International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **1**, February **2021**

ICIC International ©2021 ISSN 1349-4198
pp. 355–368

DETECTING OVERLAPPING COMMUNITIES IN NETWORKS WITH EXTREMAL OPTIMIZATION

Jin Ding[1], Shirinbaev Azizbek[1], Yongzhi Sun[1], Ping Tan[1] and Feijie Wang[2]

[1]School of Automation and Electrical Engineering
Zhejiang University of Science and Technology
No. 318, Liuhe Road, Xihu District, Hangzhou 310023, P. R. China
jding@zust.edu.cn; azik.www.94@mail.ru; sunyongzhi@hotmail.com; tanp@supcon.com

[2]Design Center
Zhejiang UniTTEC Co., Ltd.
No. 1785, Jianghan Road, Binjiang District, Hangzhou 310051, P. R. China
wfjhlx@263.net

Received September 2020; revised December 2020

Abstract. *Overlapping communities are ubiquitous in biological networks and social networks. Identifying them effectively and efficiently can help us better understand the underlying behaviors of these networks. Modularity function is a widely-used metric for evaluation of the quality of the detected communities. Due to the NP-hard property of the maximization of the modularity function, the evolutionary algorithms are often employed to tackle it. In this paper, for the first time, an EO based algorithm is proposed to optimize the modularity function and detect the overlapping communities of the networks. Firstly, a modified modularity function is defined to deal with the resolution limit problem. Secondly, the local fitness functions of nodes are defined, which can be linearly combined to form the modularity function. And thirdly, a novel mutation operator is designed which can explore the solution space effectively and efficiently. Experiments on multiple real and synthetic network datasets show the proposed EO based overlapping communities detection algorithm can converge fast and has the competitive performance compared to several other commonly-used overlapping communities detection algorithms measured on three standard metrics – Overlapping Normalized Mutual Information (ONMI), Omega (Ω) Index, and F-Score.*

Keywords: Modified modularity function, Local fitness functions of nodes, Extremal optimization, Overlapping communities detection

1. **Introduction.** Overlapping communities are often found in the biological networks and social networks, where one node can play a role in multiple functional modules, e.g., a protein can be involved in various diseases, or a person can belong to different interest groups. Detecting these overlapping communities effectively and efficiently can help us better understand the structures and functions of the underlying networks. The past few years have witnessed an explosive growth of interest for tackling this explorative task, and plenty of useful and efficient approaches [1] have been proposed to detect the overlapping communities of the networks, e.g., optimization of a quality function [2, 3, 4, 5, 6], statistical learning for a generative model [7, 8, 9, 10, 11, 12], and heuristics [13, 14, 15, 16].

A commonly-used approach to detect overlapping communities is to maximize the modularity function, which measures the difference between the real number and the expected

DOI: 10.24507/ijicic.17.01.355

number of edges in communities based on the first order null model [17]. Due to the NP-hard property of the maximization of the modularity function, the evolutionary algorithms are often employed to solve it, e.g., genetic algorithm [18] and particle swarm optimization algorithm [19]. Extremal optimization (EO for short) is a statistical physics inspired, self-organized critical, and far-from-equilibrium evolutionary algorithm [20, 21], which has been successfully applied to several classical combinatorial optimization problems and industrial optimization problems [22, 23, 24, 25, 26].

EO maintains only one chromosome during its optimization process, and needs to define the local fitness functions for each gene of the chromosome. The property of these local fitness functions is that they can be linearly combined to constitute the quality function of interest, and the optimization of the quality function is performed by the co-evolution of the genes. Noticing that the modularity function can be expressed as a linear combination of the edge number difference of each node, therefore, it is suitable to employ EO for solving the maximization of the modularity function.

In this regard, Duch and Arenas proposed an EO based solution framework to maximize the modularity function for detection of non-overlapping communities [27]. In their solution framework, EO iteratively performs the maximization of modularity function under the setting of two communities. Firstly, two communities with the maximum modularity value is found. Secondly, keep one community unchanged and split the other community into two parts through EO optimization. And this process stops until the modularity value does not increase. Lung *et al.* proposed an EO variant – noisy extremal optimization – to maximize the modularity function [28, 29]. In the work, the local fitness function of each node was defined as the difference of ratio of real edge number and total degrees of nodes in the community with and without this node.

Note that, in Duch and Arenas's work, setting the number of communities to two can limit the ability of EO in fully exploring the solution space, and probably yields a local maximal modularity value. And in Lung *et al.*'s work, the local fitness functions under this definition are not part of the modularity function, and therefore, the optimization of the local fitness functions does not necessarily lead to the optimization of the modularity function. In addition, the evaluation of the local fitness functions needs more computation compared to the local fitness functions defined as part of the modularity function. Moreover, both of the work are not developed for overlapping communities detection.

To address these issues, in this paper, we propose an EO based solution framework for the modularity maximization to detect the overlapping communities in the networks, where the number of communities can be customized and the local fitness functions of nodes can be linearly combined to form the modularity function. Firstly, a modified modularity function is defined to deal with the resolution limit problem [30]. Secondly, the local fitness functions of nodes are defined based on the modified modularity function. And thirdly, a novel mutation operator is designed, which can explore the solution space effectively and efficiently. Experiments on multiple real and synthetic network datasets demonstrate EO can converge fast, and can explore the solution space more efficiently than the genetic algorithm (GA for short). Moreover, EO based solution framework shows a competitive performance on three standard metrics – Overlapping Normalized Mutual Information (ONMI) [31], Omega (Ω) Index [32], and F-Score [32], compared to several other commonly-used overlapping communities detection algorithms. To the best of our knowledge, it is the first time to develop an EO based solution framework for detecting overlapping communities of the networks.

The rest of the paper is organized as follows. Section 2 defines a modified modularity function to deal with the resolution limit problem. Section 3 explains the mechanism of EO based solution framework for modularity maximization, which mainly includes a brief

introduction to EO, the definition of local fitness functions of nodes, the design of a novel mutation operator, and the work flow of the algorithm. In Section 4, the experiments are conducted on multiple real and synthetic networks, and the detailed discussions are made. Finally, concluding remarks are given.

2. **Modified Modularity Function.** The modularity function proposed by Newman and Girvan [4] is shown in Equation (1),

$$Q = \sum_{i=1}^{K} \left(\frac{m_i}{m} - \left(\frac{d_i}{2m} \right)^2 \right) \tag{1}$$

where K is the number of communities, m_i is the number of edges of community i, m is the number of edges of the network, and d_i is the total degree of the nodes in community i. The optimization of Equation (1) to detect the communities of the networks has the resolution limit problem [30], which means it cannot find the communities of small size. For example, Figure 1 shows a sample network of fifty nodes consisting of two complete networks of twenty nodes C_{20} and two complete networks of five nodes C_5. The ground truth is to divide the network into four communities, as the dotted lines show in Figure 1(a). However, the modularity calculated from Equation (1) is equal to 0.5286 for Figure 1(a), which is less than that of the dividing for Figure 1(b). In Figure 1(b), the network is divided into three communities, where the two complete networks of C_5 are combined and the modularity is equal to 0.5295. Notice that, in Equation (1), the computing of the term $\frac{m_i}{m} - \left(\frac{d_i}{2m}\right)^2$ only takes the number of the edges into consideration, ignoring the number of nodes in the communities [33]. Therefore, here, we incorporate the community size n_i into Equation (1) and propose a modified version of modularity function, shown in Equation (2),

$$Q_m = \sum_{i=1}^{K} \frac{1}{n_i} \left(\frac{m_i}{m} - \left(\frac{d_i}{2m} \right)^2 \right) \tag{2}$$

where n_i is the number of nodes of the community i. According to Equation (2), the modularity for the network divisions in Figure 1(a) and Figure 1(b) is equal to 0.03353

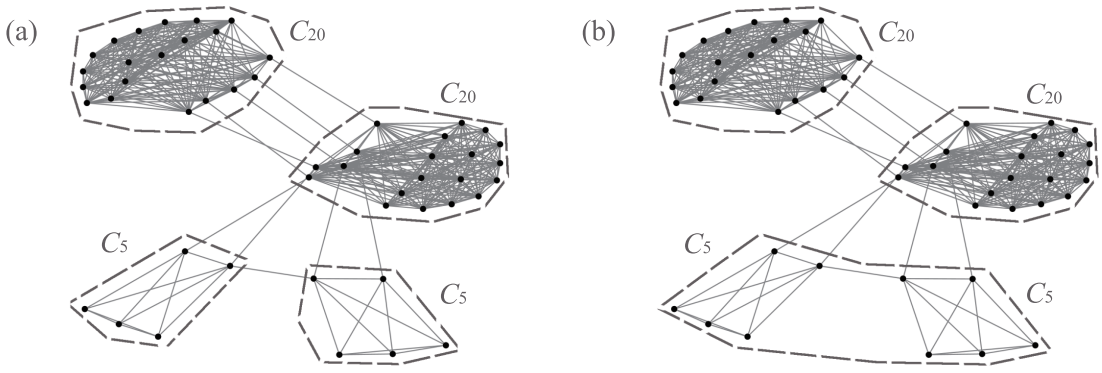

FIGURE 1. An illustrative example of resolution limit problem for Equation (1). The sample network consists of two complete networks of twenty nodes, C_{20}, and two complete networks of five nodes, C_5. (a) The network is divided into four communities, as the dotted lines show. The modularity according to Equation (1) is equal to 0.5286. (b) The network is divided into three communities, as the dotted lines show. The modularity according to Equation (1) is equal to 0.5295.

and 0.02888 respectively, which indicates the ground truth of community structure in Figure 1 can be found. Equation (2) also can be written in terms of edge pairs of the networks, shown below,

$$Q_m = \sum_{p=1}^{n}\sum_{q=1}^{n}\left[\frac{a_{pq}}{2m} - \frac{k_p k_q}{(2m)^2}\right]\delta(o_p, o_q), \tag{3}$$
$$\delta(o_p, o_q) = \begin{cases} \frac{1}{n_i}, & p \text{ and } q \text{ belong to the same community, assuming } i, \\ 0, & \text{otherwise} \end{cases}$$

where n is the number of nodes in the network, a_{pq} means the number of edges between node p and node q, k_p and k_q are the degrees of node p and node q respectively, and o_p and o_q are the community indexes that node p and node q belong to, respectively.

Note that, in this paper, we aim to detect overlapping communities in the networks. So we define the overlapping version of Equation (3) as follows,

$$Q_{mo} = \sum_{p=1}^{n}\sum_{q=1}^{n}\left\{\frac{1}{nc_p}\cdot\frac{1}{nc_q}\left[\frac{a_{pq}}{2m} - \frac{k_p k_q}{(2m)^2}\right]\sum_{i\in S}\frac{1}{n_i}\right\} \tag{4}$$

where nc_p and nc_q are the number of communities node p and node q belong to respectively, and S is the set of the indexes of the communities to which both node p and node q belong. When S is empty, the term $\sum_{i\in S}\frac{1}{n_i}$ is equal to zero. In next section, we develop an EO based solution framework to optimize Equation (4).

3. Extremal Optimization Based Solution Framework for Overlapping Communities Detection.

In this section, we apply EO to optimizing the overlapping modularity function of Equation (4) for overlapping communities detection. Firstly, we briefly introduce the general mechanism of EO, and then we give a detailed description of how to design a chromosome structure, the local fitness functions and a mutation operator of EO for the optimization of Equation (4).

3.1. Brief introduction to extremal optimization.

Different from GA, which operates on a population of chromosomes, EO evolves around a single chromosome S. Decision variables in the chromosome are called genes. A proper fitness function for genes needs to be defined as a part of the global fitness function of the chromosome. There is only one mutation operator in EO. Through mutating the worst gene successively, the chromosome S can improve its components and evolve towards the optimal solution generation by generation.

For a minimization problem with n decision variables, EO proceeds as follows [20].

Step 1: Design a structure for chromosomes, which is composed of n decision variables, a.k.a genes. Define the fitness functions and a mutation operator for genes based on the minimization problem.

Step 2: Initialize a candidate chromosome S with randomly setting the values of n genes. S is called the current chromosome. Set the best-so-far chromosome $S_{best} = S$.

Step 3: For the current chromosome S, do

(a): for each gene i in S, evaluate its fitness λ_i according to the predefined fitness function, $i = 1, 2, \ldots, n$,

(b): rank all the genes by their fitness values, and find gene j with the worst fitness value,

(c): change the value of the jth gene according to the predefined mutation operator, and then obtain a new chromosome S',

(d): set $S = S'$, and if the fitness of S' (i.e., the value of the minimization problem) is less than that of S, set $S_{best} = S'$.

Step 4: Repeat Step 3 for a number of iterations. And if a predefined threshold is reached, go to Step 5.

Step 5: Output the best-so-far chromosome S_{best}, which records the values of n decision variables and is the best-found solution for the minimization problem.

Step 3 (b) in original EO may result in the optimization procedure getting stuck in a local optimum, due to its kind of "greedy" update strategy. To avoid this, a slight modification is introduced to Step 3 (b) by making genes selected stochastically from a probability distribution $P(h) = h^{-\tau}$ [20], where h is the rank of a gene. After all genes are ranked according to their fitness values, a permutation π for genes' indexes is defined, so that the ranked fitness values can be expressed as $\lambda_{\pi(1)} < \lambda_{\pi(2)} < \cdots < \lambda_{\pi(n)}$. Assuming the gene with rank h is selected to mutate, Step 3 (b) can be modified with $j = \pi(h)$. The version of EO is called τ-EO. And in this manuscript, we adopt τ-EO to maximize the modularity function.

3.2. **Chromosome structure and local fitness function.** We can rewrite Equation (4) by introducing a binary decision matrix $\mathbf{X} \in R^{n\times K}$, shown below,

$$Q_{mo} = \sum_{p=1}^{n}\sum_{q=1}^{n}\left\{\frac{1}{\sum_{i=1}^{K} x_{pi}} \cdot \frac{1}{\sum_{i=1}^{K} x_{qi}}\left[\frac{a_{pq}}{2m} - \frac{k_p k_q}{(2m)^2}\right]\sum_{i\in S}\frac{1}{\sum_{c=1}^{n} x_{ci}}\right\} \tag{5}$$

where the element of the binary decision matrix, x_{pi}, indicates whether the node p belongs to the community i. When x_{pi} is equal to 1, the node p belongs to the community i; otherwise, it does not. The set S in Equation (5) contains the indexes of the communities to which both node p and node q belong, and can be figured out by comparing x_{pi} and x_{qi} for i from 1 to K. When S is empty, the term $\sum_{i\in S}\frac{1}{\sum_{c=1}^{n} x_{ci}}$ is equal to zero. The pth row in the binary decision matrix $\mathbf{X}$ depicts the pth node's community membership, denoted as $\mathbf{x}_p$, $p = 1, 2, \ldots, n$. The chromosome structure for optimizing Equation (5) is in the form of a matrix, with each row being a gene of the chromosome, shown in Figure 2.

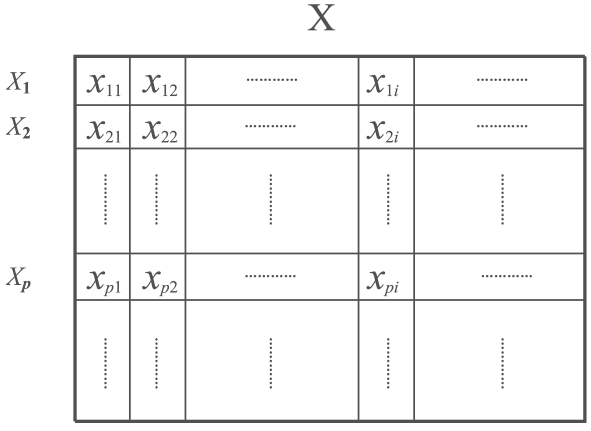

FIGURE 2. Chromosome structure of EO

A local fitness function for the pth gene can be defined as follows,

$$f_{localfitness}(\mathbf{x}_p) = \sum_{q=1}^{n}\left\{\frac{1}{\sum_{i=1}^{K} x_{pi}} \cdot \frac{1}{\sum_{i=1}^{K} x_{qi}}\left[\frac{a_{pq}}{2m} - \frac{k_p k_q}{(2m)^2}\right]\sum_{i\in S}\frac{1}{\sum_{c=1}^{n} x_{ci}}\right\}, \quad p = 1, 2, \ldots, n \tag{6}$$

which computes the summation of the difference of the real edge number and the expected edge number between node p and other nodes in a network, i.e., the contribution of node p to the global fitness function – Equation (5). Note that $Q_{mo} = \sum_{p=1}^{n} f_{localfitness}(\mathbf{x}_p)$, which means the summation of all the local fitness functions is equal to the global fitness function. This indicates the optimization of the local fitness functions can lead to the optimization of the global fitness function, which is essential for the effectiveness of EO.

3.3. **Mutation operator.** The mutation operator performs the random mutation or the dedicated mutation for the selected gene. Assuming the pth gene $\mathbf{x}_p$ is selected to mutate, i.e., the node p is set to change its community membership, it goes through a random mutation process with a probability P_{mu} or a dedicated mutation process with a probability $1 - P_{mu}$. In the random mutation process, the node p changes its community membership to the opposite or keeps its community membership unchanged with one half probability respectively. While in the dedicated mutation process, if the degree of the node p is less than a predefined threshold, make the node p belong to only one community which is identified as the one containing the most of neighbors of the node p. Otherwise, make the node p belong to multiple communities which are identified based on their contributions to the modularity value in Equation (5).

The detailed dedicated mutation process includes the following steps.

Step 1: Compute the degree of node p. If it exceeds a predefined threshold T_{deg}, go to Step 2, where the resulting community membership of node p becomes multiple. Otherwise, go to Step 3, where the resulting community membership of node p becomes single.

Step 2: Find out which community of the node p contributes negative to its fitness, and move node p out of these communities. Find out the communities to which the neighbors of node p belong, sort these communities in a descending order based on the number of node p's neighbors they have, and move node p into these communities, guaranteeing that the number of the "move out" communities is equal to the number of the "move in" communities.

Step 3: Find out the community which has the largest number of the neighbors of node p, and let node p only belong to this community.

The pseudocode for the mutation operator is shown in Algorithm 1.

3.4. **Work flow of the algorithm.** The proposed EO based overlapping communities detection algorithm maximizes Equation (5) iteratively. In each iteration, it goes through the computing of the fitness for each node, selecting a node to change its community membership, and generating a new community membership for this node. When the number of iterations reaches a predefined threshold or a certain stopping criterion is met, the algorithm ends and outputs the community memberships for all nodes and the corresponding modularity value.

The detailed work flow of the proposed algorithm is shown as follows.

Step 1: Initialize a candidate chromosome $\mathbf{X}$ with randomly setting the community memberships for n nodes. $\mathbf{X}$ is called the current chromosome. Set the best-so-far chromosome $\mathbf{X}_{best} = \mathbf{X}$.

Step 2: For the current chromosome $\mathbf{X}$, do

(a): for the community membership $\mathbf{x}_p$ of each node, evaluate its fitness using Equation (6), $p = 1, 2, \ldots, n$,

(b): rank all nodes by their fitness values in an ascending order, and get a permutation π, such that $f_{localfitness}(\mathbf{x}_{\pi(1)}) < f_{localfitness}(\mathbf{x}_{\pi(2)}) < \cdots < f_{localfitness}(\mathbf{x}_{\pi(n)})$,

Algorithm 1 Mutation operator

Input: K, $\mathbf{x}_p$, P_{mu}, T_{deg}, $\mathbf{CCB}_p$, $\mathbf{NNC}_p$
▷ K: *the number of communities*;
▷ $\mathbf{x}_p$: *node p's community membership, a K-dimensional vector*;
▷ P_{mu}: *probability for random mutation*;
▷ T_{deg}: *node degree threshold for multiple community membership or single community membership*;
▷ $\mathbf{CCB}_p$: *each community's contribution to the fitness of node p, a K-dimensional vector*;
▷ $\mathbf{NNC}_p$: *the number of node p's neighbors each community has, a K-dimensional vector*;
Output: $\mathbf{x}_p$

```
function MUTATING(K, x_p, P_mu, T_deg, CCB_p, NNC_p)
    prob ← rand()
    if prob < P_mu then                                   ▷ random mutation
        for i ← 1, K do
            if rand() > 0.5 then
                x_p(i) = 1 − x_p(i)
            end if
        end for
    else                                                  ▷ dedicated mutation
        Index^sort ← sorted community index in a descending order based on NNC_p
        if k_p > T_deg then
            for i ← 1, K do
                if CCB_p(i) < 0 then
                    x_p(i) = 0
                    for j ← 1, K do
                        if x_p(Index^sort(j)) == 0 then
                            x_p(Index^sort(j)) = 1
                        end if
                    end for
                end if
            end for
        else
            for i ← 1, K do
                if i == Index^sort(1) then
                    x_p(i) = 1
                else
                    x_p(i) = 0
                end if
            end for
        end if
    end if
    return x_p
end function
```

(c): select the jth node to mutate based on a probability distribution $P(h) = h^{-\tau}$, where h is a rank, and $j = \pi(h)$,

(d): change the community membership of the node j according to Algorithm 1, and obtain a new chromosome **X'**,

(e): set $\mathbf{X} = \mathbf{X'}$, and if the evaluation of **X'** is larger than that of **X** using Equation (5), set $\mathbf{X}_{best} = \mathbf{X'}$.

Step 3: Repeat Step 2 until a predefined threshold for the number of iterations is reached, or a predefined stopping criterion is met.

Step 4: Output the best-so-far chromosome $\mathbf{X}_{best}$, i.e., the community memberships for all nodes, and its fitness value, i.e., the modularity value of the detected overlapping community structure.

3.5. **Analysis of computational complexity.** Each iteration includes local fitness computing, gene selecting, and mutating. In local fitness computing, for one node, according to Equation (5), its time complexity is $O(Kn)$. Therefore, to compute all the nodes's fitness, its time complexity is $O(Kn^2)$. In gene selecting (i.e., node selecting), sorting the genes based on their fitness is needed before the selecting, which costs $O(n \log n)$. In mutating, compute the contributions of the selected node's communities to its fitness costs $O(Kn)$, sort the communities based on the number of the selected node's neighbors it has costs $O\left(\hat{k} + \hat{K} \log \hat{K}\right)$ ($\hat{k}$ is the degree of the selected node, and $\hat{K}$ is the number of communities the selected node belongs to), and compute the selected node's community membership costs $O(K^2)$. Generally speaking, n is much larger than K. Therefore, mutating costs $O(Kn)$. In summary, each iteration costs $O(Kn^2)$. Denote ite as the predefined number of the iterations of EO, so the time complexity of the EO based overlapping communities detection algorithm is $O(ite \times Kn^2)$.

4. **Experimental Evaluation and Discussion.** To verify the effectiveness and efficiency of the proposed EO based overlapping communities detection algorithm, the experiments are conducted on three synthetic network datasets and three real network datasets. We first examine the impacts of the parameters P_{mu} and T_{deg} on optimizing Equation (5). Then we show the convergence characteristics of EO and compare it to another popular evolutionary algorithm – genetic algorithm (GA for short). Thirdly, we make a comparison between the proposed EO based overlapping communities detection algorithm and several other baseline algorithms on three standard metrics – Overlapping Normalized Mutual Information (ONMI), Omega (Ω) Index, and F-Score. The synthetic network datasets are generated using LFR benchmark model [34], and the statistical properties of these network datasets are shown in Table 1.

LFR benchmark model is widely used to generate the networks with the desired community properties, which has the setting parameters of network size N, average degree $\bar{k}$, maximum degree $k_{\max}$, maximum community size $c_{\max}$, minimum community size $c_{\min}$, percentage of overlapping nodes O_n, number of communities the overlapping nodes belong to O_m, and mixing parameter μ, which states the ratio of inter-community edges and intra-community edges. Given a network size $N = 100$, multiple combinations of $\bar{k}$, $k_{\max}$, $c_{\max}$, $c_{\min}$, O_n, O_m, and μ can be set [27]. In our experiments, the networks with $N = 100$, $N = 1000$, and $N = 10000$ are generated using the LFR benchmark model, along with the real network datasets, aiming to provide a broad test bench for demonstrating the performance of the proposed EO based algorithm. LFR-100 is generated with $N = 100$, $\bar{k} = 10$, $k_{\max} = 20$, $c_{\max} = 30$, $c_{\min} = 15$, $O_n = 15\%$, $O_m = 2$, and $\mu = 0.1$. LFR-1000 is generated with $N = 1000$, $\bar{k} = 15$, $k_{\max} = 50$, $c_{\max} = 50$, $c_{\min} = 20$, $O_n = 10\%$, $O_m = 2$, and $\mu = 0.1$. And LFR-10000 is generated with $N = 10000$, $\bar{k} = 50$, $k_{\max} = 150$,

TABLE 1. Properties of the synthetic and real network datasets

Name	n	m	N_{GC}[1]	Avg_{ed}[2]	Avg_{cs}[3]	Avg_{ncm}[4]
LFR-100	100	507	5	0.36	23	1.15
LFR-1000	1000	7692	35	0.37	35.48	1.1
LFR-10000	10000	216972	417	0.08	92.32	3.85
Amazon [35]	334863	925872	75149	0.6	16.64	2.04
DBLP [35]	317080	1049866	13477	0.78	9.96	1.02
Youtube [35]	1134890	2987624	8385	0.56	44.24	1.62

[1] N_{GC} stands for the number of ground-truth communities.
[2] Avg_{ed} stands for the average of the community edge-density.
[3] Avg_{cs} stands for the average of community size.
[4] Avg_{ncm} stands for the average of node-community memberships.

$c_{\max} = 150$, $c_{\min} = 50$, $O_n = 15\%$, $O_m = 20$, and $\mu = 0.3$. The number of community K can be chosen from a certain range of values which is defined on the size of the given network. τ is set to 1.5 [36].

4.1. **Effects of the mutation parameters P_{mu} and T_{deg}.** According to Section 3.3, there are two preset parameters in the mutation operator of the proposed EO based overlapping communities detection algorithm. The parameter P_{mu} is the probability of performing random mutation, and the parameter T_{deg} is the degree threshold of nodes, above which performs multiple community membership mutation and below which performs single community membership mutation. By introducing a proportion variable $Prop_{deg}$, we define $T_{deg} = k_{\max} \times Prop_{deg}$. The experiments are conducted on LFR-100 and LFR-1000 network datasets to examine the effects of these two parameters on the optimization of Equation (5). P_{mu} and $Prop_{deg}$ are set from 0 to 1. Figure 3 shows Q_{mo} as a function of P_{mu} and Figure 4 shows Q_{mo} as a function of $Prop_{deg}$. The results are averaged over 100 runs.

As we can see, in Figure 3, Q_{mo} of P_{mu} from 0 to 0.4 is much better than that of P_{mu} from 0.5 to 1 in all six different cases, which means that the dedicated mutation plays a more important role in the optimization of Q_{mo} than the random mutation. Furthermore, Q_{mo} of $P_{mu} = 0.2$ is slightly better than that of other P_{mu} values. Thus, in the proposed EO based overlapping community detection algorithm, the P_{mu} is set to 0.2. In Figure 4, it is clear to see that Q_{mo} of $Prop_{deg}$ from 0.5 to 1 is much better than that of $Prop_{deg}$ from 0 to 0.4 in all six different cases, which means that to some extent, nodes with the larger degrees are probably with the multiple community memberships. Furthermore, from Figure 4, Q_{mo} of $Prop_{deg} = 0.7$ in most cases is better than that of other $Prop_{deg}$ values. Thus, in the proposed EO based overlapping community detection algorithm, the $Prop_{deg}$ is set to 0.7.

4.2. **Characteristics of convergence.** Figure 5 shows the convergence characteristics of EO based overlapping community detection algorithm. From the figure, it is clear to see that the proposed EO based algorithm is capable of converging fast. For example, in Youtube network dataset, EO converges after about 2000 iterations; in LFR-100, DBLP and Amazon network datasets, EO converges after about 5000 iterations; and in LFR-1000 and LFR-10000 network datasets, EO converges after about 10000 iterations.

We also make a comparison between EO based overlapping community detection algorithm and GA based overlapping community detection algorithm in LFR-100 network

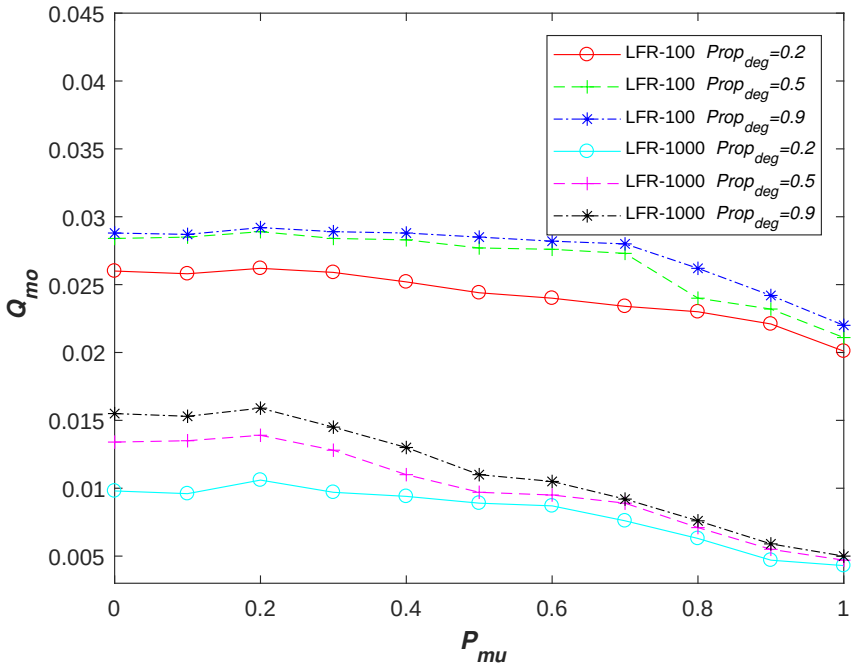

FIGURE 3. Q_{mo} as a function of P_{mu}

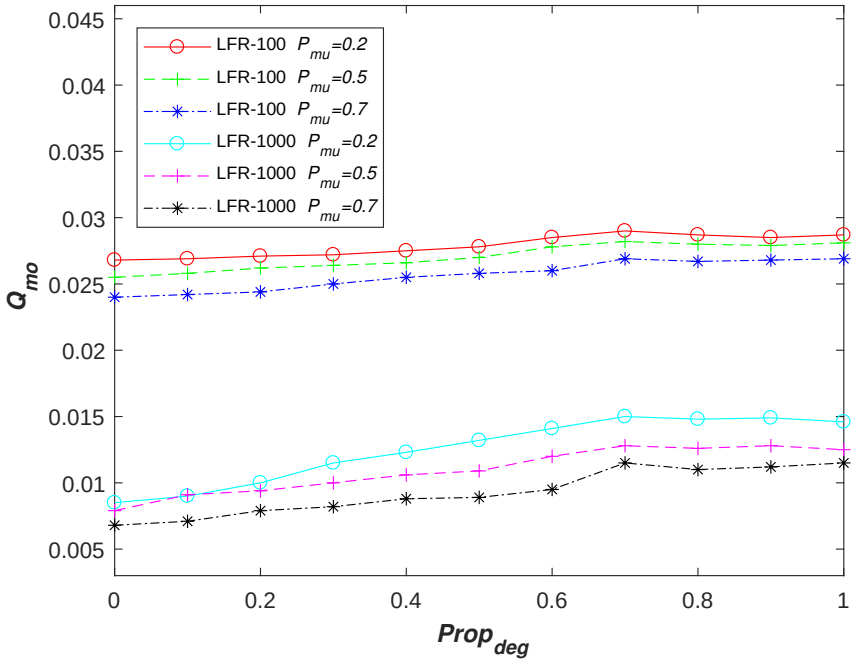

FIGURE 4. Q_{mo} as a function of $Prop_{deg}$

dataset. Figure 6 shows EO converges much faster than GA, and can obtain better Q_{mo} value.

4.3. **Comparison with other baseline algorithms.** We compare the proposed EO based overlapping community detecting algorithm with CONGA [37], CORPA [38], EAGLE [39], OSLOM [40], and SLPA [41] on three standard metrics – Overlapping Normalized Mutual Information (ONMI), Omega (Ω) Index, and F-Score. The proposed algorithm runs 100 times on each network dataset and the results are averaged. From

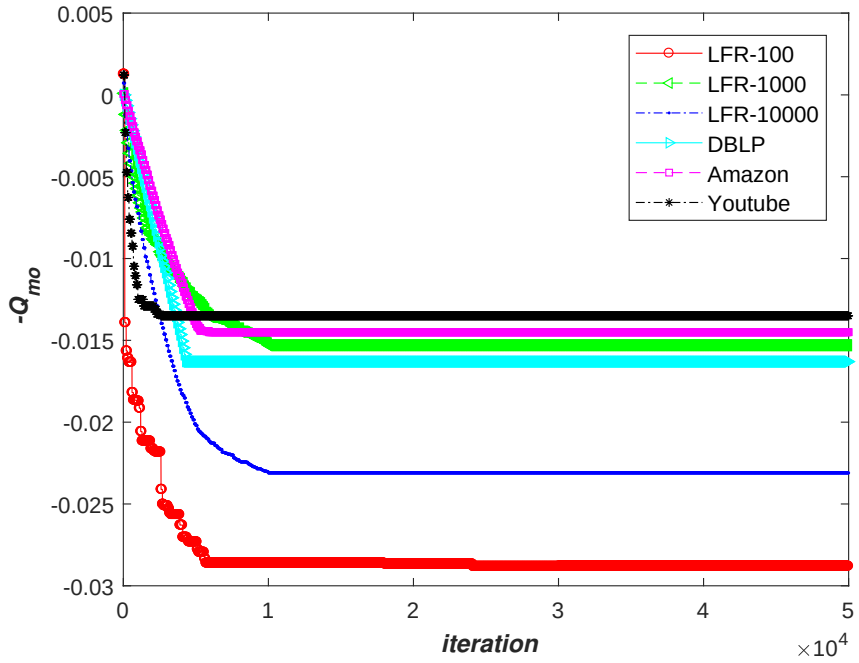

FIGURE 5. Convergence characteristics of EO

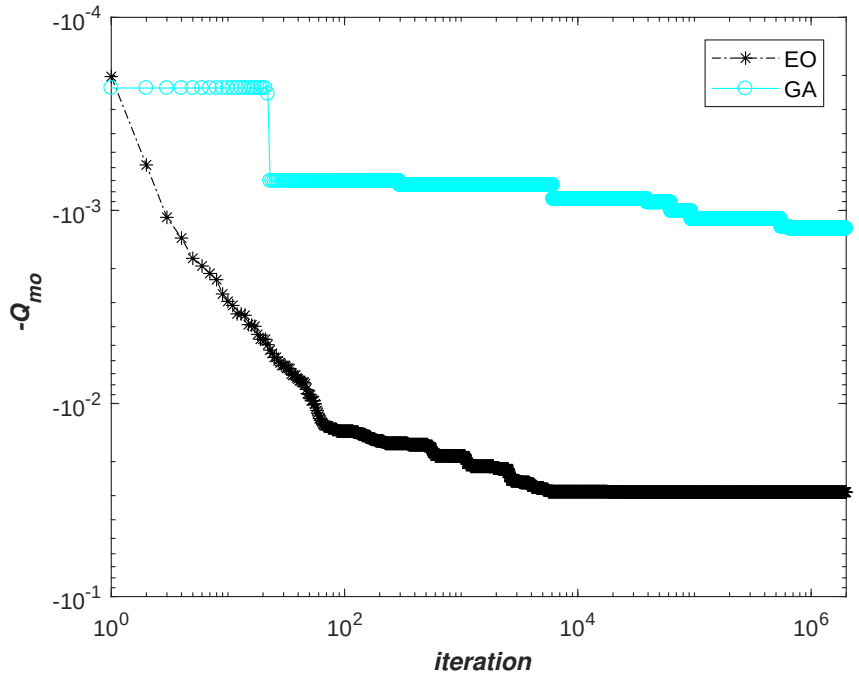

FIGURE 6. Convergence characteristics of EO and GA in LFR-100

Table 2, it is clear to see that EO can obtain better values on ONMI of LFR-1000, Ω Index of LFR-100, LFR-10000, Amazon, and DBLP, and F-Score of DBLP; OSLOM can obtain better values on ONMI of LFR-100, LFR-1000, DBLP and Youtube, Ω Index of LFR-1000, and F-Score of LFR-100, LFR-10000, DBLP, and Youtube; CORPA can obtain better values on ONMI of LFR-10000, Ω Index of Youtube, and F-Score of LFR-1000; CONGA can obtain better values on ONMI and F-Score of DBLP; EAGLE can obtain better values on ONMI and F-Score of Amazon; and SLPA can obtain better

TABLE 2. Comparison with five baseline algorithms on ONMI, Ω index, and F-Score

		LFR-100	LFR-1000	LFR-10000	Amazon	DBLP	Youtube
EO	ONMI	0.85	**0.90**	0.80	0.91	0.80	0.79
	Ω Index	**0.91**	0.89	**0.75**	**0.71**	**0.64**	0.61
	F-Score	0.93	0.91	0.85	0.94	**0.90**	0.88
CONGA	ONMI	0.49	0.57	0.58	0.62	**0.82**	0.75
	Ω Index	0.52	0.57	0.40	0.38	0.63	0.52
	F-Score	0.74	0.78	0.72	0.75	**0.90**	0.83
CORPA	ONMI	0.79	0.87	**0.82**	0.93	0.79	0.84
	Ω Index	0.80	0.89	0.70	0.66	0.60	**0.63**
	F-Score	0.87	**0.95**	0.84	0.94	0.84	0.88
EAGLE	ONMI	0.74	0.68	0.81	**0.95**	0.62	0.75
	Ω Index	0.77	0.73	0.65	0.70	0.52	0.60
	F-Score	0.80	0.83	0.83	**0.97**	0.83	0.80
OSLOM	ONMI	**0.86**	**0.90**	0.80	0.83	**0.82**	**0.80**
	Ω Index	0.90	**0.91**	0.71	0.50	0.60	0.55
	F-Score	**0.96**	0.93	**0.86**	0.88	**0.90**	**0.91**
SLPA	ONMI	0.76	0.85	0.76	0.77	**0.82**	0.79
	Ω Index	0.82	0.88	0.65	0.50	0.61	0.60
	F-Score	0.92	0.94	0.81	0.87	0.89	0.85

value on ONMI of DBLP. These results show the proposed algorithm has the competitive performance in detecting overlapping communities of the networks.

In summary, the proposed EO based algorithm for overlapping communities detection shows fast convergence characteristics, and can get better modularity value while spending less time than GA based algorithm. On three synthetic and three real network datasets with the ground-truth communities, the proposed EO based algorithm obtains the competitive values on ONMI, Ω Index, and F-Score compared to several other baseline algorithms. In the practical applications, it means the proposed algorithm can find the groups of co-purchased products in Amazon, the cliques of the co-authors in DBLP, and the groups of people who share videos of the similar topics in Youtube with good quality. These detected overlapping communities can help to design better recommendation systems and online advertising systems.

5. **Conclusion.** Community structures of the networks can help us better understand the underlying dynamics of the networks. Considering the structure of the modularity function, in this paper, we first propose an EO based algorithm to detect the overlapping communities of the networks. Firstly, a modified version of the modularity function is defined to deal with the resolution limit problem. Secondly, a chromosome structure, the local fitness functions, and a novel mutation operator of EO for maximizing modularity function are designed. Experimental results on multiple real and synthetic network datasets show the fast convergence of the proposed algorithm and the competitive performance on three standard metrics – ONMI, Ω Index, and F-Score, when compared to several other baseline overlapping communities detection algorithms. In further study, we endeavor to investigate how to combine EO with other prominent overlapping communities detection algorithms to achieve better performance.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (51677171).

REFERENCES

[1] S. Fortunato and D. Hric, Community detection in networks: A user guide, *Physics Reports*, vol.659, pp.1-44, 2016.

[2] Y.-N. Tang, J. Xiang, Y.-Y. Gao, Z.-Z. Wang, H.-J. Li, S. Chen, Y. Zhang, J.-M. Li, Y.-H. Tang and Y.-J. Chen, An effective algorithm for optimizing surprise in network community detection, *IEEE Access*, vol.7, pp.148814-148827, 2019.

[3] Z.-H. Liu, B.-J. Xiang, W.-Z. Guo, Y.-Z. Chen, K. Guo and J.-N. Zheng, Overlapping community detection algorithm based on coarsening and local overlapping modularity, *IEEE Access*, vol.7, pp.57943-57955, 2019.

[4] M. E. Newman and M. Girvan, Finding and evaluating community structure in networks, *Physical Review E*, vol.69, no.2, 026113, 2004.

[5] A. Clauset, M. E. Newman and C. Moore, Finding community structure in very large networks, *Physical Review E*, vol.70, no.6, 066111, 2004.

[6] R. Aldecoa and I. Marín, Surprise maximization reveals the community structure of complex networks, *Scientific Reports*, vol.3, no.1, pp.1-9, 2013.

[7] L.-Q. Meng, B. Fang, X.-Y. Liu, Y.-B. Shang and A. Luo, Predicting retweet behavior on social media within communities from the perspective of user behavior spreading, *ICIC Express Letters, Part B: Applications*, vol.10, no.5, pp.371-378, 2019.

[8] N. Laitonjam, W. Huáng and N. J. Hurley, Scalable inference on the soft affiliation graph model for overlapping community detection, *Asian Conference on Machine Learning*, PMLR, pp.673-688, 2020.

[9] X.-Y. Lu and B. K. Szymanski, A regularized stochastic block model for the robust community detection in complex networks, *Scientific Reports*, vol.9, no.1, pp.1-9, 2019.

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

[11] B. Karrer and M. E. Newman, Stochastic blockmodels and community structure in networks, *Physical Review E*, vol.83, no.1, 016107, 2011.

[12] P. Zhang and C. Moore, Scalable detection of statistically significant communities and hierarchies, using message passing for modularity, *Proceedings of the National Academy of Sciences*, vol.111, no.51, pp.18144-18149, 2014.

[13] P.-Z. Li, L. Huang, C.-D. Wang, J.-H. Lai and D. Huang, Community detection by motif-aware label propagation, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol.14, no.2, pp.1-19, 2020.

[14] Y. Zhang, Y.-G. Liu, J.-T. Li, J.-J. Zhu, C.-H. Yang, W. Yang and C.-B. Wen, WOCDA: A whale optimization based community detection algorithm, *Physica A: Statistical Mechanics and Its Applications*, vol.539, 122937, 2020.

[15] V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment*, vol.2008, no.10, P10008, 2008.

[16] V. A. Traag, Faster unfolding of communities: Speeding up the Louvain algorithm, *Physical Review E*, vol.92, no.3, 032801, 2015.

[17] M. E. Newman, Modularity and community structure in networks, *Proceedings of the National Academy of Sciences*, vol.103, no.23, pp.8577-8582, 2006.

[18] R. Shang, J. Bai, L. Jiao and C. Jin, Community detection based on modularity and an improved genetic algorithm, *Physica A: Statistical Mechanics and Its Applications*, vol.392, no.5, pp.1215-1231, 2013.

[19] S. Rahimi, A. Abdollahpouri and P. Moradi, A multi-objective particle swarm optimization algorithm for community detection in complex networks, *Swarm and Evolutionary Computation*, vol.39, pp.297-309, 2018.

[20] S. Boettcher and A. Percus, Nature's way of optimizing, *Artificial Intelligence*, no.1, pp.275-286, 2000.

[21] Y.-Z. Lu, M.-R. Chen and Y.-W. Chen, Studies on extremal optimization and its applications in solving realworld optimization problems, *IEEE Symposium on Foundations of Computational Intelligence (FOCI2007)*, Honolulu, HI, USA, pp.162-168, 2007.

[22] S. Boettcher and A. G. Percus, Extremal optimization at the phase transition of the three-coloring problem, *Physical Review E*, vol.69, no.6, 066703, 2004.
[23] Y.-W. Chen, Y.-Z. Lu and P. Chen, Optimization with extremal dynamics for the traveling salesman problem, *Physica A: Statistical Mechanics and Its Applications*, vol.385, no.1, pp.115-123, 2007.
[24] J. Ding, Y.-Z. Lu and J. Chu, Studies on controllability of directed networks with extremal optimization, *Physica A: Statistical Mechanics and Its Applications*, vol.392, no.24, pp.6603-6615, 2013.
[25] Y.-W. Chen, Y.-Z. Lu, M. Ge, G.-K. Yang and C.-C. Pan, Development of hybrid evolutionary algorithms for production scheduling of hot strip mill, *Computers & Operations Research*, vol.39, no.2, pp.339-349, 2012.
[26] G.-Q. Zeng, X.-Q. Xie, M.-R. Chen and J. Weng, Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems, *Swarm and Evolutionary Computation*, vol.44, pp.320-334, 2019.
[27] J. Duch and A. Arenas, Community detection in complex networks using extremal optimization, *Physical Review E*, vol.72, no.2, 027104, 2005.
[28] R. I. Lung, M. Suciu and N. Gaskó, Noisy extremal optimization, *Soft Computing*, vol.21, no.5, pp.1253-1270, 2017.
[29] N. Gaskó, R. I. Lung and M. A. Suciu, Community detection in bipartite networks using a noisy extremal optimization algorithm, *International Conference on Intelligent Systems Design and Applications*, pp.871-878, 2016.
[30] S. Fortunato and M. Barthelemy, Resolution limit in community detection, *Proceedings of the National Academy of Sciences*, vol.104, no.1, pp.36-41, 2007.
[31] A. F. McDaid, D. Greene and N. Hurley, Normalized mutual information to evaluate overlapping community finding algorithms, *arXiv Preprint arXiv:1110.2515*, 2011.
[32] J. Yang and J. Leskovec, Overlapping community detection at scale: A nonnegative matrix factorization approach, *Proc. of the 6th ACM International Conference on Web Search and Data Mining*, pp.587-596, 2013.
[33] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang and L. Chen, Quantitative function for community detection, *Physical Review E*, vol.77, no.3, 036109, 2008.
[34] A. Lancichinetti and S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, *Physical Review E*, vol.80, no.1, 016118, 2009.
[35] J. Yang and J. Leskovec, Defining and evaluating network communities based on ground-truth, *Proc. of the 12th International Conference on Data Mining*, pp.745-754, 2012.
[36] S. Boettcher and A. G. Percus, Extremal optimization for graph partitioning, *Physical Review E*, vol.64, no.2, 026114, 2001.
[37] S. Gregory, An algorithm to find overlapping community structure in networks, *European Conference on Principles of Data Mining and Knowledge Discovery*, pp.91-102, 2007.
[38] S. Gregory, Finding overlapping communities in networks by label propagation, *New Journal of Physics*, vol.12, no.10, 103018, 2010.
[39] H. Shen, X. Cheng, K. Cai and M.-B. Hu, Detect overlapping and hierarchical community structure in networks, *Physica A: Statistical Mechanics and Its Applications*, vol.388, no.8, pp.1706-1712, 2009.
[40] A. Lancichinetti, F. Radicchi, J. J. Ramasco and S. Fortunato, Finding statistically significant communities in networks, *PloS One*, vol.6, no.4, e18961, 2011.
[41] J. Xie and B. K. Szymanski, Towards linear time overlapping community detection in social networks, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.25-36, 2012.