Enhance the efficiency of heuristic algorithms for maximizing the modularity $Q$

Y. Hu$^{1}$, J. Wu$^{2}$ and Z. Di$^{1(a)}$

$^{1}$ Department of Systems Science, School of Management, Beijing Normal University - Beijing 100875, PRC
$^{2}$ Department of Physics and Astronomy, University of British Columbia - Vancouver, V6T 1Z1, B.C. Canada

received 13 June 2008; accepted in final form 3 December 2008
published online 19 January 2009

PACS 89.75.Hc – Networks and genealogical trees
PACS 89.75.Fb – Structures and organization in complex systems
PACS 89.65.Ef – Social organizations; anthropology

Abstract – The Modularity $Q$ is an important index for identifying the community structure in complex networks. Maximizing $Q$ has been proved to be an NP problem and it can only be achieved by heuristic algorithms. In this paper, the modularity maximization problem is proved to be a nonconvex quadratic programming problem. Based on this result, an approach is suggested and tested to improve the efficiency of heuristic algorithms for maximizing the modularity $Q$. Many numerical results demonstrate that the approach is effective.

Copyright © EPLA, 2009

Introduction. – Complex networks have received an enormous amount of attention in recent years [1–3]. Scientists have become interested in the study of networks describing the topologies of a wide variety of systems such as the world wide web, social and communication networks, biochemical networks and many more. Based on complex networks many quantitative methods can be applied so as to extract the characteristics embedded in the system. One of the most important quantitative methods is the analysis of the community structure [1–3]. Distinct communities within networks can loosely be defined as subsets of nodes which are more densely linked, when compared to the rest of the network. Nodes belonging to a tight-knit community are more likely to have other properties in common. In the world wide web, community analysis has uncovered thematic clusters. In biochemical or neural networks, communities may be functional groups [4,5], and separating the network into such groups could simplify the functional analysis considerably. As a result, the problem of identification of communities has been the focus of many recent efforts. Maximizing the modularity $Q$ is the most widely accepted method for detecting the community structure among many algorithms [4–22], although it has been proved that the modularity index may fail to identify small modules [23]. The modularity $Q$ was presented as an index of the community structure by Newman and Grive, and was introduced as $Q = \sum_{r} (e_{rr} - a_{r}^{2})$, where $e_{rr}$ is the fraction of links that connect two nodes inside the community $r$, $a_{r}$ the fraction of links that have one or both vertices inside the community $r$, and the sum extends to all communities $r$ in a given network. Note that this index provides a way to determine if a certain description of the graph in terms of communities is more or less accurate. Generally speaking, the larger the value of $Q$, the more accurate is the partition into communities. So, maximizing the modularity $Q$ can detect the community structures. There are many algorithms for maximizing $Q$ directly such as the extremal optimization (EO) [21], the greedy algorithms [9] and other optimal algorithms. In fact, they are usually heuristic algorithms for the modularity maximization problem and this problem has been proved to be a NPC in the strong sense by Brandes et al. [24]. So to improve the efficiency of corresponding heuristic algorithms for maximizing the modularity $Q$ is an important problem for detecting communities in complex networks. In addition, some evidences have shown that a small improvement of the modularity $Q$ could result in a remarkable change in communities (as shown in fig. 1). Thus, improving the approaches for maximizing the modularity $Q$ can help us to get more reliable communities of complex networks.

In this letter, we have drawn some conclusions about maximizing $Q$ by the detailed investigation of the mathematical structure of the modularity function. In a previous research, Newman has described the modularity function...
in a matrix form [12]. By basing on this matrix formula of the modularity function, we proved that maximizing the modularity $Q$ is equivalent to a continuous nonconvex quadratic programming. This result allows us to find the maximum modularity by maximizing other equivalent objective functions. We have done many numerical experiments on artificial and real-world networks such as the physics-economics scientists cooperation network [25], the E.coli network [26] and the college football network [5]. The results show that the approach is very helpful for two basic neighborhood transformation algorithms and for the extremal optimization (EO) algorithm for maximizing $Q$. This approach can enhance the efficiency of many heuristic algorithms.

Theorems about the modularity maximization problem. – Newman and Givan proposed the modularity $Q$ index basing on the common experience that such networks seem to have communities within themselves: subsets of nodes within which node-node connections are dense, but between which connections are less dense [5]. According to ref. [12], the modularity $Q$ can be simplified into a matrix form. Suppose we have a network $N$ which has $n$ nodes and $L$ links. It can be represented mathematically by an adjacency matrix $A$ with elements $A_{i,j} = 1$ if there is an edge from $i$ to $j$ and $A_{i,j} = 0$ otherwise. Let $d_i$ denotes the degree of node $i$ and $P$ denotes a matrix with $P_{i,j} = d_i d_j$. Then, we can easily transform the modularity function

$$Q = \frac{1}{2L} \sum_{ij} (A_{i,j} - P_{i,j}) \delta(g_i, g_j)$$

into a matrix form. In the above formula of $Q$, $g_i$ is the community to which node $i$ belongs and $\delta(r,s) = 1$, if $r = s$ and 0, otherwise.

Without losing any generality we assume that the network $N$ has $n$ communities (if the number of communities is less than $n$ we can add some communities without any node). Suppose $S = (S_1, S_2, \cdots, S_n)$ is the community structure matrix, $S_i \in \{0,1\}^n$ denotes the $i$-th community, $i = 1, 2, \cdots, n$. For example: assume $S_i = (0,1,0,1,0,\cdots,0)^T$, this indicates that community $i$ only contains two nodes which are node 2 and 4. For community $j$ without any node $S_j = (0,0,0,\cdots,0)^T$. Because a node only belongs to one community, each row of $S$ just has one 1. We use $\bar{S}$ to denote the set of all possible $S$’s. Let $B = A - P$, then the maximization of the modularity function $Q$ can be written as [12]

$$\max_{\bar{S}\in\bar{S}} Q = \text{Tr}(S^T BS),$$

where $\text{Tr}$ means trace, which denotes the sum of the diagonal entries of a matrix.

Now we will map the problem of the maximization of the modularity $Q$ to a nonconvex quadratic 0-1 programming. Define

$$\tilde{S} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{pmatrix},$$

then $\max_{\bar{S}\in\bar{S}} Q = \text{Tr}(S^T BS)$ can be written as

$$\max_{\tilde{S}\in\tilde{S}} \tilde{Q} = \tilde{S}^T \begin{pmatrix} B & & \\ & B & \\ & & B \end{pmatrix} \tilde{S},$$

subject to

$$\begin{cases} s_{1,1} + s_{1,2} + \cdots + s_{1,n} = 1, \\
    s_{2,1} + s_{2,2} + \cdots + s_{2,n} = 1, \\
    \cdots, \\
    s_{n,1} + s_{n,2} + \cdots + s_{n,n} = 1, \\
    s_{i,j} \in \{0,1\}, \quad i,j = 1,2,\cdots,n. \end{cases}$$

From the subject conditions we can easily get that the set $\bar{S}$ contains $n^n$ elements, $\bar{S} = \{S^1, S^2, \cdots, S^{n^n}\}$. According to the definition of $\bar{S}$ we also have the corresponding set $\bar{S} = \{S^1, S^2, \cdots, S^{n^n}\}$.

Theorem 1: Let $D = \text{diag}(\sum_{i=1}^n |B_{1,i}|, \sum_{i=1}^n |B_{2,i}|, \cdots, \sum_{i=1}^n |B_{n,i}|)$, then, the $\max_{\bar{S}\in\bar{S}} Q = \text{Tr}(S^T BS)$ problem is equivalent to the maximization problem of $\max_{\bar{S}\in\bar{S}} Q_1 = \text{Tr}(S^T (B + D) S)$ which can be mapped to a nonconvex quadratic continuous programming.

Proof of Theorem 1:

1. $\text{Tr}(S^T (B + D) S) = \text{Tr}(S^T BS) + \text{Tr}(S^T DS) = \text{Tr}(S^T BS) + \sum_{i=1}^n D_{i,i},$

note that $\sum_{i=1}^n D_{i,i}$ is constant when the matrix $D$ is given,

2. $\max_{\bar{S}\in\bar{S}} Q = \text{Tr}(S^T BS)$ problem is equivalent to the maximization problem of

$$\max_{\bar{S}\in\bar{S}} Q = \text{Tr}(S^T (B + D) S).$$
Obviously, for a connected but not completely connected network, \(\text{rank}(B + D) = n\). According to the Gershgorin Circle Theory [27], easily we have that \(B + D\) is a symmetrical positive matrix.

\[
\max_{S \in \mathcal{S}} Q = \text{Tr}(S^T(B + D)S) \quad \text{is equivalent to the maximization problem of} \quad \max_{S \in \mathcal{S}} Q_m = \text{Tr}(S^T(B + D)^m S).
\]

\[
\frac{\partial}{\partial S} \text{Tr}(S^T(B + D)S) = (B + D)S,
\]

Thus, the maximization problem of \(\max_{S \in \mathcal{S}} Q\) given by eq. (2) is a continuous nonconvex quadratic programming [28].

Thus, we can conclude that the maximization of modularity \(Q\) given by eq. (2) is a continuous nonconvex quadratic programming.

**Theorem 2:** For all positive integer numbers \(m\), the \(\max_{S \in \mathcal{S}} Q = \text{Tr}(S^TBS)\) problem is equivalent to the maximization problem of \(\max_{S \in \mathcal{S}} Q_m = \text{Tr}(S^T(B + D)^m S)\).

**Proof of theorem 2:** \(\max_{S \in \mathcal{S}} Q = \text{Tr}(S^TBS)\) is equivalent to \(\max_{S \in \mathcal{S}} Q_1 = \text{Tr}(S^T(B + D)S)\), with the notation of \(\bar{S}\), it is equivalent to

\[
\max_{S \in \mathcal{S}} Q_1 = \bar{S}^T \begin{pmatrix} B + D \\ \vdots \\ B + D \end{pmatrix} \bar{S},
\]

note that \(\bar{S}^T \bar{S} = \text{Tr}(S^T S) = n\),

\[
\max_{S \in \mathcal{S}} Q_1 = \bar{S} \cdot n^{m-1}.
\]

and furthermore it is equivalent to

\[
\max_{S \in \mathcal{S}} Q_m = \bar{S} \left( \begin{pmatrix} B + D \\ \vdots \\ B + D \end{pmatrix} ight)^m \bar{S} \cdot n^{m-1}.
\]

Thus, maximizing the modularity function \(Q\) given by eq. (2) is equivalent to

\[
\max_{S \in \mathcal{S}} Q_m = \text{Tr}(S^T(B + D)^m S).
\]

**Application of the theorems.** – By basing on theorem 2, maximizing \(Q\) is equivalent to \(\max_{S \in \mathcal{S}} Q_m = \text{Tr}(S^T(B + D)^m S)\). Can we enhance the efficiency of heuristic algorithms for maximizing the modularity \(Q\) by changing it into this new maximizing problem with a proper large \(m\)? There are so many heuristic algorithms for maximizing the modularity \(Q\) that we cannot investigate all of them. But if we could, we cannot promise either that our method would satisfy the future heuristic algorithms neither. But it is well known that, for many heuristic algorithms such as EO, Potts model [22] and so on, the key methods are to find optimal neighborhood transformations, where neighborhood transformation means moving a node from one community to another community at each optimizing step. So if our method is effective on the basic neighborhood transformation algorithms, it will have great possibilities to be effective on many other heuristic algorithms. We know that there are two basic neighborhood transformation algorithms. One is the random neighborhood transformation algorithm. We randomly initiate the beginning partition (with a sufficient number of groups), then at each step, we randomly choose a node from one community and move it into another one that can make the value of the objective function become larger, until moving any node cannot make the value larger any more. The other algorithm is the greedy neighborhood transformation algorithm. The corresponding process is similar to the random algorithm process, but the difference is that at each step, the node will be moved to a group that makes the value of the objective function have the largest increment. We choose four networks from different fields to test our method. One is the classical artificial random network which has \(n = 128\) nodes divided into 4 communities of 32 nodes each. The edges between two nodes are introduced with different probabilities depending on whether the two nodes belong to the same community or not: every node has \((k_{\text{intra}}) = 8\) links on average to its fellows in the same community, and \((k_{\text{inter}}) = 8\) links to the outer world. Here we chose the artificial network with the diffuse community structures to test the method. This is because when the network contains a clear community structure, \(m\) has almost no effects on the final partition.

The remaining 3 networks are the scientists cooperation network [25], the E.coli network [26] and the college football network [5]. The results show that for a properly large \(m\), our method is helpful to finding a large value of \(Q\) (as shown in fig. 2 and fig. 3) in most cases. Meanwhile, compared with the original method, the time spent in the
maximizing process would be decreased in most cases. But there are some exceptions. In some cases, it takes more time steps to reach the optimum.

We also use the extremal optimization (EO) algorithm [21] to test our method. EO was proposed by Duch and Arenas, and it is a heuristic algorithm. In their algorithm, they define a fitness of each node. For maximizing the modularity function $Q$, the fitness $f_i$ of node $i$ is defined as

$$f_i = \frac{q_i}{k_i},$$

where $k_i$ denotes the degree of node $i$, and the $q_i$ is the contribution of the individual node $i$ to $Q$. Assume $c_i$ denotes the $n$-dimensional vector in which the $i$-th element is 1, 0 otherwise, then

$$q_i = c_i^T B S_i.$$  \hfill (9)

For the maximization problem

$$\max_{S \in \mathbb{S}} Q = \text{Tr}(S^T (B + D)^m S),$$ \hfill (10)

the contribution of node $i$ is

$$q_i^m = c_i^T (B + D)^m S_i.$$ \hfill (11)

Unfortunately, we cannot use the function $f_i^m = \frac{q_i^m}{k_i}$ (as in eq. (8)) to define the fitness, because it does not satisfy the original conditions (see [21]). So we define the new fitness function as $q_i^m$ (eq. (11)). Moreover, Duch and Arenas did not define the “optimal state” quantitatively in [21]. In this paper, we think a partition process has reached the optimal state at step $t$ if the value of the objective function of $t$ is equal to or larger than each value from step $t+1$ to $t+n$, where $n$ is the node number of the network.

We investigate extremal optimization with the new fitness function (NEO) for different $m$ and compare the NEO algorithm with the EO algorithm in the above four networks. The results show that the proper larger $m$ is very helpful both for maximizing the modularity $Q$ and reducing the computing time. But similarly to the results in fig. 2 and fig. 3, for the college football network, the large $m$ has reduced the optimal value of $Q$ (as shown in fig. 4). We are not sure whether this resulted from the special structure of the network or from the effects of a too large $m$. We will investigate this in detail in future work.

Discussion and conclusion. – We prove that the modularity maximization problem is equivalent to a nonconvex quadratic programming problem. By basing on the characteristics of a nonconvex quadratic programming, we demonstrate that the modularity maximization problem is equivalent to the maximization problem $\max_{S \in \mathbb{S}} Q_m = \text{Tr}(S^T (B + D)^m S)$. This conclusion provides a simple way to improve the efficiency of the algorithms for maximizing the modularity $Q$. That is, we can use a different objective function to get an optimal division of networks. Many numerical experiments which are done in different networks include artificial networks, the scientists cooperation network the E.coli network, and the college football network. The results show that the new maximization problem with a properly large $m$ can enhance the efficiency of the heuristic algorithms for maximizing $Q$. But it is a real challenging problem to strictly give the most optimal $m$.

***

The authors want to thank Prof. M. E. J. Newman for providing the college football network and Dr. Qiang
Yuan for some useful discussion. This work is partially supported by 985 Projet and NSFC under the grants No. 70431002, No. 70771011, and No. 60534080.

REFERENCES

[1] A. Albert and A.-L. Barabási, Rev. Mod. Phys., 74 (2002) 47.
[2] M. E. J. Newman, SIAM Rev., 45 (2003) 167.
[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Phys. Rep., 424 (2006) 175.
[4] L. Danon, J. Duch, A. Arenas, and A. Diaz-Guilera, arXiv:cond-mat/0505245 (2005).
[5] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A., 99 (2002) 7821.
[6] S. Lehmann and L. K. Hansen, arXiv:physics/0701348 (2007).
[7] P. Pons and M. Latapy, Proceedings of the 20th International Symposium on Computer and Information Sciences, ISCS’05, Lect. Notes Comput. Sci., 3733 (2005) 284.
[8] F. Wu and B.A. Huberman, Eur. Phys. J. B, 38 (2004) 331.
[9] A. Clauset, Phys. Rev. E, 72 (2005) 026132.
[10] S. Muff, F. Rao, and A. Caflisch, Phys. Rev. E, 72 (2005) 056107.
[11] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A., 103 (2006) 8577.
[12] M. E. J. Newman, Phys. Rev. E, 74 (2006) 036104.
[13] C. P. Massen, D. A. Doisy, and P. K., Phys. Rev. E, 71 (2005) 046101.
[14] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, Physica A, 352 (2005) 669.
[15] M. E. J. Newman, Phys. Rev. E, 69 (2004) 066133.
[16] M. E. J. Newman and E. A. Leicht, Proc. Natl. Acad. Sci. U.S.A., 104 (2007) 9564.
[17] L. Donetti and M. A. Muniz, J. Stat. Mech. (2004) 10012.
[18] M. E. J. Newman and M. Girvan, Phys. Rev. E, 69 (2004) 026113.
[19] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U.S.A., 101 (2004) 2658.
[20] J. P. Bagrow and E. M. Bollt, Phys. Rev. E, 72 (2005) 046108.
[21] J. Duch and A. Arenas, Phys. Rev. E, 72 (2005) 027104.
[22] J. Reichardt and S. Bornholdt, Phys. Rev. Lett., 93 (2004) 218701.
[23] S. Fortunato and M. Barahémy, Proc. Natl. Acad. Sci. U.S.A., 104 (2007) 36.
[24] U. Brandes, D. Delling, M. Gaertler, R. Gorka, M. Hoefer, Z. Nikoloski, and D. Wagner, arXiv:physics/0608255 (2006).
[25] P. Zhang, M. Li, J. Wu, Z. Di, and Y. Fan, Physica A, 367 (2006) 577.
[26] http://www.nd.edu/~networks/resources.htm.
[27] S. Xu, L. Gao, and W. Zhang, Numerical Algebra (Peking University Press, Beijing) 2003.
[28] R. Horst, P. Pardalos, and N. V. Thoai, Introduction to Global Optimizations, 2nd edition (Kluwer Academic Publishers) 2000.