Research Article

A Modularity Degree Based Heuristic Community Detection Algorithm

Dongming Chen, Dongqi Wang, and Fangzhao Xia

Software College, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Dongqi Wang; wangdq@swc.neu.edu.cn

Received 11 October 2013; Revised 6 January 2014; Accepted 7 January 2014; Published 25 February 2014

Academic Editor: Yuncai Wang

Copyright © 2014 Dongming Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A community in a complex network can be seen as a subgroup of nodes that are densely connected. Discovery of community structures is a basic problem of research and can be used in various areas, such as biology, computer science, and sociology. Existing community detection methods usually try to expand or collapse the nodes partitions in order to optimize a given quality function. These optimization function based methods share the same drawback of inefficiency. Here we propose a heuristic algorithm (MDBH algorithm) based on network structure which employs modularity degree as a measure function. Experiments on both synthetic benchmarks and real-world networks show that our algorithm gives competitive accuracy with previous modularity optimization methods, even though it has less computational complexity. Furthermore, due to the use of modularity degree, our algorithm naturally improves the resolution limit in community detection.

1. Introduction

Many of the complex systems in nature and society can be thought as networks composed of nodes and edges, the Internet, metabolic networks, food web, neural networks, logistics and supply chain networks, industry cluster networks, and social organization networks are some of them [1–3]. With the in-depth study of the physical sense and mathematical properties of the network, it was discovered that many networks have a common nature, which is called the community structure. It refers that the whole network is constituted by multiple groups or clusters. Connections between nodes in the same group are relatively dense, while connections between nodes in different groups are sparse [4, 5]. Discovering community structure in networks is an important way to analyze the structure, features and functions of real networks, and it has a wide range of applications in the field of natural sciences, engineering technology, economy and sociological research.

Community detection methods usually share the metaprocedure of “trying to expand or collapse the node partitions of a given graph G , in order to optimize a given quality function, stopping when no increment is possible”

[5]. There are some commonly known quality functions that are able to quantify whether a set of entities are more related than expected and thus can be considered as a community [7–9]. Modularity is the most commonly known measure for community structure introduced by Clauset et al. [8]; the larger the value of modularity, the more accurate a partition into communities is, so it provides a way to determine if a certain description of the graph in terms of communities is more or less accurate. Due to the fact that the space of possible partitions grows faster than any power of the system size, the computational complexity of searching optimal (largest) modularity is in the nondeterministic polynomial time hard (NP-hard) class. For this reason, researchers start to use heuristic strategies to restrict the search space while finding the optimal solutions [10–13]. Duch and Arenas proposed an extremal optimization (EO) strategy for modularity optimization [14], their experimental result showed that the EO algorithm is more efficient than Newman’s fast algorithm [11] and outperforms “all previous algorithms that exist in the literature.” Kumar and Jayaraman employed Group Search Optimizer (GSO) algorithm for community detection [7]. GSO algorithm is a new algorithm in the field of evolutionary computing. GSO is a process

of obtaining optimum solution in a search space, which is analogous to search for better clustering of a network to obtain the best Newman's modularity. Kumar's approach uses optimal modularity for candidate solutions' selection and follows the GSO process to detect community structure. Evaluation result showed that GSO algorithm is capable of identifying community structures in standard benchmark datasets. The community extracting method proposed by Blondel et al. [12] focuses on dealing with large networks, a mobile phone network of 2.6 million nodes, and a web graph of 118 million nodes which were used to prove the capability of the method. These modularity optimization based approaches are somehow efficient, but are all falling into the resolution limit which is decided by the modularity itself, which means that modularity optimization approach may fail to identify the modules smaller than a scale. A significant amount of work has been carried out to solve the problem of resolution limit [15–17]. Li et al. proposed a quantitative function for community partition called modularity density [9], and it is superior to modularity. Li et al.'s work also proved modularity density is equivalent to the objective function of the kernel k-means and showed that optimizing the modularity density cannot only correctly identify the number of communities, but also can resolve detailed modules that existing approaches cannot achieve.

Here we propose a heuristic algorithm which adopts the local community structure as a basis to make it a competitively faster method and uses modularity density as measure function to ensure that it is an accurate algorithm. At the same time, our algorithm needs no prior conditions as input.

2. Definitions

As far, there is still no a strict definition about community structure. It is obvious that a reasonable community structure should have more connections in the same community than between different communities. This is to say, the community structure which has the most edges within communities and fewest edges between communities tends to be reasonable. Figure 1 shows an ordinary network diagram, where v is a community structure having n ($n = 5$) nodes, i is a node connected to v and $i \notin v$, and d equals to the degree of node i . Define $L(i, V)$ as the number of nodes in v which is directly connected to node i . It is to say, $L(i, V)$ is equal to the number of neighbors of node i which belong to community v . Now we try to add node i to community v to form a new community which is labeled as v' , then community v' will have $L(i, V)$ more inner edges than community v . Obviously, node i is supposed to be added to community v if $L(i, V)$ is large enough.

In order to quantitatively measure whether it is reasonable for node i to join community v , we use modularity density D as the measure function.

Given a network $G = (V, E)$, V is the vertex set, E is the edge set, and A is the adjacent matrix of G . If V_i and V_j are two disjoint subsets of V and $V_i \cap V_j = \emptyset$, we further define $L(V_i, V_j)$ as the number of edges which has one node in V_i

FIGURE 1: An ordinary network diagram.

and the other node in V_j . Then $L(V_i, V_j)$ is the number of inner edges of V_i and $L(V_i, \bar{V}_i)$ is the number of outer edges of V_i ($\bar{V}_i = V - V_i$) the modularity density of community V_i is defined as

$$D_i = \frac{L(V_i, V_i) - L(V_i, \bar{V}_i)}{|V_i|}. \quad (1)$$

To facilitate memory, (1) is simplified as

$$D_i = \frac{E_{in}(V_i) - E_{out}(V_i)}{\text{Num}(V_i)}. \quad (2)$$

Here, $E_{in}(V_i)$ is equal to the number of inner edges of V_i , $E_{out}(V_i)$ is equal to the number of outer edges of V_i , and $\text{Num}(V_i)$ is equal to the number of nodes that are in V_i . Then, the modularity density of community v in Figure 1 can be expressed as $D_v = (E_{in} - E_{out})/n$, where E_{in} is the number of inner edges of community v , E_{out} is the number of outer edges of community v , and n is the number of nodes in community v .

Next we try to get a constraint function for our algorithm to decide whether it is reasonable to add this specified node i to the community v .

If node i has been added to the community v to form a new community v' , then the inner edges of v' are $E'_{in} = E_{in} + L(i, v)$, the outer edges of v' are $E'_{out} = (E_{out} - L(i, v)) + (d - L(i, v)) = E_{out} + d - 2L(i, v)$, and the D value of the new community v' is

$$\begin{aligned} D'_v &= \frac{E'_{in} - E'_{out}}{n+1} = \frac{(E_{in} + L(i, v)) - (E_{out} + d - 2L(i, v))}{n+1} \\ &= \frac{E_{in} - E_{out} + 3L(i, v) - d}{n+1}. \end{aligned} \quad (3)$$

According to the theory of D value, to make v' a more reasonable community than v , $D'_v > D_v$ should be satisfied; that is,

$$\frac{E_{in} - E_{out} + 3L(i, v) - d}{n+1} > \frac{E_{in} - E_{out}}{n}. \quad (4)$$

Inequality (5) can be simplified as

$$(3L(i, v) - d)n > E_{in} - E_{out}. \quad (5)$$

The key to our approach is to traverse the nodes in the network. For each node i , we see it as an independent community at first. We first try to get the community such as v , which obtains the largest number of neighbors of node i , then we attempt to add i to v and begin to judge whether it meets the constraint of inequality (4). Node i will be added to v if the constraint is satisfied. After several times of iteration, the community structure of network will remain unchanged, then we get the final network division result.

3. MDBH Algorithm and Analysis

In this section, we are going to discuss the details about our MDBH algorithm. Suppose the number of nodes of network G is N , the degree of node i is $d(i)$. We first initialize the network as follows.

- (1) Each node i in the network independently forms a community V_i , the inner edges of V_i are $E_{in}(V_i) = 0$, the outer edges of V_i are $E_{out}(V_i) = d(i)$, and the number of nodes in V_i are $\text{Num}(V_i) = 1$.
- (2) For each node i , we maintain a collection $\{L(i, V_j) \mid i, j = 1, 2, \dots, N\}$, which means the number of neighbors of i in community V_j . If node i and node j are neighbors, initialize $L(i, V_j) = 1$, otherwise, initialize $L(i, V_j) = 0$.
- (3) The initial number of iterations of our algorithm is $p = 0$.

Algorithm 1 shows what MDBH does during a round of search. The algorithm will terminate until none of the nodes in network changes the community it belongs to after iteration p , where p is the final value of iteration number.

The initialization takes $O(N)$ to deal with N nodes and N communities; operations on them are simple numeric expressions. Our algorithm needs p iterations in total. For each iteration, MDBH tries to find maximum $L(i, V_m)$ for current node and the quantities of $L(i, V_m)$ satisfying $L(i, V_m) \neq 0$ is $\sum_{i=1}^N d(i)$, so time complexity of "Step 1" takes up to $O(p \sum_{i=1}^N d(i))$. Obviously, the time complexity for calculating (5) is $O(pN)$. The first three update operations in Step 2 are linear operations, so the time complexity is $O(3pN)$. The fourth updating operation is a linear operation too. The number of node's neighbors is the degree of the node, so it takes $O(p \sum_{i=1}^N d(i))$. Now add these results obtained by analysis up, we get the time complexity of MDBH, which is $O(N + p \sum_{i=1}^N d(i) + pN + 3pN + p \sum_{i=1}^N d(i))$, which can be simplified as $O(pN + p \sum_{i=1}^N d(i))$. If we define E as the number of edges the target network has, then $\sum_{i=1}^N d(i) = 2E$, and the time complexity of our algorithm will be $O(pN + pE)$.

The value of p is a vital factor which affects the actual speed of our algorithm. The smaller the value of p , the faster the MDBH algorithm is. In each iteration we try to add the current node to community which contains the maximum number of neighbors of the node, and the community

which is larger is more likely to meet this condition, so the potential main communities which include most of nodes in network will be formed quickly after a small amount of iterations especially when the network has a good community structure. And if the community structure of the network to be analyzed is so bad that it consists of a large amount of small communities, then p will be larger; however, it is not meaningful to detect community structure in such networks. Therefore, the value of p is supposed to be small enough to make our algorithm fast.

4. Experimental Evaluation

In order to verify the performance of MDBH algorithm, we chose two commonly known benchmark networks: "Zachary club network" and "Dolphin social network." We also built our own networks: a character relationship network of the famous Chinese classic novel book "Romance of the Three Kingdoms" and computer-generated networks with large number of nodes and clear community structures. Experiments on these networks were conducted on a typical desktop computer with a 3.0 GHz Pentium 4 processor and 3 GB of RAM.

4.1. Zachary Club Network. Early 1970s, Zachary spent two years observing the friendships between members of a karate club in a university in America, and constructing a network of relationships between them [18]. It consists of 34 members of a karate club as nodes and 78 edges representing friendship. Due to disagreement between the club's administrator and the club's instructor, the club was split into two small ones, as shown in Figure 2.

The detailed experimental process on Zachary club network is to be described below. Firstly, we gained adjacent matrix of Zachary club and initialized it. Then we began to traverse the network from the 1st node to try to find the maximum of $L(1, V_m)$. There was more than one community met this condition. The community possessing the largest summation of member's degree was chosen. For the 1st node, we got $m = 3$. We judged the constraint condition in inequality (5) and the result was $-11 > -10$; this inequality was not satisfied, so we went on with the 2nd node. When $m = 1$, the value of $L(2, V_m)$ was the largest and (5) was satisfied, so the 2nd node was removed from V_2 and was added to V_1 , after this, we followed the update process of Step 2 in Algorithm 1. Continuing with the 3rd node until all 34 nodes in the network were handled, then we got the result of the first iteration as shown in Figure 3.

Apart from the node of 24th, 25th, and 26th, other nodes in the network were partitioned to two major communities V_{02} and V_{34} ; the community structure of Zachary club tends to disclose after the first iteration. The reason is that during the first iteration, the larger communities are more likely to contain the maximum number of neighbors of the current node, so nodes in smaller community tend to be merged into a larger community. Therefore, the two major communities became stronger after the first iteration.

Input: Adjacent matrix of network G

Output: Intermediate community detection result of G

Step 1. Get community V_n which the current node i belongs to, find the maximum value of $L(i, V_m)$, $m \in \{1, 2, \dots, N\}$. Suppose S to be the summation of the degree of all members' in one community.

IF $m = N$, go to *Step 3*.

IF more than one communities make $L(i, V)$ be maximum are found

Choose the community having the largest S as V_m . If more than one communities having the largest S are found, randomly choose one as V_m .

Step 2. Calculate and determine the inequality (5) of $(3L(i, V_m) - d(i))\text{Num}(V_m) > E_{\text{in}}(V_m) - E_{\text{out}}(V_m)$, if the inequality is true, then update the network as follows:

(1) Delete node i from the original community V_n , and merge node i into community V_m , update $\text{Num}(V_n) = \text{Num}(V_n) - 1$, $\text{Num}(V_m) = \text{Num}(V_m) + 1$.

(2) $E_{\text{in}}(V_m) = E_{\text{in}}(V_m) + L(i, V_m)$, $E_{\text{out}}(V_m) = E_{\text{out}}(V_m) + d(i) - 2L(i, V_m)$.

(3) $E_{\text{in}}(V_n) = E_{\text{in}}(V_n) - L(i, V_n)$, $E_{\text{out}}(V_n) = E_{\text{out}}(V_n) - d(i) + 2L(i, V_n)$.

(4) For each neighbor node k of node i , let $L(k, V_n) = L(k, V_n) - 1$,

$L(k, V_m) = L(k, V_m) + 1$.

Step 3. Traverse the network sequentially to get next node j , if j exists, go to *Step 1*, else go to *Step 4*.

Step 4. $p = p + 1$ and end of this iteration.

ALGORITHM 1: Iteration procedure of MDBH algorithm.

FIGURE 2: Zachary club network.

FIGURE 3: Result of first iteration (Zachary network).

In the second iteration, node 24th, 25th, 26th, and 31st were removed from prior communities and merged to V_{34} in turn. It is apparent that the first three nodes are supposed to be merged to V_{34} according to the network topology. The 31st node has two edges connected with both V_{01} and V_{34} , respectively, but V_{34} became stronger after the first three nodes were added to it, so the 31st node changed its community from V_{01} to V_{34} . This adjustment reflects that our algorithm has the ability to make reasonable adjustments according to the specific changes in current community structure, and our algorithm could get the more reasonable community structure though it may make wrong decisions in previous iterations. The 9th node was merged to V_{34} in the third iteration. The 3rd node was emerged to V_{34} in the fourth iteration. None of nodes changed its community in the fifth iteration, so our algorithm stopped after the fifth iteration and got the final result as Figure 4. The result is completely consistent with the well-known algorithms for community detection when the ambiguity of the 3rd node is not considered.

4.2. Dolphin Social Network. From 1994 to 2001, Lusseau studied 62 dolphins living in Doubtful Sound, New Zealand. Through the observation of contact between them, he built a dolphin social network [19]. If two dolphins are often together, then an edge will exist between them. What is interesting, this group of dolphins is automatically differentiated into two smaller ones because of the departure of a key member of them.

Again, the proposed algorithm was applied to the Dolphin social network, and it took total of 5 iterations to get the final result. During the first iteration 57 nodes changed their communities, resulting in two large communities and

FIGURE 4: Final partition result for Zachary club network (MDBH).

6 small ones. And during the second, third, and fourth iteration, only nodes 5, 3, and 1 were separately moved to a new community. None of nodes changed their community in the fifth iteration which means that our algorithm only deals with very few nodes in the rest iterations. Network was eventually divided into two parts, with 41 nodes and 21 nodes, respectively. The final result matches the real situation very well as shown in Figure 5.

By applying our algorithm to well-known Zachary club and Dolphin social network, we found that the majority of nodes could be classified to the correct communities after the first iteration, and community structure of the network could quickly get a clear manifestation. And only few nodes need to be adjusted in the rest of the iterations. The reason why our algorithm has a good convergence rate is because possible gains in modularity density are easy to compute with the above formula and that the number of communities decreases dramatically after just a few iterations so that most of the running time is concentrated on the first few iterations.

4.3. Three Kingdoms Network (TK Network). In the first two experiments, our algorithm is proved to be capable of dividing target networks into communities as reasonable as what well-known community detection methods can do. In order to prove that our algorithm is good enough to be used to analyze more complex networks models, we built an empirical network called the “Three Kingdoms network.” The “Romance of the Three Kingdoms” is a famous classic novel describing the story of the Eastern Han Dynasty of China. The characters in this novel maintain a hierarchical community structure which is consistent with historical knowledge. We got inspired by Ravasz and Barabási’s work [20] and built the TK network based on the relationships between the characters of the first five chapters of the novel with 55 nodes and 77 edges. Each node in the network represents a character. And if two characters in the novel have direct dialogue, then an edge exists between them.

First, we applied three Kingdoms network to fast algorithm of Newman which is a typical method using modularity Q [8]. Figure 6 shows the result when getting the maximum value of Q which equals 0.35.

The network was divided into 27 small communities by fast algorithm of Newman. Communities in Figure 6 were labeled with different colors or shapes and every rectangle node labeled white representing a single community. The result shows that too many groups were found by fast algorithm and this does not meet the actual situation of that period in novel. The nodes of 3rd, 4th, and 5th are well-known brothers, but they were finally emerged into three different groups. The node of 25th ought to be a liegeman of 15th, but they were also separately divided.

According to the common understanding about the history of the Three Kingdoms Period, Newman’s fast algorithm did not give us a satisfactory result on target network.

Figure 7 shows the result of our algorithm. All six main groups were identified in Figure 7 by our algorithm after 4 iterations. They are cliques led by the nodes of 3rd, 11th, 20th, 18th, 15th, and 22nd, and they are all influential forces in the early era of the Three Kingdoms. Almost all nodes were divided into correct groups by our algorithm. Some small communities in Figure 6 were emerged into the six main groups by using the judgment of modularity density in (4). The division result in Figure 7 keeps well in line with the common understanding about the relationships of characters in the novel. So we should say that our algorithm performed better than the famous fast algorithm of Newman in real networks of three Kingdoms network.

4.4. Computer-Generated Network. From the complexity analysis, we know that the iteration time of p is a decisive factor which affects the actual speed of our algorithm. The value of p varies from one network to another, and we can hardly get it directly by analysis or calculation, so we try to get the law of p through the experiments of computer-generated network.

Generated network used here consists of three subnetworks, and the total number of nodes and edges are specified. There is only one edge linking different sub-networks. Nearly one third of total edges are within each of the three sub-networks, and they are randomly connected. Here we specify the number of nodes N in generated networks as 600, 1200, 2400, 4800, 9600, 19200, 38400, 76800, 153600, 307200, and 614400 and we specify the number of edges as $E = N \log_2 N$ and $E = 3N$, respectively.

Table 1 is the summary of the numerical results of the experiments on networks with from 600 to 9600 nodes ($E = N \log_2 N$). For each Iteration/ N the table displays the number of nodes who changed its former community to another during iterations. It took only 6 iterations for our algorithm to accomplish community detection. Figure 8 shows the partition result of the network with 4800 nodes and 56278 edges. It took only 15 seconds for our algorithm to partition this network.

In these experiments, the community structure of the network with 9600 nodes was supposed to be clearer than that of the network with 600 nodes, and the real results show that the former network takes less iteration than the later one, even though the former network has a larger scale. From this point of view, MDBH algorithm behaved

FIGURE 5: Final partition result for Dolphin social network (MDBH).

FIGURE 6: Partition result for TK network (Newman fast algorithm).

well in fast community discovery for network with clear community structure.

The statistics in Table 2 shows the number of nodes who were removed from original communities and were reassigned to other communities in each iteration where $E = 3N$. With the increase of the number of nodes, the community structure of networks in this experiment became unclear. When the number of nodes exceeds 2400, the networks were divided into a large amount of or even over a thousand small communities, but it took at most 20

iterations for our algorithm to converge, even though the network may have nearly ten thousand nodes and so unclear community structures.

Both statistics in Tables 1 and 2 show the fact that the iteration time of p in our algorithm depends more on the community structure rather than the network size N . A network with clear community structure takes only a small number of iterations and performs well in fast community detection. And even a network with unclear structure requires iterations of p which is far less than N .

FIGURE 7: Partition result of TK network (MDBH).

TABLE 1: The number of nodes who escaped to other communities during iteration ($E = N \log_2 N$).

Iteration	N				
	600	1200	2400	4800	9600
1st	31	483	1222	2426	5188
2nd	33	756	1675	3340	6720
3rd	14	395	280	368	429
4th	48	5	2	0	0
5th	1	0	0	—	—
6th	0	—	—	—	—
p	6	5	5	4	4

TABLE 2: The number of nodes who escaped to other communities during iteration ($E = 3N$).

Iteration	N				
	600	1200	2400	4800	9600
1st	457	826	1654	3092	8310
2nd	266	433	649	748	1430
3rd	167	307	372	402	551
4th	23	230	194	179	322
5th	23	187	200	140	200
6th	⋮	⋮	⋮	⋮	⋮
p	9	10	18	20	17

The truth is that our algorithm tries to decrease the number of communities in iterations. The algorithm will be stopped till there are no more changes and the modularity density

constraint is achieved. During this process, communities of communities will be built to form a hierarchical structure. The height of the hierarchy that is constructed is determined

FIGURE 8: Computer-generated network (4800 nodes, 56278 edges).

by the number of iterations and is generally a small number, as is shown in above experiments.

In order to test the running time of our algorithm, we chose 11 networks with nodes from 600 to 614400 ($E = N \log_2 N$) and did comparison between our algorithm and Newman's fast algorithm. Figure 9 summarized the comparison results of these experiments. It is demonstrated that the proposed algorithm is much faster with a linear running time in terms of the size of networks. For the network with 614400 nodes and nearly 12 million edges, it took 67.7 seconds for our algorithm to accomplish community detection which is 50 times faster than Newman's fast algorithm.

5. Conclusions

This paper proposed a community detection method called MDBH, which takes both detection accuracy and executing efficiency into account. In order to gain good partition result, the modularity density was employed as a measure function. We simplified the calculation of modularity density and constructed the heuristic search surrounding the simplified formulas. We theoretically analyzed our algorithm

FIGURE 9: Running time comparison between our algorithm and Newman's fast algorithm.

and explained the reason why MDBH is feasible for fast community detection. Experimental results showed that the proposed algorithm performs accurately and competitively

fast for discovering reasonable community structure in network.

Conflict of Interests

The authors declare that they have no financial or personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be constructed as influencing the position presented in (or the review of) the paper.

References

- [1] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.
- [2] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
- [3] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.
- [4] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical Review E*, vol. 80, no. 5, Article ID 056117, 11 pages, 2009.
- [5] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Statistical Analysis and Data Mining*, vol. 4, no. 5, pp. 512–546, 2011.
- [6] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [7] K. Kumar and V. Jayaraman, "Clustering of complex networks and community detection using group search optimization," 2013, <http://arxiv.org/abs/1307.1372>.
- [8] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 6 pages, 2004.
- [9] 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, Article ID 036109, 9 pages, 2008.
- [10] A. Arenas, J. Duch, A. Fernández, and S. Gómez, "Size reduction of complex networks preserving modularity," *New Journal of Physics*, vol. 9, article 176, 2007.
- [11] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, Article ID 066133, 5 pages, 2004.
- [12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics*, vol. 2008, no. 10, Article ID P10008, 2008.
- [13] D. Chen, Y. Fu, and M. Shang, "A fast and efficient heuristic algorithm for detecting community structures in complex networks," *Physica A*, vol. 388, no. 13, pp. 2741–2749, 2009.
- [14] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, Article ID 027104, 4 pages, 2005.
- [15] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007.
- [16] A. Geyer-Schulz, M. Ovelgönne, and M. Stein, "Modified randomized modularity clustering: adapting the resolution limit," in *Algorithms from and for Nature and Life*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 355–363, Springer, 2013.
- [17] C. Granell, S. Gomez, and A. Arenas, "Hierarchical multiresolution method to overcome the resolution limit in complex networks," *International Journal of Bifurcation and Chaos*, vol. 22, no. 7, 2012.
- [18] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
- [19] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
- [20] E. Ravasz and A. L. Barabási, "Hierarchical organization in complex networks," *Physical Review E*, vol. 67, no. 2, Article ID 026112, 7 pages, 2003.