Network community-detection enhancement by proper weighting

Alireza Khadivi, Ali Ajdari Rad, and Martin Hasler

Laboratory of Nonlinear Systems, School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

(Received 7 June 2010; revised manuscript received 22 December 2010; published 11 April 2011)

In this paper, we show how proper assignment of weights to the edges of a complex network can enhance the detection of communities and how it can circumvent the resolution limit and the extreme degeneracy problems associated with modularity. Our general weighting scheme takes advantage of graph theoretic measures and it introduces two heuristics for tuning its parameters. We use this weighting as a preprocessing step for the greedy modularity optimization algorithm of Newman to improve its performance. The result of the experiments of our approach on computer-generated and real-world data networks confirm that the proposed approach not only mitigates the problems of modularity but also improves the modularity optimization.

DOI: 10.1103/PhysRevE.83.046104 PACS number(s): 89.75.Hc, 89.65.Ef, 89.75.Fb

I. INTRODUCTION

The study of the properties of complex networks has recently attracted many researchers from various disciplines such as physics, biology, and mathematics [1–4]. A feature, observable in many networks, is the presence of community structures, i.e., clusters of vertices which are densely connected to each other while less connected to the vertices outside. The community structure identification is an important problem in a wide range of applications such as marketing in social networks and the study of protein interaction networks. Usually, the community members have more properties in common among themselves than with nonmembers and detecting community structure helps in analyzing and searching the network with less effort.

In many scenarios, community detection methods can help to unveil the functional properties of the complex networks. Besides, many of these networks such as the Internet are large scale and their size grows with time. Thus there is a necessity to devise better community detection methods which meet both speed and accuracy requirements simultaneously.

In order to estimate how much a decomposition of a network which is found by a community detection algorithm is meaningful, we need a measure. For a particular measure, the community detection algorithms can be ranked. So far, various measures have been proposed in the literature. These measures can be classified into two distinct categories. The first category consists of measures which indicate the accuracy of the found partition by comparing it to the true one. Thus before using these measures, the correct communities should already be known, which is not the case in a wide range of networks including many real-world ones. This constraint mostly limits their application to computer-generated benchmark graphs. Some examples of such measures are the *fraction of the nodes classified correctly* [5], the *Jaccard index* [6], and *normalized mutual information* [7].

The second category belongs to the measures which evaluate the quality of the found partition and are mostly used when there is no information about the correct communities. Indeed, such measures cannot provide the accuracy of the found partition since they do not assess how close it is to the true partition. Angelini *et al.* [8] have proposed a measure called *ratio association*, which is derived from the ratio of the number of edges to the number of vertices within each partition. But, the most prevalent measure, which has been used extensively in the literature, is due to Newman and Girvan [9]. This measure, called *modularity*, quantifies how much the density of the edges inside identified communities differs from the expected edge density in an equivalent network with similar number of vertices and edges but randomized edge placement. Indeed, the higher the modularity value, the better the decomposition. Therefore considering the modularity measure, the community detection problem is transformed to the modularity maximization problem. In this paper, we focus on this measure for our analysis.

It has been discovered that there is a strong correlation between structure and function of complex networks. Accordingly, a function of a complex network can be significantly influenced by its structure. Thus one can think of adjusting a proper structural property such as weighting of the links to improve a particular function of a complex network.

In this paper, we discuss how an appropriate weighting scheme can affect the problems that are associated with modularity, i.e., the resolution limit [10] and the extreme degeneracy [11]. Then, we take advantage of the local and global structural information of the network and propose a weighting scheme that not only mitigates the problematic behavior of modularity, but also remarkably improves the community detection performance of the Clauset-Newman-Moore method (CNM) [12], which is a simple greedy modularity optimization method. The proposed approach is a good compromise between speed and accuracy. Briefly summarized, a proper weighting scheme is applied on a given unweighted network and then a modularity optimization algorithm, i.e., CNM, is used to determine the communities.

II. RELATED WORK

Several algorithms have been proposed for modularity optimization [5,8,9,12–18]. The main shortcoming of most of the available algorithms is their high computational complexity. This is the main reason that prevents their use for large-size networks and restricts them to networks with a few thousand vertices at most. Some algorithms have been recently proposed that achieve acceptable performance with rather low computational cost. Duch and Arenas (DA) proposed an extremal optimization based procedure [13] that finds the community structure in $O(n^2 \log n)$ operations, where n is the number of vertices in the network, and obtained very good performance for the modularity measure. Blondel *et al.* [16] have proposed a low computational complexity method, of order $O(n \log n)$, for sparse networks, called the Louvain method, which is one of the fastest available algorithms proposed so far. This method has shown very good performance on benchmark networks [19].

Also, Newman proposed a greedy approach [18] called the fast Newman method. In particular, this method initially assumes each individual vertex to be a separate community and then, it repeatedly finds a pair of communities whose joining gives the greatest increase in modularity and joins them together. It continues joining communities until a single community is left. Finally, the community division which corresponds to the highest modularity value is chosen as the outcome. In [12], Clauset, Newman, and Moore introduced a faster implementation of the same algorithm called CNM with the computational complexity of $O(md \log n)$, where m is the number of edges and d is the depth of the dendrogram, i.e., a tree diagram used to illustrate the hierarchical arrangement of the clusters of a network. The CNM algorithm is simple, intuitive, and fast. Indeed, it is much faster than DA from a computational complexity point of view, but poorer in modularity optimization. It has been frequently reported to show poor performance on some classes of networks. The low performance can be rooted to two main sources, namely the greedy nature of the algorithm and the failure of the modularity measure to capture all needed information for perfect community detection.

The idea of reconfiguring the network to enhance a specific function is used in different eras. Two main approaches frequently applied in the literature are proper rewiring [20] and weighting of links [21]. The improvement of a network function using a proper weighting can be traced back to the training of artificial neural networks, while exploiting a weighting scheme based on structural properties of complex networks is a much newer technique. Recently, proper weighting schemes have been successfully applied to various problems, including improvement of synchronizability of complex networks [21], reservoir-based recurrent neural networks [22], and average consensus on complex networks [23]. In particular, Berry *et al.* have proposed a weighting scheme for community detection enhancement with runtime of $O(mn \log n)$ for scale free graphs [24].

III. MODULARITY ANALYSIS

In order to reveal the properties of modularity, one should know the intuition behind the terms that define it. For a particular partition of an unweighted network, the standard definition of modularity [9] is

$$Q = \sum_{i=1}^{c} \left[\frac{l_i^{\text{in}}}{L} - \left(\frac{d_i}{2L} \right)^2 \right],$$

where c is the number of communities, L is the total number of edges in the network, and l_i^{in} and $d_i = 2l_i^{\text{in}} + l_i^{\text{out}}$ are the number of edges and the sum of vertex degrees in the ith community, respectively. Also, l_i^{out} is the number of edges with one end in the ith community. The first term of the summation is indeed the fraction of edges within community i while the second term corresponds to the expected value for the same community in a random network with the same degree sequence. The definition of modularity is generalized accordingly for weighted networks. In such networks, L is the sum of weights of all edges in the network, and l_i^{in} and d_i are, respectively, the sum of weights of edges and the sum of degrees of vertices of the ith community where the degree of a vertex is defined as the sum of weights of edges connected to that vertex. An unweighted network can be considered as a special case of a weighted network with all edge weights equal to 1.

Although the standard modularity measure has been accepted as a convenient measure among researchers, it has some intrinsic shortcomings [25]. For instance, the modularity value for a particular partition of network cannot individually express how good this division is, e.g., there are partitions of instances of random networks that have high modularity values just by pure chance. Therefore a network has only an intrinsic community structure, if its modularity value is significantly larger than the (probabilistic) maximum modularity of the random network with the same expected degree sequence [26]. As a more serious criticism, it has been shown that modularity is highly correlated to the presence of a community structure only if the size of the communities is limited by a value which is dependent to the total number of edges in the network [10]. This limit on the size of the communities is counterintuitive since networks can have communities with different sizes depending on the density of intercluster and intracluster edges. There is also the recent discovery of extreme degeneracies in the modularity [11]. In other words, one can find various distinct solutions, i.e., high modularity partitions without a clear global maximum for a given network.

The ratio $R_i = l_i^{\text{out}}/l_i^{\text{in}}$ plays an important role in the mentioned problems of modularity. This ratio should be within $\lfloor 0,2 \rfloor$ in order to satisfy the criterion of the "weak" definition of community given by Radicchi [27]. Next, we will show how a proper weighting scheme can abate the resolution limit as well as extreme degeneracy problems and make modularity a more trustable measure. For the sake of simplicity, we keep the sum of weights of all edges, i.e., L of the weighted graph constant and equal to the total number of edges. Thus L is equal for both weighted and unweighted versions of the network. Also, as a general notation, all quantities that carry a hat, e.g., $\hat{R}$, belong to the weighted graph and are generalizations of the same quantities without a hat belonging to the unweighted graph, e.g., R.

A. Resolution limit

Fortunato *et al.* have discovered the resolution limit of modularity. This is an important reason why optimization performs poorly in certain circumstances [10]. It is shown in [10] that the communities determined by modularity optimization are constrained in size, i.e., l_i^{in} falls necessarily into a certain subinterval of $\lfloor 0,L \rfloor$. Therefore when the size of a true community falls outside of this subinterval, it cannot be detected by modularity optimization. The effect of introducing a proper weight on the edges of the graph is that the subinterval gets larger and therefore weighted modularity, optimization, i.e., maximizing the modularity value for the weighted version of the network, is capable of detecting communities with a larger range of sizes. We first recall the rigorous bounds on the community sizes from [10].

Theorem 1. The size of any community that contributes positively to the modularity measure is limited by

$$\frac{l^{\text{in}}}{L} < \frac{4}{(2+R)^2}, \tag{1}$$

where $R = l^{\text{out}}/l^{\text{in}}$. If merging community i and community j does not increase the modularity measure, the sizes of the two communities are bounded below by

$$\begin{aligned} \frac{l_i^{\text{in}}}{L} &\geqslant \frac{2R_{ij}}{(R_i+2)(R_j+2)}; \quad \text{for all } j, \\ \frac{l_j^{\text{in}}}{L} &\geqslant \frac{2R_{ji}}{(R_i+2)(R_j+2)}; \quad \text{for all } i, \end{aligned} \tag{2}$$

where $R_{ij} l_j^{\text{in}} = R_{ji} l_i^{\text{in}} = l_{ij}^{\text{out}}$, which is the sum of weights of edges between ith and jth communities. ■

If we weight the network such that the sum of all weights is L, if $\hat{l}_i^{\text{in}}$ and $\hat{l}_i^{\text{out}}$ are, respectively, the sum of the weights of the edges within community i and the sum of the weights of the edges connecting a vertex of community i to a vertex of another community, if $\hat{R}_i = \hat{l}_i^{\text{out}}/\hat{l}_i^{\text{in}}$, $\hat{R}_{ij} = \hat{l}_{ij}^{\text{out}}/\hat{l}_j^{\text{in}}$, and if we use the weighted modularity measure, the constraints on the community sizes corresponding to those of Theorem 1 are

$$\frac{2\hat{R}_{ij}}{(2+\hat{R}_i)(2+\hat{R}_j)} \leqslant \frac{\hat{l}_i^{\text{in}}}{L} < \frac{4}{(2+\hat{R}_i)^2}. \tag{3}$$

As we will show, the weighting we propose will in general reduce the value of $\hat{R}_i$ and therefore increase the upper bound for the size of the community. As far as the lower bound is concerned, reducing $\hat{R}_i, \hat{R}_j$, and $\hat{R}_{ij}$ will not necessarily reduce the bound. However, making the reasonable assumption that $\hat{R}_i \leqslant 2$ for all i, we can give an interval for the lower bound:

$$\frac{1}{8}\hat{R}_{ij} \leqslant \frac{2\hat{R}_{ij}}{(2+\hat{R}_i)(2+\hat{R}_j)} \leqslant \frac{1}{2}\hat{R}_{ij}. \tag{4}$$

Both endpoints of this interval decrease when $\hat{R}_{ij}$ decreases. Therefore we conclude that the lower bound for the size of the communities in general decrease when our weighting is introduced.

In summary, the introduction of our weights generally allows us to detect both larger and smaller communities by modularity optimization than would be possible without the weights.

B. Extreme degeneracy

A recent analysis on modularity is given by Good *et al.* in [11]. An important issue of modularity that has been argued is the existence of extreme degeneracies in modularity. In some cases, the penalty for joining two distinct communities can be insignificant even when modularity decreases, i.e., $\Delta Q < 0$. Thus it frequently happens that there are many graph partitions whose modularity is very close to the global maximum $Q_{\max}$. More specifically, the number of slightly suboptimal partitions increases when there are more modular groups of vertices in the network. As a result, it is more difficult to find the optimal partition among competing suboptimal ones when the network has a large number of modules. This problem is called extreme degeneracy. It should be mentioned that extreme degeneracy is problematic only when competing maxima are very different from each other. Here, we show how an efficient weighting scheme can improve this situation. To do so, we will discuss the two cases which were investigated in [11] for illustrative purposes.

The first case concerns extreme degeneracy in modular networks. Indeed, it is argued that the penalty in Q for joining two true communities i and j in a network with k communities and roughly equal community degrees, i.e., $d_i \approx 2L/k$, is bounded below by $\Delta Q_{ij} = -2k^{-2}$. The difference in modularity for joining two groups i and j is

$$\Delta Q_{ij} = \frac{l_{ij}^{\text{out}}}{L} - 2\left(\frac{d_i}{2L}\right)\left(\frac{d_j}{2L}\right). \tag{5}$$

An efficient weighting scheme that is capable of reducing the weights of intercluster edges decreases the first term of Eq. (5) and consequently increases the penalty in Q for an inappropriate community join. When the size of communities is less than a certain threshold, then the first term becomes larger than the second term and the resolution limit problem occurs, i.e., $\Delta Q > 0$. Thus one can help mitigate both extreme degeneracy and the resolution limit problem by reducing R using an efficient weighting scheme. We admit that at the same time when the number of communities increases, the second term becomes small and the penalty for joining them is reduced.

In the second case, a simple type of a hierarchical network is considered. Suppose that there are two true communities i and j, with two submodules each, so that $\{a,b\} \in i$ and $\{c,d\} \in j$. Merging the opposite pairs of submodules, i.e., a with c and b with d, will produce the penalty

$$\begin{aligned} \Delta Q = {} & \frac{\left(l_{ac}^{\text{out}} + l_{bd}^{\text{out}}\right)}{L} - \frac{\left(l_{ab}^{\text{out}} + l_{cd}^{\text{out}}\right)}{L} \\ & - 2\left(\frac{d_a - d_d}{2L}\right)\left(\frac{d_c - d_b}{2L}\right), \end{aligned} \tag{6}$$

which may produce still a smaller penalty than the first case. In this case also, if a weighting scheme can strengthen the intracommunity edges and weaken the rest, then it will decrease the first term of Eq. (6) while increasing the second one. Hence the total penalty will increase by introducing a proper weighting scheme.

Based on the arguments given in this section, we propose to use a proper weighting scheme prior to applying a community detection method based on modularity optimization. The weighting scheme should be able to discriminate between intercluster and intracluster edges. It should be able to strengthen the weights of intracluster edges while decreasing the weights of intercluster ones in order to mitigate the modularity resolution limit and extreme degeneracies. Therefore such a weighting scheme acts like a preprocessing procedure that sets up the network for the next step, i.e., community detection. Intuitively, it gives prior knowledge to community detection methods and improves their efficiency. In the following, we introduce a weighting scheme that meets the criteria outlined before.

IV. PROPOSED WEIGHTING SCHEME

In this section, we propose a weighting scheme that meets the requirements discussed in the previous section and enhances the performance of the CNM significantly. This scheme benefits from two well-known structural measures of complex networks, namely the edge betweenness centrality and common neighbor ratio measures which are described below.

For an edge e_{ij} connecting vertex i and vertex j, the edge-betweenness centrality (EBC) measure, denoted by B_{ij}, is defined as the number of shortest paths passing through e_{ij} [28]. This measure can be formally expressed as follows:

$$B_{ij} = \sum_{u \neq v \in V} \frac{\sigma_{uv}(e_{ij})}{\sigma_{uv}}, \tag{7}$$

where $\sigma_{uv}(e_{ij})$ is the number of shortest paths between vertices u and v that pass through edge e_{ij} and σ_{uv} is the total number of shortest paths between u and v.

The paths that connect vertices of distinct communities must pass through at least one intercluster edge. Bearing in mind the fact that the communities are loosely connected, one can expect that the intercluster edges have usually rather high EBC scores. On the other hand, the vertices within a community are tightly connected, so the betweenness centrality of intracluster edges is usually smaller. Consequently, it seems that the inverse of EBC is an appropriate parameter to weight the network. Edge betweenness centrality measure has been previously used in some community detection algorithms [5,29].

Although EBC conveys important information about the communities, it could be sometimes misguiding for community detection algorithms. The betweenness centrality of an edge depends on the number of vertices that are connected to the network through that edge. Thus it is easy to imagine cases where the betweenness centrality of an intercluster edge which connects a small community to the network is less than the betweenness centrality of an intracluster edge in a large community. An example is given in Fig. 1 in which there is one small community (M_1) with n_1 vertices and one large community (M_2) with n_2 vertices, each of which is connected by just one edge to the rest of the network (M_0), i.e., e_1^{out} and e_2^{out}, respectively. The total number of vertices is n. Thus supposing that between two vertices there is always only one shortest path, the betweenness centrality of e_1^{out} and e_2^{out} are $(n-n_1)n_1$ and $(n-n_2)n_2$, respectively. Now, we assume the following conditions for this network. l_2^{out} has only two neighboring edges in M_2 (e_1^{in} and e_2^{in}), the number of vertices in M_2 is three times the number of vertices in M_1 ($n_2 = 3n_1$), and finally $n_1 n_2 \ll n$. With these conditions the betweenness centrality for e_2^{out} is greater than for e_1^{out} by a factor of almost 3 while the betweenness centrality of one of e_1^{in} and e_2^{in} intracluster edges has at least half of the betweenness centrality of e_2^{out}. Hence there is at least one intracluster edge whose betweenness centrality is larger than the betweenness centrality of the intercluster edge e_1^{out}. This happens because EBC is community scale dependent.

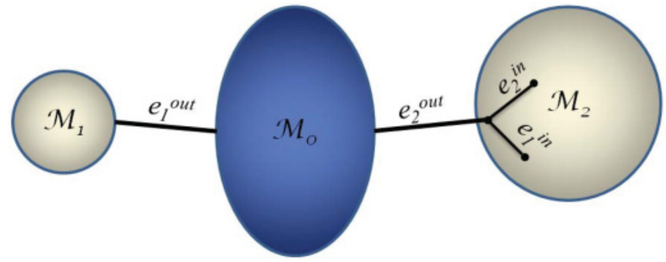

FIG. 1. (Color online) A network division with at least three communities. M_1 and M_2 are two communities which are connected to the rest of the network (M_0) each by one edge, i.e., e_1^{out} and e_2^{out}, respectively. e_2^{out} has two neighboring edges inside M_2.

Recently, we have introduced a new measure, namely common neighbor ratio (CNR) [15] which does not depend on the size of the communities and is computed locally for each edge. For a pair of vertices i and j, CNR, denoted by C_{ij}, is defined as

$$C_{ij} = \frac{2\left(A_{ij} + \sum_k A_{ik}A_{jk}\right)}{\sum_k A_{ik} + \sum_k A_{jk}}, \tag{8}$$

where A is the adjacency matrix, i.e., $A_{ik} = 1$, if there is a connection between i and k, while $A_{ik} = 0$ otherwise. This measure computes the percentage of the common neighbor vertices between i and j. Since the vertices inside a community are densely connected, CNR is supposed to be higher for such vertices and lower for vertices which are not in the same community. The denominator of the right hand side of Eq. (8) locally normalizes the number of common neighbors. Again, like EBC, although CNR contains some information about community structures it can be misguiding in some cases. In particular, on the boundaries of the communities, one can find neighboring nodes from different communities that have rather high CNR scores.

Recalling the fact that EBC is a global measure and CNR is a local measure, one can imagine that they contain some complementary information about the community structure of the network. Accordingly, our proposed weighting scheme is a combination of EBC and CNR which is as follows:

$$W_{ij} = \begin{cases} \dfrac{b_{ij}^{-\alpha} \cdot C_{ij}^{\beta}}{\sum\limits_{\substack{k,m \\ k \neq m}} b_{km}^{-\alpha} \cdot C_{km}^{\beta}} & \text{if } A_{ij} = 1 \\ 0 & \text{if } A_{ij} = 0 \end{cases} \quad ; \quad \alpha,\beta > 0, \tag{9}$$

where b_{ij} and W_{ij} are the normalized EBC, i.e., $b_{ij} = B_{ij}/B_{max}$, and assigned weight for the existing edge between vertices i and j, respectively. Both of the factors in the numerator of Eq. (9) are trying to distinguish the type of the edges by strengthening the intracluster edges and weakening the intercluster ones. The denominator of Eq. (9) normalizes the weights. For a pair of α and β values, the algorithm assigns weights based on Eq. (9) to the edges and then the CNM algorithm is used that joins the community pair that maximizes the modularity of the weighted network and, in the end, we choose the partition that has the highest modularity. It is clear that having a community configuration, the corresponding modularity measure of weighted and unweighted networks are not the same. We call the modularity measure that is calculated based on the weighted network weighted modularity while the other one is called unweighted modularity.

In particular, having a pair of α and β, the proposed algorithm has the following steps:

(a) The input is a pair of α and β and an unweighted graph, $G(V,E)$ where V and E are the set of vertices and edges of graph G, respectively.

(b) Weight the network based on weighting scheme (9).

(c) Assume that each node is a separate community.

(d) While there is more than one community, do the following:

- (i) Calculate ΔQ for all pairs of communities.
- (ii) Select the corresponding pair of communities that has the highest ΔQ.
- (iii) Join the selected pair of communities.
- (iv) Store the current partition as well as corresponding weighted modularity measures.

(e) Output the partition that results in the highest weighted modularity measure.

The runtime of the EBC calculation is $O(mn)$, while it shrinks to $O(n^2)$ for sparse graphs [30]. Therefore the runtime of the proposed weighting computation is $O(n^2)$ for sparse graphs.

The proposed algorithm has two parameters, namely α and β that control the level of contribution of the EBC and CNR parameters, respectively. It is expected that different pairs of α and β result in different outcomes. In fact, as we will show later in this paper, when the parameters of the proposed weighting scheme are well tuned, high modularity partitions are obtained at an acceptable computational effort.

In general, the optimal values for α and β parameters can be found by brute-force search or, preferably, by the Nelder-Mead method [31]. Figure 2 shows the unweighted modularity measure as a function of α and β for the United States (US) Football network [5]. Here, the unweighted modularity is chosen for the sake of comparision.

Finding the optimal modularity partition by brute force search is computationally very expensive and similarly the optimization of α and β is a time consuming task which has its own problems such as trapping in local maxima. Thus it is

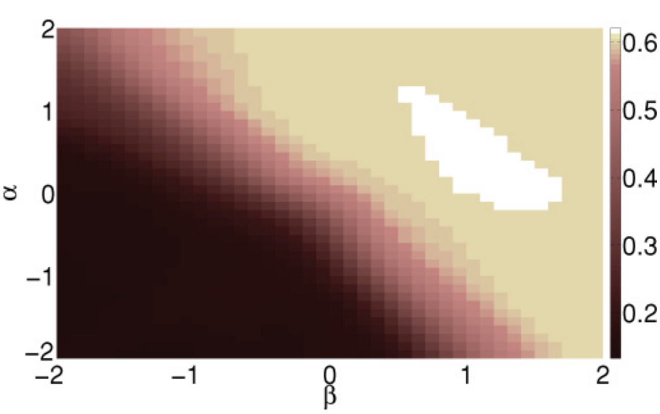

FIG. 2. (Color online) Modularity for the US Football network after weighting and applying the CNM algorithm for different values of α and β. The values are for the unweighted network as described in the text. The brighter the region, the higher the modularity value.

desired to determine α and β in a much more efficient way. In [14,15], we showed that the simple choice of $\alpha = \beta = 1$ gives acceptable results on benchmark networks. Here, based on the intuition behind our weighting scheme, we propose another heuristic approach for tuning the parameters in a computationally efficient way which gives very close results to the optimal value of modularity. We will first give the heuristic reasoning that leads to the proposed algorithm.

Consider the weights (9) of all edges in the network. They constitute a certain distribution in $\lfloor 0,1 \rfloor$ that depends on α and β. In a high modularity partition of the graph, the edges with large weights are likely to be intercluster edges and those with low weights intracluster edges. If the distribution of the weights is narrow, then different partitions have rather close modularity values, which is detrimental to modularity optimization. Therefore it seems natural to choose the parameters α and β that maximize the variance of the weight distribution. While this choice gives good results, the values of α and β chosen by the following heuristics have produced still better results.

It is a two-step procedure. In the first step, we find β via a line search such that

$$\beta = \arg\max \sum_{\substack{i>j \\ A_{ij}=1}} \gamma_{ij}\left(C_{ij}^{\beta} - \langle C^{\beta}\rangle\right)^2, \tag{10}$$

where $\gamma_{ij} = b_{ij}\, C_{ij}^{-\beta}$ and $\langle C^{\beta}\rangle$ denote the average of C_{ij}^{β} over all i and j with $A_{ij} = 1$. Roughly speaking, the values of γ are high for the intercluster edges and low for the rest, which is as desired. Finding β, the second step tries to find α such that

$$\alpha = \arg\max \sum_{\substack{i>j \\ A_{ij}=1}} \gamma_{ij}\left(b_{ij}^{\alpha} - \langle b^{\alpha}\rangle\right)^2, \tag{11}$$

where in this case $\gamma_{ij} = b_{ij}^{-\alpha}\, C_{ij}^{\beta}$ and $\langle b^{\alpha}\rangle$ is the average of b_{ij}^{α} over all i and j with $A_{ij} = 1$. Thus the values of γ are strengthening the intercluster edges and weakening the rest. Note that the optimization of α and β are not independent. In the rest of paper, we call this heuristic *max weighted variance* (MWV).

Many real-world networks are weighted. In such networks, weights incorporate invaluable information about the structure of the network [32]. Accordingly, we have also extended our method for weighted networks where the weighting scheme remains the same while the definitions of the terms of Eq. (9) are modified. Since EBC on weighted networks is derived from paths with the lowest costs, it may happen that some powerful edges, i.e., edges with high coupling value, gain zero EBC score. This phenomenon may lead to network disconnectivity. To avoid such a problem, we define $b_{ij} = (B_{ij} + B_{min})/B_{max}$, where B_{min} is the minimum nonzero edge betweenness centrality of the network. The latter term maintains the network connected while keeping the EBC measure effective.

Furthermore, we define C_{ij} for weighted networks as

$$C_{ij} = \frac{2S_{ij}\left(1 + \sum_k S_{ik}S_{jk}\right)}{\sum_k S_{ik} + \sum_k S_{jk}}, \tag{12}$$

where S_{ij} is the original weight of the edge connecting i and j nodes. It is obvious that when $S_{ij} = 0$, C_{ij} and consequently W_{ij} are both equal to zero.

It should be emphasized again that the proposed approach for finding proper pairs of α and β is an intuitive heuristic and there is no rigorous guarantee for optimal outcome. It is obvious that in applications where accuracy is vital, a more complex and costly search for α and β, e.g., Nelder-Mead, should be performed.

V. RESULTS

In this section, we will present the change in the ratio R on benchmark networks before and after applying our proposed weighting scheme. Indeed, we show that our weighting is able to reduce this ratio and meet the criteria discussed in Sec. II. Also, we will evaluate the whole procedure, i.e., weighting plus CNM on different networks including computer-generated and real-world data networks.

A. Computer-generated benchmark networks

The computer-generated graphs that we have used to evaluate our claims of this paper include the Girvan Newman (GN) bechmark graphs [5], and the Lancichinetti Fortunato Radicchi (LFR) graphs [33], and the ring of cliques networks [10] which consist of a ring of k cliques connected by one edge.

The popular GN bechmark graphs are used extensively in the literature for the sake of performance comparison between different community detection methods. These graphs consist of 128 vertices in four known communities, each of which has 32 vertices. Every intracluster edge is independently chosen with a probability $p_{\rm in}$ and every intercluster edge with a probability $p_{\rm out}$. Therefore the expected intracluster degree at each vertex is $z_{\rm in} = 31p_{\rm in}$ and the expected intercluster degree is $z_{\rm out} = 96p_{\rm out}$. Furthermore, we varied $p_{\rm in}$ and $p_{\rm out}$ in such a way that $z_{\rm out}$ varies from 0 to 8 in steps of 0.5 and $z_{\rm in} + z_{\rm out} = 16$ on average.

In [33], Lancichinetti, Fortunato, and Radicchi recently introduced a new set of computer-generated benchmark graphs, so called LFR. The LFR benchmark graphs give the users the opportunity to make more realistic graphs by tuning the relevant parameters. Such graphs are scale free with arbitrary scale factor where the degree of the vertices and the numbers of vertices in communities follow distinct power-law distributions with user defined exponents. The total number of vertices, average degree of vertices, maximum vertex degree, and finally, the mixing parameter (μ) are the other parameters that need to be set by the user. The latter varies within $\lfloor 0,1 \rfloor$ and determines the level of the fuzziness of the clusters in the network. The larger the μ, the more fuzzy the clusters. Also, the minimum and maximum of the community sizes are arbitrary to set. It is worthwhile to note that the GN benchmark is a special case of the LFR benchmark.

In the following, we will show by simulation that our proposed weighting scheme reduces R to achieve the goals discussed in Sec. II. In Fig. 3(a), the average ratio R before and after applying our weighting scheme to GN benchmark graphs is represented. To show the individual effects of EBC (CNR), we have also plotted R for when we weight each edge ij by b_{ij}^{-1} (C_{ij}). As can be seen, both reduce the average R, while the proposed heuristics do so more. The same results are obtained for the LFR benchmark graphs with the following properties. Each graph has 1000 vertices with average degree of 20 and maximum degree of 50. The power-law exponents of distributions of vertex degree and the number of vertices in each community are 2 and 1, respectively. The mixing parameter is varied between 0 and 0.5 where the average R reaches 2 for the unweighted graph. The results are presented in Fig. 3(b).

From Fig. 3, it is clear that the proposed weighting scheme has reduced the average ratio R considerably. This ratio has been reduced more than four times for LFR benchmark graphs. Also, it has been reduced more than two times for GN benchmark graphs when $z_{\rm out} \leqslant 6.5$. The results of two heuristics are close to each other but the simple choice of $\alpha = \beta = 1$ gives lower average R in both cases. As we will see later, it does not always mean that this weighting gives better results in terms of the accuracy of the found community. One reason for this phenomenon could be that this weighting may increase (decrease) the weight of a portion of intercommunity (intracommunity) edges rather than all of them.

As mentioned before, different metrics have been devised to approximate the accuracy of the community detection algorithms on the test and benchmark networks. The first one which was introduced by Girvan and Newman [5] was the

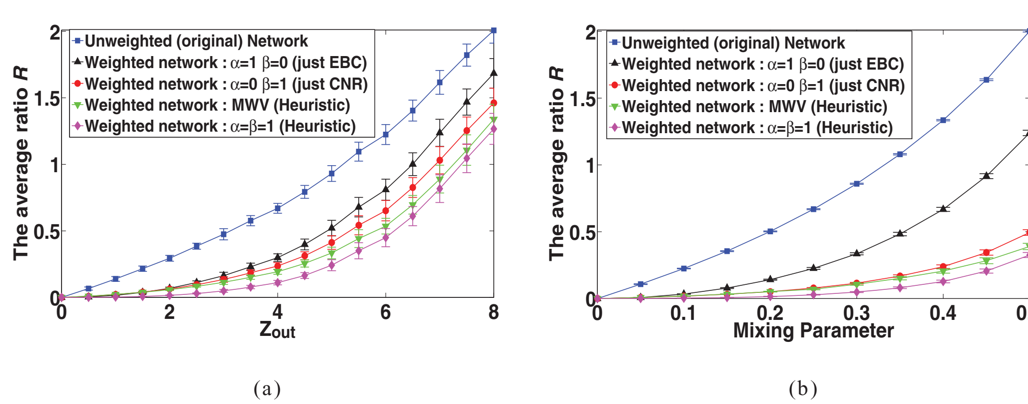

FIG. 3. (Color online) The change in the average ratio R, before and after applying the proposed weighting for (a) GN (b) LFR benchmark graphs. Each point is an average over 50 distinct graphs.

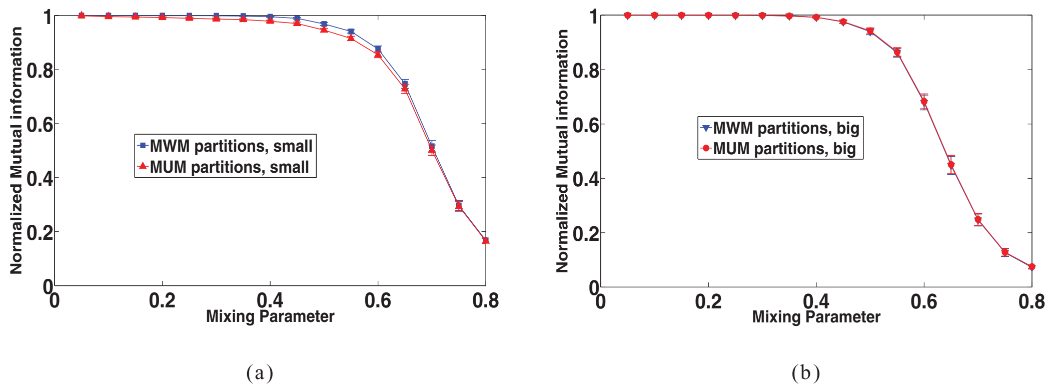

FIG. 4. (Color online) Mutual information for MWM and MUM partitions after applying max weighted variance heuristic weighting and CNM. The test is done for (a) small and (b) big community sizes of LFR benchmark graphs with $n = 1000$ vertices. Other parameters are described in the text. Each point is an average over 50 realizations.

fraction of the nodes classified correctly. Although it is a good measure for a GN benchmark, it is not well performing in the cases where two or more true communities are identified as one sole community. In addition, there exist some cases where the value of this measure changes abruptly by reassigning a vertex to a new community. Therefore some more robust metrics have been recently used to evaluate the algorithms. One of these metrics which is extensively examined in information theory is normalized mutual information (mutual information) that was introduced by Danon *et al.* in [7]. This measure takes values in $\lfloor 0,1 \rfloor$ and estimates how much the true and found communities have information in common. When they are perfectly matched, the mutual information is 1. Otherwise, the less there is a match between the found and true communities, the smaller is the value of the mutual information. We have used this measure to assess the accuracy of the results on benchmark networks. The mutual information between the partitions of the true communities X and the found communities Y is mathematically defined as

$$I(X,Y) = \frac{-2\sum_{i=1}^{c_X}\sum_{j=1}^{c_Y} N_{ij}\ln(nN_{ij}/N_{i.}N_{.j})}{\sum_{i=1}^{c_X} N_{i.}\ln(N_{i.}/n) + \sum_{j=1}^{c_Y} N_{.j}\ln(N_{.j}/n)}, \qquad (13)$$

where c_X and c_Y are, respectively, the number of communities in partitions X and Y. In this equation N is the confusion matrix where the rows and columns correspond to the true and found communities, respectively. The element N_{ij} is defined as the number of common nodes in the true community i and the found community j. Also, $N_i.$ and $N_j.$ denote the sum over the ith row and the jth column, respectively.

In order to show the effect of our weighting on modularity, we have done the following experiment. We have weighted the graph using our max weighted variance heuristic and then have applied the CNM algorithm. Then, we have chosen two distinct partitions such that one of them has the maximum weighted modularity (MWM) while the other one is the partition with the maximum unweighted modularity (MUM). For this experiment, two different LFR benchmark graphs have been engaged. These LFR graphs have the same properties of the last experiment and their distinction is the size of their communities. The number of the vertices of each community in the first graph set is between 10 and 50, while these values for the second set are 20 and 100, respectively. The first graph set has small community sizes and it is shown in [19] that it has a modularity resolution limit problem for the modularity optimization methods. On the other hand, the second data set has more diversity on the size of clusters and the modularity resolution limit has less effect on detecting the communities [19]. We denote, respectively, the first and second graph sets by *small* and *big* in the rest of the paper. The outcome of this experiment is illustrated in Fig. 4. As can be seen, the MWM partitions are more accurate than the MUM

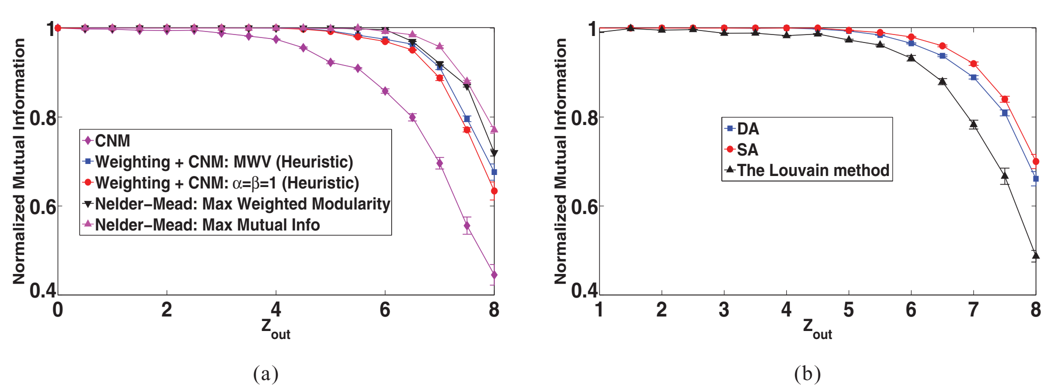

FIG. 5. (Color online) The result of (a) CNM, our heuristic weightings plus CNM, (b) DA, Louvain, and SA methods on GN benchmark graphs. Each point is averaged over 50 realizations.

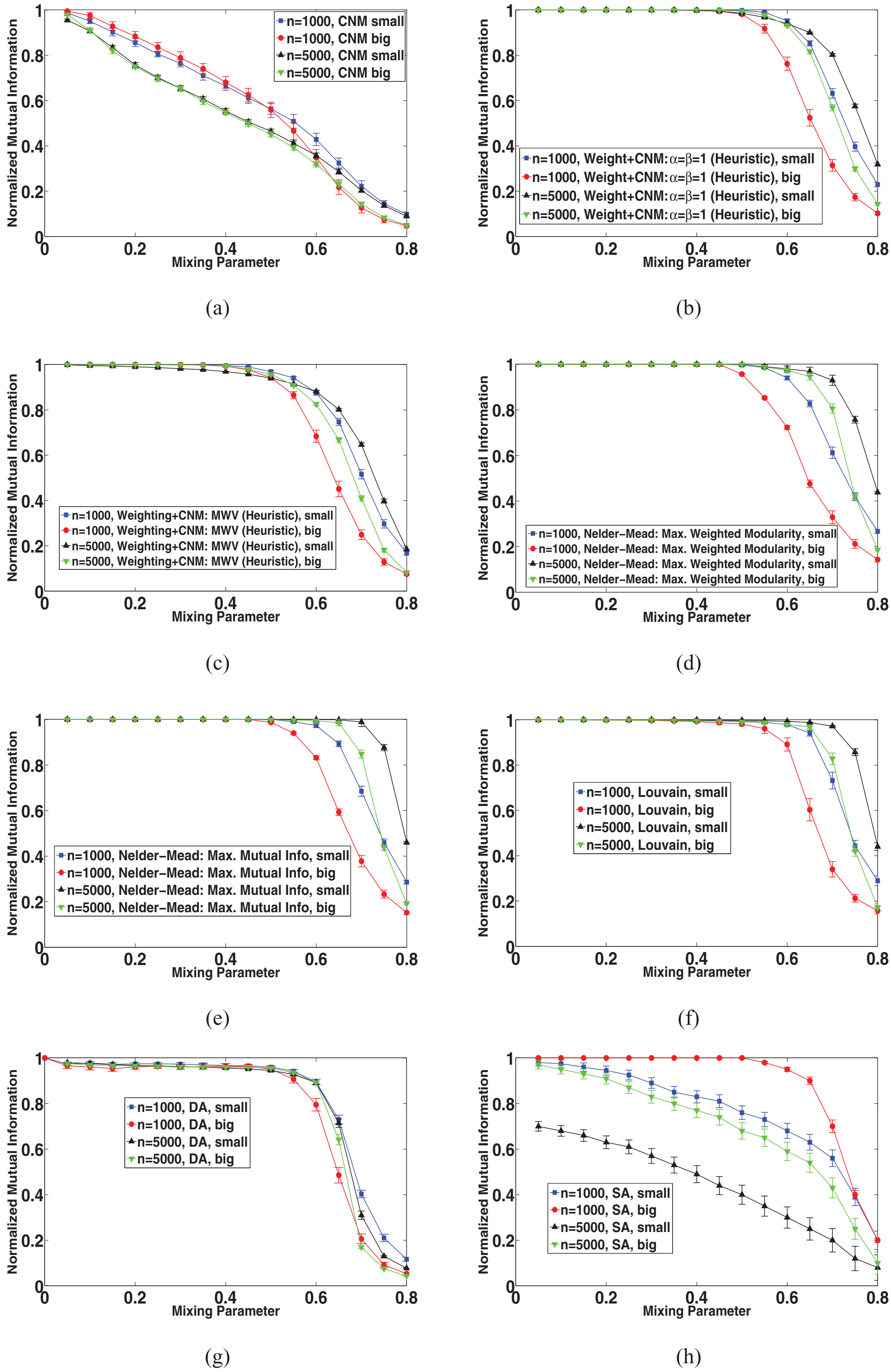

FIG. 6. (Color online) Test of algorithms (a) CNM, (b),(c) our proposed heuristic weightings and CNM, (d) maximizing weighted modularity using Nelder-Mead, (e) maximizing mutual information using Nelder-Mead, (f) Louvain method, (g) DA method, and (h) SA method on LFR benchmark graphs with $n = 1000$ and $n = 5000$ vertices and small and big community sizes. Each point is averaged over 50 realizations.

ones, i.e., MWM partitions have higher mutual information for the small graph set [Fig. 4(a)], while they are similar for the big graph set [Fig. 4(b)]. In fact, one can trust more the weighed rather than the unweighted modularity.

Figure 5 presents the results of different methods, including CNM, our heuristic weightings plus CNM, the simulated annealing method [34], the Louvain method, and the DA method on GN benchmark graphs. The simulated annealing (SA) method is one of the best methods that can disclose the communities of GN benchmark graphs [19] but in a very high computational cost. Also, the DA method is proved to be reliable on modularity optimization with lower computational complexity than SA. As can be seen, the results of our heuristics are much better than CNM and the Louvain method, better than DA, and very close to the result of SA while our runtime is much lower than that.

In [11], it is well described that the high modularity partitions do not always correspond to the best partition, and they are sometimes even far from optimal. In Fig. 5(a), we have sketched the best mutual information result we could get by optimizing using the Nelder-Mead method. It clearly outperforms the result obtained by SA in terms of accuracy of the found communities. It is worthwhile to note that these optimal mutual information partitions do not correspond to the highest modularity ones and it confirms what is argued in [11].

We have also tested our approach on the LFR benchmark graph, which is more general than the GN benchmark graph. Four groups of graphs are considered for this purpose which are the combination of graphs with 1000 and 5000 vertices with small and big community sizes. Each vertex of the graphs has an average degree of 20 and maximum degree of 50. The power-law exponents of distributions of vertex degree and the number of vertices in each community are 2 and 1, respectively. The results are drawn in Fig. 6.

The results of CNM, DA, and SA algorithms are illustrated in Figs. 6(a), 6(g), and 6(h), respectively. All these methods try to optimize the modularity function in order to resolve the communities. By looking at Figs. 6(b) and 6(c), one can find out that both of our heuristics have significantly improved the performance of CNM and they outperform the DA and SA methods. Indeed, it proves our claim that optimizing the weighted version of the modularity function leads to more accurate decomposition of the graph. Figure 6(h) shows that the SA method, which tries to find the maximal modularity partition, substantially suffers from modularity resolution limit. Since the resolution limit depends on L, it worsens the accuracy of community detection of larger graphs in the case that the community sizes are kept similar. It is clearly illustrated in Figs. 6(a) and 6(h) for CNM and SA methods, respectively. However, by applying our weighting scheme, the accuracy is improved when we increase L. This is also evidence that our weighting scheme mitigates the resolution limit problem.

In addition, we have plotted the results of optimizations of Nelder-Mead over α and β parameters in Figs. 6(d) and 6(e). It is clearly shown that the results of weighted modularity and mutual information optimizations are very close to each other and of course better than the heuristics. These results are also very close to the Louvain method, i.e., Fig. 6(f), which is proved to have an excellent performance on such benchmark graphs [19]. It can be concluded that by optimizing the weighted modularity function, significant results can be obtained.

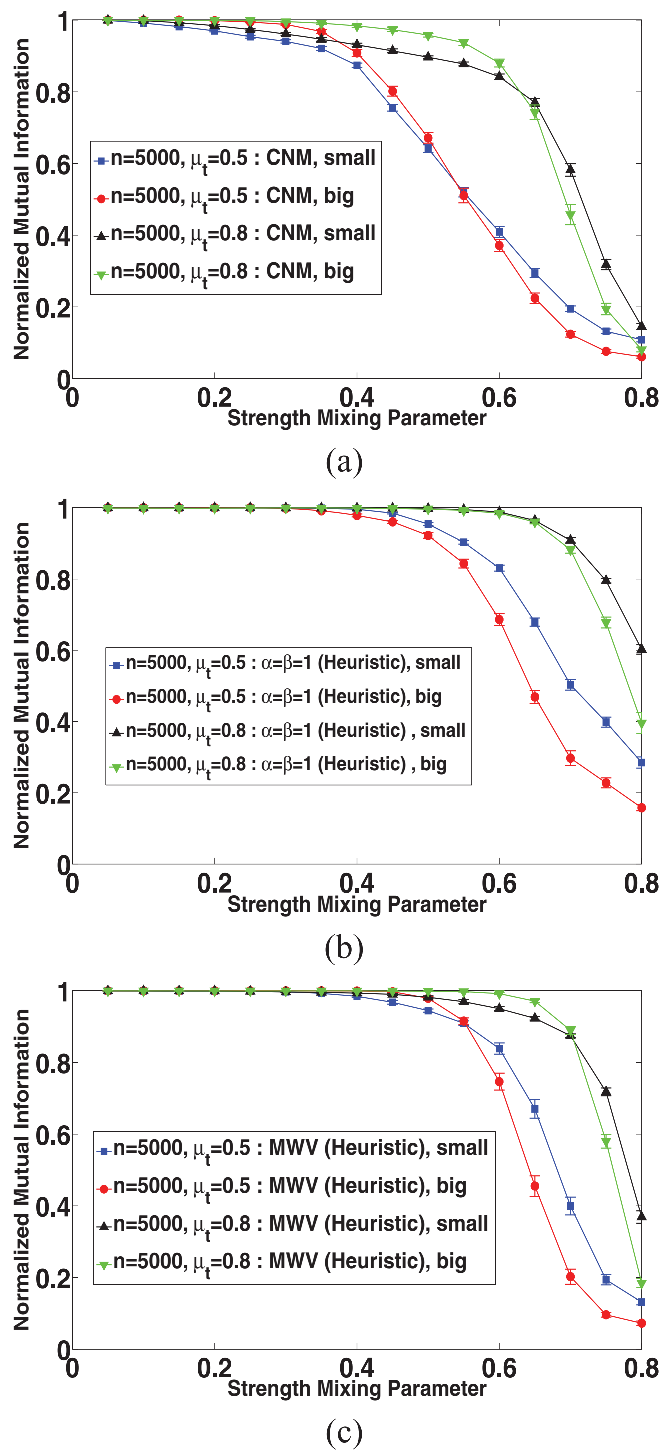

FIG. 7. (Color online) Test of algorithms (a) CNM and (b), (c) our proposed heuristic weightings and CNM on weighted LFR benchmark graphs with $n = 5000$ vertices, small and big community sizes, and the topological mixing parameter $\mu_t = 0.5$ and $\mu_t = 0.8$. Each point is averaged over 50 realizations.

Furthermore, we have tested our proposal on weighted LFR benchmark graphs which are introduced in [35]. Such graphs have two different mixing parameters. The first one, which is similar to what has been used so far, is dedicated to the topology (μ_t), while the second adjusts the strengths of the edges (μ_w). Indeed, it is on average the percentage of

TABLE I. The results of CNM and the proposed weighting heuristics on a ring of cliques graph with 1000 cliques and ten vertices per clique.

Approach	CNM	MWV (heuristic) + CNM	$\alpha = \beta = 1$ (heuristic) + CNM
NOIC	232	1000	1000
Q	0.9905	0. 9972	0.9990
Mutual information	0.88	1	1
R	0.044	0.0036	5.71×10^{-10}

the node's strength, i.e., the sum of the weights of the edges connected to the node, which lies on the intercluster edges. In our simulations, we fixed the topological mixing parameter and then ran our method on such graphs with different strength mixing parameters. The topological mixing parameter values are chosen as 0.5 and 0.8, respectively. Another parameter on these graphs is the exponent of the distribution of the strength, which was set to 1.5 in our simulations. The rest of the parameters were set as they were for unweighted LFR graphs.

The results of our proposed method and CNM on weighted LFR networks with $n = 5000$ are shown in Fig. 7. It is clearly shown that our weighting scheme improves the accuracy of the resolved communities compared to CNM. Comparing our results with what is obtained in [19], it is obvious that our method is among the best proposed methods for unveiling the communities of weighted networks.

Our last experiment on computer-generated graphs is done on a ring of cliques networks. It can be easily shown that for such networks, modularity optimization algorithms are not always able to find the cliques as communities due to the resolution limit problem. Indeed, there are some cases where the identified communities consist of more than one clique, which is undesired in reality. For example, the results of CNM and our proposal for the case of such a graph with 1000 cliques with ten vertices per clique, i.e., $n = 10\,000$, is summarized in Table I. As can be seen, the number of identified communities (NOIC) that is found by CNM is 232 out of 1000. On the other hand, both of our proposed heuristics are able to resolve all the cliques perfectly. In fact, the ratio becomes almost negligible.

VI. REAL-WORLD DATA NETWORKS

We evaluated our approach for some real-world data networks, shown in Table II. We believe that the partition with the maximum weighted modularity is more similar to the true partitions for a given network; but, for the sake of comparison with other methods, the maximum unweighted modularity function that is obtained by our proposal is presented. In addition, the best result, i.e., maximum unweighted modularity, that can be obtained when α and β were determined by Nelder-Mead optimization is given. In order to compare our method with the rest, we have provided the best published results, which were obtained by applying computationally more complex algorithms. As can be seen, the result heuristics are marginally smaller than the modularity with α and β found by the Nelder-Mead optimization, but they are always very close to the best published values, but at a smaller computational cost.

Normally, the true structure of real-world data networks is not known. However, there are a few examples available that have a known community structure. The US Football network is one of them. The data of this network were gathered by Girvan and Newman. It is a representation of the schedule of Division I American Football games in the 2000 season in the Unites States. The vertices are the teams which are in 12 distinct groups. The vertices are joined by an edge when there are regular-season games between them. Since the actual community structure of this network is available, one can evaluate the accuracy of the algorithms by comparing their mutual information index. Accordingly, we have applied our weighting scheme and compared the mutual information of both MWM and MUM partitions using Nelder-Mead optimization. The result shows that the MUM partition corresponds to ten communities with mutual information equal to 0.890 while these values for MWM partition are 11 and 0.911, respectively. Therefore we can conclude that our weighting has increased the accuracy of the community identification procedure. It is worthwhile to mention that it is also possible to find better network community division by modularity based algorithms

TABLE II. This table is a summary of results for some real-world data networks. The modularity of these networks for our proposed approach when $\alpha = \beta = 1$, for α and β found by Nelder-Mead optimization, and for α and β found by the maximum weighted variance heuristic are shown. Also, the modularity of CNM and the best modularity values which are published up to now are given for them.

Network		CNM	Best published		Q after applying weighting + CNM		
Name	Ref.	Q	Best published Q	Source	Nelder-Mead	$\alpha = \beta = 1$	MWV heuristic
Zachary's Karate Club	[36]	0.381	0.420	[37]	0.416	0.416	0.415
US Football	[5]	0.567	0.606	[37]	0.605	0.604	0.604
Les Miserables	[38]	0.501	0.561	[37]	0.560	0.532	0.539
Dolphin Social Network	[39]	0.496	0.531	[37]	0.529	0.518	0.521
Email	[40]	0.503	0.579	[37]	0.574	0.549	0.563
Jazz	[41]	0.439	0.446	[37]	0.444	0.437	0.442
PGP-key signing	[42]	0.849	0.878	[43]	0.880	0.843	0.872

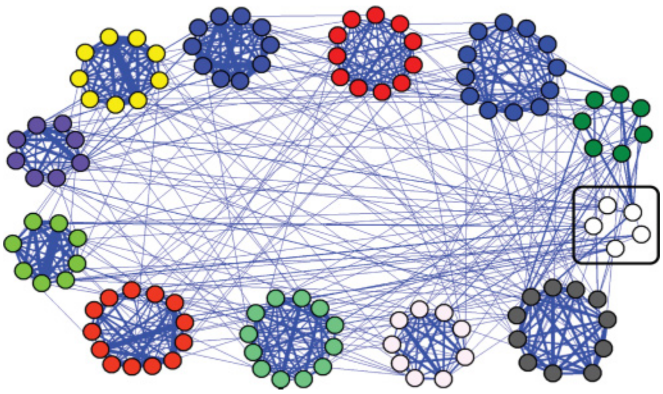

FIG. 8. (Color online) Configuration of the US Football network. Each node is a team and each set of nodes encircled represents a group. The edge widths are proportional to the corresponding MWM weight. The group surrounded by the solid box is a special group which is discussed in the text.

if the resolution of modularity, i.e., the scale at which modules are looked at, is properly chosen.

In Fig. 8, the effect of MWM weighting on this network is illustrated. The vertices represent the teams and the set of vertices encircled are the groups labeled by Girvan and Newman. The widths of edges are scaled proportional to their corresponding weights. As can be seen, as an effect of our proposed weighting scheme, most of the intracommunity edges are reinforced while the majority of the intercommunity edges are weakened. The only exception is for the community which is surrounded by the solid box. In fact, contrary to the intuition behind community definition, the vertices of this group have few connections with each other, i.e., they are loosely coupled within the cluster, while they have more links with the members of the other groups, i.e., they are rather strongly coupled with the rest. This is why our approach and many other community detection algorithms could not identify this specific group.

VII. CONCLUSION

Summarizing, we have analyzed how applying an appropriate weighting scheme can mitigate the resolution limit and extreme degeneracy problems of community detection methods that are based on modularity optimization. Also, we have introduced a weighting scheme based on EBC and CNR, which improves the performance of the Newman-Fast algorithm considerably. As was mentioned, the tunable parameters of the algorithm, i.e., α and β, need to be adjusted for the network at hand. Furthermore, to get rid of complexity of optimization algorithms, two heuristic methods for finding the parameters of the weighting scheme are introduced. The first heuristic is the simple choice of $\alpha = \beta = 1$ and the second one follows a line search to determine α and β such that the weighted variances of b^{α} and C^{β} over all the edges are maximized, respectively.

The results of the experiments show that the proposed method has a very good performance on the benchmark graphs as well as real-world data networks. It not only ameliorates the problems of modularity but also enhances the modularity optimization. Indeed, when we apply our proposed weighting and run the CNM for a given network, the results show that the MVM partition is more accurate while the modularity of the MUM partition is significantly higher than CNM and very close to the maximum modularity found for such a network. We admit that the computational complexity of the proposed weighting is not that low but it is comparable with many existing procedures. Furthermore, our primary goal in this paper was to investigate the possibility of improvement of a particular functionality, i.e., community detection, using a proper weighting scheme. In order to decrease the complexity of the proposed algorithm, as a future work we will use an estimation of EBC [44,45] instead of the original and investigate what precision of EBC estimation our proposed algorithm tolerates.

ACKNOWLEDGMENTS

The authors are thankful to M. Newman and A. Arenas for providing the network data and their codes. We are also grateful to S. Fortunato for providing the code to produce LFR benchmark graphs.

[1] S. H. Strogatz, Nature (London) **410**, 268 (2001).
[2] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, Phys. Rep. **424**, 175 (2006).
[4] A. Ajdari Rad, A. Khadivi, and M. Hasler, IEEE Circuits Syst. Mag. **10**, 26 (2010).
[5] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).
[6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. (Morgan Kaufmann, San Francisco, 2006).
[7] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech. (2005) P09008.
[8] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, and S. Stramaglia, Chaos **17**, 023114 (2007).
[9] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[10] S. Fortunato and M. Barthèlemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).
[11] B. H. Good, Y.-A. de Montjoye, and A. Clauset, Phys. Rev. E **81**, 046106 (2010).
[12] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).
[13] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).
[14] A. Khadivi, A. Ajdari Rad, and M. Hasler, in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), Paris* (2010), p. 3777.
[15] A. Khadivi and M. Hasler, in *International Conference on Communications (ICC), 2010 IEEE*, Cape Town South Africa (2010), p. 1–4.

[16] V. D. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech. (2008) P10008.
[17] J. M. Hofman and C. H. Wiggins, Phys. Rev. Lett. **100**, 258701 (2008).
[18] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).
[19] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 056117 (2009).
[20] A. Ajdari Rad, M. Jalili, and M. Hasler, Chaos **18**, 037104 (2008).
[21] M. Jalili, A. A. Rad, and M. Hasler, Phys. Rev. E **78**, 016105 (2008).
[22] A. Ajdari Rad, M. Hasler, and M. Jalili, LOG J IGPL **18**, 670 (2010).
[23] R. Olfati-Saber, J. A. Fax, and R. M. Murray, Proc. IEEE **95**, 215 (2007).
[24] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips, e-print arXiv:0903.1072 (2009).
[25] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[26] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **70**, 025101 (2004).
[27] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. USA **101**, 2658 (2004).
[28] L. C. Freeman, Sociometry **40**, 35 (1977).
[29] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, Phys. Rev. E **75**, 045102(R) (2007).
[30] U. Brandes, J. Math. Sociol. **25**, 163 (2001).
[31] J. A. Nelder and R. Mead, Comput. J. **7**, 308 (1965).
[32] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, Proc. Natl. Acad. Sci. USA **101**, 3747 (2004).
[33] A. Lancichinetti, S. Fortunato, and F. Radicchi, Phys. Rev. E **78**, 046110 (2008).
[34] R. Guimera and L. A. Nunes Amaral, Nature (London) **433**, 895 (2005).
[35] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 016118 (2009).
[36] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).
[37] G. Agarwal and D. Kempe, Eur. Phys. J. B **66**, 409 (2008).
[38] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing* (Addison-Wesley Professional, Reading, MA, 1993), p. 592.
[39] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behav. Ecol. Sociobiol. **54**, 396 (2003).
[40] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, Phys. Rev. E **68**, 065103 (2003).
[41] P. M. Gleiser and L. Danon, Adv. Complex Syst. **6**, 565 (2003).
[42] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, Phys. Rev. E **70**, 056122 (2004).
[43] P. Schuetz and A. Caflisch, Phys. Rev. E **77**, 046112 (2008).
[44] D. Bader, S. Kintali, K. Madduri, and M. Mihail, *Algorithms and Models for Web-Graph* (Springer, New York, 2007), p. 124.
[45] U. Brandes and C. Pich, Int. J. Bifurcation Chaos **17**, 2303 (2007).