	348.30	/	7336.97	/	1984.57	13741.70	/	8913.60	3005.74	4376.59 (1323.46)
Ca-HepTh2	Q_{Max}	0.4585	/	0.6166	/	0.5051	0.6064	/	0.5677	0.3957	**0.6289**
	Q_{Avg}	0.4548$^-$	/	0.6112$^-$	/	0.4960$^-$	0.5807$^-$	/	0.5650$^-$	0.3854$^-$	**0.6226**
	std	0.0030	/	0.0027	/	0.0049	0.0192	/	0.0011	0.0050	0.0051
	runtime(s)	33654.04	/	18289.50	/	3393.11	18096.00	/	6246.34	3262.02	10054.77 (2542.59)
Ca-AstroPh	Q_{Max}	0.3820	/	/	/	0.3817	/	/	0.5345	0.2552	**0.5839**
	Q_{Avg}	0.3796$^-$	/	/	/	0.3730$^-$	/	/	0.5283$^-$	0.2507$^-$	**0.5719**
	std	0.0013	/	/	/	0.0059	/	/	0.0025	0.0031	0.0053
	runtime(s)	9160.80	/	/	/	13879.13	/	/	18069.50	6126.58	31813.79 (4442.79)
+/−/≈		1/14/0	2/11/2	3/10/2	1/14/0	1/14/0	1/13/1	1/12/2	2/11/2	0/13/2	

similar to real community partition and the algorithm performs better.

B. Comparison Results Between NCMOEA and Baselines

1) Experimental Results in Terms of Modularity Q: Table II lists the modularity Q of the proposed algorithm NCMOEA and the seven baselines for community detection averaging over 15 runs on the 15 real-world networks. The Wilcoxon rank sum test is adopted to evaluate the statistical difference of the performance of comparison, with a significance level of 0.05. In Table II, the symbols "+," "−," and "=" indicate that the result of baseline algorithm is significantly better, significantly worse, and statistically similar to the result of NCMOEA. From this table, it can be found that on most of the real networks and the proposed NCMOEA achieves the best performance; on some networks (e.g., polbooks and football), NCMOEA achieves the second best, but the result is statistically similar to the best result. The symbol "/" in the table means that the algorithm takes more than 10 h for one run so that the results are not provided. Note that CoCoMi takes more than 10 h for one run on the Ca-GrQc network but uses less time on the Erdos992 network, and it is because Ca-GrQc has more edges and a higher average degree.

From Table II, the modularity of the non-EA algorithm CSE is much worse than that of most EA-based algorithms because it does not optimize the modularity directly. Compared with the three single-objective optimization algorithms CoCoMi, DCRO, and SOSFCD, it can be found that the three algorithms achieve good performance on four small networks (karate, dolphin, polbooks, and football) because these single-objective algorithms focus on the optimization of modularity and use search strategies to get solutions with better modularity. However, the search strategies in CoCoMi, DCRO, and SOSFCD take much time and they focus on the local structure adjusting too much. It can be proved by the results that on the networks with a larger size, the three single-objective algorithms are time-consuming so that they cannot get results on larger networks within 10 h (e.g., Ca-AstroPh). In addition, we can find that MOEAs get better results than the three single-objective algorithms on some large networks (e.g., email, blogs, and Erdos992).

Among the MOEA-based algorithms, NCMOEA can achieve the best results on most of networks. This is because of the proposed node classification scheme and the proposed strategies. When the scale of the network becomes greater, the structural characteristics of the nodes get more important. Classifying the nodes and using different strategies on nodes of different categories can get better performance on these networks (e.g., blogs and larger networks). The mixed representation can represent the activated community centers and the community structure. If the algorithm can find the community centers correctly, a good rough community structure can be obtained. Then, the algorithm adjusts the other nodes and finds communities with high quality. However, NCMOEA does not perform well on few networks (e.g., dolhpin and net-science). The reason is due to the fact that when the communities with high quality cannot be defined with the central nodes found by the strategy, it is difficult for NCMOEA to find good community partition because of the limit of the representation (specifically the one-to-one correspondence between the activated central nodes and the communities).

In terms of running time, it can be found that CSE is efficient on most of the networks because it is not EA-based algorithm and does not need to evaluate solutions. The three single-objective algorithms, CoCoMi, DCRO, and SOSFCD, cannot obtain the results on large networks within 10 h for one run because the local search of them takes a lot of time. The teaching and learning strategies used in MODTLBO/D also take much time. Although the node classification strategy takes time, the proposed NCMOEA still gets the comparable running time compared to MOGA-Net, MODPSO, RMOEA, and NSGA-III-KRM.

To further analyze the good performance of NCMOEA, a concrete example is given. Fig. 5 shows an example on a karate network, and the solution with the best Q value ($Q = 0.4198$) found by NCMOEA. The four communities can be seen as smaller communities divided from the two real communities. NCMOEA can find the optimal community structure of the network in all the 15 independent runs. One of the reasons is that NCMOEA finds good community centers (e.g., nodes 1 and 34), and then, the rough community structure can be found quickly. According to the analysis in [45], these central nodes usually mean key individuals in the social relationships. After finding good community centers, the proposed NCMOEA can obtain a community structure with high quality through searching the community assignment of the remaining nodes.

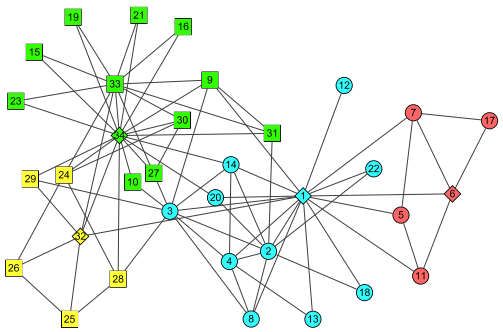

Fig. 5. Solution found by NCMOEA on Zachary's karate club social network. The different colors of the nodes represent the communities detected by NCMOEA and the shapes of nodes represent the real partition of the network. Note that the nodes with the rhombus mark are the central nodes NCMOEA found. In the real community partition of karate network, nodes 1 and 6 belong to the circle community and nodes 32 and 34 belong to the square community.

2) Experimental Results in Terms of NMI: Table III shows the NMI value of the nine comparison algorithms and the proposed NCMOEA over 15 runs on the four real-world networks with ground-truth labels. From this table, we can see that CSE and MOEAs perform better than the three single-objective algorithms, and it is because the single-objective algorithms optimize the modularity only; however, the solution with the best modularity often does not correspond to the true partition of a real-world network [24]. CSE performs better than CoCoMi and DCRO because the algorithm can enhance the ambiguous community structure, make it clearer, and then find some communities that are closer to the ground truth. The MOEAs perform better in the comparison because they can provide a set of solutions with a tradeoff on different community partitions. In addition, we can observe that NCMOEA, MODTLBO/D, RMOEA, and NSGA-III-KMR perform better than CoCoMi, DCRO, MOGA-Net, and MODPSO on most of the networks. NCMOEA can achieve the best result on the karate network and similar results on the remaining networks compared with MODTLBO/D and RMOEA. However, Q of MODTLBO/D and RMOEA is much worse than that of NCMOEA.

Furthermore, Fig. 6 presents the NMI values of all comparison algorithms on the two groups of LFR networks: 1) μ ranges from 0.1 to 0.7 at the step of 0.1 ($n = 1000$) and 2) n ranges from 500 to 2500 at the step of 500 ($\mu = 0.6$). From Fig. 6(a), it can be found that CoCoMi, MODPSO, and NCMOEA can achieve comparable good performance when μ increases from 0.1 to 0.6. When μ reaches 0.7, all comparison algorithms degrade greatly and MODPSO achieves the best performance due to the label propagation-based initialization used in the algorithm. CoCoMi performs well because of its adapted KL moving scheme, and however, it is very time-consuming. From Fig. 6(b), it can also be observed that

TABLE III

COMPARISON RESULTS OF NMI ON THE FOUR REAL-WORLD NETWORKS, WHERE SYMBOLS "+," "−," AND "≈" INDICATE THAT THE PERFORMANCE IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OF NCMOEA, RESPECTIVELY

Network	Measure	CSE	CoCoMi	DCRO	SOSFCD	MOGA-Net	MODPSO	MODTLBO/D	RMOEA	NSGA-III-KMR	NCMOEA
Karate	NMI_{Max}	0.8155	0.6873	0.6873	0.8041	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	NMI_{Avg}	0.8155^{-}	0.6873^{-}	0.6873^{-}	0.7014^{-}	0.9891$^{\approx}$	0.8925^{-}	**1.0000**$^{\approx}$	0.9500^{-}	0.9341^{-}	**1.0000**
	std	0.0000	0.0000	0.0000	0.0289	0.0420	0.1110	0.0000	0.0909	0.0836	0.0000
Dolphin	NMI_{Max}	0.7224	0.5932	0.5932	0.6169	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	NMI_{Avg}	0.7224^{-}	0.5825^{-}	0.5874^{-}	0.5421^{-}	0.9506^{-}	0.8619^{-}	**1.0000**$^{\approx}$	0.9962$^{\approx}$	0.9468^{-}	0.9926
	std	0.0000	0.0095	0.0040	0.0461	0.0653	0.1366	0.0000	0.0148	0.0719	0.0287
Polbooks	NMI_{Max}	0.4418	0.5198	0.5198	0.5359	0.5329	0.5412	0.5670	0.5512	**0.5679**	0.5412
	NMI_{Avg}	0.4418^{-}	0.5103^{-}	0.5103^{-}	0.4880^{-}	0.5054^{-}	0.5119^{-}	0.5375$^{\approx}$	0.5417^{+}	**0.5428**$^{+}$	0.5327
	std	0.0000	0.0039	0.0039	0.0270	0.0173	0.0178	0.0114	0.0057	0.0159	0.0109
Football	NMI_{Max}	0.9079	0.8903	0.9095	0.8860	0.6904	0.9269	**0.9367**	0.9361	0.8610	0.9269
	NMI_{Avg}	0.9079^{-}	0.8789^{-}	0.8430^{-}	0.8293^{-}	0.6505^{-}	0.9209^{+}	0.9272^{+}	**0.9301**$^{+}$	0.8220^{-}	0.9140
	std	0.0000	0.0167	0.0380	0.0279	0.0340	0.0058	0.0041	0.0069	0.0181	0.0090
$+/-/\approx$		0/4/0	0/4/0	0/4/0	0/4/0	0/3/1	0/3/1	1/0/3	2/1/1	1/3/0	

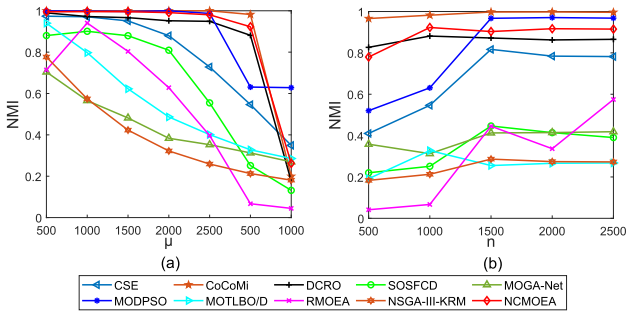

Fig. 6. NMI values of all comparison algorithms averaging over 15 runs on the two groups of LFR networks. (a) Different μ, $n = 1000$. (b) Different n, $\mu = 0.6$.

NCMOEA still performs better than most of the baseline algorithms on the networks with different sizes and get the comparably good performance as CoCoMi and MODPSO.

According to the experimental results on Q and NMI, it can be found that NCMOEA achieves the best modularity Q on most of the real networks and its performance on NMI is also comparable. The better performance of NCMOEA is attributed to the proposed node classification scheme and the proposed strategies. In Section IV-C, we demonstrate the effectiveness of these proposed strategies in NCMOEA.

C. Comparison Results Between NCMOEA and Its Variants

To verify the effectiveness of the proposed node classification strategy, the mixed representation, the crossover and mutation operators, and the initialization strategy, we design six variants of NCMOEA: the first variant NCMOEA-NC does not classify the nodes and all the nodes are regarded as the CC nodes. We do not set all the nodes as NC nodes because the mixed representation of NCMOEA cannot find any communities without central nodes. The second variant NCMOEA-RI changes the initialization strategy of NCMOEA with a totally random initialization. Specifically, we activate CC nodes randomly and assign the other nodes into the communities of the activated central nodes according to the similarity. The third variant NCMOEA-avg classifies the nodes whose centrality value is higher than the average value into CC nodes. The fourth variant NCMOEA-med classifies the nodes whose centrality value is higher than the median value into CC nodes. The fifth variant NCMOEA-NL is a variant whose representation is simplified. Specifically, the community labels of NC nodes are not in the representation and they are determined by these of their neighboring central nodes. The sixth variant NCMOEA-NG is a variant in which the genetic operators are replaced with one-way crossover and a neighbor-based mutation (i.e., change the label of the mutated node to the label of its neighbor). Note that, due to the particularity of the mixed representation, most of the existing tailored genetic operators for community detection could generate illegal individuals. In NCMOEA-NG, the one-way crossover and a neighbor-based mutation are adopted, and the illegal solutions they generated are repaired.

The results of the six variants of the proposed NCMOEA are listed in Table IV. From the table, it can be found that the performance of NCMOEA is better than the variants on most of the networks. On the four small-scale networks, there is no significant difference between NCMOEA, NCMOEA-NC, and NCMOEA-RI. When the scale of the network grows, NCMOEA-NC is not good as NCMOEA on most of the larger networks. It is because the node classification strategy can select CC nodes and reduce the search space of central nodes, which helps the algorithm find good central nodes and determine the rough structure of communities quickly. The performance of NCMOEA-RI is worse significantly than NCMOEA because the random initialization cannot find good initial active central nodes. Then, the NCMOEA-RI cannot find good solutions based on the initial structure with low quality.

To further verify the effectiveness of the classification strategy proposed in NCMOEA, two variants NCMOEA-avg and NCMOEA-med are also compared. NCMOEA-avg uses the average value of the centralities of all nodes to divide the nodes into CC nodes and NC nodes. Specifically, the nodes whose centralities are greater than the average value are considered as CC nodes. Otherwise, the remaining nodes are considered as NC nodes. Similarly, NCMOEA-med uses the median value of the centralities of all nodes to divide the nodes into the two kinds. It can be found from Table IV that the proposed NCMOEA is better than the two variants in general. The better performance of NCMOEA is due to the fact that the adopted network embedding used in the proposed classification strategy can better capture the structure characteristics of nodes on different networks. For example, the performance of NCMOEA-avg and NCMOEA-med on the networks Karate and Dolphin is worse than that of NCMOEA since both of them classify some real central nodes incorrectly

TABLE IV

Q VALUES OF NCMOEA AND ITS VARIANTS ON THE 15 REAL-WORLD NETWORKS, WHERE SYMBOLS "+," "−," AND "≈" INDICATE THAT THE PERFORMANCE IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OF NCMOEA, RESPECTIVELY

Network	Measure	NCMOEA-NC	NCMOEA-RI	NCMOEA-avg	NCMOEA-med	NCMOEA-NL	NCMOEA-NG	NCMOEA
Karate	Q_{Max}	**0.4198**	**0.4198**	0.3974	0.4132	**0.4198**	**0.4198**	**0.4198**
	Q_{Avg}	**0.4198**$^{\approx}$	**0.4198**$^{\approx}$	0.3974^{-}	0.4004^{-}	0.4185^{-}	0.4190$^{\approx}$	**0.4198**
	std	0.0000	0.0000	0.0000	0.0057	0.0020	0.0017	0.0000
Dolphin	Q_{Max}	0.5265	0.5265	0.5220	0.5220	0.5210	0.5220	**0.5268**
	Q_{Avg}	**0.5224**$^{\approx}$	0.5221$^{\approx}$	0.5212^{-}	0.5209^{-}	0.5197^{-}	0.5212^{-}	0.5223
	std	0.0017	0.0014	0.0012	0.0014	0.0008	0.0012	0.0012
Polbooks	Q_{Max}	**0.5270**	**0.5270**	0.5222	0.5269	**0.5270**	0.5269	**0.5270**
	Q_{Avg}	**0.5269**$^{\approx}$	**0.5269**$^{\approx}$	0.5200^{-}	0.5268^{-}	0.5265^{-}	0.5268^{-}	**0.5269**
	std	0.0000	0.0000	0.0020	0.0001	0.0001	0.0002	0.0000
Football	Q_{Max}	**0.6046**	0.6044	**0.6046**	0.5993	0.6043	0.6044	**0.6046**
	Q_{Avg}	0.6040$^{\approx}$	0.6044$^{\approx}$	0.6016^{-}	0.5982^{-}	0.6013^{-}	0.6036^{-}	**0.6044**
	std	0.0010	0.0001	0.0036	0.0014	0.0032	0.0013	0.0001
Email	Q_{Max}	0.5583	0.3672	**0.5704**	0.5674	0.4913	0.4807	0.5702
	Q_{Avg}	0.5559^{-}	0.3379^{-}	**0.5648**$^{\approx}$	0.5618$^{\approx}$	0.4805^{-}	0.4696^{-}	0.5603
	std	0.0025	0.0171	0.0033	0.0033	0.0075	0.0107	0.0061
Net-science	Q_{Max}	0.9300	0.8210	0.9137	**0.9302**	0.7402	0.7554	0.9133
	Q_{Avg}	0.9171^{+}	0.7957^{-}	0.9074$^{\approx}$	**0.9232**$^{+}$	0.7166^{-}	0.7431^{-}	0.9048
	std	0.0084	0.0113	0.0053	0.0050	0.0246	0.0088	0.0059
Hamsterster	Q_{Max}	0.5259	0.3648	0.5322	0.5322	0.4306	0.4128	**0.5369**
	Q_{Avg}	0.5180^{-}	0.3484^{-}	0.5215^{-}	0.5213^{-}	0.4149^{-}	0.3998^{-}	**0.5273**
	std	0.0058	0.0124	0.0072	0.0063	0.0108	0.0117	0.0044
Fb-tvshow	Q_{Max}	0.8454	0.6469	0.8451	0.8484	0.7857	0.7875	**0.8505**
	Q_{Avg}	0.8426^{-}	0.6186^{-}	0.8399^{-}	0.8424$^{\approx}$	0.7793^{-}	0.7788^{-}	**0.8456**
	std	0.0019	0.0180	0.0032	0.0049	0.0066	0.0074	0.0032
Blogs	Q_{Max}	0.8326	0.7326	0.8284	0.8296	0.7868	0.7645	**0.8349**
	Q_{Avg}	0.8217^{-}	0.7152^{-}	0.8227^{-}	0.8236^{-}	0.7813^{-}	0.7545^{-}	**0.8272**
	std	0.0045	0.0113	0.0044	0.0041	0.0051	0.0067	0.0040
Ca-GrQc	Q_{Max}	0.8136	0.5986	**0.8256**	0.8202	0.7253	0.7289	0.8214
	Q_{Avg}	0.8109$^{\approx}$	0.5795^{-}	**0.8183**$^{+}$	0.8163$^{\approx}$	0.7196^{-}	0.7193^{-}	0.8139
	std	0.0021	0.0130	0.0022	0.0032	0.0036	0.0071	0.0045
Erdos992	Q_{Max}	0.6912	0.5924	0.6961	0.6941	0.6346	0.6445	**0.6978**
	Q_{Avg}	0.6789$^{\approx}$	0.5816^{-}	**0.6862**$^{\approx}$	0.6833$^{\approx}$	0.6273^{-}	0.6335^{-}	0.6861
	std	0.0103	0.0046	0.0061	0.0077	0.0064	0.0073	0.0091
Ca-HepTh1	Q_{Max}	0.7201	0.4944	0.7206	0.7182	0.5967	0.6140	**0.7274**
	Q_{Avg}	0.7114$^{\approx}$	0.4837^{-}	**0.7157**$^{\approx}$	0.7137$^{\approx}$	0.5868^{-}	0.6066^{-}	0.7131
	std	0.0050	0.0046	0.0039	0.0021	0.0074	0.0041	0.0063
PGP	Q_{Max}	0.8440	0.6959	0.8450	0.8455	0.8070	0.7983	**0.8488**
	Q_{Avg}	0.8330^{-}	0.6698^{-}	0.8409^{-}	0.8369^{-}	0.8017^{-}	0.7900^{-}	**0.8437**
	std	0.0065	0.0128	0.0030	0.0053	0.0047	0.0058	0.0038
Ca-HepTh2	Q_{Max}	0.6170	0.3859	0.6261	0.6271	0.5612	0.5526	**0.6289**
	Q_{Avg}	0.6137^{-}	0.3670^{-}	0.6222$^{\approx}$	**0.6228**$^{\approx}$	0.5537^{-}	0.5421^{-}	0.6226
	std	0.0029	0.0150	0.0028	0.0040	0.0056	0.0074	0.0051
Ca-AstroPh	Q_{Max}	**0.5936**	0.3882	0.5910	0.5927	0.4376	0.4314	0.5839
	Q_{Avg}	0.5839^{+}	0.3739^{-}	0.5853^{+}	**0.5874**$^{+}$	0.4194^{-}	0.3903^{-}	0.5719
	std	0.0044	0.0084	0.0048	0.0030	0.0193	0.0211	0.0053
$+/-/\approx$		1/7/7	0/11/4	2/8/5	2/7/6	0/15/0	0/14/1	

and these centers cannot be found in the following evolutionary optimization process.

NCMOEA-NL and NCMOEA-NG are designed for verifying the effectiveness of the proposed representation and genetic operators. From Table IV, it can be found that NCMOEA performs better than NCMOEA-NL on all networks. The better performance is due to the fact that the labels of the NC nodes in the mixed representation used in NCMOEA make it possible for the algorithm to finely adjust the community structure. However, in NCMOEA-NL, the simplification of the representation ignores the influence of the NC nodes, so it cannot perform well. NCMOEA-NG's performance is also worse than NCMOEA, and it is because the adopted existing operators just consider the community assignment of nodes and the neighborhood information but ignore the structural difference between the two kinds of nodes (i.e., central nodes and NC nodes).

V. CONCLUSION AND FUTURE WORK

In this article, we have proposed a node classification-based MOEA named NCMOEA for community detection, where different kinds of nodes with different structural characteristics are searched in different ways. In order to consider the structural differences between the nodes, a node classification strategy was proposed to classify the nodes into two categories (i.e., CC nodes and NC nodes). For the CC nodes, the algorithm focus on the active states of them and the rough community structure can be determined by the active nodes. For the NC nodes, they cannot be selected as centers of communities and the algorithm just considers which community they should belong to. Then, in the evolutionary process, the algorithm chose some of the CC nodes to activate and then assigned the NC nodes into the communities. In addition, a mixed representation, genetic operators, and initialization strategy were proposed in the algorithm to better search the two kinds of nodes in different ways. The experimental results on 15 real-world networks and several synthetic networks have demonstrated that the proposed NCMOEA exhibits better performance than nine baseline algorithms on most situations. Six variants of the algorithm have also been implemented for comparison and the results have verified that the effectiveness of the proposed node classification strategy, the mixed representation, the crossover and mutation operators, and the initialization strategy.

There are still some works related to the node classification search scheme, which deserves to be further explored. In this article, we have adopted an embedding method called struc2vec and proposed the classification strategy to classify the nodes according to the structural characteristics of the nodes. Other node classification strategy, which can fit more complex networks and find good CC node sets, effectively needs to be developed in the future. In addition, designing a much more efficient algorithm by using some index and similarity-based techniques for finding CC nodes and NC nodes will be one of the future works. Another direction worth considering is to use the proposed search scheme for overlapping community detection problems or on other kinds of networks (e.g., signed networks and dynamic networks). Furthermore, the search scheme considering the structural characteristics of the nodes can also be extended for some other tasks, such as influence maximization [54] and control of networks [55].

REFERENCES

[1] B. A. Attea et al., "A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions," *Swarm Evol. Comput.*, vol. 63, Jun. 2021, Art. no. 100885.

[2] K. Kandhway and J. Kuri, "Using node centrality and optimal control to maximize information diffusion in social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1099–1110, Jul. 2017.

[3] X. Zeng, X. Zhang, and Q. Zou, "Integrative approaches for predicting microRNA function and prioritizing disease-related microRNA using biological interaction networks," *Briefings Bioinformat.*, vol. 17, no. 2, pp. 193–203, 2016.

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

[5] 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.

[6] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, P. 10008, Oct. 2008.

[7] Y. Su, C. Liu, Y. Niu, F. Cheng, and X. Zhang, "A community structure enhancement-based community detection algorithm for complex networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 5, pp. 2833–2846, May 2021.

[8] A. Mahmood and M. Small, "Subspace based network community detection using sparse linear coding," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 801–812, Mar. 2016.

[9] Z.-H. Deng, H.-H. Qiao, Q. Song, and L. Gao, "A complex network community detection algorithm based on label propagation and fuzzy C-means," *Phys. A, Stat. Mech. Appl.*, vol. 519, pp. 217–226, Apr. 2019.

[10] R. Lambiotte, J. C. Delvenne, and M. Barahona, "Random walks, Markov processes and the multiscale modular organization of complex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 2, pp. 76–90, Jul. 2014.

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

[12] S. He et al., "Cooperative co-evolutionary module identification with application to cancer disease module discovery," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 874–891, Dec. 2016.

[13] H. Chang, Z. Feng, and Z. Ren, "Community detection using dual-representation chemical reaction optimization," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4328–4341, Dec. 2017.

[14] J. Xiao, C. Wang, and X. K. Xu, "Community detection based on symbiotic organisms search and neighborhood information," *IEEE Trans. Comput. Soc. Syst.*, vol. 6, no. 6, pp. 1257–1272, Oct. 2019.

[15] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, 2007.

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

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

[18] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, Feb. 2014.

[19] D. Chen, F. Zou, R. Lu, L. Yu, Z. Li, and J. Wang, "Multi-objective optimization of community detection using discrete teaching–learning-based optimization with decomposition," *Inf. Sci.*, vol. 369, pp. 402–418, Nov. 2016.

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

[21] X. Liu, Y. Du, M. Jiang, and X. Zeng, "Multiobjective particle swarm optimization based on network embedding for complex network community detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 437–449, Apr. 2020.

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

[23] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, 2008, Art. no. 046110.

[24] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 418–430, Jun. 2012.

[25] C. Shi, Z. Yan, Y. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Appl. Soft Comput.*, vol. 12, pp. 850–859, Feb. 2012.

[26] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. 3rd Annu. Conf. Genetic Evol. Comput.*, 2001, pp. 283–290.

[27] B. A. Attea, W. A. Hariz, and M. F. Abdulhalim, "Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks," *Swarm Evol. Comput.*, vol. 26, pp. 137–156, Feb. 2016.

[28] Z. Zhang, P. Cui, X. Wang, J. Pei, X. Yao, and W. Zhu, "Arbitrary-order proximity preserved network embedding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Y. Guo and F. Farooq, Eds., Jul. 2018, pp. 2778–2786.

[29] H. Ma, H. Yang, K. Zhou, L. Zhang, and X. Zhang, "A local-to-global scheme-based multi-objective evolutionary algorithm for overlapping community detection on large-scale complex networks," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5135–5149, May 2021.

[30] Y. Luo et al., "A reduced mixed representation based multi-objective evolutionary algorithm for large-scale overlapping community detection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2021, pp. 2435–2442.

[31] 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.

[32] J. Sun, W. Zheng, Q. Zhang, and Z. Xu, "Graph neural network encoding for community detection in attribute networks," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7791–7804, Aug. 2021.

[33] Y. Yin, Y. Zhao, H. Li, and X. Dong, "Multi-objective evolutionary clustering for large-scale dynamic community detection," *Inf. Sci.*, vol. 549, pp. 269–287, Mar. 2021.

[34] T. Zhang and B. Wu, "A method for local community detection by finding core nodes," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 1171–1176.

[35] S. Y. Bhat and M. Abulaish, "OCTracker: A density-based framework for tracking the evolution of overlapping communities in OSNs," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 501–505.

[36] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 4, pp. 1021–1033, Dec. 2018.

[37] X. You, Y. Ma, and Z. Liu, "A three-stage algorithm on community detection in social networks," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104822.

[38] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "*struc2vec*: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 385–394.

[39] W. Luo, Z. Yan, C. Bu, and D. Zhang, "Community detection by fuzzy relations," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 478–492, Apr. 2020.
[40] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 2010.
[41] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. 19th Int. Conf. Mach. Learn.*, 2002, pp. 315–322.
[42] 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.
[43] J. Xiao, Y.-J. Wang, and X.-K. Xu, "Fuzzy community detection based on elite symbiotic organisms search and node neighborhood information," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2500–2514, Jul. 2022.
[44] T. Shaik, V. Ravi, and K. Deb, "Evolutionary multi-objective optimization algorithm for community detection in complex social networks," *Social Netw. Comput. Sci.*, vol. 2, no. 1, pp. 1–25, Feb. 2021.
[45] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Apr. 1977.
[46] D. Lusseau, "The emergent properties of a dolphin social network," *Proc. Roy. Soc. London B, Biol. Sci.*, vol. 270, no. 2, Nov. 2003.
[47] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.
[48] Z. Ding, X. Zhang, D. Sun, and B. Luo, "Overlapping community detection based on network decomposition," *Sci. Rep.*, vol. 6, no. 1, p. 24115, Jul. 2016.
[49] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. AAAI*, 2015, pp. 1–15. [Online]. Available: https://networkrepository.com
[50] J. Leskovec and A. Krevl, "SNAP datasets: Stanford large network dataset collection," Jun. 2014. [Online]. Available: http://snap.stanford.edu/data
[51] X. Guardiola, R. Guimera, A. Arenas, A. Diaz-Guilera, D. Streib, and L. Amaral, "Macro-and micro-structure of trust networks," 2002, *arXiv:cond-mat/0206240*.
[52] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 9, Sep. 2005, Art. no. P09008.
[53] 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.
[54] L. Ma et al., "Influence maximization in complex networks by using evolutionary deep reinforcement learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, Jan. 13, 2022, doi: 10.1109/TETCI.2021.3136643.
[55] L. Ma, J. Li, Q. Lin, M. Gong, C. A. Coello Coello, and Z. Ming, "Cost-aware robust control of signed networks by using a memetic algorithm," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4430–4443, Oct. 2020.

Haipeng Yang received the B.Sc. degree from the School of Computer Science and Technology, Anhui University, Hefei, China, in 2019, where he is currently pursuing the Ph.D. degree.

His research interests include multiobjective optimization and social network analysis.

Bin Li received the B.Sc. degree from the School of Computer Science and Technology, Henan University of Technology, Zhengzhou, China, in 2020. He is currently pursuing the master's degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His current research interests include multiobjective optimization methods and their application in complex network analysis.

Fan Cheng received the B.Sc. and M.Sc. degrees from the Hefei University of Technology, Hefei, China, in 2000 and 2003, respectively, and the Ph.D. degree from the University of Science and Technology of China, Hefei, in 2012.

He is currently a Full Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has published over 40 papers in refereed conferences and journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON BIG DATA, IEEE Computational Intelligence Magazine (CIM), *Applied Soft Computing*, and *Information Sciences*. His main research interests include machine learning, imbalanced classification, multiobjective optimization, and complex networks.

Peng Zhou (Member, IEEE) received the B.Sc. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has published several papers in highly regarded conferences and journals, including IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, International Joint Conferences on Artificial Intelligence (IJCAI), the Association for the Advancement of Artificial Intelligence (AAAI), SIAM International Conference on Data Mining (SDM), and IEEE International Conference on Data Mining (ICDM). His research interests include machine learning, data mining, and artificial intelligence.

Renzhi Cao received the B.Sc. degree in computer science from Anhui Normal University (AHNU), Wuhu, China, in 2008, the M.S. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2011, and the Ph.D. degree from the University of Missouri (MU), Columbia, MO, USA, in 2016.

He is currently an Associate Professor with the Department of Computer Science, Pacific Lutheran University, Tacoma, WA, USA. His research interest is mainly focused on developing and applying machine learning and data mining techniques to solve complex problems in different fields, such as protein structure/function predictions and complex network analysis.

Lei Zhang (Member, IEEE) received the B.Sc. degree from Anhui Agriculture University, Hefei, China, in 2007, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2014.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has published more than 70 papers in refereed journals and conferences, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, ACM Transactions on Knowledge Discovery Data (ACM TKDD), IEEE Computational Intelligence Magazine (CIM), *Information Sciences*, the Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conferences on Artificial Intelligence (IJCAI), and ACM SIGKDD Conference on Knowledge Discovery and Data Mining (ACM SIGKDD). His main research interests include multiobjective optimization and its applications, data mining, machine learning, social network analysis, and recommendation.

Dr. Zhang is a member of Association for Computing Machinery (ACM). He was a recipient of the ACM CIKM'12 Best Student Paper Award.