CAAI Transactions on Intelligence Technology

Special Issue: Bio-inspired Computing for Learning and Optimisation

Common strategy to improve community detection performance based on the nodes' property

ISSN 2468-2322
Received on 13th November 2016
Revised on 7th January 2017
Accepted on 9th March 2017
doi: 10.1049/trit.2017.0003
www.ietdl.org

Wei Du[1,2], *Xiaochen He*[1,2] ✉

[1]*School of Public Policy and Administration, Xi'an Jiaotong University, Xi'an, Shaanxi, People's Republic of China*
[2]*Center for Administration Complexity Science, Xi'an Jiaotong University, Xi'an, Shaanxi, People's Republic of China*
✉ *E-mail: hexiaochen121vip@163.com*

Abstract: Improving the community detection algorithm is of great importance. The authors propose a novel method based on the nodes' property in order to detect the community structure. Given a detected community structure, in which nodes have their community signals, the value of the modularity can be changed if a node's community sign change to other communities' signs. Accordingly, the new method readjusts the affiliation between a node and its community in order to raise the modularity value. Experimental results of the detection for a list of open-source networks show that the proposed algorithm can detect better community structure than classic methodologies based on modularity.

1 Introduction

Most of the social networks do not consist of an undifferentiated mass of nodes, but have some subgroup structures. Within those subgroups, the network connections are denser, while the connections are sparser between them. This complex characteristic of the network is called community structure and becomes the key objectives of network analysis in various domains such as computer science, biology and sociology, and has attracted considerable attention in the complex network [1]. Community structure measurement and detection are key issues in this research field. The modularity Q defined by Girvan and Newman is regarded as a normative criterion for measurement of a characteristic of community structure [2], even though it may have some drawbacks [3, 4]. In many domains and disciplines, community structure detecting methods have become the hot spot [5].

According to the operation of the community splitting or combining during the detection process, the detection algorithms can be classified as three categories [6]: (i) the 'top-down dividing' strategy, which takes the whole network as a single community at the initial, and then it divides a big community into small subgroups repetitively [2, 7]. The 'top-down dividing' strategy needs to compute the edge-betweenness which has high computational complexity, and the modularity cannot be fully exploited either. (ii) The 'bottom-up merging' strategy, which treats each node as a community at the initial, and then merges those small communities to form bigger ones [1, 8–10]. This strategy overcomes the disadvantages of the former strategy, but different orderings of merging may lead to different detections. (iii) The 'mixed optimising' strategy absorbing both advantages of the mentioned two strategies. The heuristic methods by optimising the modularity such as simulated annealing techniques [11] and genetic algorithm [12] are representative ones of this strategy. Each community structure detection algorithm mainly solves two problems synchronously: one is to set the number of network communities and the other is to rightly distribute the nodes to one of the communities. These two issues always correlate and influence each other.

Currently, most algorithms for detecting community structure are designed based on maximising modularity value. Since there are $\sum_{k=1}^{n}(1/k!)\sum_{j=1}^{k}\binom{k}{j}j^{n}$, different possible community structures for a network with size on n [13], it suffers an non-deterministic polynomial (NP)-hard problem to detect the community structure [14]. Moreover, thus it is difficult to get a theoretical global optimal solution. A series of research focus on improving methods of merging or partitioning communities based on the three-mentioned categories of algorithms. However, most of them ignore the impact of moving nodes to the different community on the detecting results. Actually, Raddichi *et al.* have defined a community in a strong and weak sense based on node properties and stressed the node degree distribution in the community will affect the community characteristics [15].

Given community structure detection, we discuss the influence of putting network nodes into different communities and come up with a theoretical mechanism. An improved algorithm based on the mechanism is also proposed in order to raise the modularity value. The rest of this paper is divided into three parts. The second part is to introduce the mechanism of the algorithm based on node characteristics after discussing the definition of modularity. The third part is to compare the community structure detection results between improved algorithm and existing algorithms using the classical network data and to clarify the availability and validity of our improved algorithm. The fourth part is the conclusions and further researches.

2 Algorithm

Given a network $G(V, \boldsymbol{A})$, where V represents the sets of nodes, while $\boldsymbol{A}$ represents the set of edges, which is presented as an adjacency matrix. Given a community structure with the number of communities m, the value of modularity defined by Girvan and Newman [2] can be formulated as

$$Q = \sum_{p=1}^{m}\left[e_{pp} - \left(\sum_{q=1}^{m} e_{pq}\right)^{2}\right] \tag{1}$$

In which $e_{pp} = (\|A_{pp}\|/\|A\|)$ represents the proportion of edges within community p and $e_{pq} = (\|A_{pq}\|/2\|A\|)$ represents the proportion of edges between communities p and q. $\|A\| = (1/2)\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}$ denotes the total number of edges, $\|A_{pp}\| = (1/2)\sum_{i\in V_p}\sum_{j\in V_p} a_{ij}$ represents the number of edges within cluster p, while $\|A_{pq}\| = (1/2)\sum_{i\in V_p}\sum_{j\in V_q} a_{ij}$ represents the number of edges between clusters p and q.

Fig. 1 gives an illustration of the mechanism of modularity. The node i belongs to cluster p is presented. $k_{ip} = \left(\sum_{j\in V_p} a_{ij}/2\|A\|\right)$ and $k_{iq} = \left(\sum_{j\in V_q} a_{ij}/2\|A\|\right)$ represents the proportion of node i's a connection within V_p and V_q, respectively.

If node i is removed from cluster p and is set as an independent cluster, the change of the modularity value can be formulated as

$$\begin{aligned}\Delta Q_{p\to i} &= \left(e_{pp} - \left(\sum_{j=1}^{m} e_{pj}\right)^2\right) + (0 - k_i^2)\\ &\quad - \left(e_{pp} + 2k_{ip} - \left(\sum_{j=1}^{m} e_{pj} + k_i\right)^2\right)\\ &= -2k_{ip} + 2k_i\sum_{q=1}^{m} e_{pj}\end{aligned} \tag{2}$$

Next, if node i is merged to the cluster q, the change of modularity can be formulated as

$$\begin{aligned}\Delta Q_{i\to q} &= \left(e_{qq} + 2k_{iq} - \left(\sum_{j=1}^{m} e_{qj} + k_i\right)^2\right)\\ &\quad - \left(e_{qq} - \left(\sum_{j=1}^{m} e_{qj}\right)^2\right) - (0 - k_i^2)\\ &= 2k_{iq} - 2k_i\sum_{j=1}^{m} e_{qj}\end{aligned} \tag{3}$$

Combining the above two operations together, namely removing node i from cluster p and putting into cluster q, the modularity value will be changed as

$$\begin{aligned}\Delta Q &= \Delta Q_{p\to i\to q} = \Delta Q_{i\to q} + \Delta Q_{p\to i}\\ &= 2k_{iq} - 2k_{ip} + 2k_i\left(\sum_{j=1}^{m} e_{pj} - \sum_{j=1}^{m} e_{qj}\right)\end{aligned} \tag{4}$$

The result with $\Delta Q > 0$ means that node i is more appropriate to locate in cluster q rather than p, while if the degree of node i equals to 1 or $\Delta Q < 0$, node i is more appropriate to stay in p.

The above discussion mainly focuses on the effect of moving one node to other communities. Actually, this effect can be also applied for moving more nodes because ΔQ can be superposed. For convenience, we focus on the mechanism of moving nodes one by one in this paper. Given a community structure detected by some optimisation algorithms, we propose a common strategy to improve the classic detecting algorithms based on the effect of changing the node's cluster property (see Fig. 2).

First, we input the detecting result operated by the traditional algorithm. Then, we test each node i and judge whether moving node i from community p to a new community will promote the modularity. If the move increases the modularity, we choose to move i from p to q that maximises the increase of modularity. This operation will be repeated for each community $p = 1, 2, \ldots, m$, where m represents the number of community.

It should be noted that this new method is operated based on the detected community structure. In other words, it is a tool to improve the detection result based on other detecting algorithms. For a network with m communities, the proposed algorithm will explore at most $n \times m$ times. The complexity of the proposed strategy will be much smaller than $O(n^2)$ because of $m \ll n$. Since the complexity of most detection algorithms is $O(n^2)$, the new algorithm based on our approach will not significantly increase the complexity of the original algorithm.

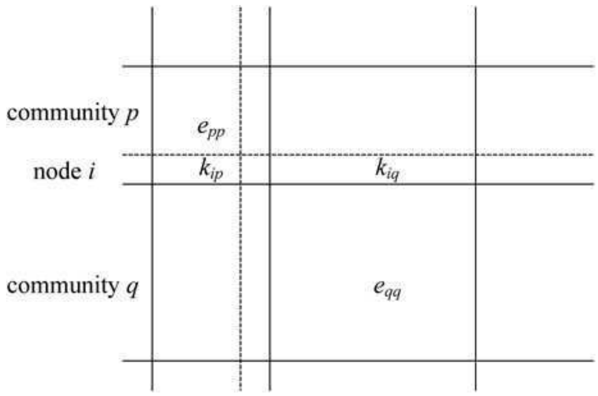

Fig. 1 *Relationship between node i and the community structure. Since this paper is an extended conference paper, we reused this figure in this conference paper [16]*

1: **Input:** ***COM(c)***,which is the node set in community c, c=1,2,…m ;
2: **repeat**
3: **for** each node i in ***COM(p)*** **do**
4: computer $\Delta Q_{p\to i\to q}$ according to equation 4;
5: $\Delta Q_{\vec{q}} \leftarrow \max_q \Delta Q_{p\to i\to q}$;
6: **if** $\Delta Q_{\vec{q}} > 0$
7: move node i from $COM(p)$ to $COM(\vec{q})$;
8: **end if**
9: **end for**
10: **until** c=m
11: **Output:** ***COM***

Fig. 2 *Framework of the new algorithm based on the node's property*

3 Experimental results

In the part of experiments, our proposed approach is compared with four traditional algorithms: the first is Girvan and Newman's algorithm based on betweenness analysis (denoted as G) with the complexity of $O(m^2 n)$, where m is the number of network edges and n is the number of nodes [2]; the second is Newman *et al.*'s algorithm (denoted as N) based on modularity change costing $O[(m+n)n]$ [1]; the third is Clauset *et al.* algorithm (denoted as A) modified from N algorithm with the running time $O(md \log n)$, where d represents the depth of the dendrogram describing the community structure [8]; the fourth is Blondel *et al.*'s heuristic method (denoted as B); and the complexity of which is $O(n^2)$ [10]. G algorithm is a typical 'top-down dividing' strategy, while N, A and B algorithm are very popular 'bottom-up merging' strategy.

The G algorithm result is obtained by Ucinet 6.212, and other experiments are run in MATLAB on a PC with Intel^{R} Core^{TM} 2 DUO CPU T6400@2.00 GHz.

3.1 Computer-generated network

In this paper, the computer-generated network in our experiment is a benchmark network proposed by Lancichinetti *et al.* [17]. The network contains 128 nodes, four communities that each has 32 nodes. The degree of each node is 16, which is divided into inside link and outer link by a fraction μ. It can be expressed as

$$k_{\text{out}} = \mu k_i \quad k_{\text{in}} = (1-\mu)k_i \tag{5}$$

where k_{out} are the links of node i with the other nodes of the network and k_{in} are the links of node i with the other nodes of its community. When μ<0.5, the links of a node connected nodes inside its group are more than the links connected nodes in the other three communities. Moreover, we use this network to test the efficiency of the modified algorithm.

To estimate the accuracy of the experimental results, we introduce an index *normalised mutual information* (NMI) to calculate the similarity between the actual community structure and the detecting community structure [18]. It is formulated as

$$I(A,\ B) = \frac{-2\sum_{i=1}^{cA}\sum_{j=1}^{cB} C_{ij}\ \log\left(C_{ij}N/C_{i.}C_{.j}\right)}{\sum_{i=1}^{cA} C_{i.}\ \log\left(C_{i.}/N\right) + \sum_{j=1}^{cB} C_{.j}\ \log\left(C_{.j}/N\right)} \tag{6}$$

where cA (cB) is the number of groups in community partition A(B), whereas C_{i}. ($C_{\cdot i}$) is the sum of elements of C in row i (column j) and N is the number of nodes. A bigger $I(A, B)$ means better detection results. $I(A, B) = 1$ means that the algorithm can correctly detect the real partition of the network; and with a smaller, $I(A\ B)$, the efficiency of the algorithm is worst.

Then we test these algorithms on 11 sets of computer-generated networks. We employ a mixing parameter μ to represent the fog level of the characteristic of community structure. Each experiment is run for ten independent times.

Fig. 3 shows the average NMI for A algorithm and its modified algorithm for different mixing parameter.

From the result, the detecting result of A algorithm is unstable when the value of mixing parameter μ is smaller than 0.35, but the result of the modified algorithm can always find the correct community partition (NMI equal 1). When μ is bigger than 0.35, the community structure begins to fuzzy. The NMI of the two algorithms is decreased significantly, which means that it is harder to detect the community structure by the two algorithms. However, the detecting result of the modified algorithm is always superior to the result of A algorithm, obtained a greater NMI. This feature can be shown in Fig. 4, which is the NMI versus mixing parameter μ for N algorithm and its modified algorithm.

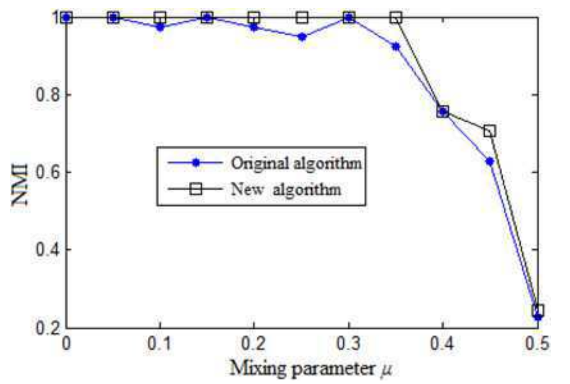

Fig. 3 *NMI versus mixing parameter μ for A algorithm and its modified algorithm. Since this paper is an extended conference paper, we reused this figure in this conference paper [16]*

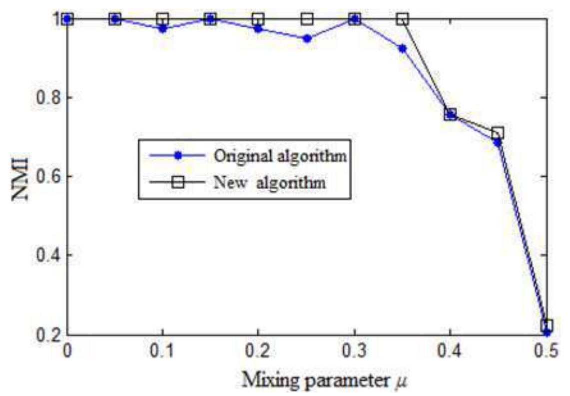

Fig. 4 *NMI versus mixing parameter μ for N algorithm and its modified algorithm. Since this paper is an extended conference paper, we reused this figure in this conference paper [16]*

However, in Fig. 4 when μ is ≥4, the advantage of two algorithms begins to disappear. For example, when $\mu = 0.4$, two algorithms get the same NMI value. This means that the detection result is the same. Moreover, for the subsequent μ, the difference between the two algorithms' NMI value is still very small. So when the community structure begins to fuzzy, for N algorithm, the efficiency of the modified algorithm is not very obvious.

Fig. 5 shows the average NMI for B algorithm and its modified algorithm.

As shown in Fig. 5, the B algorithm and its modified algorithm can detect the correct community structure when μ<0.4. Moreover, the modified algorithm can still detect the community structure correctly when $\mu = 0.45$. Moreover, at this time the community structure has been fuzzy and is hard to be detected. It should be noted that when μ equals to 0.5, each node has half of its relationships connecting with nodes from other communities, in which circumstance the community structure is fuzzy. So algorithms can hardly find the actual partition.

3.2 Standard networks

Furthermore, we tested some real-world networks from Pajek and Ucinet. Table 1 shows the different parameters for these networks, where the original asymmetric networks are symmetrised.

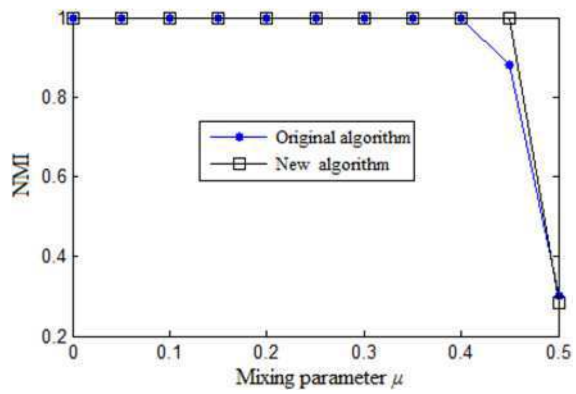

Fig. 5 *NMI versus mixing parameter μ for B algorithm and its modified algorithm. Since this paper is an extended conference paper, we reused this figure in this conference paper [16]*

Table 1 Parameters for the classic networks

Network	Scale	Degree	Density	Source
Dolphins	62	2.5645	0.042	Data[a]
Lesmis	77	10.6494	0.1401	Data
Drugnet	293	1.9386	0.0066	Ucinet
Zachary	34	4.5882	0.139	Ucinet
1crn	327	2.0612	0.0063	Pajek
ADF073	262	2.0458	0.0078	Pajek
B	111	3.4775	0.0316	Pajek
BKHAM	44	4.0455	0.0941	Pajek
BKOFF	40	6.15	0.1577	Pajek
C	65	3.8462	0.0601	Pajek
Cc	62	4.6452	0.0762	Pajek
CENPROD	131	9.6031	0.0739	Pajek
Dnet	180	2.5333	0.0142	Pajek
GR3_53	144	5	0.035	Pajek
GR3_60	120	3	0.0252	Pajek
KAPTAIL	39	8.1026	0.2132	Pajek
MREZA3	144	3.6667	0.0256	Pajek
Nooy	85	1.9059	0.0227	Pajek

Note: For the standard network provided by Ucinet [http://www.analytictech.com/downloaduc6.htm] and Pajek [http://vlado.fmf.uni-lj.si/pub/networks/pajek/default.htm], we have done symmetrical transform for the non-symmetric 0–1 network. Since this paper is an extended conference paper, we reused this table in this conference paper [16].
[a]www-personal.umich.edu/~mejn/.

CAAI Trans. Intell. Technol., 2017, Vol. 2, Iss. 2, pp. 68–72

Table 2 Comparison between original algorithm and a new algorithm

Network	G algorithm			N algorithm		
	Original algorithm	New algorithm	*T*-test	Original algorithm	New algorithm	*T*-test
Dolphins	0.519	0.5194	3.43**	0.4955	0.4955	1.64 +
Lesmis	0.538	**0.5481**	—	0.5006	**0.5498**	—
Drugnet	0.74	**0.7426**	—	0.7448	**0.7455**	—
Zachary	0.409	**0.4112**	—	0.3807	**0.3813**	—
1crn	0.874	**0.8766**	—	0.8827	**0.8828**	—
ADF073	0.883	0.8826	—	0.8810	**0.8814**	—
B	0.632	0.6323	—	0.6096	0.6096	—
BKHAM	−0.002	**0.000**	—	0.1800	**0.196**	—
BKOFF	0.339	**0.3439**	—	0.3413	**0.3429**	—
C	0.556	**0.5683**	—	0.5778	0.5778	—
Cc	0.556	**0.5683**	—	0.5778	0.5778	—
CENPROD	0.11	**0.1288**	—	0.2746	**0.278**	—
Dnet	0.6	**0.612**	—	0.6548	**0.6576**	—
GR3_53	0.675	0.675	—	0.6519	0.6519	—
GR3_60	0.673	0.6731	—	0.6739	**0.6761**	—
KAPTAIL	0.227	**0.2481**	—	0.2910	**0.2961**	—
merza3	0.681	0.6809	—	0.6796	0.6796	—
Nooy	0.808	0.8081	—	0.8081	0.8081	—

Network	A algorithm			B algorithm		
	Original algorithm	New algorithm	*T*-test	Original algorithm	New algorithm	*T*-test
Dolphins	0.4955	0.4955	1.83*	0.5188	**0.5233**	2.14*
Lesmis	0.5006	**0.5498**	—	0.5556	0.5556	—
Drugnet	0.7454	**0.7455**	—	0.7067	0.7067	—
Zachary	0.3807	**0.3813**	—	0.4188	0.4188	—
1crn	0.8819	**0.8820**	—	0.8011	**0.8039**	—
ADF073	0.8815	**0.8818**	—	0.8354	0.8354	—
B	0.6096	0.6096	—	0.6234	**0.6261**	—
BKHAM	0.1800	**0.196**	—	0.2067	0.2067	—
BKOFF	0.3413	**0.3478**	—	0.3676	0.3676	—
C	0.5778	**0.5799**	—	0.5651	**0.5772**	—
Cc	0.5778	**0.5799**	—	0.5578	**0.5643**	—
CENPROD	0.2893	**0.2951**	—	0.2902	0.2902	—
Dnet	0.6548	**0.6554**	—	0.6499	0.6499	—
GR3_53	0.6440	**0.6457**	—	0.6616	**0.6815**	—
GR3_60	0.6739	**0.6746**	—	/	/	—
KAPTAIL	0.2910	**0.2961**	—	0.3215	0.3215	—
Merza3	0.6796	0.6796	—	0.6139	0.6139	—
Nooy	0.8081	0.8081	—	0.8081	0.8081	—

Note: G algorithm result is obtained by Ucient 6.212; the values of the modularity were calculated to three decimal places. Here, '/' represent the algorithm cannot obtain a result; ***means the corresponding $p<0.001$; **means the corresponding $p<0.01$; *means the corresponding $p<0.05$; and + means the corresponding $p<0.1$. The values in bold represent the modularity value. Since this paper is an extended conference paper, we reused this table in this conference paper [16].

The detecting results of the modularity value acquired by four classic algorithms are shown in the column 'Original algorithm' separately, while the results of our proposed algorithm are listed in column 'New algorithm' in Table 2.

There are different community detection results for the same network by using these four algorithms. According to the value of the modularity, B algorithm performs better than others. According to the *T*-test results, our proposed approach can significantly improve the efficiency of the original algorithms. Specifically, the new algorithm has improved the detecting results of 61.1% networks for G algorithm, 61.1% for N algorithm, 77.8% for A algorithm and 35.3% for B algorithm.

4 Conclusions

In this paper, we present an improved approach for detecting community structure based on the node properties. Given a detected community, moving one node from a cluster to other ones may have an impact on the change of the value of modularity. A new approach based on this mechanism is proposed to improve the efficiency of the existing community structure detecting algorithm. Experiments show that our proposed algorithm can significantly improve the community detection result with the same time complexity.

The community detection is an NP-hard problem. For a given network, theoretically, it is hard to obtain an optimal detection result of modularity. So the improvement of algorithm efficiency and accuracy of the precise result are always the important research problem in community structure. It is a useful attempt that this paper proposed an improvement algorithm based on the node feature. However, our algorithm is a correction after other algorithms obtain the community detection result. How to merge our algorithm to another algorithm or effectively combine our improvement strategy into another algorithm in algorithm execution should be considered in further studies.

5 Acknowledgments

This work was jointly supported by the Young Fund of the Social Science of Ministry of Education of China (12YJC840005), the National Natural Science Foundation of China (71071128), the Key Program of the National Social Science Foundation of China (12AZD110) and the Fundamental Research Funds for the Central Universities of Ministry of Education of China (2011jdgz08). This

paper is an extended conference paper in [16] and thus some parts may cite the Springer book.

6 References

[1] Newman, M.E.J.: 'Fast algorithm for detecting community structure in networks', *Phys. Rev. E*, 2004, **69**, (6), p. 066133

[2] Girvan, M., Newman, M.E.J.: 'Community structure in social and biological networks'. Proc. National Academy of Sciences of the United States of America, 2002, vol. 99, no. 12, pp. 7821–7826

[3] Fortunato, S., Barthélemy, M.: 'Resolution limit in community detection'. Proc. National Academy of Sciences of the United States of America, 2007, vol. 104, no. 1, pp. 36–41

[4] Li, Z., Zhang, S., Wang, R., *et al*.: 'Quantitative function for community detection', *Phys. Rev. E*, 2008, **77**, (2), pp. 257–260

[5] Newman, M.E.J, Barabàsi, A.L., Watts, D.J.: 'The structure and dynamic of networks' (Princeton University Press, NJ, 2006)

[6] Santo, F.: 'Community detection in graphs', *Physics (College Park MD)*, 2010, **486**, (3–5), pp. 75–174

[7] White, S., Smyth, P.: 'A spectral clustering approach to finding communities in graphs'. SIAM Int. Conf. Data Mining, Newport Beach, CA, 2005

[8] Clauset, C., Newman, M.E.J, Moore, C.: 'Finding community structure in very large networks', *Phys. Rev. E*, 2004, **70**, pp. 264–277

[9] Wang, X., Chen, G., Lu, H.: 'A very fast algorithm for detecting community structures in complex networks', *Physica A*, 2007, **384**, (2), pp. 667–674

[10] Blondel, V.D., Guillaume, J.L., Lambiotte, R., *et al*.: 'Fast unfolding of communities in large networks', *J. Stat. Mech. Theory Exp.*, 2008, **2008**, (10), pp. 155–168

[11] Medus, A., Acuna, G., Dorso, C.O.: 'Detection of community structures in networks via global optimization', *Physica A*, 2005, **358**, (2–4), pp. 593–604

[12] Tasgin, M.: 'Community detection model using genetic algorithm in complex networks and its application in real-life networks'. MS thesis, Graduate Program in Computer Engineering, Bogazici University, 2005

[13] Richard, O.D., Peter, E.H., David, G.S.: 'Pattern classification' (John Wiley & Sons Inc., New York, 2001, 2nd edn.)

[14] Brandes, U., Delling, D., Gaertler, M., *et al*.: 'Maximizing modularity is hard', *Physics (College Park MD)*, 2006, arXiv:physics/0608255v2

[15] Radicchi, F., Castellano, C., Cecconi, F., *et al*.: 'Defining and identifying communities in networks', *Proc. Natl. Acad. Sci. USA*, 2004, **101**, (9), pp. 2658–2663

[16] Du, W., He, X.: 'A common strategy to improve community detection performance based on the nodes' property', in Gong, M., Pan, L., Song, T., *et al*. (Eds.): 'Bio-inspired computing – theories and applications' (Communications in Computer and Information Science, Springer, Singapore, 2016), p. 682

[17] Lancichinetti, A., Fortunato, S., Radicchi, F.: 'New benchmark in community detection', *Phys. Rev. E*, 2008, **78**, (4), pp. 561–570

[18] Danon, L., Díaz-Guilera, A., Duch, J., *et al*.: 'Comparing community structure identification', *J. Stat. Mech. Theory Exp.*, 2005, **2005**, (9), p. 09008