A Maximal Clique Based Multiobjective Evolutionary Algorithm for Overlapping Community Detection

Xuyun Wen, *Student Member, IEEE*, Wei-Neng Chen, *Member, IEEE*, Ying Lin, *Member, IEEE*, Tianlong Gu, Huaxiang Zhang, Yun Li, *Member, IEEE*, Yilong Yin, and Jun Zhang, *Senior Member, IEEE*

***Abstract*—Detecting community structure has become one important technique for studying complex networks. Although many community detection algorithms have been proposed, most of them focus on separated communities, where each node can belong to only one community. However, in many real-world networks, communities are often overlapped with each other. Developing overlapping community detection algorithms thus becomes necessary. Along this avenue, this paper proposes a maximal clique based multiobjective evolutionary algorithm (MOEA) for overlapping community detection. In this algorithm, a new representation scheme based on the introduced maximal-clique graph is presented. Since the maximal-clique graph is defined by using a set of maximal cliques of original graph as nodes and two maximal cliques are allowed to share the same nodes of the original graph, overlap is an intrinsic property of the maximal-clique graph. Attributing to this property, the new representation scheme allows MOEAs to handle the overlapping community detection problem in a way similar to that of the separated community detection, such that the optimization problems are simplified. As a result, the proposed algorithm could detect overlapping community structure with higher partition accuracy and lower computational cost when compared with the existing ones. The experiments on both synthetic and real-world networks validate the effectiveness and efficiency of the proposed algorithm.**

***Index Terms*—Clique-based representation, maximal-clique graph, multiobjective evolutionary algorithm (MOEA), overlapping community detection.**

I. Introduction

RECENT years have witnessed many researches that modeled real-world systems in nature and society as networks to capture the intricate properties of these complex systems, where objects are represented as nodes and the interactions among the objects are represented as edges [1]–[5]. Uncovering the community structure of complex networks is helpful for understanding complex systems. Researches on analyzing community structure thus gained growing attention during the past decades [6]–[9]. Traditionally, much of the focus within community detection is on the separated communities, where each node can belong to only one community [6], [7], [10], [11]. However, in many real-world networks, communities are often overlapped with each other [12]–[15]. For example, people in social networks always belong to several groups, simultaneously, such as family, friends and colleagues. For this reason, recent studies have paid much attention to overlapping community detection and developed various algorithms from different perspectives, including clique percolation [12], link partitioning [13], local expansion and optimization [16], [17], and label propagation [18].

The community detection problem can be formulated as an optimization problem [7], and such a problem is always NP-hard [19]. Therefore, some researchers introduced evolutionary algorithms (EAs) into this field and developed several promising methods [9], [20], [21]. In addition, it is widely accepted that a community should have dense intraconnections and sparse interconnections, implying that two conflicting objectives should be optimized simultaneously in community detection, i.e., maximizing internal links and minimizing external links [6], [22], [23]. Therefore, the community detection problem can also be modeled as a multiobjective optimization problem (MOP). Along this line, several multiobjective EAs (MOEAs) [19], [24]–[28] have been proposed. However, most of them focused on separated community detection and failed to detect overlapping community structures.

Manuscript received February 24, 2016; revised June 30, 2016; accepted August 19, 2016. Date of publication September 1, 2016; date of current version May 25, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61622206, Grant 61332002, Grant U1201258, and Grant 61309003. *(Corresponding authors: Wei-Neng Chen; Jun Zhang.)*

X. Wen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Sun Yat-sen University, Guangzhou 510006, China.

W.-N. Chen and J. Zhang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: cwnraul634@aliyun.com; junzhang@ieee.org).

Y. Lin is with the Department of Psychology, Sun Yat-sen University, Guangzhou 510006, China.

T. Gu is with the School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China.

H. Zhang is with the School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China.

Y. Li is with the School of Computer Science and Network Security, Dongguan University of Technology, Dongguan 523808, China.

Y. Yin is with Shandong University, Jinan 250100, China.

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2016.2605501

In fact, one obstacle for applying MOEAs to overlapping community detection is the representation scheme of the individual. The existing representation approaches can be broadly divided into two major classes, i.e., prototype-based approaches [28] and node-based approaches [24]–[27]. In the prototype-based approaches [28], each gene of an individual represents the information of one community, e.g., the coordinates of the community center. Though suitable for overlapping community detection, this representation scheme has some shortcomings and limitations, such as tending to capture round-shaped community, requiring to set the number of communities in advance, and increasing the difficulty in designing evolutionary operators. Furthermore, when adopting prototype-based approaches as the representation scheme, the community detection is generally converted to a data clustering problem, which is based on the network information such as spectrum [28]. During this transformation, some valid network information may be lost [28].

Unlike the prototype-based representation, genes of an individual in the node-based approaches correspond to the community information of the nodes in the network [24]–[27]. Under this scheme, there are two types of approaches: 1) direct and 2) indirect. For the direct node-based approach [24]–[27], each gene is an integer representing the community information of the corresponding node, such as the label of the community this node belongs to [27] or the label of a node that belongs to the same community with this node [24]–[26]. However, since this representation scheme can only ensure every node to be assigned to one community, it is not suitable for overlapping community detection. For the indirect node-based approach [27], each gene of an individual is a random integer within the number of nodes of the network and thus a decoder is needed to transform them to the corresponding community information. In the decoding process, each node is allowed to belong to multiple communities, such that this representation approach can be used for overlapping community detection. However, the introduction of the decoder in the evolution process brings in two main drawbacks. First, since the fitness computation is directly related to the decoder, the decoding method has a significant influence on partition accuracy. Second, the decoding process is executed for each individual in each generation, leading to a high computational complexity of the algorithm.

To address the aforementioned issues of representation schemes, this paper introduces the maximal-clique graph, which uses a set of maximal cliques as nodes and links among maximal cliques as edges. Then based on the maximal-clique graph, a clique-based representation scheme is proposed, where each gene of the individual represents the community label of the corresponding maximal clique. Since two maximal cliques are allowed to share the same nodes of the original graph, overlap is an intrinsic property of the nodes of the maximal-clique graph, which exactly characterizes the overlapping communities. Attributing to this property, the new representation scheme allows MOEAs to handle the overlapping community detection problem in a way similar to that of the separated community detection, which not only simplifies the optimization problems, but also overcomes some limitations of the existing representation schemes. Compared with the prototype-based representation, the clique-based approach is not restricted by community shapes and requires no prior knowledge on the community structure. Compared with the indirect node-based representation, the clique-based approach does not need to decode individuals in the evolution process, which largely lowers the computational cost of the algorithm.

Afterwards, the clique-based representation scheme and the corresponding evolutionary operators are coupled with the framework of MOEA, constituting a maximal clique based MOEA (MCMOEA), for overlapping community detection. The experiments on synthetic networks and real-world networks validate that MCMOEA is effective and efficient. Comparisons with other five representative algorithms show that MCMOEA is competitive and promising.

The rest of this paper is organized as follows. Section II briefly describes the community detection problem, and introduces the objective functions and the framework of MOEA used in this paper. Section III gives a detailed description of the proposed maximal-clique graph. In Section IV, the details of MCMOEA are presented. In Section V, the performance of MCMOEA is evaluated on both synthetic and real-world networks and the comparisons are made between MCMOEA and other five representative methods. Finally, the conclusion is given in Section VI.

II. Background

This section introduces the necessary background knowledge for understanding the proposed MCMOEA. First, the definition of network community used in this paper is clarified. Then the multiobjective model of the community detection problem is given. In the end, the framework of MOEA used in this paper is briefly reviewed.

A. Definition of Network Community

A network can be modeled as a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes, $E = \{(v_i, v_j)|v_i, v_j \in V \text{ and } i \neq j\}$ is the set of links, called edges, and N is the number of nodes. Generally, a community in a network is regarded as a group that has dense intra-links and sparse inter-links. To make the definition more clear, Radicchi *et al.* [10] gave a quantitative description of the network community based on the node degree. Let $A = [A_{ij}]_{N\times N}$ be the adjacent matrix of G. For an unweighted graph, $A_{ij} = 1$, if $(v_i, v_j) \in E$; otherwise, $A_{ij} = 0$. Suppose S is a subgraph of G, and then for any node $v_i \in S$, the internal and external degrees of node v_i can be denoted as $k_S^{\text{in}}(v_i) = \sum_{v_i,v_j\in S} A_{ij}$ and $k_S^{\text{out}}(v_i) = \sum_{v_i\in S, v_j\notin S} A_{ij}$, respectively. Then S is a community in a strong sense if

$$\forall v_i \in S, k_S^{\text{in}}(v_i) > k_S^{\text{out}}(v_i). \tag{1}$$

In other words, every node in a strong community has more intraconnections than interconnections. In contrast, S is a community in a weak sense if

$$\sum_{v_i\in S} k_S^{\text{in}}(v_i) > \sum_{v_i\in S} k_S^{\text{out}}(v_i). \tag{2}$$

That is, the sum of internal degrees of nodes in a weak community is larger than that of external degrees. Considering the former community definition is too strict, the latter one is adopted in this paper [27].

B. Community Detection Problem

The community detection problem can be modeled as an MOP with two objectives [19], [24]–[28]. One objective is to maximize the link density among nodes in the same community (intra-link density), while the other is to minimize the link density among nodes in different communities (inter-link density). A number of different criteria have been proposed for measuring intra-link and inter-link densities [19], [25], [26]. In this paper, the kernel k-means (*KKM*) [29] and ratio cut (*RC*) [26] are, respectively, adopted for measuring these two densities. Given a graph G and a partition C with t communities, let $V_p \subseteq V$ be the set of nodes in a community $p(p = 1, 2, \ldots, t)$ and $\overline{V_p} = V - V_p$ be the set of nodes that are not in p. The *KKM* and the *RC* values of C can be calculated as

$$KKM = 2(N - t) - \sum_{p=1}^{t} \frac{L(V_p, V_p)}{|V_p|} \tag{3}$$

$$RC = \sum_{p=1}^{t} \frac{L(V_p, \overline{V_p})}{|V_p|} \tag{4}$$

where $L(V_p, V_p) = \sum_{v_i, v_j \in V_p} A_{ij}$ and $L(V_p, \overline{V_p}) = \sum_{v_i \in V_p, v_j \in \overline{V_p}} A_{ij}$ are the sum of internal and external link strengths of nodes in V_p, respectively, and "$| \cdot |$" denotes the size of a set. Through (3) and (4), we can see that a small *KKM* value indicates that the communities in C have high intra-link densities, while a small *RC* value indicates that the communities in C have low inter-link densities. Therefore, through using (3) and (4) as the objectives, the community detection problem can be formulized as an MOP that seeks minimization on both objectives.

C. MOEA/D

The main focus of this paper is to propose a new representation scheme for MOEAs to solve the overlapping community problems and address the limitations of the existing approaches. Therefore, based on the proposed representation scheme, different MOEAs could be adopted to implement MCMOEA, such as MOEA based on decomposition (MOEA/D) [30], non-dominated sorting genetic algorithm II (NSGA-II) [31], strength Pareto EA II [32], and MOEA with double-level archives (MOEA_DLA) [33]. However, to facilitate illustration, one of the most widely used MOEAs with relatively low computational complexity, i.e., MOEA/D [19], [26], [27], is chosen as the representative in the following section for illustrating the structure of MCMOEA. The implementation details of MCMOEA with the frameworks of other MOEAs are provided in Section V-D. Therefore, in this section, only the procedure of MOEA/D is reviewed.

As a decomposition-based method, MOEA/D [30] decomposes an MOP into several scalar optimization subproblems and optimizes them simultaneously. Each individual in the population of MOEA/D is associated with a subproblem and assigned an n-dimensional weight vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, where n is the number of objective functions. Based on the distance between the weight vectors, the neighborhood relationships among subproblems are determined. Considering the fact that neighboring subproblems should have similar optimal solutions, each individual is optimized using only the information of its neighbors.

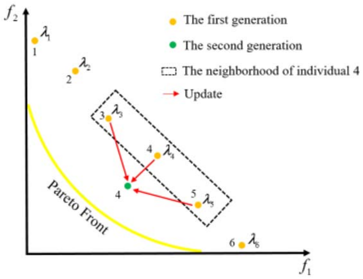


Fig. 1. Simple illustration of MOEA/D.

Fig. 1 exemplifies the idea of MOEA/D using an MOP with two objectives. In Fig. 1, the yellow curve represents the Pareto front (PF). Dots in different colors represent individuals in different generations and red arrows indicate the evolutionary direction. The population evolution in MOEA/D consists of three steps. First, six individuals, each of which is associated with a subproblem, are initialized and assigned different weight vectors $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_6$. Second, the neighborhood of each subproblem is determined according to the distance between the weight vectors. For example, supposing that the neighborhood size is 3, individuals 3 and 5 are selected as the neighbors of individual 4. Third, each individual is optimized based on the information of its neighbors, e.g., individual 4 is updated based on individuals 3–5. For more detailed descriptions of MOEA/D, please refer to [30].

III. Maximal-Clique Graph

In this section, the definition and the construction method of the maximal-clique graph are introduced. As the basic unit of the maximal-clique graph, the concept of the maximal clique is given at first. Given a graph, a clique is a complete subgraph in which every two nodes are adjacent [34]. k-clique means the size of the clique is k. A maximal clique is a special clique which cannot be extended by adding any other nodes [34]. Take Fig. 2 as an example. Nodes v_5, v_6, and v_7 compose a 3-clique, but it is not a maximal clique as it can be extended to a 4-clique by adding node v_8.

A recent study revealed that a community is typically composed of several cliques which share many common nodes [12]. Therefore, as one type of cliques, maximal cliques can be viewed as the essential unit of a community. Additionally, a node or an edge in a graph may belong to multiple maximal cliques. For example in Fig. 2, node v_3 is shared by maximal cliques ①, ②, and ③, and the edge between v_5 and v_6 is shared by maximal cliques ③ and ④. Using maximal cliques to represent communities thus facilitates

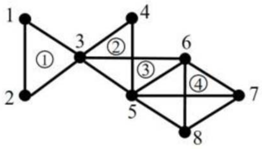


Fig. 2. Examples of maximal cliques. Nodes v_1, v_2, and v_3 constitute maximal clique ①; nodes v_3, v_4, and v_5 constitute maximal clique ②; nodes v_3, v_5, and v_6 constitute maximal clique ③; nodes v_5, v_6, v_7, and v_8 constitute maximal clique ④.

revelation of the overlapping structure. In order to detect communities from the perspective of maximal cliques, this paper proposes to convert the original graph $G = (V, E)$ into a maximal-clique graph $G^c = (V^c, E^c)$, where $V^c = \{v_1^c, v_2^c, \ldots, v_M^c\}$ is named as *clique nodes* and $E^c = \{(v_m^c, v_n^c) | v_m^c, v_n^c \in V^c \text{ and } m \neq n\}$ is named as *clique edges* or *clique links*, respectively. The conversion method contains three steps: 1) determining the clique nodes; 2) measuring the link strength between clique nodes; and 3) determining the clique edges based on the distribution of link strength. Detailed implementation of these three steps is described in the following parts and the corresponding flowchart is presented in Fig. S1 in the supplementary material.

A. Determining Clique Nodes of Maximal-Clique Graph

Determining clique nodes is the first step to build a maximal-clique graph. In this paper, a set of maximal cliques in the original graph G are adopted as clique nodes of the corresponding maximal-clique graph G^c. Since each maximal clique is one of the largest cliques the nodes in G belong to, this paper transforms the determination process of clique nodes into finding all the largest clique(s) that each node in G belongs to, and designs the corresponding method as shown in Algorithm 1.

In Algorithm 1, the clique nodes are determined in descending order according to their clique sizes. Since the nodes with higher degrees are more likely to constitute larger maximal cliques, the determination process first calculates the degree of each node in G and sorts the nodes in descending order of their degrees (lines 2–4). Suppose that the number of nodes of G is N and the largest node degree is $k_{\max}$, then the size of cliques in G is not larger than $(k_{\max} + 1)$. With k descending from $(k_{\max}+1)$ to 1, the determination process searches for the k-clique(s) of each node. As every two nodes in a clique are adjacent, the degrees of all nodes in a k-clique must be larger than $(k - 1)$. Additionally, since only the largest clique(s) for each node are interested, the searching process stops seeking smaller cliques for a node if it has been assigned to the larger ones. Therefore, the determination process only searches for k-clique(s) of a node when the following three conditions are satisfied: 1) the node degree is not smaller than $(k - 1)$ (lines 7–9); 2) the node has not been assigned to any cliques (lines 10–12); and 3) at least $(k - 1)$ adjacent nodes have a degree no smaller than $(k - 1)$ (lines 13–16). If all these conditions are satisfied, the determination process transforms the problem of finding all the k-clique(s) that contain this node to that of searching for the $(k - 1)$-clique(s) constituted by its neighbors (line 17). Then all the discovered k-cliques are added to the set of clique nodes (lines 18 and 19).

Algorithm 1 Determining the Clique Nodes of G^c

```
Input: Original graph $G = (V, E)$;
Output: Set of clique nodes $V^c$;
 1: $V^c \leftarrow \phi$;
 2: Calculate the degree $k(v_i)$ of each node $v_i \in V$;
 3: $k_{\max} \leftarrow \max_{v_i \in V} k(v_i)$;
 4: Sort the nodes in descending order of the degree;
 5: for $k = k_{\max} + 1$ to 1 do
 6:     for each node $v_i \in V$ do
 7:         if $k(v_i) < k - 1$
 8:             no more $k$-cliques exist and goto Outer loop (Line 22);
 9:         end if
10:         if $v_i$ has been assigned to one clique node
11:             goto Inner loop (Line 20);
12:         end if
13:         $Neigh(v_i) \leftarrow \{v_j | (v_j$ is adjacent to $v_i$ and $k(v_j) \geq k - 1\}$;
14:         if $|Neigh(v_i)| < k - 1$
15:             $v_i$ cannot constitute $k$-cliques and goto Inner loop (Line 20);
16:         end if
17:         if the nodes in $Neigh(v_i)$ can constitute $q$ $(k - 1)$-cliques $(q \geq 1)$
18:             $V^c \leftarrow V^c \cup \{v_i, ((k-1) - \text{clique})_1\} \cup, \ldots, \cup \{v_i, ((k-1) - \text{clique})_q\}$;
19:         end if
20:         Inner loop;
21:     end for
22:     Outer loop;
23: end for
```

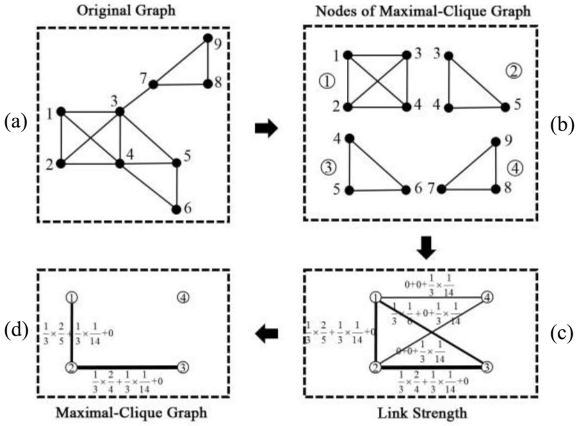

Original Graph
Nodes of Maximal-Clique Graph
Link Strength
Maximal-Clique Graph
(a)
(b)
(c)
(d)
1
2
3
4
5
6
7
8
9
①
②
③
④
$0+0+\frac{1}{3}\times\frac{1}{14}$
$\frac{1}{3}\times\frac{1}{6}+0+\frac{1}{3}\times\frac{1}{14}$
$\frac{1}{3}\times\frac{2}{5}+\frac{1}{3}\times\frac{1}{14}+0$
$0+0+\frac{1}{3}\times\frac{1}{14}$
$\frac{1}{3}\times\frac{2}{4}+\frac{1}{3}\times\frac{1}{14}+0$

Fig. 3. Illustration of constructing the maximal-clique graph G^c from the original graph G. (a) Original graph G contains 9 nodes and 14 edges. (b) Clique nodes of G^c contains one 4-clique and three 3-cliques. (c) Link strength between different clique nodes is calculated and given beside each line. (d) w_{thr} is set to the averaged link strength of the graph in (c) and the final maximal-clique graph is determined.

Figs. 3(a) and (b) exemplify the above determination process of clique nodes. Fig. 3(a) shows the original graph with 9 nodes and 14 edges. The corresponding clique nodes are given in Fig. 3(b), including one 4-clique and three 3-cliques. As can be seen, each node of the original graph is assigned to at least one clique node (i.e., maximal clique) of the corresponding maximal-clique graph. For example, nodes v_7, v_8, and v_9 belong to clique node v_4^c, while nodes v_4 and v_5 belong to both clique nodes v_2^c and v_3^c.

B. Measuring Link Strength Between Cliques Nodes

Basic observations on a graph indicate that the link strength between two cliques depends on the ratios of

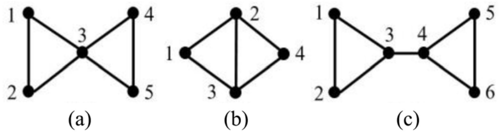


Fig. 4. Examples of (a) overlapping nodes, (b) overlapping edges, and (c) joint edges.

overlapping nodes, overlapping edges, and joint edges. Given two cliques, an overlapping node is a common node of both cliques, an overlapping edge is a common edge of both cliques, and a joint edge is an edge that connects a node in one clique with a node in the other. Taking two 3-cliques as an example in Fig. 4, node v_3 in Fig. 4(a) is an overlapping node, the edge between nodes v_2 and v_3 in Fig. 4(b) is an overlapping edge, and the edge between nodes v_3 and v_4 in Fig. 4(c) is a joint edge. Since each clique node in a maximal-clique graph is a clique, the link strength between two clique nodes can be evaluated in the same way as evaluating the link strength between two cliques. Given two clique nodes v_m^c and $v_n^c(m \neq n)$, the ratios of overlapping nodes, overlapping edges, and joint edges are calculated as

$$\ell_{\text{on}}\left(v_m^c, v_n^c\right) = \frac{N\left(v_m^c \bigcap v_n^c\right)}{N\left(v_m^c\right) + N\left(v_n^c\right) - N\left(v_m^c \bigcap v_n^c\right)} \tag{5}$$

$$\ell_{\text{oe}}\left(v_m^c, v_n^c\right) = \frac{\sum_{v_i, v_j \in \left(v_m^c \cap v_n^c\right)} A_{ij}}{\sum_{v_i, v_j \in V} A_{ij}} \tag{6}$$

$$\ell_{\text{je}}\left(v_m^c, v_n^c\right) = \frac{\sum_{v_i \in \left(v_m^c - v_n^c\right), v_j \in \left(v_n^c - v_m^c\right)} A_{ij}}{\sum_{v_i, v_j \in V} A_{ij}} \tag{7}$$

where $N(v_m^c)$ and $N(v_n^c)$ return the number of original nodes in clique nodes v_m^c and v_n^c, respectively, and $N(v_m^c \cap v_n^c)$ returns the number of original nodes in the intersection between clique nodes v_m^c and v_n^c. A is the adjacent matrix of the original graph and $A_{ij} = 1$, if $(v_i, v_j) \in E$; otherwise, $A_{ij} = 0$. Based on ℓ_{on}, ℓ_{oe}, and ℓ_{je}, the link strength $L(v_m^c, v_n^c)$ between v_m^c and v_n^c can be defined as a weighted sum of these three components

$$L\left(v_m^c, v_n^c\right) = \alpha \ell_{\text{on}}\left(v_m^c, v_n^c\right) + \beta \ell_{\text{oe}}\left(v_m^c, v_n^c\right) + \gamma \ell_{\text{je}}\left(v_m^c, v_n^c\right) \tag{8}$$

where $\alpha, \beta, \gamma \in [0, 1]$ are the weights that control the impact of overlapping nodes, overlapping edges, and joint edges on the link strength, respectively. Note that $\alpha + \beta + \gamma = 1$. In this paper, for simplicity, all the weights are set to 1/3, namely, we consider that these three parts have equal contributions to the link strength between two clique nodes. Explicitly, we can derive that $L(v_m^c, v_n^c)$ is within [0, 1), and only if there are no overlapping nodes, overlapping edges, and joint edges between v_m^c and v_n^c, $L(v_m^c, v_n^c) = 0$.

In Fig. 3(c), the nonzero link strength between the clique nodes in Fig. 3(b) is given. Taking the link between clique nodes v_1^c and v_2^c as an example, there are two overlapping nodes and one overlapping edge, but no joint edges. The total number of original nodes in clique nodes v_1^c and v_2^c is 5. The number of original edges in G is 14. Hence, $\ell_{\text{on}} = 2/5, \ell_{\text{oe}} = 1/14$, and $\ell_{\text{je}} = 0$, and the link strength between v_1^c and v_2^c is $L(v_1^c, v_2^c) = 1/3(2/5 + 1/14 + 0)$.

C. Determining Clique Edges of Maximal-Clique Graph

According to the previous two steps, the clique nodes and the link strength between them are obtained. Some links are strong, while the others are weak or even close to zero. If all the links with nonzero strength are admitted as clique edges, the following two problems may occur when detecting communities. First, the computational cost increases due to the large number of clique edges. Second, noise usually exists in a complex system, and thus weak links can interrupt the detection of community structure. In order to avoid the above problems, a threshold w_{thr} is operated on the clique edges to counteract the influence of noise and lower the complexity. A link is admitted as a clique edge only if its strength is beyond w_{thr}. The setting of w_{thr} will be discussed in Section V-B.

After determining clique edges, the original graph G is converted into a maximal-clique graph G^c. The link strength is considered as the weight of the corresponding edge. Take Fig. 3(d) as an example. With w_{thr} set as the average link strength, only two links in Fig. 3(c) are admitted as clique edges, resulting in a maximal-clique graph with four clique nodes and two clique edges.

IV. MCMOEA

This section details the implementation of the proposed MCMOEA, including the representation scheme, evolutionary operators, and the overall procedure.

A. Representation Scheme

How to represent solutions in EAs has a great influence on the design of evolutionary operators and the algorithmic efficiency. As stated in Section I, the existing representation schemes, either prototype-based or node-based, have certain limitations when applied to the overlapping community detection. To address these limitations, this paper proposes a new representation approach based on the introduced maximal-clique graph, namely, clique-based representation. In the proposed representation, each gene of an individual is an integer that represents the community label of the corresponding clique node of the maximal-clique graph. Thus, every clique node can only be assigned to a unique community, which is similar to the separate community detection. However, as the maximal-clique graph has the property that clique nodes are allowed to share the same original nodes, the overlapping nodes shared by different clique nodes can actually be assigned to multiple communities. Taking Fig. 5 as an example, node v_4 is shared by both clique nodes v_1^c and v_2^c [Fig. 5(b)]. Since v_1^c is assigned to community "1" and v_2^c is assigned to community "2" [Fig. 5(c)], node v_4 belongs to both communities "1" and "2" [Fig. 5(d)]. The detailed population initialization is described as follows.

Let $\boldsymbol{I}_i = (g_{i1}, g_{i2}, \ldots, g_{iM})$ be the ith individual in the population of MCMOEA, where g_{ij} is the jth gene of $\boldsymbol{I}_i(j = 1, 2, \ldots, M, i = 1, 2, \ldots, PS, M$ is the number of clique nodes of the maximal-clique graph and PS is the size of population). The initialization of $\boldsymbol{I}_i$ is conducted by using a local algorithm in [16] to form a locally optimal partition of clique nodes.

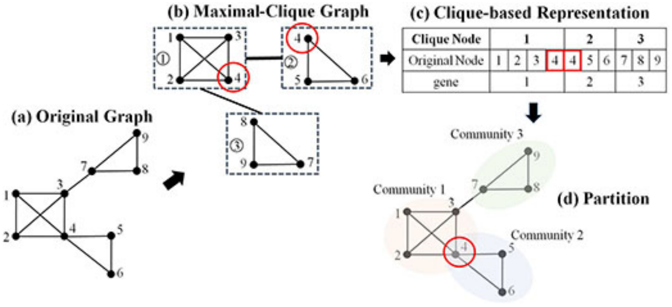


Fig. 5. Example for the clique-based representation. (a) Original graph G. (b) Corresponding maximal-clique graph G^c is constructed based on G and each node in G is allowed to be assigned to multiple clique nodes of G^c, e.g., node v_4 belongs to both clique nodes v_1^c and v_2^c. (c) Proposed clique-based representation is adopted to assign each clique node an integer to represent the community label this node belongs to. For example, the community label of clique node v_1^c is 1, indicating that v_1^c belongs to community "1." (d) Partition of the original graph G is directly determined based on the individual.

In detail, the algorithm builds the partition C as follows. First, the M clique nodes are randomly permuted and C is initialized as an empty set. Then, following the permutation order, each unassigned clique node v_m^c is assigned to the first community whose cohesiveness can be improved by including v_m^c. If v_m^c cannot improve the cohesiveness of any existing communities, a new community that only contains v_m^c is added to C. Considering that the maximal-clique graph is a weighted graph, the cohesiveness of a community c is measured as in [17]

$$F(c) = \frac{W_{\text{in}}(c)}{W_{\text{in}}(c) + W_{\text{out}}(c)} \tag{9}$$

where $W_{\text{in}}(c)$ and $W_{\text{out}}(c)$ are the total weights of internal edges and external edges of c, respectively. Based on the obtained partition C, the individual $\boldsymbol{I}_i$ can be initialized by setting each gene g_{im} as the community label of v_m^c in C.

The proposed clique-based representation scheme has three features. First, instead of using the original nodes, the proposed approach uses clique nodes (i.e., maximal cliques) as the basic unit of the representation. Since overlap is an intrinsic property of clique nodes in the maximal-clique graph, clique-based representation enables MOEAs to handle the overlapping community detection problem in a way similar to that of the separated community detection. Second, the clique-based representation does not require decoding individuals in the population evolution process, which largely lowers the computational complexity of MOEAs for overlapping community detection when compared with the indirect node-based approach. Third, the clique-based representation scheme has no limitations on the shape of communities and needs no prior community information, making it superior to the prototype-based representation scheme.

B. Evolutionary Operators

Evolutionary operators, including crossover and mutation, are the most important components of EAs, which significantly influence the population diversity and the convergence speed. Hence, it is crucial to select appropriate evolutionary operators for the algorithm. In this paper, to fit the proposed clique-based representation scheme, we adopt the one-way crossover operator [9] and design a new mutation operator based on the maximal-clique graph.

1) Crossover: In many classic EAs, the crossover operator generates offspring by randomly exchanging genes of two parental individuals [24], [35]. However, such an idea of the crossover is not suitable for the proposed clique-based representation since community labels in different individuals are not compatible. Take the two individuals A and B in Fig. 6(a) as an example. According to the clique-based representation scheme, both individuals indicate that clique nodes v_1^c, v_4^c, and v_6^c belong to the same community, but the community label is "0" in individual A but "1" in individual B. In this case, randomly exchanging genes between the two individuals can easily break the promising community structure formed by v_1^c, v_4^c, and v_6^c. Therefore, this paper employs the one-way crossover operator introduced in [9]. The one-way crossover randomly selects two parental individuals from the population, with one set as the source (denoted as $\boldsymbol{I}_s$), and the other set as the destination (denoted as $\boldsymbol{I}_d$). Then a clique node is randomly selected as the crossover seed e. Let l be the community label of e in $\boldsymbol{I}_s$ and $\boldsymbol{\kappa}$ be the set of clique nodes in $\boldsymbol{I}_s$ that have the same community label as l. An offspring $\boldsymbol{I}'_d$ is generated by modifying $\boldsymbol{I}_d$ so that all the clique nodes in $\boldsymbol{\kappa}$ are relabeled as l. By doing so, the offspring can merge the community structures of both parents. Using the two individuals in Fig. 6(a) as parents, Fig. 6(b) shows an example of offspring generation utilizing the above one-way crossover, given that v_1^c is selected as the crossover seed (i.e., $e = v_1^c$ and $l = 0$). For each individual, the crossover operator is executed with a predefined possibility p_c.

2) Mutation: A new mutation operator is proposed in this paper to fit the clique-based representation. The essential idea is to improve the communities with low cohesiveness by including clique nodes that have strong connections with them. In detail, for each individual $\boldsymbol{I}_i$ in the population, the mutation operator first calculates the cohesiveness of each community in $\boldsymbol{I}_i$ according to (9). Then an M-dimensional vector $\boldsymbol{r}_i$ is generated, in which each element r_{ij} is a random number uniformly distributed in [0, 1]. r_{ij} is compared with the cohesiveness of the community $c_{g_{ij}}$ that the clique node v_j^c belongs to. If r_{ij} is larger than $F(c_{g_{ij}})$, a clique node v_k^c adjacent to v_j^c is randomly selected through the roulette wheel selection. The larger the weight on the clique edge (v_j^c, v_k^c), the higher the probability that v_k^c will be selected. The mutation operator then changes the community label of v_j^c (i.e., g_{ij}) to that of v_k^c (i.e., g_{ik}). By doing so, each community in $\boldsymbol{I}_i$, especially the one with low cohesiveness, is given a chance to adjust its structure, which leads to a possible improvement in the overall fitness.

C. Overall Procedure of MCMOEA

Following the framework of MOEA/D [30], MCMOEA is implemented using the clique-based representation scheme and the evolutionary operators described above. As MOEA/D, MCMOEA needs to decompose the problem of overlapping

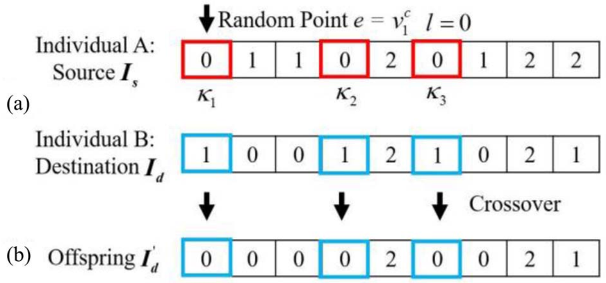


Fig. 6. Simple illustration of the one-way crossover operator. (a) Parental individuals. (b) Offspring.

community detection into several scalar optimization subproblems. Considering that the shape of PF is unknown and the weighted sum approach only works well for concave PF, Tchebycheff approach [27] is used in MCMOEA for decomposition. Let $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^{PS}$ be a set of evenly spread weight vectors, where $\boldsymbol{\lambda}^i = (\lambda_1^i, \lambda_2^i)$ $(\lambda_1^i, \lambda_2^i \in [0, 1]$, and $\lambda_1^i + \lambda_2^i = 1)$ and PS is the size of population (i.e., the number of subproblems). Based on the maximal-clique graph and two objective functions defined in (3) and (4), a subproblem can be formulated as

$$\text{minimize } g^{\text{te}}\big(\boldsymbol{x}|\lambda^i, \boldsymbol{z}^*\big) = \max\big\{\lambda_1^i \big|KKM(\boldsymbol{x}) - z_1^*\big|, \lambda_2^i \big|RC(\boldsymbol{x}) - z_2^*\big|\big\} \quad (10)$$

where $\boldsymbol{x}$ is a solution to the problem, $\boldsymbol{z}^* = (z_1^*, z_2^*)$ is the reference point, i.e., the best values found so far for the KKM and RC, respectively, and $i = 1, 2, \ldots, PS$. MCMOEA approximates the PF by minimizing the PS scalar subproblems defined above using a population with PS individuals.

As shown in Algorithm 2, MCMOEA contains three steps: 1) construction of the maximal-clique graph; 2) initialization; and 3) population evolution. In the first step, the maximal-clique graph G^c is constructed from the original graph G using the method introduced in Section III. In the second step, the population with PS individuals is initialized according to the clique-based representation scheme proposed in Section IV-A. PS weight vectors $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_{PS}$ are assigned to the PS individuals $\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_{PS}$, respectively. Then, the neighborhood $B(i)$ of each weight vector is determined by Euclidean distance. The reference point $\boldsymbol{z}^*$ is initialized by the KKM and RC values of $\boldsymbol{I}_1$ and the set of non-dominated solutions, NS, is initialized as an empty set. In the third step, the population is evolved through the evolution operators described in Section IV-B. A new individual $\boldsymbol{z}$ is generated by performing crossover and/or mutation on each existing individual $\boldsymbol{I}_i$. After evaluating $\boldsymbol{z}$ with the two objective functions (i.e., KKM and RC), the reference point, the neighborhood of $\boldsymbol{I}_i$, and the set of NS are updated accordingly.

The computational complexity of MCMOEA can also be analyzed from the above three steps. Suppose that there are N nodes in the original graph G, M clique nodes in the corresponding maximal-clique graph G^c, and the maximum node degree in G is $k_{\max}$. In the first step, the computational complexities of determining the clique nodes, measuring the link strength, and determining the clique edges are $O(N \times k_{\max}^3)$, $O(M^2)$, and $O(M)$, respectively. Hence, the computational complexity of the first step is $O(\max\{N \times k_{\max}^3, M^2\})$. In the second step, the computational complexity is bounded by the operation of population initialization which takes $O(M^2 \times PS)$. In the third step, the time complexity of crossover and mutation operators is related to M and can be realized in linear time, namely, $O(M)$. The other operations can be finished in constant time. Therefore, the time complexity of the third step is bounded by $O(M \times PS \times gen_{\max})$ (PS is the size of population and $gen_{\max}$ is the maximum number of generations). To sum up, the overall computational complexity of MCMOEA is $O(\max\{N \times k_{\max}^3, M^2 \times PS, M \times PS \times gen_{\max}\})$. It should be noted that M is usually smaller than or equal to N.

Algorithm 2 MCMOEA

```
Input:
• $G = (V, E)$: the original graph;
• $gen_{\max}$: the maximum number of generations;
• $PS$: the size of population;
• $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^{PS}$: a uniform spread of $PS$ weight vectors;
• $T$: the size of the neighborhood of each weight vector;
Output:
• $NS$: the set of non-dominated solutions.
1. Construction of the Maximal-clique Graph $G^c$:
    Determine the set of clique nodes $V^c$;
    Measure the link strength between clique nodes;
    Determine the set of clique edges $E^c$ based on the link strength.
2. Initialization:
    Initialize the population $P = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_{PS}\}$, where each individual
      $\boldsymbol{I}_i = (g_{i1}, g_{i2}, g_{i3}, \ldots, g_{iM})$ represents the current solution to the $i$-th
      subproblem;
    Initialize the neighborhood of each weight vector. For each $i =$
      $1, 2, \ldots, PS$, set $B(i) = \{i_1, i_2, \ldots, i_T\}$, where $\boldsymbol{\lambda}^{i_1}, \boldsymbol{\lambda}^{i_2}, \ldots, \boldsymbol{\lambda}^{i_T}$ are
      the $T$ closest weight vectors to $\boldsymbol{\lambda}^i$ in the Euclidean space;
    Initialize $z^*$ as $(KKM(\boldsymbol{I}_1), RC(\boldsymbol{I}_1))$;
    Initialize $NS$ as an empty set.
3. Population Evolution :
  for $g = 1$ to $gen_{\max}$
      for $i = 1$ to $PS$
          Generate a random number $r_i$ from $U(0,1)$;
          if $r_i < p_c$
             Randomly choose another individual from $B(i)$ and generate
             an offspring $\boldsymbol{x}$ by conducting the one-way crossover on this
             individual and $\boldsymbol{I}_i$;
          else
             Set $\boldsymbol{x}$ as $\boldsymbol{I}_i$;
          end if
          Generate an $M$-dimensional random vector $\boldsymbol{r}_i$ from $U^M(0, 1)$;
          for $j = 1$ to $M$
             if $r_{ij} > F(c_{g_{ij}})$
                Generate a new individual $z$ by mutating $\boldsymbol{x}$;
             end if
          end for
          if $KKM(z) < z_1^*$, set $z_1^*$ as $KKM(z)$;
          if $RC(z) < z_2^*$, set $z_2^*$ as $RC(z)$;
          for each individual $\boldsymbol{I}_j \in B(i)$
             if $g^{te}(\boldsymbol{I}_j|\lambda^i, z^*) > g^{te}(z|\lambda^i, z^*)$, replace $\boldsymbol{I}_j$ with $z$;
          end for
          Remove from $NS$ all the solutions that are dominated by $z$;
          Add $z$ to $NS$ if no solutions in $NS$ can dominate $z$;
      end for
  end for
$U$(0,1): the normalized uniform distribution.
```

V. Experiments and Discussion

In this section, a series of experiments are designed based on synthetic and real-world networks to validate the performance

TABLE I
PARAMETER SETTINGS OF SYNTHETIC NETWORKS

Part	Experiments		$\boldsymbol{\mu}$	$\boldsymbol{O_n}$	$\boldsymbol{O_m}$	$\boldsymbol{N}$	Other Parameters
1	Investigation of $\boldsymbol{w_{\text{thr}}}$		{0.1, 0.5}	{0.1N, 0.5N}	{2, 8}	10^3	$k = 20$, $k_{\max} = 50$, $\tau_1 = 2$, $\tau_2 = 1$, $N = 10^3$: $c_{\min} = 20$, $c_{\max} = 50$ $N = 5\times10^3$: $c_{\min} = 30$, $c_{\max} = 70$ $N = 10^4$: $c_{\min} = 40$, $c_{\max} = 100$
2	Investigation of gNMI and Q_{ov}		{0.1, 0.3, 0.5}	{0.1N, 0.3N, 0.5N}	{2, 4, 6, 8}	10^3	
3	Investigation of the different MOEAs	1. Solution Quality	{0.1, 0.3, 0.5}	{0.1N, 0.3N, 0.5N}	{2, 4, 6, 8}	10^3	
		2. Speed	0.1	0.1N	2	{10^3, 5×10^3}	
4	The performance of MCMOEA on synthetic networks with different $\boldsymbol{\mu}$		{0.1, 0.2, 0.3, 0.4, 0.5}	{0.1N, 0.5N}	{2, 8}	10^3	
	The performance of MCMOEA on synthetic networks with different $\boldsymbol{O_n}$		{0.1, 0.5}	{0.1N, 0.2N, 0.3N, 0.4N, 0.5N}	{2, 8}	10^3	
	The performance of MCMOEA on synthetic networks with different $\boldsymbol{O_m}$		{0.1, 0.5}	{0.1N, 0.5N}	{2, 3, 4, 5, 6, 7, 8}	10^3	
	The performance of MCMOEA on synthetic networks with different $\boldsymbol{N}$		{0.1, 0.5}	{0.1N, 0.5N}	{2, 8}	{10^3, 5×10^3, 10^4}	
5	Comparison	1. Solution Quality	{0.1, 0.3, 0.5}	{0.1N, 0.3N, 0.5N}	{2, 4, 6, 8}	10^3	
		2. Speed	0.1	0.1N	2	{10^3, 5×10^3, 10^4}	

of the proposed MCMOEA. First, the generation model of synthetic networks and evaluation indexes in use are introduced. Second, the influences of the parameter w_{thr}, evaluation indexes and MOEA frameworks on the performance of MCMOEA are experimentally tested. Third, the performance of MCMOEA is evaluated on synthetic networks with different characteristics and compared with other five representative algorithms. Finally, MCMOEA is applied to four real-world networks and the benefit of using a multiobjective framework to detect overlapping communities is illustrated.

In all experiments, the parameters of MCMOEA are set as follows. PS and $gen_{\max}$ are set as 100 and 50, respectively [27]. The crossover probability is set as a relatively high value, 0.7, to increase the population diversity [27]. The neighborhood size T is set as 20 according to the suggestion of MOEA/D [30]. All the experiments are carried out on computers with Intel Core i5-3470 (3.20 GHz) CPU, 16 GB RAM, and Ubuntu 12.04 LTS 64-bit operating system.

A. Synthetic Networks and Evaluation Indexes

In this paper, the Lancichinetti–Fortunato–Radicchi (LFR) model [36] is adopted to produce synthetic networks. A synthetic network under LFR can be described as $\text{LFR}(N, k, k_{\max}, \tau_1, \tau_2, c_{\min}, c_{\max}, \mu, O_n, O_m)$. N is the number of nodes. k and $k_{\max}$ are the average node degree and the maximum node degree, respectively. τ_1 and τ_2 are the exponents of the power law distributions that the node degrees and the community sizes, respectively follow. $c_{\min}$ and $c_{\max}$ are the minimum and the maximum size of each community. $\mu \in [0, 1]$ is a mixing parameter that controls the average ratio of the external links to the total links of each node. If $\mu = 0$, all the edges in the network are intraconnections. If $\mu = 1$, all the edges are interconnections. Therefore, a larger μ indicates a more ambiguous community structure. O_n and O_m are two parameters specially defined for controlling the overlapping rate of communities in the network. O_n is the number of overlapping nodes, evaluating overlapping density among communities. Similar to μ, the higher the value of O_n, the more ambiguous the community structure is. O_m, namely, overlapping membership, is the number of communities to which each overlapping node belongs. The difficulty of the detection problem increases with the rise of O_m.

Once the above parameters are determined, a synthetic network can be generated from the LFR model using the method in [36]. In this paper, the settings of LFR parameters follow the suggestions in [27] and [37]. k, $k_{\max}$, τ_1, and τ_2 are fixed at 20, 50, 2, and 1, respectively. N is set at three levels: 1000, 5000, and 10000. $(c_{\min}, c_{\max})$ is set as (20, 50), (30, 70), and (40, 100) at different levels of N, respectively, since the community size generally slightly increases with the scale of the network. μ varies from 0.1 to 0.5, O_n varies from $0.1N$ to $0.5N$, and O_m varies from 2 to 8. In each experiment, a set of networks with different settings of N, μ, O_n, and O_m is considered, which is summarized in Table I. The effectiveness and efficiency of MCMOEA can thus be studied at different levels of network scale, community ambiguity, and overlapping rate.

As a multiobjective algorithm, MCMOEA yields a set of non-dominated solutions, none of which is superior to the other on both objectives of KKM and RC. In order to facilitate comparison with algorithms that treat overlapping community detection as a single-objective problem, an evaluation index is needed to select one solution from the set of non-dominated solutions. In this paper, the generalized normalized mutual information (gNMI) [16] and the modularity (Q_{ov}) [38] are adopted as the evaluation criteria. The first index, gNMI, evaluates the similarity between the true partition and the detected one, which is only suitable when the true community structure is already known. The second index, Q_{ov}, measures the difference between the fraction of edges within the given communities and the expected fraction if edges are distributed at random. Obviously, Q_{ov} can be used without knowing the true community structure. Therefore, gNMI can only be used in the experiments on synthetic networks, while Q_{ov} can be used on both synthetic and real-world networks. Both gNMI and Q_{ov} range from 0 to 1. The higher values of the gNMI and Q_{ov} are, the better the quality of a solution will be.

B. Investigation of the Parameter w_{thr}

As stated in Section III-C, w_{thr} is a threshold of link strength for determining clique edges of the maximal-clique graph. Only links with strength beyond w_{thr} will be admitted as clique edges. The setting of w_{thr} thus has a direct control over the distribution of clique edges. It may also have an indirect yet significant influence on the algorithmic performance since MCMOEA operates on the maximal-clique graph. To study the effect of $w_{\rm thr}$ and find its appropriate setting, the performance of MCMOEA using different $w_{\rm thr}$ is compared on eight synthetic networks. For detailed settings of the synthetic networks, please refer to part 1 of Table I. On each network, MCMOEA is tested with $w_{\rm thr}$ rising from 0 to $30w_{\rm avg}$, where $w_{\rm avg}$ is the average link strength of clique nodes in the maximal-clique graph. For each value of $w_{\rm thr}$, MCMOEA is run for 30 independent times and the average gNMI value is reported for comparison. Fig. S2 in the supplementary material, plots the average gNMI as a function of $w_{\rm thr}/w_{\rm avg}$.

From Fig. S2 in the supplementary material, it can be observed that in most cases the MCMOEA with extreme $w_{\rm thr}$ values performs the worst, e.g., $w_{\rm thr} = 0$ or $w_{\rm thr} = 30w_{\rm avg}$. This observation is not surprising. If $w_{\rm thr}$ is too small, a large number of links between clique nodes are kept as clique edges, which interferes community detection as the noise between communities increases. On the opposite, if $w_{\rm thr}$ is too large, most links between clique nodes are filtered out, which causes possible loss of useful information for revealing the actual community structure. Besides the above observation, it is also noticed that the curves in Fig. S2 in the supplementary material, have consistent shapes. That is, the average gNMI value rises from $w_{\rm thr} = 0$, reaches the top at about $w_{\rm thr} = w_{\rm avg}$, remains steady for a small interval, and then drops until $w_{\rm thr} = 30w_{\rm avg}$. Such a phenomena suggests that $w_{\rm thr} = w_{\rm avg}$ is a generally good setting no matter what kind of network it is. Hence, the $w_{\rm thr}$ is always set to $w_{\rm avg}$ in the following experiments.

Comparing the curves in Fig. S2 in the supplementary material, it can also be summarized that the decreasing trend after the turning point becomes less steep as μ, O_n, or O_m increases. As stated in Section V-A, larger values of μ, O_n, and O_m indicate a more ambiguous community structure and a higher overlapping rate. In this case, a larger $w_{\rm thr}$ is preferred for filtering out more links so that the community structure becomes clearer.

C. Investigation of gNMI and Q_{ov}

Since gNMI and Q_{ov} are both evaluation indexes suitable for synthetic networks, an experiment is designed here to investigate the differences between the partitions chosen by gNMI and Q_{ov}. To make a comprehensive test, μ, O_n, and O_m are all set to different levels, which are {0.1, 0.3, 0.5}, {0.1*N*, 0.3*N*, 0.5*N*}, and {2, 4, 6, 8}, respectively. *N* is set to 1000. The detailed parameter settings of the synthetic networks are listed in part 2 of Table I. On each network, MCMOEA is run for 30 independent times. For each run, the best gNMI value and the corresponding Q_{ov} (gNMI_Q_{ov}) value, as well as the best Q_{ov} value and the corresponding gNMI (Q_{ov}_gNMI) value are recorded. The average results of each index over 30 runs are reported in Fig. S3 and Table SI in the supplementary material. From Fig. S3 in the supplementary material, it can be observed that the difference between the best gNMI and Q_{ov}_gNMI, or between the best Q_{ov} and gNMI_Q_{ov} is slight, indicating the solutions chosen by gNMI and Q_{ov} are similar.

Additionally, to make a further comparison, this paper also examines the cumulative distributions of community sizes of partitions chosen by gNMI and Q_{ov} and compares them with the known ground truth. The results are presented in Figs. S4–S6 in the supplementary material. As can be seen, compared to Q_{ov}, the cumulative distribution curves of gNMI are closer to those of the true partitions on almost all networks, implying that gNMI may be slightly more suitable as the evaluation index for synthetic networks. Therefore, in the following experiments, gNMI is used as the criterion on synthetic networks and Q_{ov} is used on real-world networks.

D. Comparisons of MCMOEAs Based on Different MOEAs

As stated in Section II-C, different kinds of MOEAs are applicable for the proposed MCMOEA [39]–[43]. Therefore, in this section, except MOEA/D, other state-of-the-art MOEAs are adopted to implement MCMOEA for investigating the performance of MCMOEA based on different MOEAs. Among the existing MOEAs, NSGA-II [31] is a classical domination-based MOEA, which has been widely used in many fields and shown excellent performance. MOEA_DLA [33] is a new MOEA, which achieves a competitive performance by taking advantages of both MOEA/D and NSGA-II. Therefore, this paper chooses NSGA-II and MOEA_DLA as the representatives of MOEAs to implement MCMOEA and compares their performance with MOEA/D-based MCMOEA. The implementation details of MCMOEA using NSGA-II and MOEA_DLA as the MOEA frameworks are, respectively, provided in Algorithms S1 and S2 in the supplementary material.

The parameter settings of the synthetic networks are listed in part 3.1 of Table I. To make a comprehensive comparison, μ, O_n, and O_m are all set to different levels, which are {0.1, 0.3, 0.5}, {0.1*N*, 0.3*N*, 0.5*N*}, and {2, 4, 6, 8}, respectively, and *N* is set to 1000. For the sake of fairness, parameters of the three MCMOEAs are set to the same values, i.e., $PS = 100$, $gen_{\max} = 50$, and $p_c = 0.7$. On each network, each MCMOEA is run for 30 independent times and the average gNMI is reported in Table SII in the supplementary material. Additionally, the computational speed of three MCMOEAs is also compared on several networks with different scales. The corresponding network information is listed in part 3.2 of Table I. The comparison results are reported in Table SIII in the supplementary material.

From Tables SII and SIII (in the supplementary material), we can see that, MOEA_DLA-based MCMOEA shows the best performance on partition accuracy, and MOEA/D-based MCMOEA consumes the least computational time. Overall, it can be observed that the differences among these three MCMOEAs on both partition quality and computational speed are slight. Therefore, in the following experiments, the one with the least time, i.e., MOEA/D-based MCMOEA is used as the representative to further validate the performance of MCMOEA.

E. Experiments on Synthetic Networks

In this section, based on the synthetic networks, two experiments are designed to test the effectiveness and efficiency of MCMOEA. The first experiment makes a comprehensive analysis of how MCMOEA performs on networks with different properties, including community ambiguity (μ), overlapping rate (O_n and O_m), and network scale (N). The second experiment compares the performance of MCMOEA with other five representative algorithms.

1) Performance of MCMOEA on Synthetic Networks: To well observe how the performance of MCMOEA varies on networks with the change of either μ, O_n, O_m, or N separately, the method of controlling variables is adopted here. That is to say, when investigating one parameter, the other ones are held constant. Therefore, this experiment is divided into four parts.

1) Verifying the performance of MCMOEA on networks with different μ and constant O_n, O_m, and N.
2) Verifying the performance of MCMOEA on networks with different O_n and constant μ, O_m, and N.
3) Verifying the performance of MCMOEA on networks with different O_m and constant μ, O_n, and N.
4) Verifying the performance of MCMOEA on networks with different N and constant μ, O_n, and O_m.

In each part, to give a comprehensive result, the fixed parameters except N are all set to two values, which are the minimum and maximum of their feasible ranges, respectively. As for N, considering the computational cost, it is always set to 1000 in the first three parts. The detailed parameters setting of the synthetic networks is presented in part 4 of Table I. For each network, 30 independent runs of MCMOEA are conducted and the average gNMI is used as result.

First, the performance of MCMOEA on networks with different μ is investigated. O_n, O_m, and N are set to $\{0.1N, 0.5N\}$, $\{2, 8\}$, and 1000, respectively. Through increasing μ from 0.1 to 0.5 with the step of 0.1, the performance of MCMOEA is evaluated on each network and the results are shown in Fig. S7 in the supplementary material. As can be seen, all curves in Fig. S7 in the supplementary material, present similar tendency that the performance of MCMOEA naturally decreases with the increase of μ, because a higher value of μ indicates a more ambiguous community structure.

Second, the performance of MCMOEA on networks with different O_n is evaluated. μ, O_m, and N are set to $\{0.1, 0.5\}$, $\{2, 8\}$, and 1000, respectively. The performance of MCMOEA is displayed in Fig. S8 in the supplementary material with O_n increasing from $0.1N$ to $0.5N$ with the step of $0.1N$. Obviously, we can see that with O_n increasing, MCMOEA performs worse and worse. This is not surprising because a larger O_n implies a higher overlapping density, and such a high overlapping density usually blurs the boundary of the communities and increases the challenge to the community detection.

Third, the performance of MCMOEA on networks with different O_m is tested. μ, O_n and N are set to $\{0.1, 0.5\}$, $\{0.1N, 0.5N\}$, and 1000, respectively. O_m varies from 2 to 8 with the interval 1. The results are presented in Fig. S9 in the supplementary material. As can be seen, the performance of MCMOEA decreases with the increase of O_m. The fundamental reason is that, with the increase of O_m, it becomes harder and harder to successfully detect all the communities that each overlapping node belongs to, resulting in a lower partition accuracy.

Finally, we evaluate how the performance of MCMOEA changes on networks with the increase of N. μ, O_n, and O_m are set to $\{0.1, 0.5\}$, $\{0.1N, 0.5N\}$, and $\{2, 8\}$, respectively. N increases from 1000 to 5000 and 10000. The results are shown in Fig. S10 in the supplementary material. As can be seen, when μ, O_n, and O_m are all small, MCMOEA shows similar performance on networks with different N, such as $\mu = 0.1$, $O_n = 0.1N$, and $O_m = 2$. However, when the value of μ [observing Figs. S10(a)–(d) in the supplementary material, separately] becomes large, the performance of MCMOEA decreases with the increase of N. The similar conclusions can also be obtained on O_n [comparing Fig. S10(a) with (c) or Fig. S10(b) with (d) in the supplementary material] and O_m [comparing Fig. S10(a) with (b) or Fig. S10(c) with (d) in the supplementary material]. The fundamental reason of these phenomena lies in insufficient population diversity for the large scale networks. As we know, a larger N usually means a longer length of the individual and more communities. Meanwhile, with the increase of μ (O_n or O_m), the number of communities also increases. However, for the same population size and evolutionary generation of MCMOEA, the increases of the individual length and the number of communities inevitably result in a decrease of population diversity, leading to a decrease in the quality of the final solutions.

In conclusion, we can obtain that MCMOEA works well on synthetic networks with different combinations of μ, O_n, O_m, and N. For example, in Fig. S7 in the supplementary material, when O_n, O_m, and N are set to $0.1N$, 2, and 1000, the value of gNMI is always larger than 0.85, even when μ increases to 0.5. In Fig. S8 in the supplementary material, when $\mu = 0.1$, $O_m = 2$, and $N = 1000$, the value of gNMI is larger than 0.8 even increasing O_n to $0.5N$. Likewise, as can be seen in Fig. S9 in the supplementary material, the value of gNMI is also larger than 0.8 even O_m increases to 8 with $\mu = 0.1$, $O_n = 0.1N$, and $N = 1000$. Additionally, even though all the parameters are set to the maximum of their feasible ranges, i.e., $\mu = 0.5$, $O_n = 0.5N$, $O_m = 8$, and $N = 10000$, the value of gNMI is still larger than 0.5 as shown in Fig. S10 in the supplementary material.

2) Comparisons With Other Representative Algorithms: In this section, the performance of MCMOEA is compared with other five algorithms: 1) clique percolation method (CPM) [12]; 2) link clustering [13]; 3) speaker-listener-based information propagation algorithm (SLPA) [18]; 4) MOEA based on the signed similarity for detecting communities from social networks (MEA*s*_SN) [27]; and 5) improved multiobjective quantum-behaved particle swarm optimization (IMOQPSO) [28]. The five algorithms are selected for comparison as they are either state-of-the-art methods for overlapping community detection or relevant to MCMOEA in some sense. In detail, CPM [12] first introduced the conception of the clique into the field of

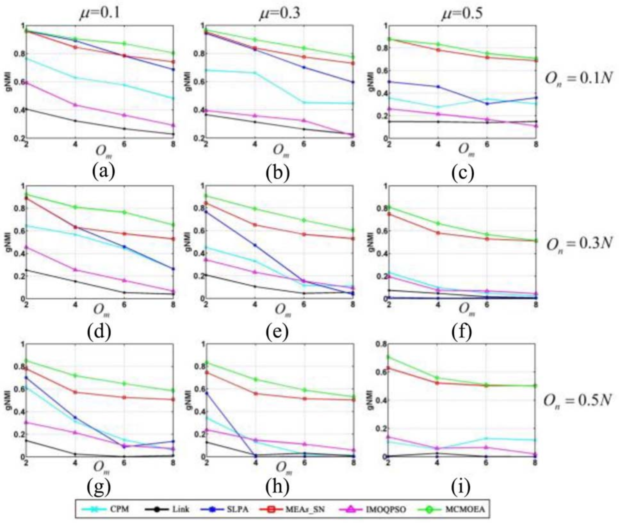


Fig. 7. Comparison results of gNMI values between MCMOEA and other five representative algorithms. (a) $\mu = 0.1$, $O_n = 0.1N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (b) $\mu = 0.3$, $O_n = 0.1N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (c) $\mu = 0.5$, $O_n = 0.1N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (d) $\mu = 0.1$, $O_n = 0.3N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (e) $\mu = 0.3$, $O_n = 0.3N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (f) $\mu = 0.5$, $O_n = 0.3N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (g) $\mu = 0.1$, $O_n = 0.5N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (h) $\mu = 0.3$, $O_n = 0.5N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$. (i) $\mu = 0.5$, $O_n = 0.5N$, $O_m = \{2, 4, 6, 8\}$, and $N = 1000$.

community detection. Link clustering [13] is featured by using links instead of nodes to discover community structures. As for SLPA [18], a previous review [37] claimed that it is a competitive method for overlapping community detection. MEA*s*_SN [27] and IMOQPSO [28], [44]–[47] are two representatives of MOEAs applicable to overlapping community detection. For each algorithm in comparison, the code is provided by its authors. CPM, link clustering, and SLPA are implemented with C++, MEA*s*_SN with C, and IMOQPSO with MATLAB. The tunable parameters of each algorithm are set as the suggestion of the corresponding paper.

The parameter settings of the synthetic networks used in this section are listed in part 5.1 of Table I. In this experiment, the networks are all with 1000 nodes. μ, O_n, and O_m are set to {0.1, 0.3, 0.5}, {0.1*N*, 0.3*N*, 0.5*N*}, and {2, 4, 6, 8}, respectively, which are the minimum, the median and the maximum of their feasible ranges, to represent the networks with different properties. Hence, these algorithms are compared on totally $3 \times 3 \times 4 = 36$ networks with different levels of community ambiguity and overlapping rate. By doing so, it is expected that the comparison can be comprehensive and thorough. For the sake of fairness, on each network, the algorithms with adjustable control parameters (i.e., CPM, link clustering, and SLPA) report their best results among different parameter settings, and the results of nondeterministic algorithms (i.e., SLPA, MEA*s*_SN, IMOQPSO, and MCMOEA) are averaged over 30 independent runs. The comparison results of gNMI among these algorithms are presented in Fig. 7 and Table SIV in the supplementary material. Additionally, in order to provide insight into the behaviors of different algorithms, this paper also examines the cumulative distribution of community sizes of each algorithm and compares it with the known ground truth on each network. The results are shown in Figs. S11–S13 in the supplementary material.

From Fig. 7, three observations can be obtained. First, as expected, the performance of all the algorithms decreases as either of the three network parameters (μ, O_n, or O_m) increases. Second, MCMOEA and MEA*s*_SN substantially outperform CPM, link clustering, and IMOQPSO on all the total 36 networks. Comparing with SLPA, MCMOEA and MEA*s*_SN achieve similar performance when μ, O_n, and O_m are all small. However, with either of the three parameters (μ, O_n, or O_m) increases, the performance of SLPA decreases dramatically and becomes much worse than MCMOEA and MEA*s*_SN. Third, when compared MCMOEA with MEA*s*_SN, MCMOEA performs better than, or at least as well as MEA*s*_SN on all networks. Moreover, MCMOEA is more robust to the variations of O_n and O_m. Take the first column of Fig. 7 as an example [i.e., Figs. 7(a), (d), and (g)]. With μ and O_m fixed and O_n rising from 0.1*N* to 0.5*N*, the change of gNMI for MCMOEA is smaller than that of MEA*s*_SN. Take Fig. 7(d) as another example. With μ and O_n fixed and O_m rising from 2 to 8, the performance of MCMOEA degrades more gently than that of MEA*s*_SN. From Figs. S11–S13 in the supplementary material, it can be observed that when μ, O_n, and O_m are all small, the cumulative distributions of community sizes of CPM, SLPA, MEA*s*_SN, and MCMOEA are similar and close to the true community size structures. With the increase of either μ, O_n, or O_m, all curves gradually deviate from the known ground truth. However, compared to other algorithms, the curves of MCMOEA are closer to those of the true partitions on almost all networks.

Although MCMOEA shows great superiority in gNMI to other algorithms, recent studies [48]–[50] pointed out that gNMI may have the selection bias problem that tends to choose solutions with more communities. That is to say, a partition with a higher number of communities is more likely to obtain a larger gNMI value. To avoid this possible unfairness caused by gNMI, another evaluation index, called scaled normalized mutual information (FNMI) [49], is also adopted to evaluate the performance of all algorithms in comparison. As an adjustment of gNMI, FNMI [49] can overcome the defect of gNMI to some extent by punishing partitions that have a number of communities either too higher or too lower than the true number. Similar to gNMI, FNMI is in the range of [0, 1] and a larger FNMI represents a better partition. The comparison results of FNMI among all algorithms are presented in Fig. S14 and Table SV in the supplementary material. From Fig. S14 in the supplementary material, it can be observed that the comparison results of FNMI are roughly similar to those of gNMI. For example, the performance of CPM, link clustering and IMOQPSO are much worse than those of MCMOEA and MEA*s*_SN on all networks. SLPA has the similar performance to MCMOEA and MEA*s*_SN when μ, O_n, and O_m are all small. However, with the increase of either μ, O_n, or O_m, its performance dramatically decreases and becomes much worse than MCMOEA and MEA*s*_SN. The only difference lies in the relationship between the performance of MEA*s*_SN and MCMOEA. When measured by gNMI, MCMOEA performs better than, or at least similar to MEA*s*_SN on almost all

TABLE II
COMPUTATIONAL TIME OF MCMOEA AND MEA*s*_SN

Time(s) Algorithm	*N*=1000	*N*=5000	*N*=10000
MCMOEA	23.30	1122.97	7974.53
MEA*s*_SN	140.77	12056.93	102073.90

networks as shown in Fig. 7. However, when measured by FNMI, MCMOEA shows worse performance than MEA*s*_SN on networks with a larger μ. To uncover the fundamental reason of such difference, this paper examines both community information of partitions obtained by MCMOEA and MEA*s*_SN, including the number of communities (c_n), the minimum community size ($c_{\min}$), and the maximum community size ($c_{\max}$), and then compares them with the known ground truth. As shown in Table SVI in the supplementary material, it can be noticed that when μ is large, $c_{\max}$ of MEA*s*_SN is much larger than that of both MCMOEA and the true partition, indicating MEA*s*_SN tends to merge small communities into a large one on the networks with a high value of μ. As a result, the obtained c_n is decreased and its difference with the true number of the communities is shortened. In this case, the punishment of FNMI for the partition is alleviated, and thus MEA*s*_SN obtains a high FNMI value.

Based on the above results, we can see that only MEA*s*_SN can be comparable with the proposed MCMOEA on partition accuracy. However, as stated in Section I, the indirect representation scheme adopted in MEA*s*_SN leads to a high computational complexity as $O(N^2 \times PS \times gen_{\max})$ [27]. Nevertheless, the developed MCMOEA only costs $O(\max\{N \times k_{\max}^3, M^2 \times PS, M \times PS \times gen_{\max}\})$, which is determined by three parts, i.e., the determination process of clique nodes of the maximal-clique graph, population initialization, and evolutionary operators. Here, N is the number of nodes of the original graph, $k_{\max}$ is the maximum node degree of the original graph, and M is the number of clique nodes of the corresponding maximal-clique graph. PS is the size of population and $gen_{\max}$ is the maximum number of generations. Since M is usually smaller than or equal to N, both $O(M^2 \times PS)$ and $O(M \times PS \times gen_{\max})$ are much less than $O(N^2 \times PS \times gen_{\max})$. Thus, $O(N \times k_{\max}^3)$ is the key differentiator between MEA*s*_SN and MCMOEA. According to the analysis of pseudo code of Algorithm 1, it can be obtained that $O(N \times k_{\max}^3)$ is the worst case complexity of the clique nodes determination process. Additionally, in most large real-world networks, $k_{\max}$ is generally kept at 10^2, which is much smaller than N and does not largely increase with the rise of N.[1] Therefore, the computational complexity of MCMOEA is much lower than that of MEA*s*_SN. Moreover, the larger the network is, the more obvious superiority of MCMOEA to MEA*s*_SN will be. To further validate the above analysis, an experiment is conducted here to compare the actual run time between these two algorithms on several synthetic networks with different scales. The detailed network information is listed in part 5.2 of Table I. On each network, MCMOEA and MEA*s*_SN are, respectively, run for 30 independent times with equal computational budget ($PS = 100$ and $gen_{\max} = 50$) and the average computational time is shown in Table II for comparison. As can be seen, the computational time of MCMOEA is always much less than that of MEA*s*_SN when *N* increases from 1000 to 10,000 and the advantage of MCMOEA enlarges with the increase of *N*. These experimental results are consistent with the theoretical analysis of the computational complexity, which demonstrates the proposed clique-based representation is beneficial for reducing the computational cost of MOEAs for overlapping community detection. All the above observations indicate that MCMOEA is a competitive and promising method for overlapping community detection.

TABLE III
INFORMATION OF TWO LARGE REAL-WORLD NETWORKS

Network	*N*	*E*	d_{ave}
Word Association	10617	63785	12.02
Scientific Collaborators	31163	120029	7.70

F. Experiments on Real-World Networks

For further validation, MCMOEA is applied to detect the community structures of four real-world networks, including two small networks and two large networks. The two small networks are about word association [13], each baring one of the following two overlapping community structures: 1) two communities share multiple nodes and 2) one node belongs to several communities. The two large networks are about word association [51] and scientific collaborators [52]. The word association network was created by the University of South Florida and University of Kansas, which adopted words as stimulus and asked participants to write the first word that came into mind. The scientific collaborator network describes coauthorships between scientists that post preprints on the condensed matter E-print archive. Information about the two large real-world networks is presented in Table III. Since the actual community structures of these real-world networks are unknown, Q_{ov} is adopted as the evaluation index. Additionally, the distributions of community sizes (nodes per community) and overlapping membership (communities per node) in large real-world networks are often found to follow power laws approximately [12]. Hence, for the two large networks, the cumulative distributions of community sizes and overlapping membership are also presented.

Fig. 8 shows an example of the partitions found by MCMOEA for the two small networks. In Fig. 8(a), the partition divides the network into two communities, one of which is related to “juice” and the other is related to “mixture”. “blend” and “blender” are detected as the overlapping nodes of the two communities. Such a rational partition shows that MCMOEA can deal with the situation when two communities have multiple overlapping nodes. In Fig. 8(b), the network is partitioned into four communities, each of which represents one meaning of the word “brush”. “Brush” is detected as the overlapping node of the four communities. Such a partition indicates that MCMOEA can also capture the community structure where one node belongs to several communities.

As for the word association network, the partition found by MCMOEA achieves a Q_{ov} of 0.15. Figs. 9(a) and (b) present the cumulative distributions of community sizes and overlapping

[1] snap.stanford.edu/data/.

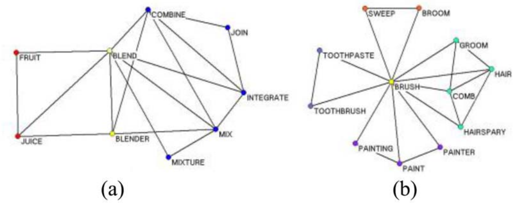


Fig. 8. Two simple examples of the word association network. Circles in different colors represent different communities, except yellow ones, which are overlapping nodes between different communities. (a) MCMOEA successfully captures the community structure that two communities share multiple nodes ($Q_{ov} = 0.34$). (b) MCMOEA successfully captures the community structure that one node belongs to multiple communities ($Q_{ov} = 0.38$).

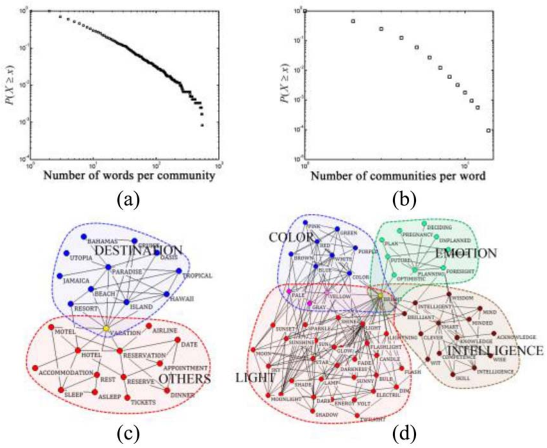


Fig. 9. Analysis of the partition found by MCMOEA for the word association network ($Q_{ov} = 0.15$). (a) Cumulative distribution of community sizes. (b) Cumulative distribution of overlapping membership of nodes. (c) Communities containing the word "vacation". (d) Communities containing the word "bright".

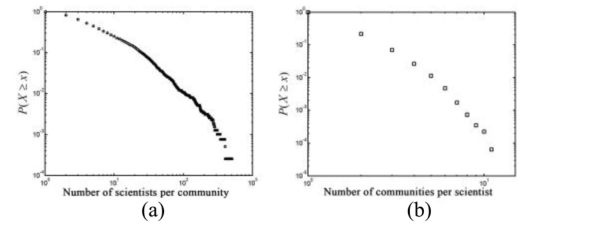


Fig. 10. Analysis of the partition found by MCMOEA for the scientific collaborators network ($Q_{ov} = 0.48$). (a) Cumulative distribution of community sizes. (b) Cumulative distribution of overlapping membership of nodes.

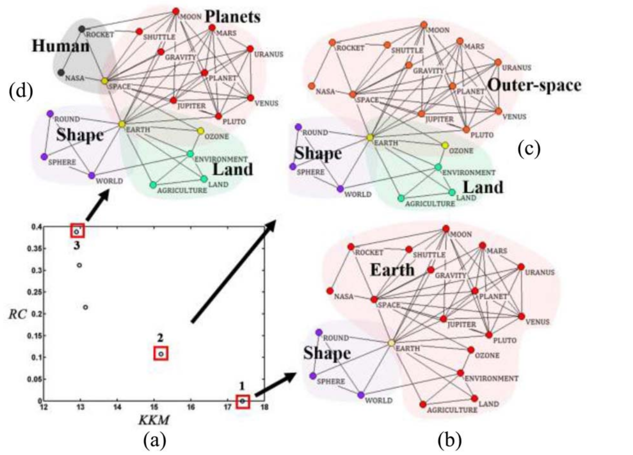


Fig. 11. Example of the hierarchical structure behind the non-dominated solutions obtained by MCMOEA. Nodes in different communities are color coded and the yellow ones are overlapping nodes between different communities. (a) Set of non-dominated solutions. (b) Partition corresponding to solution 1 ($Q_{ov} = 0.46$). (c) Partition corresponding to solution 2 ($Q_{ov} = 0.43$). (d) Partition corresponding to solution 3 ($Q_{ov} = 0.38$).

membership obtained from the partition, respectively. As can be observed, both of them are close to the power law distribution. Besides, since the nodes in the word association network are plain English words, the rationality of the partition can be evaluated based on the meanings of the communities. Using "vacation" and "bright" as examples, Figs. 9(c) and (d) draw the communities that the two words belong to, respectively. As shown in Fig. 9(c), "vacation" belongs to two communities, one of which contains tourist destinations, such as "Island" and "Hawaii," while the other contains things and actions that one may take during a vacation. Fig. 9(d) shows that the word "bright" is involved in four communities, i.e., "color," "light," "intelligence," and "emotion," each of which represents one meaning of "bright". Besides "bright", MCMOEA also successfully detects other overlapping nodes between the communities of "color" and "light," i.e., "pale," "yellow," and "gray." All the above indicates that MCMOEA is able to find a reasonable partition for the word association network.

For the scientific collaborators network, the partition found by MCMOEA achieves a Q_{ov} of 0.48. Figs. 10(a) and (b) show the cumulative distributions of community sizes and overlapping membership, respectively. As can be seen, both of the cumulative distributions approximately follow the power law distribution. The above results indicate that MCMOEA can find a rational partition for the large real-world networks.

G. Benefit of Using Multiobjective Framework

As a multiobjective algorithm, MCMOEA generates a set of non-dominated solutions, each of which corresponds to a partition of the given network. Previous studies claimed that the non-dominated solutions have a hierarchical structure [24], [27] and thus, can reveal the community structure of the given network on different levels. To validate such a benefit, MCMOEA is applied to a small hierarchical word association network. The five non-dominated solutions found by MCMOEA are presented in Fig. 11(a). Figs. 11(b)–(d) are the visualization of the partitions corresponding to three non-dominated solutions, which are labeled as 1, 2, and 3, respectively. As shown in Fig. 11(b), solution 1 partitions the network into two communities, namely, "shape" and "earth." Fig. 11(c) shows that solution 2 further divides the community "earth" in solution 1 into two new communities "land" and "outer-space." Solution 3 continues to divide the community "outer-space" in solution 2 into two new communities "human" and "planets," as shown in Fig. 11(d). It can be concluded that the community structure becomes more and more concrete from solutions 1–3. All the three solutions are rational partitions, but they reveal the community structure of the network on different levels. The above example confirms that MCMOEA can provide hierarchical partitions of the given network and thus allows users to explore the network on different levels.

VI. Conclusion

This paper proposes a novel MCMOEA, for overlapping community detection. In MCMOEA, we introduce the maximal-clique graph by using a set of maximal cliques as nodes and the links among maximal cliques as edges. Then based on the maximal-clique graph, a clique-based representation scheme is proposed. Since two maximal cliques are allowed to share the same nodes of the original graph, overlap is an intrinsic property of the nodes of the maximal-clique graph, which exactly characterizes the overlapping communities. Attributing to this property, the new representation scheme allows MOEAs to handle the overlapping community detection problem in a way similar to that of the separated community detection, such that the optimization problem is simplified. As a result, MCMOEA could detect overlapping community structure with higher partition accuracy and lower computational cost when compared with the existing algorithms. Experiments on synthetic and real-world networks show the effectiveness and efficiency of MCMOEA. Comparisons with other five representative algorithms also confirm that MCMOEA is competitive and promising.

References

[1] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the world-wide Web," *Nature,* vol. 401, no. 6749, pp. 130–131, Sep. 1999.

[2] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. Comput. Commun. Rev.*, Cambridge, MA, USA, 1999, pp. 251–262.

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

[4] A. C. Gavin *et al.*, "Functional organization of the yeast proteome by systematic analysis of protein complexes," *Nature*, vol. 415, no. 6868, pp. 141–147, 2002.

[5] H. Q. Dinh *et al.*, "An effective method for evolving reaction networks in synthetic biochemical systems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 374–386, Jun. 2015.

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

[7] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E.*, vol. 69, no. 6, Jun. 2004, Art. no. 066133.

[8] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104.

[9] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms," in *Proc. Eur. Conf. Complex Syst.*, Apr. 2006.

[10] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proc. Nat. Acad. Sci. USA.*, vol. 101, no. 9, pp. 2658–2663, 2004.

[11] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

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

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

[14] S. Kelley, M. Goldberg, M. Magdon-Ismail, K. Mertsalov, and A. Wallace, "Defining and discovering communities in social networks," in *Handbook of Optimization in Complex Networks*. New York, NY, USA: Springer, 2012, pp. 139–168.

[15] F. Reid, A. McDaid, and N. Hurley, "Partitioning breaks communities," in *Mining Social Networks and Security Informatics*. Dordrecht, The Netherlands: Springer, 2013, pp. 79–105.

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

[17] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. Magdon-Ismail, and N. Preston, "Finding communities by clustering a graph into overlapping subgraphs," in *Proc. IADIS AC*, vol. 5. 2005, pp. 97–104.

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

[19] 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.

[20] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Proc. Parallel Problem Solving Nat.*, Dortmund, Germany, 2008, pp. 1081–1090.

[21] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Phys. Rev. E.*, vol. 84, no. 5, Nov. 2011, Art. no. 056101.

[22] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E.*, vol. 80, no. 5, Nov. 2009, Art. no. 056117.

[23] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

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

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

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

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

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

[29] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, and S. Stramaglia, "Identification of network modules by optimization of ratio association," *Chaos*, vol. 17, no. 2, 2007, Art. no. 023114.

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

[31] 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.

[32] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design Optim. Control*, Athens, Greece, 2001, pp. 95–100.

[33] N. Chen *et al.*, "An evolutionary algorithm with double-level archives for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1851–1863, Sep. 2015.

[34] P. R. J. Östergård, "A fast algorithm for the maximum clique problem," *Discrete Appl. Math.*, vol. 120, nos. 1–3, pp. 197–207, Aug. 2002.

[35] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proc. 1st Int. Conf. Genet. Algorithm. Appl.*, vol. 154. Pittsburgh, PA, USA, 1985, pp. 154–159.

[36] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E.*, vol. 80, no. 1, 2009, Art. no. 016118.

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

[38] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A.*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.

[39] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

[40] J. Branke, S. Greco, R. Słowński, and P. Zielniewicz, "Learning value functions in interactive evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 88–102, Feb. 2015.

[41] J. E. Fieldsend and R. M. Everson, "The rolling tide evolutionary algorithm: A multiobjective optimizer for noisy optimization problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 103–117, Feb. 2015.

[42] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, in press, 2016.

[43] Q. Yang *et al.*, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, in press, 2016.

[44] L. Jiao, Y. Li, M. Gong, and X. Zhang, "Quantum-inspired immune clonal algorithm for global optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1234–1253, Oct. 2008.

[45] Y. Li, L. Jiao, R. Shang, and R. Stolkin, "Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation," *Inf. Sci.*, vol. 294, pp. 408–422, Feb. 2015.

[46] W.-N. Chen *et al.*, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, Apr. 2010.

[47] W.-N. Chen *et al.*, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241–258, Apr. 2013.

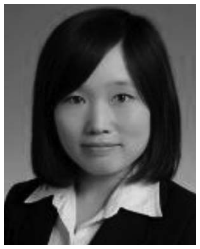
[48] S. Romano, J. Bailey, X. V. Nguyen, and K. Verspoor, "Standardized mutual information for clustering comparisons: One step further in adjustment for chance," in *Proc. 31st Int. Conf. Mach. Learn. JMLR W&CP*, vol. 32. Beijing, China, 2014, pp. 1143–1151.

[49] A. Amelio and C. Pizzuti, "Is normalized mutual information a fair measure for comparing community detection methods?" in *Proc. IEEE. Conf. Adv. Social Netw. Anal. Min.*, Paris, France, 2015, pp. 1584–1585.

[50] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, no. 10, pp. 2837–2854, 2010.

[51] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber, "The University of South Florida free association, rhyme, and word fragment norms," *Behav. Res. Methods Instrum. Comput.*, vol. 36, no. 3, pp. 402–407, Aug. 2004.

[52] S. Warner, "E-prints and the open archives initiative," *Library Hi Tech*, vol. 21, no. 2, pp. 151–158, 2003.



Xuyun Wen (S'14) received the M.S. degree from Beijing Normal University, Beijing, China, in 2014. She is currently pursuing the Ph.D. degree at Sun Yat-sen University, Guangzhou, China.

She is also a Research Assistant with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. Her current research interests include evolutionary algorithms and their applications on real-world problems.

Wei-Neng Chen (S'07–M'12) received the Bachelor's and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively.

He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. He has published over 50 papers in international journals and conferences. His current research interests include swarm intelligence algorithms and their applications on cloud computing, operations research, and software engineering.

Dr. Chen was a recipient of the IEEE Computational Intelligence Society Outstanding Dissertation Award in 2016 for his doctoral thesis and the National Science Fund for Excellent Young Scholars in 2016.

Ying Lin (M'12) received the Ph.D. degree in computer applied technology from Sun Yat-sen University, Guangzhou, China, in 2012.

She is currently an Assistant Professor with the Department of Psychology, Sun Yat-sen University. Her current research interests include computational intelligence and its applications in network analysis and cognitive diagnosis.

Tianlong Gu received the M.Eng. degree from Xidian University, Xi'an, China, in 1987, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 1996.

From 1998 to 2002, he was a Research Fellow with the School of Electrical and Computer Engineering, Curtin University of Technology, Perth, WA, Australia, and a Post-Doctoral Fellow with the School of Engineering, Murdoch University, Perth. He is currently a Professor with the School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin, China. His current research interests include formal methods, data and knowledge engineering, software engineering, and information security protocol.

Huaxiang Zhang received the Ph.D. degree from Shanghai Jiaotong University, Shanghai, China, in 2004.

He is currently a Professor with the School of Information Science and Engineering, Shandong Normal University, Jinan, China, where he was an Associate Professor with the Department of Computer Science from 2004 to 2005. He has authored over 100 journal and conference papers and has been granted eight invention patents. His current research interests include machine learning, pattern recognition, evolutionary computation, and Web information processing.

Yun Li (S'87–M'90) received the B.S. degree in radio electronics science from Sichuan University, Chengdu, China, in 1984, the M.Eng. degree in electroni engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, in 1987, and the Ph.D. degree in parallel processing for control engineering from the University of Strathclyde, Glasgow, U.K., in 1990.

Currently, he is a Professor in the School of Computer Science and Network Security, Dongguan University of Technology, Dongguan, China. Before that, he was a Professor in the School of Engineering, University of Glasgow, Glasgow, U.K. From 1989 to 1990, he was with U.K. National Engineering Laboratory and Industrial Systems and Control Ltd., Glasgow. He joined the University of Glasgow as a Lecturer in 1991, served as the two-year Founding Director of the University of Glasgow Singapore, Singapore, from 2011 to 2013, and was an Interim/Founding Director of the University's first joint programme in China in 2013, with the University of Electronic Science and Technology (UESTC), Chengdu, China. He established Evolutionary Computation workgroups for the IEEE Control System Society and European Network of Excellence in Evolutionary Computing (EvoNet) in 1998 and served on the Management Board of EvoNet from 2000 to 2005. He has been a Visiting Professor with Kumamoto University, Kumamoto, Japan, UESTC, and Sun Yat-sen University, Guangzhou, China. He has supervised over 20 Ph.D. students and has over 200 publications.

Prof. Li is a Chartered Engineer in the U.K.

Yilong Yin received the Ph.D. degree from Jilin University, Changchun, China, in 2000.

He is the Director of the Machine Learning and Applications Group and a Professor with Shandong University, Jinan, China. From 2000 to 2002, he was a Post-Doctoral Fellow with the Department of Electronic Science and Engineering, Nanjing University, Nanjing, China. His current research interests include machine learning, data mining, computational medicine, and biometrics.

Jun Zhang (M'02–SM'08) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Professor with the South China University of Technology, Guangzhou, China. His current research interests include computational intelligence, cloud computing, wireless sensor networks, operations research, and power electronic circuits. He has authored seven research books and book chapters, and over 50 IEEE TRANSACTIONS papers in the above areas.

Prof. Zhang was a recipient of the National Science Fund for Distinguished Young Scholars in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He was also appointed as the Changjiang Chair Professor in 2013. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS. He is the Founding and the Current Chair of the IEEE Guangzhou Subsection, the IEEE Beijing (Guangzhou) Section Computational Intelligence Society Chapters, and the ACM Guangzhou Chapter.