A Hybrid-based Conical Area Evolutionary Algorithm for Community Detection from Signed Social Networks

Weiqin Ying*, *Member, IEEE*, Pengfei Chao*, Yuehong Xie*, Zhenyu Wang*, Yu Wu†
*School of Software Engineering, South China University of Technology, Guangzhou 510006, China
†School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China

***Abstract*—Mining and analyzing community structures on social networks has drawn a great deal of attention during the past decades. Various social relationships, such as friends and foes, can be abstracted as signed social networks (SNs) containing both positive and negative links. However, most of existing community detection (CD) methods are designed primarily for unsigned networks containing only positive links. Therefore, it is significant to explore and design effective and efficient CD methods for SNs. In this paper, we first utilize decomposable characteristic of modularity Q to establish a bi-objective model for community detection in SNs. Afterwards, a conical area evolutionary algorithm based on Q-modularity (CAEAh-SN) is developed to solve this bi-objective model. Furthermore, a new tournament selection mechanism based on Q-modularity is applied to accelerate the convergence of modularity Q. Experimental results on both benchmark networks and random generated large SNs indicate that, compared with the existing algorithm MEAs-SN, CAEAh-SN not only achieves better community structures in term of both Q-modularity and NMI metrics, but also has stronger robustness to the SNs with high noises.**

***Index Terms*—Signed Social Networks, Community Detection, Bi-objective Optimization, Evolutionary Algorithms**

I. Introduction

In the real world, entities and relationships between them can be, in general, represented as various social networks (SN), such as friendship networks, Internet, neural networks and even metabolic networks. Digging out useful data information from various networks has received considerable attention among researchers. With the deep study of social networks, it is found that the majority of social networks have the characteristics of community structures, and each of them can be divided into many communities in various sizes. In most situations, the higher connection density between nodes implies a higher possibility they lie in the same community [1]–[4].

Specifically, a social network where there exists both positive and negative edges is called a signed social network (SNs). For instance, there are not only "like", "respect", "cooperation", "promote" on behalf of positive connections, but also "hate", "contempt", "competition", "inhibition" on behalf of negative connections in an interpersonal network. In unsigned networks, the community structure is defined as a group of nodes or vertices which have dense connections within groups and sparse connections between groups. Whereas for SNs, communities are defined not only by the density of links but also by the tendencies of links. That is to say, the links should be densely positive and sparsely negative in a community while densely negative and sparsely positive between communities. Compared with unsigned social networks, SNs reflect the real social networks more realistically [5]. In view of this consideration, methods for community detection from SNs have been the focus of many recent efforts [6]–[11].

In the last few decades, many classic approaches based on graph segmentation, such as betweenness centrality, minimum-cut and modularity optimization, have been proposed to uncover community structure in networks. In these classic approaches, the formation of communities can be attributed to the ternary closure relationship and the competition of nodes [12], [13]. Therefore, these classic approaches based on graph segmentation focus on identifying the strength of the edges between nodes.

In the betweenness centrality approach, the betweenness centrality index is essential in the analysis of social networks, but costly to compute. Motivated by the fast-growing need to compute centrality indices SNs on large, Ulrik et al. [14] presented a faster algorithm for betweenness centrality. This efficient algorithm is based on a new accumulation technique with a recursion formula that integrates well with traversal algorithms solving the single-source shortest-paths problem, and thus exploiting the sparsity of typical instances.

The minimum-cut graph partitioning divides the nodes to minimize the connection between the clusters and is suitable for networks where the number of partitions is known in advance and the size of each partition is similar. Based on minimum-cut graph partitioning, Shi et al. [15] treated image segmentation as a graph partitioning problem and further proposed a novel global criterion, the normalized cut, for solving the perceptual grouping problem in vision. This criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups, which aims at extracting the global impression of an image.

Corresponding author: Y. Wu (wuyu@gzhu.edu.cn)

This work was supported in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2015A030313204, in part by the Pearl River S&T Nova Program of Guangzhou under Grant 2014J2200052, in part by the National Natural Science Foundation of China under Grant 61203310 and Grant 61503087, in part by the Fundamental Research Funds for the Central Universities, SCUT, under Grant 2017MS043, in part by the Guangdong Province Science and Technology Project under Grant 2015B010131003 and in part by the China Scholarship Council (CSC) under Grant 201406155076 and Grant 201408440193.

The community detection can be regarded as community prediction when the number of communities is unknown. In particular, Girvan and Newman [16] presented the concept of modularity, one of the most known criteria for community detection, to evaluate the quality of obtained communities. The emergence of modularity makes community prediction possible. The Louvain method [17] was propose for fast unfolding of communities in large networks by the optimization of modularity. This method includes two phases. First, each node of the network is assigned a different community. For each node, if the gain of modularity takes place by removing this node from its community and by placing it in the community of one of its neighbours, the Louvain method adds this node to the community of this neighbour. This process is repeated iteratively until no positive gain is possible. The second phase considers the communities formed in first phase as new nodes. That is, this phase builts communities of communities. Therefore, the topological structures of networks consist in hierarchical communities.

According to the intrinsic properties, the problem of CD can be generally simulated as a multi-objective optimization problem (MOP) [18]. Evolutionary algorithms (EAs) are popular in solving MOPs and have made some achievements on the CD [19]–[28]. Consequently, several CD algorithms based on EAs have been proposed during the past decade.

Pizzuti [29] proposed a multiobjective genetic algorithm to uncover community structure in complex network based on the degrees of nodes. To be specific, this method optimizes two objective functions, the community score and the community fitness, which measure the quality of the division in communities of a network and the number of external links respectively. Through the use of Pareto optimality theory, a partitioning of the network is identified from a set of solutions that maximizes connections inside each community and minimizes the number of links between the communities. Moreover, it returns a set of solutions at different hierarchical levels rather than a single partitioning of the network. That is, the number of communities is determined by the optimal compromise values of the objective functions. He et al. [30] proposed another evolutionary community detection algorithm in social networks, which guided the evolutionary process using two fitness functions defined, respectively, as the average proportion of significant connection within a community and the mean vagueness of the community boundary between any two communities.

However, the above two CD algorithms based on EAs are only available for unsigned social networks. In recent years, Liu et al. developed a multi-objective evolutionary algorithm based on similarity for community detection from signed social networks, referred as MEAs-SN [28]. First, the original similarity is extended to the signed similarity based on the social balance theory in MEAs-SN. Then, in consideration of the natural contradiction between positive and negative links, two conflict objective functions are designed, respectively, for maximizing the sum of positive similarities within communities and maximizing the sum of negative similarities between communities. In this way, an original problem of detecting communities from SNs is modeled as a multi-objective one. Furthermore, the framework of a multi-objective evolutionary algorithm based on decomposition (MOEA/D) [31] is adopted to solve this multi-objective problem. In their rigorous experiments, the results on both benchmark networks and large-scale synthetic networks with 1000, 5000, and 10000 nodes show the effectiveness and efficacy of MEAs-SN. However, the signed similarity between a pair of vertices is based on whether they have the common neighbors and whether both of them have the same sign of positive links with the common neighbors. As a consequence, the performance of MEAs-SN decreases seriously when the community structure is sparse or the majority of the similarities between vertexes equal to 0.

In recent years, a conical area evolutionary algorithm (CAEA) [32] has been further proposed to enhance the runtime efficiency and population diversity of MOEA/D for bi-objective optimization by employing an efficient scheme of cone decomposition. Different from MOEA/D, CAEA not only partitions a MOP into N scalar subproblems, but also assigns an exclusive decision subregion to each subproblem. In addition, each subproblem uses a conical area indicator as its scalar objective to find a local non-dominated solution within its associated decision subset.

In this paper, a hybrid of signed similarity and the intrinsic weight method based on CAEA, referred to as CAEAh-SN, is further developed for CD from SNs. This approach adopts the framework of CAEA to improve the runtime efficiency and the solution quality of MEAs-SN.

The rest of this paper is organized as follows. Section II introduces the definition of CD from SNs and the strategy of conical decomposition. Section III describes objective functions and tournament selection designed for CD from SNs. The details of CAEAh-SN are given in Section **??**. Section V discusses the experimental results and the performance comparisons. Finally, the conclusions are given in Section VI.

II. Preliminaries

A. Community Detection from Signed Social Networks

A signed network $\mathbf{G} = \{V, E, w\}$. $V = \{v_1, v_2, \cdots, v_n\}$ is the set of nodes, $E = \{(v_i, v_j) | v_i, v_j \in V \wedge i \neq j\}$ is the set of edges, and $w_{ij} \neq 0$ is the weight of the edge between nodes v_i and v_j. The weight can be larger than 0 (positive relationship) or smaller than 0 (negative relationship). If all weights are larger than 0, $\mathbf{G}$ is an unsigned network. Otherwise it is a signed network. Let $C = \{C_1, C_2, \cdots, C_m\}$ be a set of communities in $\mathbf{G}$, that is, $C_a \subset V$ for $k = 1, 2, \cdots, m$. The problem of CD from SNs can be defined as follows:

$$\begin{cases} w_{ij} > 0, & v_i \in C_l \wedge v_j \in C_l \\ w_{ij} < 0, & v_i \in C_l \wedge v_j \in C_a \wedge l \neq k \end{cases} \tag{1}$$

where $l, k = 1, 2 \cdots, m$. In other words, the problem aims to find a community structure that maximizes both the sum of positive links within communities and that of negative links between communities.

B. Cone Decomposition

In the cone decomposition scheme of CAEA, a utopia point and a nadir point are firstly required in the objective space in order to partition the decision space. Let $A \subseteq \Omega$ be a set of solutions. The utopia point $\mathbf{f}^{\triangle}(A)$ and the nadir point $\mathbf{f}^{\triangledown}(A)$ over A are defined as follows:

$$\begin{aligned} &\mathbf{f}^{\triangle}(A) = (f_1^{\triangle}, f_2^{\triangle}),\\ &\mathbf{f}^{\triangledown}(A) = (f_1^{\triangledown}, f_2^{\triangledown}),\\ where\ &f_i^{\triangle} = \underset{x \in A}{min} f_i(x),\\ &f_i^{\triangledown} = \underset{x \in A}{max} f_i(x), \qquad i = 1, 2. \end{aligned} \tag{2}$$

For convenience, CAEA transforms an objective vector $\mathbf{y}$ to $\overline{\mathbf{y}}$ by $\overline{\mathbf{y}} = \mathbf{y} - \mathbf{f}^{\triangle}(A)$. After this transformation, the utopia point is at the origin (0,0). The observation vector for any point $\overline{\mathbf{y}} = (\overline{y}_1, \overline{y}_2)$ is $\overline{\mathbf{V}}(\overline{\mathbf{y}}) = (\overline{v}_1(\overline{\mathbf{y}}), \overline{v}_2(\overline{\mathbf{y}}))$, where $\overline{v}_i(\overline{\mathbf{y}}) = \frac{\overline{y}_i}{\overline{y}_1 + \overline{y}_2}$, and $i = 1, 2$. It means that all the observation vectors are on the line $\overline{v}_1(\overline{\mathbf{y}}) + \overline{v}_2(\overline{\mathbf{y}}) = 1$. Given a prescribed number of divisions N, a series of uniformly distributed observation vectors can be defined as $\overline{\mathbf{V}}^{(k)} = (\frac{k}{N-1}, 1 - \frac{k}{N-1}), k = 0, 1, \cdots, N-1$, called the center observation vector, which induces a resulting conical subregion $C^{(k)}$ where the observation vector of any point is closer to $\overline{\mathbf{V}}^{(k)}$ than to the other center vectors.

In CAEA, each sub-problem aims to search a local optimal solution with the minimal conical area in its associated conical sub-region. Given a solution $\overline{\mathbf{y}}' \in C^{(k)}, 0 \leq k \leq N-1$, the conical area for $\overline{\mathbf{y}}'$ in $C^{(k)}$ is the area of the portion $\tilde{C}(\overline{\mathbf{y}}') = \{\overline{\mathbf{y}} \in C^{(k)} | \neg(\overline{\mathbf{y}}' \prec \overline{\mathbf{y}}) \wedge \overline{\mathbf{y}} \prec \overline{\mathbf{y}}^r\}$, written as $s_k(\overline{\mathbf{y}}')$, where $\overline{\mathbf{y}}^r$ is the reference point, which should be an approximate infinity point dominated by all feasible solutions. The *k-th* conical sub-problem $g^{cone,k}(\mathbf{x})$ associated with an exclusive decision sub-set $\Omega^{(k)}$ can be thereby described in the following form:

$$\begin{aligned} &minimize \quad g^{cone,k}(\mathbf{x}) = s_k(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{\triangle}(\Omega)),\\ &subject\ to \quad \mathbf{x} \in \Omega^{(k)}. \end{aligned} \tag{3}$$

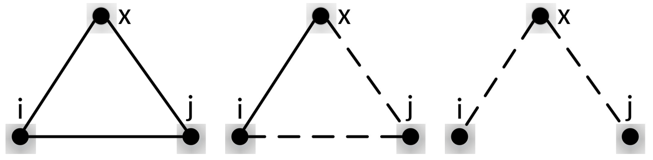
To be specific, it is inferred by further derivation that if $\overline{\mathbf{y}}' \in C^{(k)}$, $1 \leq k \leq N-2$, the conical area for $\overline{\mathbf{y}}'$ can be efficiently calculated as follows:

$$s_k(\overline{\mathbf{y}}') = 0.5\frac{1}{a}\overline{y}_1'^{\,2} + 0.5b\overline{y}_2'^{\,2} - \overline{y}_1'\overline{y}_2', \tag{4}$$

where $a = \frac{k-0.5}{N-k-0.5}$ and $b = \frac{k+0.5}{N-k-1.5}$. Otherwise, if $\overline{\mathbf{y}}' \in C^{(0)}$, $s_0(\overline{\mathbf{y}}') = 0.5b\overline{y}_2'^{\,2} + (\overline{y}_2^r - \overline{y}_2')\overline{y}_1'$ can be reached. Similarly, it can be obtained that $s_{N-1}(\overline{\mathbf{y}}') = 0.5\frac{1}{a}\overline{y}_1'^{\,2} + (\overline{y}_1^r - \overline{y}_1')\overline{y}_2'$ in the case $\overline{\mathbf{y}}' \in C^{(N-1)}$.

III. SIMILARITY AND WEIGHT HYBRID-BASED MULTI-OBJECTIVE MODEL

This paper builds a multi-objective model based on hybrid of similarity and weight for CD from SNs. Instead of using weight of two adjacent nodes directly, Huang et al. proposed structural similarity to denote the local connectivity density of any two adjacent nodes in a unsigned weighted undirected network. Based on the social balance theory, Liu et al. designed a signed similarity to make it applicable in signed networks. Specifically, They classified the relationships between a pair of nodes i, j and their mutual neighbor x into three cases. Fig. 1 gives the example for these three nodes' relationships. Thus, the signed similarity is defined as



(a) (b) (c)

Fig. 1. Example for three nodes' relationships. Solid lines are positive and dashed lines are negative. (a)Both i, x and j, x are friends. (b) i, x are friends and j, x are foes. (c)Both i, x and j, x are foes.

$$S_{signed}(i,j) = \frac{\sum\limits_{x \in \Gamma(i) \cap \Gamma(j)} \Psi(x)}{\sqrt{\sum\limits_{x \in \Gamma(i)} w^2(i,x)} \cdot \sqrt{\sum\limits_{x \in \Gamma(j)} w^2(j,x)}}, \tag{5}$$

where

$$\Psi(x) = \begin{cases} 0 & w(i,x) < 0 \, and \, w(j,x) < 0 \\ w(i,x) \cdot w(j,x) & \text{otherwise} \end{cases}$$

It is worth noting that, the total signed similarity of i, j is depended on the number of their common neighbors and the property of their connections. Therefore, the performance of MEAs_SN turns down when detecting SNs that the community structure is sparse or the most of connections are negative. Given this situation, we use the weight of this pair nodes instead of no relationship between i and j in case (c). So a hybrid of signed similarity and the intrinsic weight between i and j to denote strength of these two adjacent nodes, is calculated as

$$SW(i,j) = \frac{\sum\limits_{x \in \Gamma(i) \cap \Gamma(j)} \Psi(x)}{\sqrt{\sum\limits_{x \in \Gamma(i)} w^2(i,x)} \cdot \sqrt{\sum\limits_{x \in \Gamma(j)} w^2(j,x)}}, \tag{6}$$

where

$$\Psi(x) = \begin{cases} |w(i,j)| \cdot w(i,j & w(i,x) < 0 \, and \, w(j,x) < 0 \\ w(i,x) \cdot w(j,x) & otherwise \end{cases}$$

In general, the links should be densely positive and sparsely negative in a community while densely negative and sparsely positive between communities. In view of this consideration, a strength-based multi-objective model is designed in this paper. The model consists of the following two objective functions:

$$\begin{aligned} &minimize \quad F(C) = \{f^+(C), f^-(C)\}\\ &subject\ to \quad C = (c_1, c_2, ..., c_m) \in \Omega, \end{aligned} \tag{7}$$

where

$$f^+(C) = \frac{1}{m}\sum_{a=0}^{m} \frac{PS_{out}^{C_a}}{PS_{in}^{C_a} + PS_{out}^{C_a}} \tag{8}$$

$$f^-(C) = \frac{1}{m}\sum_{a=0}^{m} \frac{NS_{in}^{C_a}}{NS_{in}^{C_a} + NS_{out}^{C_a}} \tag{9}$$

$$PS_{in}^{C_a} = \sum_{v_i, v_j \in C_a} max(SW(i,j), 0), \tag{10}$$

$$NS_{in}^{C_a} = \sum_{v_i, v_j \in C_a} min(SW(i,j), 0), \tag{11}$$

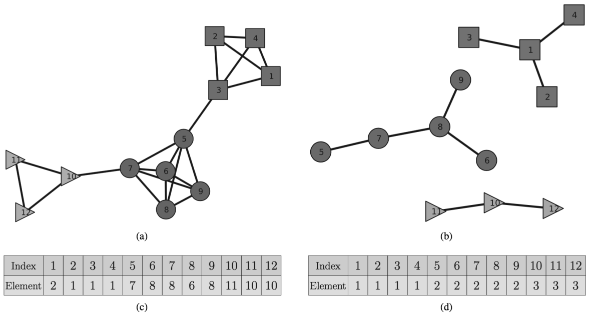
$$PS_{out}^{C_a} = \sum_{v_i \in C_a \wedge v_j \in C_b \wedge a \neq b} max(SW(i,j), 0), \tag{12}$$

$$NS_{out}^{C_a} = \sum_{v_i \in C_a \wedge v_j \in C_b \wedge a \neq b} min(SW(i,j), 0), \tag{13}$$

$(v_i, v_j) \in E$ is the set of edges, C is a possible partition in the partition space Ω of network, and C_a is one of communities in C. $PS_{in}^{C_a}$ and $NS_{in}^{C_a}$ denote the sums of positive and negative strength, respectively, in community C_a, while $PS_{out}^{C_a}$ and $NS_{out}^{C_a}$ represent the sums of positive and negative strength, respectively, between C_a and the other communities. $f^+(C)$ implies maximizing the tendency of forming the community structure C, and $f^-(C)$ denotes minimizing the tendency of destroying the community structure C. These two objectives are often contradict to each other and both are located in an interval [0,1]. Specifically, the strength-based multi-objective model has three major advantages for CD from SNs. Foremost, since two objectives are derived from the hybrid of signed similarity and weight, it guarantees the robustness of the model. Afterwards, the multi-objective model can be solved by multi-objective evolutionary algorithms which possess better mechanisms of diversity maintenance than single objective ones. In addition, the optimization of the first objective function $f^+(C)$ tends to divide a network into lager communities, while the optimization of another objective function $f^-(C)$ tends to divide a network into small communities. Thus we can obtain a set of tradeoff solutions, which represent the network partitions at different resolution scales.

IV. Proposed Algorithm: CAEAh-SN

A. *Strength-based Initializer of Chromosome*



(a)

(b)

Index	1	2	3	4	5	6	7	8	9	10	11	12
Element	2	1	1	1	7	8	8	6	8	11	10	10

(c)

Index	1	2	3	4	5	6	7	8	9	10	11	12
Element	1	1	1	1	2	2	2	2	2	3	3	3

(d)

Fig. 2. Example for locus-based adjacency representation. Solid lines are positive and dashed lines are negative. (a) A network with 3 communitie. (b) The locus-based representation for a solution. (c)Three connected components representing the three communities. (d) The corresponding genotype of this network

The representation and initialization of chromosome have a major impact on the effectiveness and efficiency of EAs for CD. The proposed algorithm, CAEAh-SN, adopts a vector with n elements to directly represent the community structure of an individual **x**:

$$\mathbf{x} = (x_1, x_2, ..., x_n), \tag{14}$$

where x_i denotes that node v_i belongs to C_{x_i}, $1 \leq i \leq n$. Therefore **x** is also the actual community structure. For the sake of producing a better initial population, it is necessary to design a initializer based on the network information. The population initialization is described as follows.

Let $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_n\}$ be a permutation of all nodes in V (i.e., a permutation of $\{1, 2, ..., n\}$). For an individual **x** of the initial population, **x** is conducted by the initializer from $\boldsymbol{\pi}$. Then the initializer repeats this process until N individuals are produced.

Huang et al. used a similarity-based quality function for any local community C_a, called the tightness [33], to design the initializer. Formally, the original tightness for community C_a is defined as follow:

$$T(C_a) = \frac{S_{in}^{C_a}}{S_{in}^{C_a} + S_{out}^{C_a}}, \tag{15}$$

where $S_{in}^{C_a}$ and $S_{out}^{C_a}$ mean, respectively, the internal and external similarities of community C_a in an unsigned network. Liu et al. further introduced the similarity-based signed tightness [28] in MEAs-SN to extend T to the signed situation. The similarity-based signed tightness for community C_a is labeled as

$$T_{signed}(C_a) = \frac{P_{in}^{C_a} - N_{in}^{C_a}}{P_{in}^{C_a} - N_{in}^{C_a} + P_{out}^{C_a}}, \tag{16}$$

where $P_{in}^{C_a}$ and $P_{out}^{C_a}$ denote the positive internal and external similarities of community C_a, respectively, and $N_{in}^{C_a}$ defines the negative internal similarity of C_a. Based on this form, we propose a strength-based signed tightness, written as T_s, Formally, the strength-based signed tightness for community C_a is calculated as follow:

$$T_s(C_a) = \frac{PS_{in}^{C_a} - NS_{in}^{C_a}}{PS_{in}^{C_a} - NS_{in}^{C_a} + PS_{out}^{C_a}}, \tag{17}$$

where $PS_{in}^{C_a}$, $NS_{in}^{C_a}$ and $PS_{out}^{C_a}$ have been introducted in Eqs. (10), (11) and (12), respectively.

It is worth noting that the internal positive ratio $PS_{in}^{C_a}$ and the external negative ratio $NS_{in}^{C_a}$ contribute to the total tightness proportionally to the positive and negative strengths. As a consequence, even negative weights in signed SNs can be also naturally handled for the strength-based tightness. In particular, a larger tightness $T_s(C_a)$ is obtained if $PS_{in}^{C_a}$ is far greater than $PS_{out}^{C_a}$ and $NS_{out}^{C_a}$ is far greater than $NS_{in}^{C_a}$, which means that most positive links lie inside community C_a and most negative links lie outside C_a. Obviously, $T_s(C_a)$ is in the range of $[0, 1]$, and a higher value of $T_s(C_a)$ implies a higher quality of community C_a.

Let $C = \{c_1, c_2, ..., c_m\}$ be a set of communities. Afterwards, for each node v_{π_i} in $\boldsymbol{\pi}$ in order, the initializer checks whether node v_{π_i} can increase the tightness T_s of current community C_a, that is, whether the following equation can be satisfied:

$$T_s(C_a \cup \{v_{\pi_i}\}) > T_s(C_a). \tag{18}$$

As long as a community is found to satisfy Eq. 18, the process terminates and this node is added to this community. Otherwise this node itself forms a new community and this new community is added to C. The details of the initializer are presented in Algorithm 1.

Algorithm 1: Strength-based Initializer

```
Output: $C = \{c_1, c_2, ..., c_m\}$;
$C \leftarrow \phi$; $jcomm \leftarrow null$;
for $i \leftarrow 1$ to $n$ do
    for $k \leftarrow 1$ to $|C|$ do
        if $T_s(C_k \cup \{v_{\pi_i}\}) > T_s(C_k)$ then
            $C_k \leftarrow C_k \cup \{v_{\pi_i}\}$; update $T_s(C_k)$;
            $jcomm \leftarrow k$; break;
        end
    end
end
if $jcomm = null$ then
    $C \leftarrow C \cup \{v_{\pi_i}\}$;
end
```

B. Evolutionary Operators

We adopt the one-way crossover operator introduced in [34]. The one-way crossover randomly selects two parental individuals from the population, with one individual $\mathbf{x}^{p1}$ set as the initial child individual $\mathbf{x}^c$. Then a node is randomly selected as the crossover seed i. Let $find\_community(\mathbf{x}^{p2}[i])$ return the nodes set (denoted as C_{find}) of community $\mathbf{x}^{p2}[i]$. Thereafter, all the nodes in C_{find} of $\mathbf{x}^{p2}$ are also assigned to the same community in the initial child individual $\mathbf{x}^c$ chromosome.

Algorithm 2 describes the procedure of the crossover operator. It can be easily inferred from Algorithm 2 that the crossover operator has the time complexity of $O(n)$, which increase linearly with the size of network.

Algorithm 2: Crossover operator

```
Input: $n$: the number of network nodes; $\mathbf{x}^{p1}, \mathbf{x}^{p2}$: a pair
        of parental individuals.
Output: $\mathbf{x}^c$:the child individual.
for $j \leftarrow 1$ to $n$ do
    $\mathbf{x}^c[j] \leftarrow \mathbf{x}^{p1}[j]$;
end
$i \leftarrow rand()\%n$;
$C_{find} \leftarrow find\_community(\mathbf{x}^{p2}[i])$;
for $v_k$ in $C_{find}$ do
    $\mathbf{x}^c[k] \leftarrow \mathbf{x}^{p2}[i]$;
end
return $\mathbf{x}^c$
```

Based on signed tightness, mutation operator traverses all the neighbors of vertex v_i, where index i is a random integer in the range $[1, n]$. If the neighbor increases the tightness of community C that v_i belongs to, then this neighbor is added to C and the traversal terminates. The process of mutation is shown in Algorithm 2, where $find\_neighbor(v_i)$ returns all neighbors of node v_i.

The time complexity of the mutation operator is $O(L)$, which mainly depends on the number of vertex v_i's neighbors.

Algorithm 3: Mutation operator

```
Input: $n$: the number of network nodes; $\mathbf{x}^c$: the child
        individual.
Output: $\mathbf{x}^c$: the child individual.
$\mathbf{x} \leftarrow null$;
for $i \leftarrow 1$ to $n$ do
    $C_{find} \leftarrow find\_community(\mathbf{x}^c[i])$;
    if $unirand() > 0.1$ or $unirand() < T_s(C_{find})$ then
        continue;
    end
    if $\mathbf{x} = null$ then
        $\mathbf{x} \leftarrow \mathbf{x}^c$;
    end
    $j \leftarrow roulette\_neighbor(i)$;
    $\mathbf{x}[i] \leftarrow \mathbf{x}^c[j]$;
end
$\mathbf{x}^c \leftarrow \mathbf{x}$;
return $\mathbf{x}^c$;
```

C. Procedure of CAEAh-SN

It is verified that CAEA shows an excellent performance and has lower computational cost for MOPs, especially for bi-objective problems. Thus, CAEAh-SN is implemented under the framework of CAEA. It is worth noting that important concepts of CAEA have been described in Section II, individual initializer and evolutionary operators are given in Section III, and this subsection describes the pseudo-code of CAEAh-SN.

In Algorithm 4, $\lceil \cdot \rceil$ is ceil, $d(\cdot, \cdot)$ is the Euclidean distance between two vectors, iL is set to increase the selected probability of the individual in edge subregion, and k_1 and k_2 denote respectively the index of subregion where solution c' and c^{k1} belong to.

$$k_1 = \lfloor \frac{(N-1)(f_1(\mathbf{x}^c) - f_1^{\triangle})}{f_1(\mathbf{x}^c) - f_1^{\triangle} + f_2(\mathbf{x}^c) - f_2^{\triangle}} + \frac{1}{2} \rfloor \qquad (19)$$

$$k_2 = \lfloor \frac{(N-1)(f_1(\mathbf{x}^{k_1}) - f_1^{\triangle})}{f_1(\mathbf{x}^{k_1}) - f_1^{\triangle} + f_2(\mathbf{x}^{k_1}) - f_2^{\triangle}} + \frac{1}{2} \rfloor \qquad (20)$$

Obviously, the first part is the initialization process from step 1 to 18, and the second part is evolutionary process from step 19 to 34. In the evolutionary part, CAEAh-SN uses the new tournament selection mechanism (i.e., QmodulTour). QmodulTour not only accelerates the convergence rate of the modularity Q value but also simplifies the calculation since it uses the objective function of individual for comparison directly.

V. Experiments and Analysis

In this section, the performance of CAEAh-SN is evaluated on both small benchmark real SNs and large synthetic SNs, and compared with MEAs-SN and the Louvain method. In our experiments, the population sizes of CAEAh-SN and MEAs-SN are set as $N = 100$. For CAEAh-SN and MEAs-SN, the termination criterion is satisfied when the number of evolutionary generations reaches 100 on every SNs. The

Algorithm 4: Procedure of CAEAh-SN

```
   Input: $\mathbf{G} = \{V, E, w\}$; $N$: the size of population; $Gen$: the number of generations;
   Output: The final population.
1  Generate $N$ initial solutions $\mathbf{C} = \{C^0, C^1, ..., C^{N-1}\}$ by Algorithm 1
2  for $i \leftarrow 1$ to $2$ do
3      $f_i^{\triangle} \leftarrow \underset{0\leq j\leq N-1}{min} f_i(C^j)$; $f_i^{\triangledown} \leftarrow \underset{0\leq j\leq N-1}{max} f_i(C^j)$;
4  end
5  $\mathbf{f}^{\triangle} \leftarrow (f_1^{\triangle}, f_2^{\triangle}), \mathbf{f}^{\triangledown} \leftarrow (f_1^{\triangledown}, f_2^{\triangledown})$
6  Initialize the population $P = \{\mathbf{x}^0, \mathbf{x}^1, ..., \mathbf{x}^{N-1}\}$ where $\mathbf{x}^0, \mathbf{x}^1, ..., \mathbf{x}^{N-1} = null$. Set $\mathbf{C}' = \mathbf{C}$.
7  for $k \leftarrow 0$ to $N-1$ do
8      if $\boldsymbol{C}^{(k)} \neq \phi$ then
9          $\mathbf{x}^k \leftarrow \arg \underset{C\in\mathbf{C}}{min}\, s(\mathbf{f}(C) - \mathbf{f}^{\triangle})$;
10         $\mathbf{C}' \leftarrow \mathbf{C}' - \mathbf{x}^k$;
11     end
12 end
13 for $k \leftarrow 0$ to $N-1$ do
14     if $\mathbf{x}^k = null$ then
15         $\mathbf{x}^k \leftarrow \arg \underset{C\in\mathbf{C}'}{min}\, d(\overline{V}(\mathbf{f}(C) - \mathbf{f}^{\triangle}), \overline{V}^{(j)})$;
16         $\mathbf{C}' \leftarrow \mathbf{C}' - \mathbf{x}^k$;
17     end
18 end
19 $p_1 \leftarrow currentGen\%N$, $p_2 \leftarrow rand()\%N$;
20 Evolution: obtain $\mathbf{x}^c$ after the crossover and mutation of $\mathbf{x}^{p_1}, \mathbf{x}^{p_2}$; Set $nE \leftarrow nE + 1$;
21 Update $\mathbf{f}^{\triangle}, \mathbf{f}^{\triangledown}$
22 for $i \leftarrow 1$ to $2$ do
23     if $f_i(\mathbf{x}^c) < f_i^{\triangle}$ then
24         $f_i^{\triangle} \leftarrow f_i(\mathbf{x}^c)$;
25     end
26     if $f_i(\mathbf{x}^c) > f_i^{\triangledown}$ then
27         $f_i^{\triangledown} \leftarrow f_i(\mathbf{x}^c)$;
28     end
29 end
30 Update the solution $\mathbf{x}^k$: $k_1$ is the index of subregion where $\mathbf{x}^c$ lies and $k_2$ is the index of subregion where $\mathbf{x}^{k_1}$ lies
31 if $k_1 \neq k_2 \wedge s(\mathbf{f}(\mathbf{x}^c) - \mathbf{f}^{\triangle}) < s(\mathbf{f}(\mathbf{x}^{k_1}) - \mathbf{f}^{\triangle})$ then
32     $\mathbf{x}^{k_1} \leftarrow \mathbf{x}^c$;
33 end
34 If evolution to $Gen$ generation, then stop and output the best solution, otherwise go to step 19
```

other parameters of MEAs-SN and the Louvain method use the corresponding recommended values provided, respectively, by [28] and [17]. CAEAh-SN is implemented in C++, MEAs-SN is implemented in C, and a special version for SNs of Louvain method is implemented in Pajek [35] All the three algorithms run on an Intel Core i5-3470 3.20 GHz PC with 4GB RAM and Windows 7 OS. To assess the overall performances of algorithms, a total of 20 statistically independent runs of each algorithm have been executed for each SNs.

A. Performance Indicators

A commonly used normalized mutual information (NMI) measure [36] and the modularity, Q [37], are chosen as the performance indicators in our experiments for CD from SNs.

Based on the information theory, the NMI indicator measures the difference between the community structure obtained by an algorithm and the real structure through a confusion matrix M. In matrix M, each row represents one real community, each column denotes an obtained community, and the element M_{ij} refers to the number of nodes which lie in the real community C_i^* in the real structure C^* and also belong to the community C_j in an obtained structure C. Given the real structure C^* and an obtained structure C for a SNs, the NMI between them is defined as follows:

$$NMI(C^*, C) = \frac{-2\sum_{i=1}^{m^*}\sum_{j=1}^{m} M_{ij}\log(\frac{nM_{ij}}{M_{i\cdot}M_{\cdot j}})}{\sum_{i=1}^{m^*} M_{i\cdot}\log(\frac{M_{i\cdot}}{n}) + \sum_{j=1}^{m} M_{\cdot j}\log(\frac{M_{\cdot j}}{n})}, \tag{21}$$

where n is the number of nodes, m^* represents the number of the real communities in C^*, and m indicates the number of the obtained community in C, $N_{i\cdot}$ and $N_{\cdot j}$ denote the sums of elements in the i-th row and the j-th column, respectively, in matrix M. If the obtained structures is identical to the real network structure, the value of NMI takes its maximum 1. In contrast, when they are completely different, NMI is equal to 0.

Góme z et al. presented a reformulation of modularity for the analysis of community structures in signed situations. Let c_i denote that node v_i is assigned to the community C_{c_i}, $w = (w_{ij})$ be the weighted adjacency matrix of a SNs where every element w_{ij} represents the value of the weight in the link between node v_i and v_j (0 if no link exists), and $w_{ij}^+ = max\{0, w_{ij}\}$ and $w_{ij}^- = max\{0, -w_{ij}\}$, respectively, indicate the positive and negative portions of w_{ij} such that $w_{ij} = w_{ij}^+ + w_{ij}^-$. As a result, the sums of the positive and negative weights of node v_i are, respectively, $w_i^+ = \sum_j w_{ij}^+$ and $w_i^- = \sum_j w_{ij}^-$. Furthermore, the total positive and negative weights of the SNs equal to $w^+ = \sum_i w_i^+$ and $w^- = \sum_i w_i^-$, respectively. Afterwards, the total signed modularity Q [37] for the SNs can be expressed in the following mathematical form:

$$Q = \frac{1}{w^+ + w^-}\sum_i\sum_j\Big[w_{ij} - (\frac{w_i^+ w_j^+}{w^+} - \frac{w_i^- w_j^-}{w^-})\Big]\delta(c_i, c_j), \tag{22}$$

where $\delta(c_i, c_j)$ takes the value 1 if nodes v_i and v_j are in the same community, i.e., $c_i = c_j$, otherwise it takes value 0. Therefore, it does not require the reference real structure.

A larger value of the modularity usually indicates a better structure. In practice, the values of modularity Q for the real structures of most networks are typically in the range from 0.3 to 0.7.

B. Experiments on Benchmark Signed Networks

Four benchmark signed networks, including two illustrative SNs [38] with 28 nodes and two real SNs, are first detected in our experiments. Two real SNs are the Slovene Parliamentary Party Network [39] and the Gahuku-Gama Subtribes Network [40]. Specifically, the former describes the relationship of the 10 parties of the Slovenian Parliament in 1994, and the latter represents the political alliance and class struggle of the 16 subtribes of Gahuku-Gama.

TABLE I
THE MEDIAN SCORES OF BOTH Q AND NMI ACHIEVED BY CAEAh-SN FOR FOUR BENCHMARK SNS

Network	Q	NMI
$Illustrated\ signed\ network\ I$	0.564286	1
$Illustrated\ signed\ network\ II$	0.521267	1
$Slovene\ Parliamentary\ Party$	0.454733	1
$Gahuku - Gama\ Subtribes$	0.431034	1

Fig. 3. Community structures with the median NMI scores obtained by CAEAh-SN for two illustrative SNs. (a) Illustrated signed network I. (b) Illustrated signed network II.

Fig. 4. Community structures with the median NMI scores obtained by CAEAh-SN for two real SNs. (a) Slovene Parliamentary Party Network. (b) Gahuku-Gama Subtribes Network.

Table I records the median scores of both Q and NMI achieved by CAEAh-SN for four benchmark SNs. Afterwards, Fig. 3 and 4 plot the community structures with the median NMI scores obtained by CAEAh-SN, respectively, for two illustrative SNs and two real SNs where solid lines denote positive links and dashed lines indicate negative links. It can be observed from Fig. 3 and 4 that the Illustrated signed network I and II and the Gahuku-Gama Subtribes Network are divided into three communities and the Slovene Parliamentary Party Network is into two communities, which are perfectly consistent with the numbers of communities in the real partitions of four networks. Furthermore, Table I indicates that the achieved NMI scores reach 1 for all these four SNs and the obtained Q scores are in an expected normal range from 0.43 to 0.56. It implies that the partitions detected by CAEAh-SN are absolutely the same as the real community structures of four networks.

C. Experiments on Synthetic Signed Networks

1) SNs Generator: In order to systematically test the performance of CAEAh-SN on larger SNs, a popular SRN generator proposed in [38] is used to generate larger synthetic SNs with the given community structures in our experiments. This SNs generator combines the Lancichinetti-Fortunato-Radicchi (LFR) benchmark with the generator proposed in [38] and is formulated as $SRN(n, K, maxK, \tau_1, \tau_2, minc, maxc, \mu, P^+, P^-)$. Among the above input parameters, n is the total number of nodes in the SNs, K and $maxK$ denotes the mean and maximum of degrees of nodes in the network, τ_1 and τ_2 are the respective exponential distribution coefficients of degree and community size, and $minc$ and $maxc$ indicate the smallest and largest sizes of communities. In addition, μ is the fraction of edges across communities for each node, whose increase will reduce the cohesiveness of communities. Furthermore, P^+ and P^- denotes the proportions of positive connections across communities and negative connections within communities, respectively. In general, the higher value of each of μ, P^+ and P^- will result in the greater ambiguity of the community structure, thereby a larger difficulty in detecting the correct community structure.

2) Experiments on Synthetic SNs with Low Noise: In our experiments, three synthetic networks with different sizes, referred to as SRN1000, SRN5000, and SN10000, are first generated by the following parameters:

$$SRN(1000, 20, 50, 2, 1, 20, 50, 0.1, 0, 0), \tag{23}$$

$$SRN(5000, 20, 50, 2, 1, 40, 100, 0.1, 0, 0), \tag{24}$$

$$SRN(10000, 20, 50, 2, 1, 40, 100, 0.1, 0, 0). \tag{25}$$

Figure 5 presents the topological structures plotted by the Pajek for four synthetic SNs where the number of edges are 1046, 5272, 34190, and 68736, respectively, for these four SNs. Note that all four synthetic networks are in low noise levels since they are generated through the very small values of the last three parameters, i.e., $\mu = 0.1$, $P^+ = 0$, and $P^- = 0$.

Fig. 5. Topological structures of four synthetic SNs.(a) SN100. (b) SN500. (c) SN1000. (d) SN2000.

The community structures with the median NMI scores detected by CAEAh-SN for four synthetic SNs are plotted by the Pajek in Fig. 6 where each color represents a community.

It can be seen that community structures are much clear after the detection. Four SNs are divided into 3, 17, 24 and 52 communities respectively and most of positive links are distributed in communities, and negative links are distributed between communities.

Fig. 6. Partitions of four synthetic SNs obtained by CAEAh-SN. Solid lines are positive and dashed lines are negative.(a)SN100. (b)SN500. (c)SN1000. (d)SN2000.

At first, we discuss the results of the CAEAh-SN and MEAs-SN on these four synthetic SNs. In practice, value of Q for such networks typically is in the range from 0.3 to 0.7 and higher values are rare [41]. Fig. 7 gives the comparison between CAEAh-SN and MEAs-SN on Q values and NMI.It is evident from Fig. 7 that CAEAh-SN obtains higher Q values than MEAs-SN for all synthetic SNs, which suggests that CAEAh-SN achieves batter partitions than MEAs-SN. The Q obtained by CAEAh-SN of SN1000 and SN2000 are greater than 0.8 and the Q of network SN100 and SN500 are more than 0.5. It is clear that the performance of CAEAh-SN is batter than MEAs-SN. In the comparison of NMI values, CAEAh-SN is also superior to MEAs-SN.

Fig. 7. Comparison between CAEAh-SN and MEAs-SN on four synthetic SNs. (a)Q. (b)NMI.

Table II shows the average CPU time in seconds spend by two algorithms for each test problem. As it turns out, the time taken by two algorithms increases quickly when the network size becomes larger. Table II clearly indicates that the computational time of CAEAh-SN is always less than that of MEAs-SN when n increases from 1000 to 10000.

TABLE II
COMPUTATIONAL TIME OF CAEAh-SN AND MEAs-SN

Algorithm	SN1000	SN5000	SN10000
$CAEAh-SN$	0.5832	18.6738	85.6885
$MEAs-SN$	4.7379	72.0092	193.4043

3) Experiments on Synthetic SNs with High Noise: In our experiments, a group of synthetic networks are first generated by the following parameters:

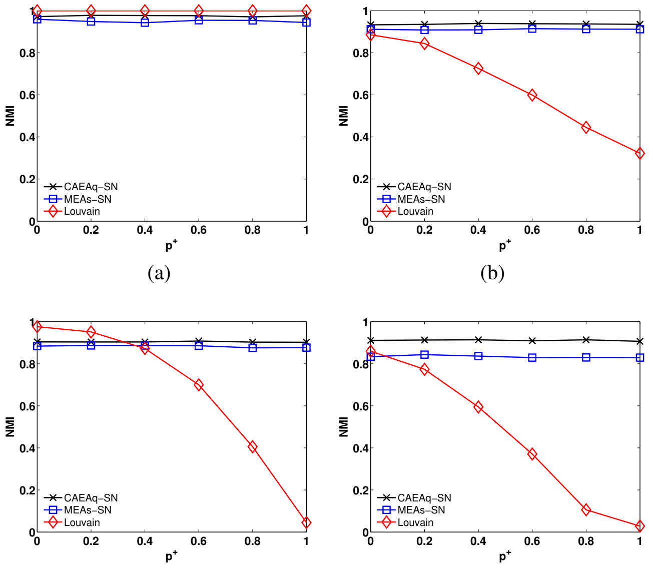
$$SN(1000, 20, 50, 2, 1, 20, 50, \mu, P^+, P^-) \qquad (26)$$

Note that, 90 combinations of three parameters, μ, P^+, and P^-, are applied to systematically validate the robustness of CAEAh-SN to noises. To be specific, μ, P^+, and P^- take values, respectively, from $\{0.1, 0.3, 0.5\}$, $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, and $\{0.0, 0.2, 0.4, 0.6, 0.8\}$.

In order to evaluate the performance of CAEAh-SN systematically, we change the value of μ, P^+, and P^- to destroy SN100 and SN500. P^+ is in the range of $[0.0, 1.0]$, P^- is [0.0 0.8], μ is in the range of $[0.1, 0.5]$ and they are all in the step of 0.2. For each combination of these three parameters, 30 independent runs of CAEAh-SN and MEAs-SN are conducted, and the averaged Q is reported in Figs. **??** and **??**.

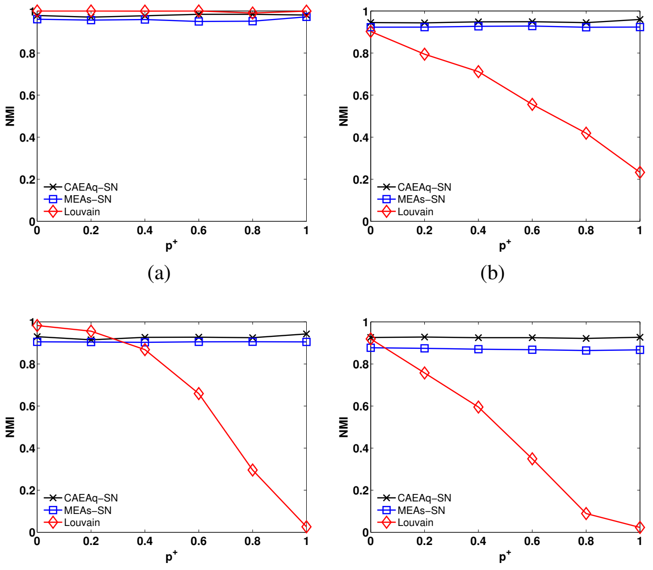
On the whole, the Q of CAEAh-SN and MEAs-SN reduce gradually with the increase of μ, which implies a network with a lower density is more difficult to detect. For different combinations of μ, P^+ and P^-, CAEAh-SN still works better on these networks than MEAs-SN. For example, for SN100 and SN500 when $P^- = 0.2$, Q of CAEAh-SN are in the vicinity of 0.2 while Q obtained by MEAs-SN are near 0.1 and some even are close to 0. It is necessary to emphasize that when $P^- > 0.2$ the Q of MEAs-SN are almost 0 while the most of Q obtained by CAEAh-SN are acceptable and some batter Q are more than 0.1 for SN100 and SN500. Moreover, in some extreme case (i.e., $p^+ = 1.0, p^- = 0.0$), CAEAh-SN obtains batter Q while MEAs-SN fails to find results whose $Q \neq 0$. In general, CAEAh-SN obtains obviously greater Q than MEAs for all the dynamic synthetic SNs.

The results also indicate that two algorithms are sensitive to P^- and insensitive to P^+. With the increase of P^-, the noise level increases, which destroys the network community structure and increases the ambiguous of the community structure significantly. For instance, when $P^- \geq 0.4$, all the values of Q are smaller than 0.3. Relatively speaking, comparison of performance is more meaningful when P^- takes $\{0, 0.2\}$. Similarly, most of the Q are smaller than 0.3 when $\mu \geq 0.5$. Therefore, the separate comparisons of performance is discussed when μ takes $\{0.1, 0.3\}$ and P^- takes $\{0, 0.2\}$.



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

Fig. 8. Further comparison on SN5000. (a)$\mu = 0.1, P^- = 0.0$. (b)$\mu = 0.1, P^- = 0.2$. (c)$\mu = 0.3, P^- = 0.2$. (d)$\mu = 0.3, P^- = 0.4$.



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

Fig. 9. Further comparison on SN10000. (a)$\mu = 0.1, P^- = 0.0$. (b)$\mu = 0.1, P^- = 0.2$. (c)$\mu = 0.3, P^- = 0.2$. (d)$\mu = 0.3, P^- = 0.4$.

Figs. 8 and 9 show further comparison on 5000 and 10000 nodes of synthetic SNs between CAEAh-SN, MEAs-SN and Louvain method. Calculations indicate that NMI of the first two algorithms in SRN5000 is more stable than it in SN100 when P^- and μ are fixed. CAEAh-SN obtains obviously

TABLE III
THE COMPARISON RESULTS OF NMI OF THREE ALGORITHMS ON SN1000 NETWORKS

$Parameters$			P^{+}					
μ	P^{-}	$Algorithm$	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
0.1	**0.0**	CAEAh-SN	0.991114	0.994233	0.987117	0.989815	0.991579	0.993818
		MEAs-SN	0.971001	0.944494	0.975134	0.955544	0.969295	0.968892
		Louvain	**1**	**1**	**1**	**1**	**1**	**1**
0.1	**0.2**	CAEAh-SN	**0.963302**	**0.959600**	**0.959524**	**0.954450**	**0.970490**	**0.959076**
		MEAs-SN	0.950464	0.942959	0.949074	0.939577	0.941886	0.946633
		Louvain	0.882081	0.834460	0.771962	0.643372	0.533471	0.422818
0.1	**0.4**	CAEAh-SN	**0.982295**	**0.979831**	**0.974207**	**0.972996**	**0.979924**	**0.984062**
		MEAs-SN	0.758618	0.783190	0.765030	0.763224	0.743427	0.733261
		Louvain	0.837580	0.716590	0.619024	0.532111	0.411566	0.260161
0.1	**0.6**	CAEAh-SN	**0.985829**	**0.984057**	**0.990403**	**0.987543**	**0.989358**	**0.986422**
		MEAs-SN	0.814883	0.823792	0.754369	0.718447	0.670672	0.609954
		Louvain	0.806120	0.567816	0.446409	0.325257	0.236715	0.137532
0.1	**0.8**	CAEAh-SN	0.840622	0.864914	**0.853220**	**0.862451**	**0.853750**	**0.845105**
		MEAs-SN	**0.873774**	**0.865268**	0.826151	0.781367	0.737088	0.719427
		Louvain	0.739021	0.533837	0.339940	0.205034	0.158304	0.113824
0.3	**0.0**	CAEAh-SN	0.971497	0.956623	0.964652	0.961201	0.963699	0.970149
		MEAs-SN	0.979256	0.904717	0.933336	0.962014	0.915149	0.918842
		Louvain	**1**	**1**	**1**	**0.994774**	**0.966240**	**0.982148**
0.3	**0.2**	CAEAh-SN	0.890920	0.875444	**0.916736**	0.912256	**0.928410**	**0.912521**
		MEAs-SN	0.894593	0.889660	0.908407	**0.917575**	0.894546	0.906661
		Louvain	**0.971276**	**0.926507**	0.863408	0.699961	0.426157	0.123345
0.3	**0.4**	CAEAh-SN	**0.933084**	**0.924302**	**0.921779**	**0.922807**	**0.912561**	**0.923177**
		MEAs-SN	0.633303	0.657864	0.659809	0.679410	0.728945	0.775453
		Louvain	0.831889	0.770093	0.603767	0.398792	0.197796	0.076608
0.3	**0.6**	CAEAh-SN	**0.938855**	**0.912575**	**0.923704**	**0.939762**	**0.926758**	**0.943550**
		MEAs-SN	0.652345	0.629666	0.584359	0.577576	0.599031	0.659067
		Louvain	0.780352	0.553986	0.311570	0.174698	0.108710	0.052787
0.3	**0.8**	CAEAh-SN	0.704138	**0.687552**	**0.714041**	**0.752901**	**0.751024**	**0.717609**
		MEAs-SN	**0.774194**	0.681446	0.580598	0.521343	0.598653	0.636236
		Louvain	0.740240	0.380090	0.171606	0.125529	0.087198	0.074037
0.5	**0.0**	CAEAh-SN	0.914627	0.915391	0.919352	0.887540	0.889974	0.906223
		MEAs-SN	0.864806	0.900087	0.915137	0.895802	0.889345	0.844126
		Louvain	**1**	**1**	**1**	**1**	**0.985132**	**0.989914**
0.5	**0.2**	CAEAh-SN	0.745901	0.729494	0.741702	0.738369	0.759910	0.759521
		MEAs-SN	0.791783	0.795920	0.783898	**0.788930**	**0.772516**	**0.766636**
		Louvain	**0.993928**	**0.982256**	**0.932168**	0.705027	0.263902	0.009384
0.5	**0.4**	CAEAh-SN	0.794617	0.796439	**0.786368**	**0.794242**	**0.779132**	**0.806895**
		MEAs-SN	0.690155	0.694694	0.694957	0.691602	0.725536	0.760993
		Louvain	**0.911017**	**0.802321**	0.535053	0.325710	0.103395	0.026297
0.5	**0.6**	CAEAh-SN	**0.767368**	**0.796543**	**0.808819**	**0.825660**	**0.841407**	**0.804394**
		MEAs-SN	0.662859	0.626340	0.574299	0.589592	0.640577	0.639253
		Louvain	0.767311	0.479374	0.244850	0.137148	0.095462	0.028386
0.5	**0.8**	CAEAh-SN	0.544676	**0.573281**	**0.668767**	**0.701072**	**0.697983**	**0.689796**
		MEAs-SN	0.701814	0.505738	0.487485	0.479673	0.469466	0.475954
		Louvain	**0.711435**	0.300863	0.170825	0.106195	0.059554	0.035665

TABLE IV
THE COMPARISON RESULTS OF MODULARITY OF THREE ALGORITHMS ON SN1000 NETWORKS

$Parameters$			P^+					
μ	P^-	$Algorithm$	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
0.1	**0.0**	CAEAh-SN	0.425509	0.426189	0.422164	0.424542	0.422506	0.421245
		MEAs-SN	0.412342	0.417905	0.420392	0.423038	0.428601	0.388554
		Louvain	**0.432988**	**0.432900**	**0.432460**	**0.431458**	**0.430686**	**0.428878**
0.1	**0.2**	CAEAh-SN	0.107424	0.108362	0.103430	0.101102	0.104087	0.095657
		MEAs-SN	0.109136	0.108356	0.105006	**0.107340**	**0.106764**	**0.103660**
		Louvain	**0.114077**	**0.110016**	**0.107128**	0.098348	0.095409	0.084879
0.1	**0.4**	CAEAh-SN	-0.110909	-0.111019	-0.108502	-0.105500	-0.111728	-0.113077
		MEAs-SN	-0.048594	-0.052136	-0.049868	-0.048986	-0.045293	-0.041051
		Louvain	**0.031038**	**0.036926**	**0.042866**	**0.049581**	**0.054418**	**0.060804**
0.1	**0.6**	CAEAh-SN	-0.285827	-0.282797	-0.290970	-0.289384	-0.290531	-0.289748
		MEAs-SN	-0.148397	-0.160081	-0.125442	-0.120454	-0.092117	-0.075886
		Louvain	**-0.014784**	**0.006334**	**0.019578**	**0.033691**	**0.042524**	**0.049738**
0.1	**0.8**	CAEAh-SN	-0.195599	-0.218945	-0.211289	-0.212345	-0.217337	-0.205812
		MEAs-SN	-0.238332	-0.237352	-0.225645	-0.186293	-0.150625	-0.141921
		Louvain	**-0.021470**	**-0.001106**	**0.017053**	**0.031744**	**0.038493**	**0.042010**

higher Q than MEAs-SN for all the dynamic synthetic SNs.

In Fig. **??**, the performance of CAEAh-SN is not only batter than MEAs-SN, but also more stable than MEAs-SN. Two algorithms are incomparable with each other only when $\mu = 0.1, P^- = 0.2$, and $P^+ = 0.8$.

In Fig. **??**, the performance of CAEAh-SN outperforms MEAs-SN in the mass. It is worth noting that MEAs-SN fails to obtain partitions of which Q are not 0.

As a whole, the proposed CAEAh-SN obtains higher Q than MEAs-SN in terms of the solution quality, which implies the detected communities from SNs have better partitions by CAEAh-SN. Furthermore, CAEAh-SN is computationally much more efficient than MEAs-SN and shows a strong anti-perturbation performance and faster convergence rate.

4) Comparison between CAEAh-SN and Louvain Method: Besides comparing modularity of different algorithms, we also obtained NMI of detected communities from large-scale synthetic SNs. NMI measures the difference between the partition of community obtained by algorithms and the real partition. In this experiment, a series of large-scale SNs containing 10000 vertices are used to compare the performance of CAEAh-SN and Louvain Method, which can be formulated as the follow:

$$SN(10000, 5, 15, 2, 1, 40, 100, 0.1, P^+, P^-). \tag{27}$$

Then, P^+ increases from 0.0 to 1.0 in the step of 0.2, and P^- increases from 0.0 to 0.4 in the step of 0.2. For each network, two algorithms are executed 10 times. The average NMI values over these 10 runs are reported in Fig. **??**.

Fig. **??** clearly shows both CAEAh-SN and Louvain Method are almost always higher than 0.9 and close to 1.0 when $P^- = 0.0$ and P^+ increases from 0.0 to 1.0. However, Louvain Method is so sensitive to the increase of P^+ that it's solutions drop dramatically when $P^- = 0.2$ and $P^- = 0.4$. For example, the NMI of CAEAh-SN is still higher than 0.8 but the NMI of Louvain Method drop to 0.2 when $P^- = 0.2$ and $P^+ = 1.0$. More importantly, when $P^- = 0.2$ and $P^- = 0.4$, the NMI of CAEAh-SN is better than that of Louvain method when P^+ increases from 0.0 to 1.0 in most cases. The obtained results show that CAEAh-SN has stronger robustness to the SNs with high noises

VI. CONCLUSION

In this paper, CAEAh-SN is proposed to extend the decomposition-based CAEA for CD from SNs. In CAEAh-SN, the modularity Q is partitioned into Q^+ and Q^-. The problem behaves like a bi-objective problem where Q^+ represents the development of the network towards the formation of the community, and Q^- destroys the formation of the community. Moreover, an improved version of AreaTour algorithm which has been applied in CAEA algorithm, called QmodulTour, uses the modularity Q to select individuals. QmodulTour avoids the deviation of the evolution vector and improves computational efficiencies. The experimental results indicate that CAEAh-SN offers significantly higher Q than MEAs-SN in benchmark signed networks, synthetic networks with the general parameters, and synthetic networks with the dynamic parameters. Furthermore, our future research will focus on the large scale of SNs.

REFERENCES

[1] M. E. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.

[2] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

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

[4] H. Q. Dinh, N. Aubert, N. Noman, T. Fujii, Y. Rondelez, and H. Iba, "An effective method for evolving reaction networks in synthetic biochemical systems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 374–386, 2015.

[5] J. Wu, L. Zhang, Y. Li, and Y. Jiao, "Partition signed social networks via clustering dynamics," *Physica A*, vol. 443, pp. 568–582, 2016.

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

[7] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, no. 4, p. 046110, 2008.

[8] L. Gulikers, M. Lelarge, and L. Massoulié, "A spectral method for community detection in moderately sparse degree-corrected stochastic block models," *Adv. Appl. Probab.*, vol. 49, no. 3, pp. 686–721, 2017.

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

[10] M. Newman, "Community detection in networks: Modularity optimization and maximum likelihood are equivalent," *arXiv preprint arXiv:1606.02319*, 2016.

[11] P. Zhang, C. Moore, and M. Newman, "Community detection in networks with unequal groups," *Phys. Rev. E*, vol. 93, no. 1, p. 012303, 2016.

[12] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[13] J. Su and T. C. Havens, "Quadratic program-based modularity maximization for fuzzy community detection in social networks," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1356–1371, 2015.

[14] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of mathematical sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[16] M. E. Newman, "Analysis of weighted networks," *Phys. Rev. E*, vol. 70, no. 5, p. 056131, 2004.

[17] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech: Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.

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

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

[20] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Cybern.*, 2017.

[21] A. Amelio and C. Pizzuti, "An evolutionary and local refinement approach for community detection in signed networks," *Int. J. Artif. Intell. Tools*, vol. 25, no. 04, p. 1650021, 2016.

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

[23] X. Wen, W.-N. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin, and J. Zhang, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, 2017.

[24] 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, 2014.

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

[26] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for constraint satisfaction problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 1, pp. 54–73, 2006.

[27] J. Liu, W. Zhong, L. Jiao, and X. Li, "Moving block sequence and organizational evolutionary algorithm for general floorplanning with arbitrarily shaped rectilinear blocks," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 630–646, 2008.

[28] 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, 2014.

[29] C. Pizzuti, "A multi-objective genetic algorithm for community detection in networks," in *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*. IEEE, 2009, pp. 379–386.

[30] T. He and K. C. C. Chan, "Evolutionary community detection in social networks," in *Evolutionary Computation*, 2014, pp. 1496–1503.

[31] 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, 2007.

[32] Y. Weiqin, X. Xing, F. Yuxiang, and W. Yu, "An efficient conical area evolutionary algorithm for bi-objective optimization," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 8, pp. 1420–1425, 2012.

[33] J. Huang, H. Sun, Y. Liu, Q. Song, and T. Weninger, "Towards online multiresolution community detection in large-scale networks," *PLoS One*, vol. 6, no. 8, p. e23829, 2011.

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

[35] A. Mrvar and V. Batagelj, "Pajek and pajek-xxl. programs for analysis and visualization of very large networks. reference manual," 2013.

[36] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech: Theory Exp.*, vol. 2005, no. 09, p. P09008, 2005.

[37] S. Gómez, P. Jensen, and A. Arenas, "Analysis of community structure in networks of correlated data," *Phys. Rev. E*, vol. 80, no. 1, p. 016114, 2009.

[38] B. Yang, W. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, 2007.

[39] A. Ferligoj and A. Kramberger, "An analysis of the slovene parliamentary parties network," in *Developments in statistics and methodology*, 1996, pp. 105–144.

[40] K. E. Read, "Cultures of the central highlands, new guinea," *Southwestern Journal of Anthropology*, vol. 10, no. 1, pp. 1–43, 1954.

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

Weiqin Ying received the B.Sc. degree in computer science and technology from Chongqing University, Chongqing, China, in 2001, the M.Sc. degree and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2005 and 2009, respectively.

He is currently an Associate Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include evolutionary and genetic algorithms, multi-objective optimization and decision, distributed computing, deep learning, and relevant real-world applications.

Pengfei Chao received the B.Sc. degree in automation from Henan Polytechnic University, Jiaozuo, China, in 2016, and is currently pursuing his M.Eng. degree in software engineering from the School of Software Engineering, South China University of Technology, Guangzhou, China.

His current research interests include evolutionary algorithms and their applications on real-world problems.

Yuehong Xie received the B.Sc. degree and the M.Eng. degree in software engineering from South China University of Technology, Guangzhou, China, in 2014 and 2017, respectively.

His current research interests include multi-objective optimization, many-objective optimization, evolutionary algorithms and constrained handling techniques.

Zhenyu Wang received the B.Sc. degree in computer science and technology from Xiamen University, Xiamen, China, in 1987, the M.Sc. degree in computer application and the Ph.D. degree in computer architecture from Harbin Institute of Technology, Harbin, China, in 1990 and 1993, respectively.

He is currently a Professor and the dean with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include distributed computing, cloud computing, big data technologies, service-oriented architecture, large-scale application design and development, and multi-objective optimization.

Yu Wu received the B.Sc. degree in computer science and technology from Chongqing University, Chongqing, China, in 2001, the M.Sc. degree and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2005 and 2009, respectively.

He is currently an Associate Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include evolutionary and genetic algorithms, multi-objective optimization and decision, distributed computing, deep learning, and relevant real-world applications.