Eur. Phys. J. B **60**, 83–88 (2007)
DOI: 10.1140/epjb/e2007-00313-2

THE EUROPEAN
PHYSICAL JOURNAL B

Deterministic modularity optimization

S. Lehmann[1,2,a] and L.K. Hansen[1]

[1] Informatics and Mathematical Modeling, Technical University of Denmark, Building 321, 2800 Kgs. Lyngby, Denmark
[2] Center for Complex Network Research and Department of Physics, Northeastern University, Boston, MA 02115, USA

Received 7 May 2007
Published online 16 November 2007 – © EDP Sciences, Società Italiana di Fisica, Springer-Verlag 2007

Abstract. We study community structure of networks. We have developed a scheme for maximizing the modularity Q [1] based on mean field methods. Further, we have defined a simple family of random networks with community structure; we understand the behavior of these networks analytically. Using these networks, we show how the mean field methods display better performance than previously known deterministic methods for optimization of Q.

PACS. 89.75.Hc Networks and genealogical trees – 05.10.-a Computational methods in statistical physics and nonlinear dynamics

1 Introduction

A theoretical foundation for understanding complex networks has developed rapidly over the course of the past few years [2–4]. More recently, the subject of detecting network *communities* has gained an large amount of attention, for reviews see references [5,6]. Community structure describes the property of many networks that nodes divide into modules with dense connections between the members of each module and sparser connections between modules.

In spite of a tremendous research effort, the mathematical tools developed to describe the structure of large complex networks are continuously being refined and redefined. Essential features related to network structure and topology are not necessarily captured by traditional global features such as the average degree, degree distribution, average path length, clustering coefficient, etc. In order to understand complex networks, we need to develop new measures that capture these structural properties. Understanding community structures is an important step towards developing a range of tools that can provide a deeper and more systematic understanding of complex networks. One important reason is that modules in networks can show quite heterogenic behavior [7], that is, the link structure of modules can vary significantly from module to module. For such heterogenic systems, global measures can be directly misleading. Also, in practical applications of network theory, knowledge of the community structure of a given network is important. Access to the modular structure of the internet could help search engines supply more relevant responses to queries on terms that belong to several distinct communities[1]. In biological networks, modules can correspond to functional units of some biological system [8].

2 The modularity

Identifying communities in a graph has a long history in mathematics and computer science [9,5]. One obvious way to partition a graph into C communities is distribute nodes into the communities, such that the number of links connecting the different modules of the network is minimized. The minimal number of connecting links is called the *cut size* R of the network.

Consider an unweighted and undirected graph with n nodes and m links. This network can be represented by an adjacency matrix $\mathbf{A}$ with elements

$$A_{ij} = \begin{cases} 1, \text{ if there is a link joining nodes } i \text{ and } j; \\ 0 \;\; \text{otherwise.} \end{cases} \tag{1}$$

This matrix is symmetric with $2m$ entries. The degree k_i of node i is given by $k_i = \sum_j A_{ij}$. Let us express the cut-size in terms of $\mathbf{A}$; we find that

$$R = \frac{1}{2}\sum_{i,j} A_{ij}[1 - \delta(c_i, c_j)], \tag{2}$$

where c_i is the community to which node i belongs and $\delta(\alpha,\beta) = 1$ if $\alpha = \beta$ and $\delta(\alpha,\beta) = 0$ if $\alpha \neq \beta$. Minimizing

[a] e-mail: slj@imm.dtu.dk

[1] Some search engines have begun implementing related ideas, see for example *Clusty, the Clustering Engine* (http://clusty.com/). There is, however, still considerable room for improvement.

R is an integer programming problem that can be solved exactly in polynomial time [10]. The leading order of the polynomial, however, is n^{C^2} which very expensive for even very small networks. Due to this fact, much work in graph partitioning has been based on spectral methods (more below).

Newman has argued [5,11,7] that R is not the right quantity to minimize in the context of complex networks. There are several reasons for this: First of all, the notion of cut-size does not capture the essence of our 'definition' of network as a tendency for nodes to divide into modules with dense connections between the members of module and sparser connections between modules. According to Newman, a good division is not necessarily one, in which there are few edges between the modules, it is one where there are fewer edges than expected. The fact that modules in complex networks can have heterogeneous structures was recently demonstrated quantitatively by Guimerá et al. [12]. There are other problems with R: If we set the community sizes free, minimizing R will tend to favor small communities, thus the use of R forces us to decide on and set the sizes of the communities in advance.

As a solution to these problems, Girvan and Newman propose the modularity Q of a network [1], defined as

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - P_{ij}]\delta(c_i, c_j). \tag{3}$$

The P_{ij}, here, are a null model, designed to encapsulate the 'more edges than expected' part of the intuitive network definition. It denotes the probability that a link exists between node i and j. Thus, if we know nothing about the graph, an obvious choice would be to set $P_{ij} = p$, where p is some constant probability. However, we know that the degree distributions of real networks are often far from random, therefore the choice of $P_{ij} \sim k_i k_j$ is sensible; this model implies that the probability of a link existing between two nodes is proportional to the degree of the two nodes in question. We will make exclusive use of this null model in the following; the properly normalized version is $P_{ij} = (k_i k_j)/(2m)$. It is axiomatically demanded that that $Q = 0$ when all nodes are placed in one single community. This constrains the P_{ij} such that

$$\sum_{ij} P_{ij} = 2m, \tag{4}$$

we also note that $\mathbf{P} = (\mathbf{P})^T$, which follows from the symmetry of $\mathbf{A}$.

We can formulate optimization of Q as a matrix problem. We define a matrix, called the modularity matrix $\mathbf{B} = \mathbf{A} - \mathbf{P}$ and an $(n \times C)$ *community matrix* $\mathbf{S}$. Each column of $\mathbf{S}$ corresponds to a community of the graph and each row corresponds to a node, such that the elements

$$S_{ic} = \begin{cases} 1, \text{ if node } i \text{ belongs to community } c; \\ 0 \ \text{ otherwise.} \end{cases} \tag{5}$$

Since each node can only belong to one community, the columns of $\mathbf{S}$ are orthogonal and $\mathrm{Tr}(\mathbf{S}^T\mathbf{S}) = n$. The δ-symbol in equation (3) can be expressed as

$$\delta(c_i, c_j) = \sum_{k=1}^{C} S_{ik} S_{jk}, \tag{6}$$

which allows us to express the modularity compactly as

$$Q = \frac{1}{2m}\sum_{i,j=1}^{n}\sum_{k=1}^{C} B_{ij} S_{ik} S_{jk} = \frac{\mathrm{Tr}(\mathbf{S}^T\mathbf{B}\mathbf{S})}{2m}. \tag{7}$$

This is the quantity that we wish to maximize.

Note that the structure of $\mathbf{B}$ allows for much faster calculation than one might naively expect [11]. We can write the product of $\mathbf{B}$ and a vector $\mathbf{v}$ as

$$\mathbf{B}\mathbf{v} = \mathbf{A}\mathbf{v} - \frac{\mathbf{k}(\mathbf{k}^T\mathbf{v})}{2m}. \tag{8}$$

This way the multiplication is divided into (i) sparse matrix product with the adjacency matrix that takes $O(m+n)$, and (ii) the inner product $\mathbf{k}^T\mathbf{v}$ that takes $O(n)$. Thus the entire product $\mathbf{B}\mathbf{v}$ scales like $O(m+n)$.

The question of finding the optimal Q is a discrete optimization problem. We can estimate the size of the space we must search to find the maximum. The number of ways to divide n vertices into C non-empty sets (communities) is given by the Stirling number of the second kind $S_n^{(C)}$ [13]. Since we do not know the number of communities that will maximize Q before we begin dividing the network, we need to examine a total of $\sum_{C=2}^{n} S_n^{(C)}$ community divisions [14]. Even for small networks, this is an enormous space, which renders exhaustive search out of the question. Therefore all optimization of Q in real world networks must rely on approximate methods.

There are serious criticisms of the choice of Q as a sensible objective function for finding communities in the first place. Criticism has been raised by Fortunato and Barthélemy [15] who point out that the Q measure has a resolution limit. This stems from the fact that the null model $P_{ij} \sim k_i k_j$ can be misleading. In a large network, the expected number of links between two small modules is small and thus, a single link between two such modules is enough to join them into a single community. A variation of the same criticism has been raised by Rosvall and Bergstrom [16]. These authors point out that the normalization of P_{ij} by the total number of links m has the effect that if one adds a distinct (not connected to the remaining network) module to the network being analyzed and partition the whole network again allowing for an additional module, the division of the original modules can shift substantially due to the increase of m.

In spite of these problems, the modularity is a highly interesting method for detecting communities in complex networks when we keep in mind the limitations pointed out above. What makes the modularity particularly interesting is the fact that it estimates the optimal number of communities for a given network. Classical clustering methods, such as K-means clustering [17] and spectral clustering [18,19] do not possess this ability and has a maximum value of their objective function, when the network is divided into $C = n$ communities. The modularity Q has a maximum value at $C \ll n$ because of the $P_{ij}/(2m)$ term; thus, the ability to estimate the number of communities is closely related to the conceptual problems with Q mentioned in the previous paragraph—this relationship is an interesting subject for further study.

3 Mean field optimization

Simulated annealing was proposed by Kirkpatrick et al. [20] who noted the conceptual similarity between global optimization and finding the ground state of a physical system. Formally, simulated annealing is a stochastic method that maps the global optimization problem onto a physical system by identifying the cost function with the energy function and by considering this system to be in equilibrium with a heat bath of a given temperature T. By annealing, i.e., slowly lowering the temperature of the heat bath, the probability of the ground state of the physical system grows towards unity. This is contingent on whether or not the temperature can be decreased slowly enough such that the system stays in equilibrium, i.e., that the probability is Gibbsian

$$P(\mathbf{S}|T) = \frac{1}{Z}\exp\left(\frac{1}{T}Q(\mathbf{S})\right) = \frac{1}{Z}\exp\left(\frac{\mathrm{Tr}(\mathbf{S}^T\mathbf{B}\mathbf{S})}{2mT}\right). \tag{9}$$

Here, Z is a constant ensuring proper normalization. Kirkpatrick et al. realized the annealing process by Monte Carlo sampling. The representation of the constrained modularity optimization problem is equivalent to a C-state Potts model. In the context of complex networks, simulated annealing was first suggested by [21,22], later work includes Reichardt and Bornholdt, see e.g., [23].

Mean field annealing is a deterministic alternative to Monte Carlo sampling for combinatorial optimization and has been pioneered by Peterson et al. [24,25]. Mean field annealing avoids extensive stochastic simulation and equilibration, which makes the method particularly well suited for optimization. There is a close connection between Gibbs sampling and MF annealing. In Gibbs sampling, every variable is updated by random draw of a Potts state with a conditional distribution,

$$P(S_{i1}, ..., S_{iC}|\mathbf{S}_{\{-i\}}, T) = \frac{P(\mathbf{S}|T)}{\sum_{S_{i1},...,S_{iC}} P(\mathbf{S}|T)}, \tag{10}$$

where the sum runs over the C values of the i'th Potts variable and $\mathbf{S}_{\{-i\}}$ denotes the set of Potts variables excluding the i'th node. As noted by [23], Eq. (10) is local in the sense that the part of the energy function containing variables not connected with the i'th variable cancels out in the fraction. The mean field approximation is obtained by computing the conditional mean of the set of variables coding for the i'th Potts variable using equation (10) and approximating the Potts variables in the conditional probability by their means [25]. This leads to a simple self-consistent set of non-linear equations for the means,

$$\mu_{ik} = \frac{\exp(\phi_{ik}/T)}{\sum_{k'=1}^{C}\exp(\phi_{ik'}/T)}, \quad \phi_{ik} = \sum_j \frac{B_{ij}}{2m}\mu_{jk}. \tag{11}$$

For symmetric connectivity matrices with $\sum_j B_{ij} = 0$, the set of mean field equations has the unique high-temperature solution $\mu_{ik} = 1/C$. This solution becomes unstable at the mean field critical temperature, $T_c = b_{\max}/C$, determined by the maximal eigenvalue $b_{\max}$ of $\mathbf{B}$.

In addition to being deterministic, the mean field algorithm is fast. Each synchronous iteration (see Sect. 5 for details on implementation) requires a multiplication of $\mathbf{B}$ by the mean vector μ. As we have seen, this operation can be performed in $O(m+n)$ time using the trick in equation (8). In these experiments, we have used a fixed number of iterations of the order of $O(n)$, which gives us a total of $O((m+n)n)$.

The most popular deterministic method in the literature was suggested by Newman [11,7] and consists of a combination of a spectral approach and Kernighan-Lin type optimization; we call this optimization scheme the 'KLN-algorithm'. Spectral optimization and the KLN-algorithm both have a complexity of $O(m(m+n))$. We note that the KLN-algorithm suffers from one important drawback: Due to intrinsic problems with spectral methods, Newman uses repeated bisection of the network to determine the communities. However, repeated bisection procedure offers no guarantee, for example, that that the best division into three groups can be arrived at by finding by first determining the best division into two and then dividing one of those two again. In fact, it is straight forward to construct examples where a sub-optimal division into communities is obtained when using repeated network bisection [7,23].

4 A simple network

We will perform our numerical experiments on a simple model of networks with communities. This model network consists of C communities with n_c nodes in each, the total network has $n = n_c C$ nodes. Without loss of generality, we can arrange our nodes according to their community; a sketch of this type of network is displayed in Figure 1. Communities are defined as standard random networks, where the probability of a link between two nodes is given by p, with $0 < p \leq 1$. Between the communities the probability of a link between is given by some fraction f of p with $0 \leq f \leq 1$. For such a network, Q is given by

$$Q(C, f) = \frac{1}{1+(C-1)f} - \frac{1}{C}, \tag{12}$$

which is independent of p. Thus, for this simple network, the only two relevant parameters are the number of communities and the density of the inter-community links relative to the intra-community strength.

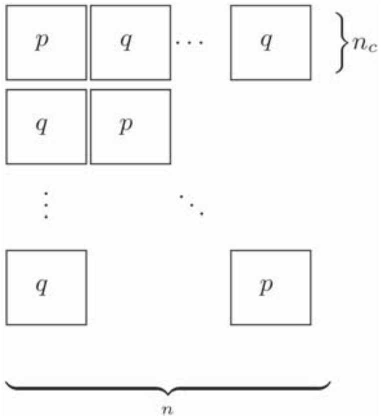

Fig. 1. A sketch of the simple network model. The figure displays the structure of the adjacency matrix with nodes arranged according to community. Inside each community (the blocks) along the diagonal, the probability of a link between two nodes is p and between communities, the probability of a link is q.

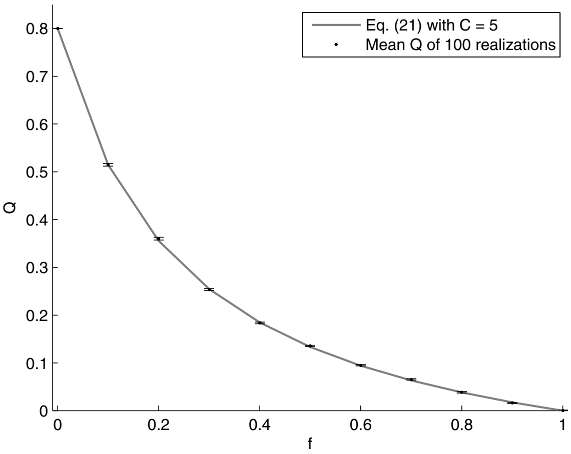

Fig. 2. Equation (12) and Q_{design}. This figure displays Q as a function of f (the relative probability of a link between communities), with $C = 5$ for the simple network defined in Figure 1. The solid line is given by equation (12) and the black dots with error-bars are mean values of Q_{design} in realizations of the simple network with $p = 1/10$ and $n = 500$; each data-point is the mean of 100 realizations. The error bars are calculated as the standard deviation divided by square root of the number of runs.

If we design an adjacency matrix according to Figure 1, we can calculate the value $Q_{\text{design}} = \text{Tr}(\mathbf{S}_{\text{d}}^T \mathbf{B} \mathbf{S}_{\text{d}})/(2m)$, where $\mathbf{S}_{\text{d}}$ is a community-matrix that reflects the designed communities. Values of Q_{design} should correspond to equation (12). We see in Figure 2 that this expectation is indeed fulfilled. The solid curve is Q as a function of f with $C = 5$. The black dots with error-bars are mean values of Q_{design} in realizations of the simple network with $p = 1/10$ and $n = 500$; each data-point is the mean of 100 realizations and the error bars are calculated as the standard deviation divided by square root of the number of runs. The correspondence between prediction and experiment is quite compelling.

We note that the value of Q_{design} can be lower than the actual modularity found for the network by a good algorithm: We can imagine that fluctuations of the inter-community links could result in configurations that would yield higher values of Q. This is always the case when $f \geq 0.3$. We can quantify this quite precisely. Reichardt and Bornholdt [23] have demonstrated that random networks can display significantly larger values of Q due to fluctuations; when $f = 1$, our simple network is precisely a random network (see also related work by Guimerà *et al.* [21]). In the case of the network we are experimenting on, ($n = 500$, $p = 1/10$), they predict $Q \approx 0.13$.

Thus, we expect that the curve for $Q(f, C)$ with fixed C will be deviate from the Q_{design} displayed in Figure 2; especially for values of f that are close to unity. The line will decrease monotonically from $Q(0, C) = 1 - 1/C$ towards $Q(1, C) = 0.11$ with the difference becoming maximal as $f \to 1$.

5 Numerical experiments

In this section we wish to compare the known deterministic methods for optimization of the modularity. We know that the running time of mean field method scales like that of the spectral solution. In order to compare the precision of the mean field solutions to the solutions stemming from spectral optimization, we have created a number of test networks with adjacency matrices designed according to Figure 1. We have created 100 test networks using parameters $n_c = 100$, $C = 5$, $p = 0.1$ and $f \in [0, 1]$. Varying f over this interval allows us to interpolate between a model with C disjunct communities and a random network with no community structure.

We applied the following three algorithms to our test networks

1. Spectral optimization,
2. Spectral optimization and the KLN-algorithm, and
3. Mean field optimization.

Spectral optimization and the KLN-algorithm were implemented as prescribed in [11]. The nC non-linear mean field annealing equations were solved approximately using a $D = 300$-step *annealing schedule* linear in $\beta = 1/T$ starting at β_c and ending in $3\beta_c$ at which temperature the majority of the mean field variables are saturated. The mean field critical temperature $T_c = b_{\max}/C$ is determined for each connectivity matrix. The synchronous update scheme defined as parallel update of all means at each of the D temperatures

$$\begin{aligned} \mu_{ik}^{(d+1)} &= \frac{\exp(\phi_{ik}^{(d)}/T)}{\sum_{k'=1}^{C} \exp(\phi_{ik'}^{(d)}/T)} \\ \phi_{ik}^{(d)} &= \sum_j \frac{B_{ij}}{2m} \mu_{jk}^{(d)} \end{aligned} \tag{13}$$

can grow unstable at low temperatures. A slightly more effective and stable update scheme is obtained by selecting random fractions $\rho < 1$ of the means for update in

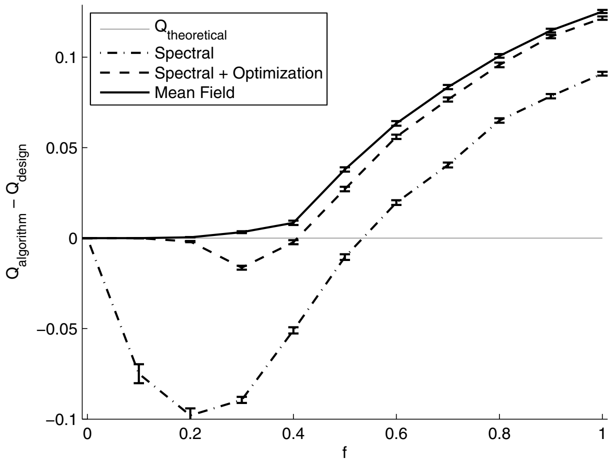

Fig. 3. Comparing spectral methods with the mean field solution. The networks were created according to the simple model, using parameters $n_c = 100$, $C = 5$, $p = 0.1$ and $f \in [0, 1]$. All data points display the point-wise differences between the value of $Q_{\text{algorithm}}$ found by the algorithm in question and Q_{design}. The error-bars are calculated as in Figure 2. The dash-dotted red line shows the results for the spectral method. The dashed blue line shows the results for the spectral optimization followed by KLN post-processing. The solid black curve shows the results for the mean field optimization. The grey, horizontal line corresponds to the theoretical prediction (Eq. (12)) for the designed communities.

$1/\rho$ steps at each temperature. We use $\rho = 0.2$ in the experiments reported below. A final $T = 0$ iteration, equivalent to making a decision on the node community assignment, completes the procedure. We *do not* assume that actual the number of communities $C < C_{\max}$ is known in advance. In these experiments we use $C_{max} = 8$. This number is determined after convergence by counting the number of non-empty communities

The results of the numerical runs are displayed in Figure 3. This figure shows the point-wise differences between the value of $Q_{\text{algorithm}}$ found by the algorithm in question and Q_{design} plotted as a function of the inter-community noise f. The line of $Q_{\text{algorithm}} - Q_{\text{design}} = 0$ thus corresponds to the curve plotted in Figure 2. We see from Figure 3 that the mean field approach uniformly out-performs both spectral optimization and spectral optimization with KLN post-processing. We also ran a Gibbs sampler [23] for with a computational complexity equivalent to the mean field approach. This lead to communities with Q slightly lower than the mean field results, but still better than spectral optimization with KLN post-processing.

We note that the obtained $Q_{\text{algorithm}}$ for a random network ($f = 1$) is consistent with the prediction made by Reichardt and Bornholdt [23]. We also see that the optimization algorithms can exploit random connections to find higher values of $Q_{\text{algorithm}}$ than expected for the designed communities Q_{design}. In the case of the mean field algorithm this effect is visible for values of f as low as 0.2.

Figure 4 shows the median number of communities found by the various algorithms as a function of f. It is evident from Figures 3 and 4 that around $f = 0.3$ — for this particular set of parameters — the distribution of nodes into communities that maximizes Q is no longer identical to the designed community structure. Spectral clustering with and without the KLN algorithm find values $Q_{\text{algorithm}}$ that are significantly lower than Q_{desigm}. The mean field algorithm manages to find a value of $Q_{\text{algorithm}}$ that is higher than the designed Q but does so by creating extra communities. As $f \to 1$ it becomes more and more difficult to recover the designed number of communities.

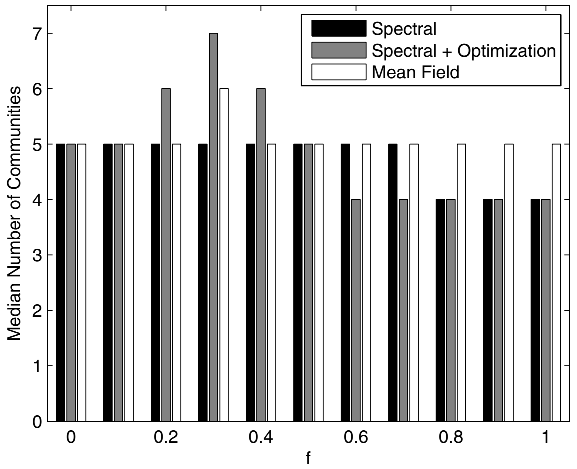

Fig. 4. The median number of communities found by the various algorithms. The panel shows the median number of communities as a function of the relative fraction of inter-community links f. All optimization schemes consistently pick four or five communities for the highest values of f. This finding is consistent with theoretical and experimental results by Reichardt and Bornholdt [23]. Note that the fact that 5 communities are detected does not imply that the communities detected correspond to the designed communities. In fact, for $f \geq 0.3$, they do not.

6 Discussion and conclusions

We have introduced a deterministic mean field annealing approach to optimization of modularity Q. We have evaluated the performance of the new algorithm within a family of networks with variable levels of inter-community links, f. Even with a rather costly post-processing approach, the spectral clustering approach suggested by Newman is consistently out-performed by the mean field approach for higher noise levels. Spectral clustering without the KLN post-processing finds much lower values of Q for all $f > 0$.

Speed is not the only benefit of the mean field approach. Another advantage is that the implementation of mean field annealing is rather simple and similar to Gibbs sampling. Careful Gibbs sampling and annealing will lead to solutions with higher modularity than those obtained by the mean field method, however, for shorter simulations the mean field method can be more efficient, because of it avoids time consuming equilibration. Careful comparison requires optimization of annealing schedules and is beyond the scope of the present communication.

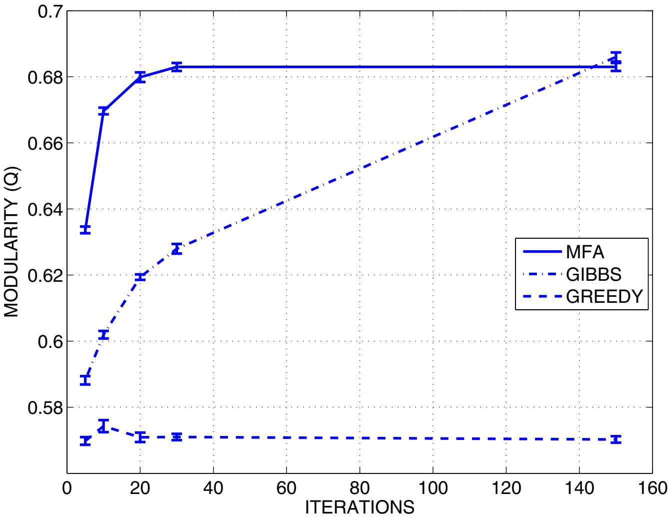

Fig. 5. Comparison of the modularity performance for mean field annealing, Gibbs sampling, and a 'greedy' optimizer. The greedy algorithm corresponds to Gibbs sampling at $T = 0$. The graph describes the co-authorship network of the Los Alamos condensed matter preprint archive, considering articles published between April 1998 and February 2004 [26], it has $n = 30\,561$ nodes, and $m = 125\,959$ links. The mean field method provides good modularity solutions for very few iterations, for the present graph the Gibbs sampling scheme outperforms mean field annealing at 150 iterations. The best modularity solutions we found in this network after extensive Gibbs sampling have $Q \equiv 0.71$. The zero temperature greedy search does not produce useful modularity solutions here.

In Figure 5 we give an example of the relative performances of mean field annealing, Gibbs sampling, and a 'greedy' algorithm, for co-authorship network of the Los Alamos condensed matter preprint archive, considering articles published between April 1998 and February 2004 [26]; this graph is rather large, with $n = 30\,561$ nodes, and $m = 125\,959$ links. The co-authorship graph was also studied in [27,23].

In this example we set $C = 30$. The greedy algorithm performs a very local optimization corresponding to Gibbs sampling at $T = 0$ and does not produce useful modularity solutions in the given graph. The mean field method produces high modularity solutions for very few iterations, making it of high interest for time critical applications, for example in on-line clustering for search engines.

As we have noted above, the modularity measure Q may need modification in specific non-generic networks. In that case, we note that the mean field method is quite general and can be generalized to many other measures.

This work is supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound', `www.intelligentsound.org` (STVF No. 26-04-0092) and by the Danish Natural Science Research Council.

References

1. M.E.J. Newman, M. Girvan, Phys. Rev. E **69**, 026113 (2004)
2. R. Albert, A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002)
3. S.N. Dorogovtsev, J.F.F. Mendes, Adv. Phys. **51**, 1079 (2002)
4. M.E.J. Newman, SIAM Rev. **45**, 167 (2003)
5. M.E.J. Newman, Eur. Phys. J. B **38**, 321 (2004)
6. L. Danon, J. Duch, A. Diaz-Guilera, A. Arenas, J. Stat. Mechanics, P09008 (2005)
7. M.E.J. Newman, Phys. Rev. E **74**, 036104 (2006)
8. A.-L. Barabási, Z.N. Oltvai, Nature Reviews Genetics **5**, 101 (2004)
9. F.K.R. Chung, *Spectral Graph Theory* (American Mathematical Society, 1997)
10. O. Goldscmidt, D.S. Hochbaum, *Polynomial algorithm for the k-cut problem*, in *Proceedings of the 29th Annual IEEE Symposium on the Foundations of Computer Science* (Institute of Electrical and Electronics Engineers, 1988), p. 444
11. M.E.J. Newman, *Proceedings of the National Academy of Sciences, USA* **103**, 8577 (2006)
12. R. Guimerá, M. Sales-Pardo, L.A.N. Amaral, Nature Physics **3**, 63 (2007)
13. Mathworld, `http://mathworld.wolfram.com/`
14. M.E.J. Newman, Phys. Rev. E **69**, 066133 (2004)
15. S. Fortunato, M. Barthelemy, *Resolution limit in community detection*, *Proceedings of the National Academy of Sciences USA* **104**, 36 (2007)
16. M. Rosvall, C.T. Bergstrom, *An information-theoretic framework for resolving community structure in complex networks*, 2006
17. J.B. MacQueen, *Some methods for classification and analysis of multivariate observations*, in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press, Berkeley, 1967), Vol. 1, p. 281
18. M. Fiedler, Czechoslovak Mathematical J. **23**, 298 (1973)
19. A. Pothen, H. Simon, K.-P. Liou, SIAM J. Matrix Analysis and Applications **11**, 430 (1990)
20. S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecchi, Science **220**, 671 (1983)
21. R. Guimerá, M. Sales-Pardo, L.A.N. Amaral, Phys. Rev. E **70**, 025101 (2004)
22. R. Guimerá, L.A.N. Amaral, Nature **433**, 895 (2005)
23. J. Reichardt, S. Bornholdt, Phys. Rev. E **74**, 016110 (2006)
24. C. Peterson, J.R. Anderson, Complex Systems **1**, 995 (1987)
25. C. Peterson, B. Söderberg, Int. J. Neural Syst. **1**, 3 (1989)
26. S. Warner, Library Hi Tech. **21**, 151 (2003)
27. G. Palla, I. Derényi, I. Farkas, T. Vicsek, Nature **435**, 814 (2005)