Computational Complexities of Optimization Problems Related to Model Based Clustering of Networks

Bhaskar DasGupta

Abstract An extremely popular model-based graph partitioning approach that is used for both biological and social networks is the so-called modularity optimization approach originally proposed by Newman and its variations. In this chapter, we review several combinatorial and algebraic methods that have been used in the literature to study the computational complexities of these optimization problems.

1 Introduction

For complex systems of interaction in biology and social sciences, modeled as networks of pairwise interactions of components, many successful approaches to mathematical analysis of such networks rely upon viewing them as composed of subnetworks or modules whose behaviors are simpler and easier to understand. Coupled with appropriate interconnections, the goal is to deduce emergent properties of the complete network from the understanding of these simpler subnetworks. Such modular decomposition of networks appears quite often in the application domain. For example, in social networks it is a common practice to partition the nodes of a network into modules called communities such that nodes within each community are related more closely to each other than to nodes outside the community [14, 17, 21, 35–37, 42], and similarly in regulatory networks modular decomposition has been used in studying "monotone" parts of the dynamics of a biological system [12, 16] and more generally in studying a network in terms of interconnectivity of smaller parts with well-understood behaviors [22, 43]. These problems are also closely connected to many partitioning problems in graphs based on local densities studied in other computer science

Bhaskar DasGupta
Department of Computer Science, University of Illinois at Chicago, Chicago, IL 50507, USA, e-mail: bdasgup@uic.edu

applications. Simplistic definitions of modules traditionally studied in the computer science literature, such as cliques, unfortunately do not apply well in the context of biological and social networks and therefore alternate methodologies are most often used [14, 17, 21, 35–37, 42]. As in virtually all works on network partitioning and community detection, we consider a *static* model of interaction in which the network connections do not evolve over time. In this chapter we focus on one approach of modular analysis of networks, namely the *model-based* approach.

2 Model-based Decomposition

In the context of biological or social interaction networks, an important problem is to *partition* the nodes into a set of so-called "communities" or "modules" of "statistically significant" interactions. Such partitions facilitate studying interesting properties of these graphs in their applications, such as studying the behavioral patterns of a group of individuals in a society, and serve as important steps towards computational analysis of these networks. The *general* model-based decomposition approach can be described in the following manner:

- We have an appropriate "global null model" $\mathcal{G}$ of a background random graph providing, *implicitly* or *explicitly*, the probability $p_{u,v}$ of an edge between two nodes u and v.
- The general goal is to place nodes in the same module if their interaction patterns are significantly stronger than those inferred by $\mathcal{G}$ and in different modules if their interaction patterns are significantly weaker than those inferred by $\mathcal{G}$. No a priori assumptions are made about the number of modules as opposed to some other traditional graph clustering approaches.

As an example of applicability of the above framework of model-based clustering framework, consider the following maximization version of the standard $\{+,-\}$-correlation clustering that appears in the computer science literature extensively [5, 9, 46]:

Input: an undirected graph $G = (V, E)$ with each edge $\{u, v\} \in E$ having a label $\ell_{u,v} \in \{1, -1\}$.

Valid solution: a partition $V_1, \ldots, V_k$ of V.

Objective: *maximize* $\sum_{i=1}^{k} \sum_{u,v \in V_i} \ell_{u,v}$.

The above problem can be placed in the above model-based clustering framework in the following manner:

- Let H be the graph consisting of all edges labeled 1 in G.
- Let $p_{u,v} = \begin{cases} 0, \text{ if } \ell_{u,v} = 1 \\ 1, \text{ otherwise} \end{cases}$

- Let the modularity of a partition V_i be $\mathsf{M}(V_i) = \sum_{u,v \in V_i} \left(a_{u,v} - p_{u,v}\right)$ where $a_{u,v} = \begin{cases} 1, \text{ if } \{u,v\} \text{ is an edge of } H \\ 0. \text{ otherwise} \end{cases}$
- Let the total modularity of the partition $V_1, \ldots, V_k$ be defined as $\sum_{i=1}^{k} \mathsf{M}(V_i)$.

As is well known, *every graph decomposition procedure has both pros and cons, and there exists no universal decomposition procedure that works for every application*. Any decomposition method that relies on a global null model such as the one currently discussed suffers from the drawback that each node can get attached to any other node of the graph; for another possible criticism, see [18]. To design and analyze a model-based decomposition, one faces *at least* the following three choices, each being influenced by the appropriateness in the corresponding applications:

($C1$) What should be an appropriate null model $\mathcal{G}$?
($C2$) How should we precisely measure the statistical significance ("fitness") of an *individual* module of the given graph ?
($C3$) How should we *combine* the fitnesses of individual modules to get a total fitness value for the entire network ?

In this chapter, we begin with a specific choice of ($C1$)–($C3$) that leads us to the so-called *modularity clustering*, an extremely popular decomposition method in practice in the context of both social networks [1, 32, 37, 38] and biological networks [22, 43]. Subsequently, we discuss a few other choices for ($C1$)–($C3$). An algorithm $\mathcal{A}$ for a maximization (resp., minimization) problem is said to have an approximation ratio of ε (or simply an ε-approximation) provided $\mathcal{A}$ runs in polynomial time in the size of the input and produces a solution with an objective value no smaller than $1/\varepsilon$ times (resp., no larger than ε times) the value of the optimum. We assume that the reader is familiar with standard concepts in algorithmic design and analysis such as found in textbooks [13, 19, 48].

3 Basic Modularity Clustering

To simplify discussion, suppose that our input is an undirected unweighted graph[1] $G = (V, E)$ of n nodes and m edges, let $A = [a_{u,v}]$ denote the *adjacency matrix* of G, *i.e.*, $a_{u,v} = \begin{cases} 1, \text{ if } (u,v) \in E \\ 0, \text{ otherwise} \end{cases}$ and let d_u denote the degree of node u.

[1] The definitions can be easily generalized for directed and weighted graphs; see Section 3.5.

3.1 Definitions

In the basic version of modularity clustering as proposed by Newman and others [21, 32, 35, 36, 38], the following options for ($C1$)–($C3$) were selected.

Choice for ($C1$): The null model $\mathcal{G}$ is dependent on the *degree-distribution* of the given graph G and is given by $p_{u,v} = \frac{d_u d_v}{m}$ *with* $u = v$ *being allowed.* Such a null model preserves the distribution of the degree of each node in the given graph *in expectation*, *i.e.*, $\sum_{v \in V} p_{u,v} = d_u$.

Choice for ($C2$): If nodes u and v belong to the same partition, then one would expect $a_{u,v}$ to be significantly higher than $p_{u,v}$. This is captured by adding the term $a_{u,v} - p_{u,v}$ to the objective value of the decomposition. Thus, for a subset of nodes $V' \subseteq V$, its fitness is given by $\mathsf{M}(V') = \sum_{u,v \in V'} (a_{u,v} - p_{u,v})$.

Choice for ($C3$): A partition $\mathcal{S} = \{V_1, \ldots, V_k\}$ of nodes[2] has a total fitness ("modularity") of

$$\mathsf{M}(\mathcal{S}) = \frac{1}{2m} \sum_{i=1}^{k} \mathsf{M}(V_i) = \frac{1}{2m} \sum_{i=1}^{k} \left(\sum_{u,v \in V_i} \left(a_{u,v} - \frac{d_u d_v}{2m} \right) \right) \tag{1}$$

and our goal is to *maximize* $\mathsf{M}(\mathcal{S})$ over all possible partitions $\mathcal{S}$ of V. The $\frac{1}{2m}$ factor is introduced only for a min-max normalization of the measure [23] so that $0 \leq \max_{\mathcal{S}} \{\mathsf{M}(\mathcal{S})\} < 1$.

Formally, the modularity clustering (Mc) problem is defined as follows:

Problem name: modularity clustering (Mc).
Input: an undirected graph $G = (V, E)$.
Valid solution: a partition $\mathcal{S} = \{V_1, \ldots, V_k\}$ of V.
Objective: *maximize* $\mathsf{M}(\mathcal{S}) = \frac{1}{2m} \sum_{i=1}^{k} \left(\sum_{u,v \in V_i} \left(a_{u,v} - \frac{d_u d_v}{2m} \right) \right)$.

In the sequel, we will use OPT to denote the maximum modularity value $\max_{\mathcal{S}} \{\mathsf{M}(\mathcal{S})\}$ of a given graph G. $\mathsf{M}(\mathcal{S})$ can be equivalently represented via simple algebraic manipulation [8, 15, 37, 38] as

$$\mathsf{M}(\mathcal{S}) = \sum_{i=1}^{k} \left[\frac{m_i}{m} - \left(\frac{D_i}{2m} \right)^2 \right] \tag{2}$$

where m_i is the number of weights of edges whose *both* endpoints are in the cluster V_i and $D_i = \sum_{v \in V_i} d_v$ is the sum of degrees of the nodes in V_i.

Yet another equivalent way to represent $\mathsf{M}(\mathcal{S})$, which was found to be quite useful in proving NP-completeness when inputs are restricted to graphs with the

[2] Each V_i is usually called a "cluster".

maximum degree of any node bounded by a constant, is the following. Let m_{ij} denote the number of edges one of whose endpoints is in V_i and the other in V_j and $D_i = \sum_{v \in V_i} d_v$ denote the sum of degrees of nodes in cluster V_i. Then,

$$\mathrm{M}(V_i) = \frac{1}{2m}\left(\sum_{u \in V_i, v \notin V_i}\left(\frac{d_u d_v}{2m} - a_{u,v}\right)\right)$$

and this gives us the following third equation of modularity (note that now each pair of clusters contributes to the sum in Equation (3) *exactly once*):

$$\mathrm{M}(\mathcal{S}) = \sum_{V_i, V_j\,:\, i<j}\left(\frac{D_i D_j}{2m^2} - \frac{m_{ij}}{m}\right) \tag{3}$$

An important special case of the Mc problem arises [8, 15] if we restrict the maximum number of partitions of V to some pre-specified value κ. This special case, referred to as the *modularity κ-clustering* (κ-Mc) problem, is thus formally defined as follows.

Problem name: modularity κ-clustering (κ-Mc).
Input: an undirected graph $G = (V, E)$.
Valid solution: a partition $\mathcal{S} = \{V_1, \ldots, V_k\}$ of V with $k \leq \kappa$.
Objective: *maximize* $\mathrm{M}(\mathcal{S}) = \frac{1}{2m}\sum_{i=1}^{k}\left(\sum_{u,v \in V_i}\left(a_{u,v} - \frac{d_u d_v}{2m}\right)\right)$.

In the sequel, we will use OPT_κ to denote the maximum modularity value of the modularity κ-clustering problem for a given graph. The usefulness of the κ-Mc problem in designing approximation algorithms for the Mc problem is brought out by the following lemma.

Lemma 1. [15] *For any $\kappa \geq 1$,* $\mathrm{OPT}_\kappa \geq \left(1 - \frac{1}{k}\right)\mathrm{OPT}$.

Thus, in particular, $\mathrm{OPT}_2 \geq \mathrm{OPT}/2$ and, for large enough κ, OPT_κ approximates OPT very well.

3.2 Absolute Bounds for OPT *and* OPT_κ

Although it is difficult to specify accurately the range of values that OPT or OPT_κ may take for general graphs, it is possible to derive some bounds when the given graph G has some specific topologies. For example, bounds of the following kinds were demonstrated in [8, 15].

- If G is a complete graph then OPT = 0.
- If G is an union of k *disjoint* cliques each with n/k nodes then $\mathrm{OPT} = 1 - \frac{1}{k}$.

- If G is a d-regular graph (*i.e.*, a graph in which every node has a degree of *exactly* d), then
$$\text{OPT} > \frac{0.26}{\sqrt{d}}, \text{ if } n > 40d^9$$
$$\text{OPT} > \frac{0.86}{d} - \frac{4}{n}, \text{ otherwise}$$
- If G is a graph in which every node has a degree of *at most* d and $d < \frac{\sqrt[5]{n}}{16 \ln n}$, then $\text{OPT} > \frac{1}{8d}$.
- For any graph G and any κ, $0 \leq \text{OPT}_\kappa \leq 1 - \frac{1}{\kappa}$.

3.3 Computational Hardness Results

3.3.1 NP-hardness Results

It was shown in [8] that computing OPT is NP-complete for sufficiently dense graphs (graphs in which nodes have degrees roughly $\Omega\left(\sqrt{n}\right)$ for every node) and this NP-completeness result for dense graphs holds even if one wishes to compute just OPT_2. A basic idea behind many of these reductions is that large size cliques of the graph are properly contained within a community. The authors in [15] show that computing OPT_2 is NP-complete even if the given graph is G sparse and regular, namely even if G is a d-regular graph for any fixed $d \geq 9$. The NP-completeness proof in [15] for sparse graphs, motivated by the proof for this case in [8], is from the *graph bisection* problem for 4-regular graphs which is known to be NP-complete [28]. Intuitively, in this reduction an optimal solution for the modularity 2-clustering problem is constrained to have *exactly* the same number of nodes in each community.

3.3.2 Beyond NP-hardness: APX-hardness Results

A minimization problem is said to be APX-hard if it cannot be approximated within a factor of $1+\varepsilon$ for some constant $\varepsilon > 0$ under the assumption of P $\neq$ NP. The authors in [15] showed that computing OPT_κ for any $\kappa > 1$ is is APX-hard for dense regular graphs, namely for d-regular with $d = n-4$. This approximation gap is derived from the following approximation gap of the maximum independent set problem for 3-regular graphs [11]:

Problem name: Maximum Independent Set for 3-regular graphs (3-MIS).
Input: a graph $H = (V, E)$ that is 3-regular, *i.e.*, every node has a degree of exactly 3.
Valid solution: a subset $V' \subset V$ of nodes such that every pair of nodes u and v in V' is *independent*, *i.e.*, $\{u, v\} \notin E$.
Objective: *maximize* $|V'|$.

Approximation gap as derived in [11] : NP-hard to decide if $\max_{V' \subseteq V}\left\{|V'|\right\} \geq \frac{95}{194}|V|$ or if $\max_{V' \subseteq V}\left\{|V'|\right\} \leq \frac{94}{194}|V|$.

The reduction is carried out by providing the edge-complement of the graph H as the input graph G to the Mc problem, *i.e.*, the input to Mc is $G = (V, E)$ with $E = \{\{u, v\} | u, v \in V, \{u, v\} \notin F\}$. The reduction was completed in [15] by proving the following bounds for any κ:

- If $\max_{V' \subseteq V}\left\{|V'|\right\} \geq \frac{95}{194}|V|$ then $\mathrm{OPT}_\kappa > \frac{0.9388}{|V|-4}$.
- If $\max_{V' \subseteq V}\left\{|V'|\right\} \leq \frac{94}{194}|V|$ then $\mathrm{OPT}_\kappa < \frac{0.9382}{|V|-4}$.

This provides the desired inapproximability result with $\varepsilon = 1 - \frac{0.9388}{0.9382} \approx 0.0006$. The intuition behind a proof of the above bounds is that, for the type of sparse graphs H that is considered in the reduction, edge-complements of large-size independent set of nodes in H must be properly contained within a cluster of G and that $\mathrm{OPT}_\kappa \leq \mathrm{OPT}_2$ for any $\kappa > 2$.

3.4 Approximation Algorithms

In this section, we review several combinatorial and algebraic method for designing approximation algorithms for the Mc and κ-Mc problems.

3.4.1 Greedy Heuristics

As a first attempt at designing approximation algorithms for Mc, one may be tempted to use a greedy approach of the following type that can easily be implemented to run in $\mathrm{O}\left(n^2 \log n\right)$ time [8]:

---

1. Start with each node being a separate cluster. Let $C^0 = \left\{\{v\} | v \in V\right\}$ be this initial clustering.

2. **for** $i = 1, 2, \ldots, n-1$ **do**

- Merge two clusters of C^{i-1} that yield a clustering with the largest increase or the smallest decrease in modularity.
- Let C^i be the new clustering obtained.

endfor

3. Return $\max_i \left\{\mathrm{M}\left(C^i\right)\right\}$ as the solution.

Consider the graph $G = (V, E)$ consisting of the union of two *disjoint* cliques V_1 and V_2, each having $n/2$ nodes, along with $n/2$ additional edges corresponding to an arbitrary maximum bipartite matching $\{\{u, v\} | u \in V_1, v \in V_2\}$ among nodes in V_1 and V_2. Brandes *et al.* [8] observed that the above greedy approach has

an unbounded approximation ratio on this graph by showing that the greedy algorithm obtains a modularity value of 0 even though OPT is very close to $1/2$. Thus, greedy approaches do not seem very promising in designing algorithms with bounded approximation ratios.

3.4.2 Linear Programming Based Approach

It is possible to formulate the modularity clustering problem with arbitrarily many clusters as an integer linear program (ILP) in the following manner. For every two distinct nodes $u, v \in V$, let $x_{u,v}$ be a Boolean variable defined as:

$$x_{u,v} = \begin{cases} 0, & \text{if } u \text{ and } v \text{ belong to the same cluster} \\ 1, & \text{otherwise} \end{cases}$$

One constraint of partitioning the nodes into clusters is the so-called "triangle inequality" constraint:

> if u, v and v, z belong to the same cluster then u, z must also belongs to the same cluster.

This is easily described by the linear (inequality) constraint $x_{u,z} \leq x_{u,v} + x_{v,z}$. Noting that $1 - x_{u,v}$ is the contribution of a pair of distinct nodes u, v to the modularity value computed by Equation (1), we arrive as the following equivalent ILP formulation of the Mc problem [1, 8, 15]:

$$\begin{array}{l} \textit{maximize} \displaystyle\sum_{u,v \in V:\, u \neq v} \left(\frac{a_{u,v} - \frac{d_u d_v}{2m}}{2m} \right) (1 - x_{u,v}) - \sum_{v \in V} \frac{d_v^2}{2m} \\ \textit{subject to} \\ \quad \forall u \neq v \neq z : \; x_{u,z} \leq x_{u,v} + x_{v,z} \\ \quad \forall u \neq v : \; x_{u,v} \in \{0, 1\} \end{array}$$

However, solving an ILP exactly is in general an NP-hard problem. A natural approach is therefore to consider the linear programming (LP) relaxation of the ILP obtained by replacing the constraints "$\forall u \neq v$: $x_{u,v} \in \{0, 1\}$" by $\forall u \neq v$: $0 \leq x_{u,v} \leq 1$, solving this LP in polynomial time [26] and then use some type of "rounding" scheme to convert fractional values of variables to Boolean values[3]. The authors in [1] used such a LP-relaxation with several rounding schemes for empirical evaluations.

Unfortunately, [15] showed that this LP-relaxation based approach, irrespective of the rounding scheme used, may not be a very good choice for designing approximation algorithms with good guaranteed approximation ratio in the following manner. Let OPT_f denote the optimal objective value of the LP obtained from the ILP. Then, it was shown in [15] that, for every $d > 3$ and for all sufficiently

[3] See [48, part II] for further details of such an approach.

large n, there exists a d-regular graph with n nodes such that the integrality gap $\mathsf{OPT}_f/\mathsf{OPT}$ is $\Omega(\sqrt{d})$, and thus an approximation ratio of $o(\sqrt{n})$ would be impossible to achieve irrespective of the rounding scheme used.

3.4.3 Spectral Partitioning Approach

Spectral partitioning methods for graph decomposition problems are well-known [41, 45]. This approach was first suggested by Newman in [37] for the 2-Mc problem but a theoretical analysis of the approximation ratio of this approach is *not* yet known. Consider the $n \times n$ symmetric matrix $W = [w_{u,v}]$ with $w_{u,v} = a_{u,v} - \frac{d_u d_v}{2m}$, and suppose that W has an eigenvector $\mathbf{u}_i$ with a corresponding eigenvalue b_i for $i = 1, 2, \ldots, n$. For every node $u \in V$, let x_u be a selection variable defined as:

$$x_u = \begin{cases} -1, & \text{if } u \text{ is assigned to cluster 1 } (V_1) \\ 1, & \text{if } u \text{ is assigned to cluster 2 } (V_2 = V \setminus V_1) \end{cases}$$

and let $X = [x_u]$ be the $1 \times n$ column vector of these selection variables such that $X = \sum_{i=1}^{n} a_i \mathbf{u}_i$ with $a_i = \mathbf{u}_i^{\mathrm{T}} X$. Then, it can be shown that $\mathsf{M}(S) = \dfrac{1}{4m} \displaystyle\sum_{i=1}^{n} (\mathbf{u}_i^{\mathrm{T}} X)^2 b_i$. Thus, one would like to select X proportional to the eigenvector with the largest eigenvalue to maximize $\mathsf{M}(S)$. However, such an eigenvector will in general have entries that are *not* ± 1 but real values. This would therefore require exploring some non-trivial "rounding scheme" for such an eigenvector to covert the real values of the components of the eigenvector to ± 1 such that the new value of objective does not decrease too much; currently no such rounding scheme is known.

This approach can also be applied to the Mc problem by using the same approach recursively to decompose the clusters V_1 and V_2 adjusting the objective function to reflect the fact that certain edges have been deselected by the partitioning, and continuing in this fashion until the modularity value cannot be improved further.

3.4.4 Quadratic Programming Based Approach

Using the fact that $\mathsf{OPT}_2 \geq \mathsf{OPT}/2 \geq \mathsf{OPT}_\kappa/2$ for any $\kappa > 2$, it follows that an algorithm for 2-Mc having an approximation ratio of ε also provides an algorithm for κ-Mc having an approximation ratio of 2ε. The quadratic programming based approach discussed in this section provides an approximation algorithm for 2-Mc, thereby also providing an approximation algorithm for κ-Mc for any $\kappa > 2$. As in the previous section, for every $u \in V$ let x_u be a selection variable defined as:

$$x_u = \begin{cases} -1, & \text{if } u \text{ is assigned to cluster 1 } (V_1) \\ 1, & \text{if } u \text{ is assigned to cluster 2 } (V_2 = V \setminus V_1) \end{cases}$$

Then, since $\sum_{u,v\in V}\left(a_{u,v}-\frac{d_u d_v}{2m}\right)=0$, Equation (1) can be rewritten for the 2-Mc problem as

$$\mathsf{M}(\mathcal{S})=\frac{1}{4m}\left(\sum_{u,v\in V} w_{u,v}(1+x_u x_v)\right)=\frac{1}{4m}\sum_{u,v\in V} w_{u,v}x_u x_v=\mathbf{x}^{\mathrm{T}}W\mathbf{x} \tag{4}$$

where $w_{u,v}=\frac{a_{u,v}-\frac{d_u d_v}{2m}}{4m}$, $W=[w_{u,v}]\in\mathbb{R}^{n\times n}$ is the corresponding symmetric matrix of $w_{u,v}$'s and $\mathbf{x}\in\{-1,1\}^n$ is a column vector of the indicator variables. Note that the $w_{u,v}$ values can be positive or negative, but $w_{u,u}=-\frac{d_u^2}{2m}$ is always negative.

Equation (4) describes a quadratic form with *arbitrary real* coefficients. As a first attempt, one might be tempted to use an existing semi-definite programming (SDP) based approximation on quadratic forms to obtain an efficient algorithm. However, a direct application of many previously known results on SDP based approximation is not possible. For example, the results in [10] cannot be directly applied since the diagonal entries $w_{u,u}$ are negative, the results in [40] cannot be directly applied since the coefficient matrix W is not necessarily positive-semidefinite, and even the elegant results on Grothendieck's inequality in [4] cannot be applied because we do not have a bipartition of the nodes.

However, the authors in [15] was able adopt the techniques in [4, 10] in a non-trivial manner to provide a *randomized* approximation algorithm with an approximation ratio of ρ, where

$$\mathbb{E}[\rho]=\begin{cases}8.4\ln d=\mathrm{O}(\log d), & \text{if } G \text{ is a } d\text{-regular graph with } d<\frac{n}{2\ln n}\\ \mathrm{O}(\log d_{\max}), & \text{if } d_{\max}, \text{ the maximum degree over all nodes, is at most } \frac{\sqrt[5]{n}}{16\ln n}\end{cases}$$

We briefly outline the proof for the $\mathrm{O}(\log d)$ bound when G is d-regular with $d<\frac{n}{2\ln n}$. Consider the matrix $W'=\left[w'_{u,v}\right]$ where $w'_{u,v}=\begin{cases}0, & \text{if } u=v\\ w_{u,v}, & \text{otherwise}\end{cases}$. First, it is shown that if $\mathsf{OPT}_2=\max_{\mathbf{x}\in\{-1,1\}^n}\mathbf{x}^{\mathrm{T}}W\mathbf{x}$ and $\mathsf{OPT}'_2=\max_{\mathbf{x}\in\{-1,1\}^n}\mathbf{x}^{\mathrm{T}}W'\mathbf{x}$ then $\mathsf{OPT}'_2>\mathsf{OPT}_2-\frac{1}{n}$. Then, the following lower bound on OPT_2 is derived:

$$\mathsf{OPT}_2>\begin{cases}0.13/\sqrt{d}, & \text{if } n>40d^9\\ \frac{0.43}{d}-\frac{2}{n}, & \text{otherwise}\end{cases}$$

This shows that it suffices to approximate OPT'_2. Note that the diagonal entries of the matrix W' are now zeroes and $\mathsf{OPT}'_2=\Omega(1/d)$. Next, we utilize the following algorithmic result on quadratic forms proven in [4, 10]. Consider the following randomized approximation algorithm:

Randomized approximation algorithm in [4, 10] for computing
$$\mathrm{OPT}'_2 = \max_{\mathbf{x}\in\{-1,1\}^n} \mathbf{x}^{\mathrm{T}}\mathbf{W}'\mathbf{x} = \max_{\forall \mathbf{u}:\, \mathbf{x_u}\in\{-1,1\}} \sum_{\mathbf{u},\mathbf{v}\in\mathbf{V}} \mathbf{w}'_{\mathbf{u},\mathbf{v}}\mathbf{x_u}\mathbf{x_v}$$

1. Solve the following maximization problem

$$\begin{aligned} &\textit{maximize} \sum_{\substack{u,v\in V \\ u\neq v}} w'_{u,v} X_u X_v \\ &\textit{subject to} \\ &\quad \forall u \in V:\ X_u \in \mathbb{R}^n \\ &\quad \forall u \in V:\ X_u \text{ is a symmetric positive semi-definite matrix} \end{aligned}$$

in polynomial time using the semidefinite programming approach[4]. Let the solution vectors be X^*_u for $u \in V$.

2. Select a suitable real number $T > 1$.

3. Let $\mathbf{r}$ be a vector selected uniformly over the n-dimensional unit-norm hypersphere.

4. Set $x_u = \begin{cases} 1, & \text{if } \mathbf{Y}_u\mathbf{r} > T \\ -1, & \text{if } \mathbf{Y}_u\mathbf{r} < -T \end{cases}$

Otherwise, if $-T \le \mathbf{Y}_u\mathbf{r} \le T$, set $x_u = \begin{cases} 1 & \text{with probability } \frac{1}{2} + \frac{\mathbf{Y}_u\mathbf{r}}{2T} \\ -1 & \text{with probability } \frac{1}{2} - \frac{\mathbf{Y}_u\mathbf{r}}{2T} \end{cases}$

5. Return $\{x_u \mid u \in V\}$ as the solution.

The bounds in [4, 10] imply that the above algorithm return a solution satisfying

$$\mathbb{E}\left[\sum_{u,v\in V} w'_{u,v}x_u x_v\right] \ge \frac{\mathrm{OPT}'_2}{T^2} - 8\,\mathrm{e}^{-T^2/2}\left(\sum_{u,v\in V}\left|w'_{u,v}\right|\right)$$

The proof can then be completed by showing that $\sum_{u,v\in V}\left|w'_{u,v}\right| < 2$ and selecting $T = \sqrt{4\ln d}$.

3.4.5 Other Heuristic Approaches

Other approaches for solving the Mc problem include:

- simple heuristics without any guarantee of performance, and
- simulated-annealing type approaches that are exhaustive and slow [22] and therefore difficult to apply to to large-scale networks with thousands of nodes.

[4] See [48, Chapter 26].

3.5 Extensions to Directed or Weighted Networks

An extension of the basic modularity clustering to a more general weighted directed network is easy and was done by Leicht and Newman [32] in the following manner. Suppose that our input is a directed weighted graph $G = (V, E, w)$ of n nodes where $w\colon E \mapsto {}^{+}$ denotes a function giving a positive weight to every edge in E, and let $A = [a_{u,v}]$ denote the weighted adjacency matrix of G, $\left(i.e.,\ a_{u,v} = \begin{cases} w(u,v), & \text{if } (u,v) \in E \\ 0, & \text{otherwise} \end{cases} \right)$. Let $d_u^{\text{in}} = \sum_{(v,u)\in E} w(v,u)$ and $d_u^{\text{out}} = \sum_{(u,v)\in E} w(u,v)$ denote the *weighted* in-degree and the *weighted* out-degree of node u, respectively, and let $m = \sum_{(u,v)\in E} w_{u,v}$ denote the sum of weights of all the edges. Then, Equation (1) computing the modularity value of a cluster $C \subseteq V$ needs to be modified as

$$\mathsf{M}(C) = \frac{1}{m}\left(\sum_{u,v\in C} \left(a_{u,v} - \frac{d_u^{\text{out}} d_v^{\text{in}}}{m} \right) \right)$$

The authors in [15] showed that with some effort almost all our computational complexity results for modularity clustering on undirected networks can be extended to directed weighted networks.

4 Other Model-based Graph Decomposition

In this section we discuss a few other choices for the ($C1$)–($C3$) items for model-based graph decomposition.

4.1 Alternate Null Models (Alternate Choices for ($C1$))

A natural objection to the basic modularity clustering is that the background degree-dependent null model may not be appropriate in all applications. We discuss a few other choices that have been explored in the literature.

4.1.1 Scale-free Null Model

The choice of the linear preferential attachment model for the class of scale-free networks [6] may not be an appropriate choice since Karrer and Newman [27] showed that this may not provide a new null model. However, it is still an open question as to whether other generative models for scale-free networks, such as the "copy" model by Kumar *et al.* [30] in which new nodes choose an existing

node at random and copy a fraction of the links of this node, provide a new and useful null model.

4.1.2 Classical Erdös-Rényi Null Models

A theoretically appealing choice is the classical Erdös-Rényi random graph model, *e.g.*, the random graph $G(n,p)$ in which each possible edge $\{u,v\}$ is selected uniformly and randomly with a probability of p. Although the Erdös-Rényi model has a rich and beautiful theory [7] with significant applications in other areas of computer science, it is by now agreed upon that such a model may be inadequate in many social and biological network applications. Nonetheless, a formal investigation of such a null model is of independent theoretical interest, and may provide insight regarding the properties that an appropriate null model must satisfy. If p is selected such that the expected number of edges of the random graph is equal to the number of edges of the given graph, then maximizing modularity with this new null model is precisely the same as maximizing modularity in an appropriate regular graph [15]; otherwise, however, it is not clear what the complexity of computing this new modularity value is.

4.1.3 Application Specific Null Models

Sometimes null models motivated by specific applications in biology and social sciences are used by the researchers. Two such null models are described next.

Null Models for Transcriptional and Signaling Biological Networks

One of the most frequently reported topological characteristics of such networks is the distribution of in-degrees and out-degrees of nodes, which is close to a power-law or a mixture of a power law and an exponential distribution [2, 20, 33]. Specifically, in biological applications, metabolic and protein interaction networks are heterogeneous in terms of node degrees and exhibit a degree distribution that is a mixture of a power law and an exponential distribution [2, 20, 24, 33, 34], whereas transcriptional regulatory networks exhibit a power-law out-degree distribution and an exponential in-degree distribution [31, 44]. Based on these types of known topological characterizations, Albert *et al.* [3] suggested some degree distributions and network parameters for generating random transcriptional and signaling networks for the null model. Random networks with prescribed degree distributions can be generated in a variety of ways, *e.g.*, by using the method suggested by Newman, Strogatz and Watts in [39].

Markov-chain Null Model

In this method, a random network for the null model is generated by starting with the given input network $G = (V, E)$ and repeatedly swapping randomly chosen pairs of connections in the following manner [25]:

---

repeat

- Select two edges, $\{a, b\}$ and $\{c, d\}$ randomly and uniformly among all edges in E.
- **If** $a = c$ or $b = d$ or $\{a, d\} \in E$ or $\{b, c\} \in E$
  then discard this pair of edges
  else add the edges $\{a, d\}$ and $\{b, c\}$ to E
  delete the edges $\{a, b\}$ and $\{c, d\}$ from E

until a specified percentage of edges of G has been replaced

4.2 Alternate Fitness Measures (Alternate Choices for (C2)–(C3))

Exact or approximate solutions to the modularity measure as described by (1) may tend to produce many trivial clusters of single nodes. For example, DasGupta and Desai in [15] showed that if the maximum node degree $d_{\max}$ of G satisfies $d_{\max} < \frac{\sqrt[5]{n}}{16 \ln n}$, then there is a clustering in which every cluster except one consists of a single node and the modularity value is at least 25% of the optimal. One reason for such a consequence is due to the fact that the fitness measure for a modularity clustering is the *sum* of fitnesses of individual clusters (*i.e.*, for a clustering $\mathcal{S} = \{V_1, V_2, \ldots, V_k\}$, $\mathrm{M}(\mathcal{S})$ is the summation of $\mathrm{M}(V_i)$'s), and one moderately large cluster sometimes over-compensates the negative effects of many small clusters.

Based on these observations, it is reasonable to investigate other suitable choices of the function that combines the individual fitness values into a global fitness measure without sacrificing the quality of the optimal decomposition. Some reasonable choices include the max-min objective, namely $\mathrm{M}^{\text{max-min}}(\mathcal{S}) = \min_{V_i \in \mathcal{S}} \mathrm{M}(V_i)$, and the average objective, namely $\mathrm{M}^{\text{average}}(\mathcal{S}) = \frac{\sum_{i=1}^{k} \mathrm{M}(V_i)}{k}$. DasGupta and Desai investigated the max-min objective in [15] and showed that the max-min objective indeed avoids generating small-size trivial clusters and the optimal objective value for max-min objective is precisely scaled by a factor of 2 from that of the objective of the basic modularity clustering, thereby keeping the overall quantitative measure the same

5 Conclusion and Further Research

There is still a large gap between the 1.0006 factor inapproximability result and logarithmic factor approximation algorithm known for modularity clustering problems. Designing better scalable algorithms for these problems would enable one to apply this method to much larger networks than that is currently done. A few interesting directions for future algorithmic research are as follows:

- Is it possible to do a non-trivial analysis of the spectral partitioning approach discussed in Section 3.4.3, perhaps by using the techniques presented in analysis of the spectral method for MAX-CUT such as in [47] ?
- Is it possible to augment the ILP formulation for modularity clustering as discussed in Section 3.4.2 with additional redundant constraints using the cutting plane approach [29] to decrease the integrality gap substantially and perhaps thereby obtaining an improved approximation algorithm ?

Acknowledgements The author was supported by NSF grant IIS-1160995.

References

1. G. Agarwal and D. Kempe, *Modularity-Maximizing Graph Communities via Mathematical Programming*, European Physics Journal B, 66/3, 2008.
2. R. Albert and A.-L. Barabási. *Statistical mechanics of complex networks*, Reviews of Modern Physics, 74 (1), 47-97, 2002.
3. R. Albert, B. DasGupta, R. Dondi, S. Kachalo, E. Sontag, A. Zelikovsky and K. Westbrooks. *A Novel Method for Signal Transduction Network Inference from Indirect Experimental Evidence*, Journal of Computational Biology, 14 (7), 927-949, 2007.
4. N. Alon and A. Naor. *Approximating the cut-norm via Grothendieck's inequality*, proceedings of the 36th ACN Symposium on Theory of Computing, 72-80, 2004.
5. N. Bansal, A. Blum, and S. Chawla. *Correlation clustering*, Machine Learning, 56 (1-3), 89-113, 2004.
6. A.-L. Barabási and R. Albert. *Emergence of scaling in random networks*, Science, 286, 509-512, 1999.
7. B. Bollobás. *Random Graphs* (2nd ed.), Cambridge University Press, 2001.
8. U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski and D. Wagner. *On Modularity Clustering*, IEEE Transaction on Knowledge and Data Engineering, 20 (2), 172-188, 2007.
9. M. Charikar, V. Guruswami, and A. Wirth. *Clustering with qualitative information*, proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 524-533, 2003.
10. M. Charikar and A. Wirth. *Maximizing quadratic programs: extending Grothendieck's inequality*, proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 54-68, 2004.
11. M. Chlebík and J. Chlebíková. *Complexity of approximating bounded variants of optimization problems*, Theoretical Computer Science, 354 (3), 320-338, 2006.
12. T. Coleman, J. Saunderson and A. Wirth. *Local-Search 2-Approximation for 2-Correlation-Clustering*, proceedings of the 16th annual European symposium on Algorithms, LNCS, 5193, 308-319, 2008.

13. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, 2nd edition, MIT Press, 2001.
14. L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. *Comparing community structure identification*, Journal of Statistical Mechanics, P09008, 2005.
15. B. DasGupta and D. Desai. *Complexity of Newman's Community Finding Approach for Social Networks*, Journal of Computer and System Sciences, 79 (1), 50-67, 2013.
16. B. DasGupta, G. Andres Enciso, E. Sontag and Y. Zhang. *Algorithmic and Complexity Results for Decompositions of Biological Networks into Monotone Subsystems*, Biosystems, 90 (1), 161-178, 2007.
17. G. W. Flake, S. R. Lawrence, C. L. Giles, and F. M. Coetzee. *Self-organization and identification of Web communities*, IEEE Computer, 35, 66-71, 2002.
18. S. Fortunato and M. Barthélemy. *Resolution limit in community detection*, Proc. Natl. Acad. Sci., 104(1), 36-41, 2007.
19. M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.
20. L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. McDaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath, N. Ioime, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. DaSilva, J. Zhong, C. A. Stanyon, R. L. Finley, K. P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shimkets, M. P. McKenna, J. Chant and J. M. Rothberg. *A protein interaction map of Drosophila melanogaster*, Science, 302 (5651), 1727-1736, 2003.
21. M. Girvan and M. E. J. Newman. *Community structure in social and biological networks*, Proc. Natl. Acad. Sci, 99, 7821-7826, 2002.
22. R. Guimer'a, M. Sales-Pardo and L. A. N. Amaral. *Classes of complex networks defined by role-to-role connectivity profiles*, Nature Physics, 3, 63-69, 2007.
23. J. Hann and M. Kamber. *Data Mining: Concepts and Techniques*, Morgan Kaufman Publishers, 2000.
24. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A.-L. Barabási. *The large-scale organization of metabolic networks*, Nature, 407, 651-654, 2000.
25. R. Kannan, P. Tetali and S. Vempala. *Markov-chain algorithms for generating bipartite graphs and tournaments*, Random Structures and Algorithms, 14, 293-308, 1999.
26. N. Karmarkar. *A new polynomial-time algorithm for linear programming*, Combinatorica, 4, 373-395, 1984.
27. B. Karrer and M. E. J. Newman. *Random graph models for directed acyclic networks*, Physical Review E, 80, 046110, 2009.
28. D. Kefeng, Z. Ping and Z. Huisha. *Graph Separation of* 4*-regular Graphs is* NP*-complete*, Journal of Mathematical Study, 32 (2), 1999.
29. J. E. Kelley, Jr. *The Cutting-Plane Method for Solving Convex Programs*, Journal of the Society for Industrial and Applied Mathematics, 8 (4), 703-712, 1960.
30. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins and E. Upfal. *Stochastic models for the web graph*, proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 57-65, 2000.
31. T. I. Lee, M. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford and R. A. Young. *Transcriptional regulatory networks in Saccharomyces cerevisiae*, Science, 298 (5594), 799-804, 2002.
32. E. A. Leicht and M. E. J. Newman. *Community Structure in Directed Networks*, Physical Review Letters, 100, 118703, 2008.
33. S. Li, C. M. Armstrong, N. Bertin, H. Ge, S. Milstein, M. Boxem, P.-O. Vidalain, J.-D. J. Han, A. Chesneau, T. Hao, D. S. Goldberg, N. Li, M. Martinez, J.-F. Rual, P. Lamesch, L. Xu, M. Tewari, S. L. Wong, L. V. Zhang, G. F. Berriz, L. Jacotot, P. Vaglio, J. Reboul, T.

Hirozane-Kishikawa, Q. Li, H. W. Gabel, A. Elewa, B. Baumgartner, D. J. Rose, H. Yu, S. Bosak, R. Sequerra, A. Fraser, S. E. Mango, W. M. Saxton, S. Strome, S. van den Heuvel, F. Piano, J. Vandenhaute, C. Sardet, M. Gerstein, L. Doucette-Stamm, K. C. Gunsalus, J. W. Harper, M. E. Cusick, F. P. Roth, D. E. Hill and M. Vidal. *A map of the interactome network of the metazoan C. elegans*, Science, 303, 540-543, 2004.

34. A. Ma'ayan, S. L. Jenkins, S. Neves, A. Hasseldine, E. Grace, B. Dubin-Thaler, N. J. Eungdamrong, G. Weng, P. T. Ram, J. J. Rice, A. Kershenbaum, G. A. Stolovitzky, R. D. Blitzer and R. Iyengar. *Formation of regulatory patterns during signal propagation in a Mammalian cellular network*, Science, 309 (5737), 1078-1083, 2005.
35. M. E. J. Newman. *The structure and function of complex networks*, SIAM Review, 45, 167-256, 2003.
36. M. E. J. Newman. *Detecting community structure in networks*, Eur. Phys. J. B, 38, 321-330, 2004.
37. M. E. J. Newman. *Modularity and community structure in networks*, Proc. Natl. Acad. Sci., 103, 8577-8582, 2006.
38. M. E. J. Newman and M. Girvan. *Finding and evaluating community structure in networks*, Physical Review E, 69, 026113, 2004.
39. M. E. J. Newman, S. H. Strogatz and D. J. Watts. *Random graphs with arbitrary degree distributions and their applications*, Physical Review E, 64 (2), 026118-026134, 2001.
40. Y. Nesterov. *Semidefinite relaxation and nonconvex quadratic optimization*, Optimization Methods and Software, 9, 141-160, 1998.
41. A. Pothen, D. H. Simon and K. P. Liou. *Partitioning sparse matrices with eigenvectors of graphs*, SIAM Journal of Matrix Analysis and Applications, 11, 430-452, 1990.
42. U. N. Raghavan, R. Albert and S. Kumara. *Near linear time algorithm to detect community structures in large-scale networks*, Physical Review E, 76, 036106, 2007.
43. E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai and A.-L. Barabási. *Hierarchical Organization of Modularity in Metabolic Networks*, Science, 297 (5586), 1551-1555, 2002.
44. S. S. Shen-Orr, R. Milo, S. Mangan and U. Alon. *Network motifs in the transcriptional regulation network of Escherichia coli*, Nature Genetics, 31, 64-68, 2002.
45. H. D. Simon and S. H. Teng. *How good is recursive bisection*, SIAM Journal on Scientific Computing, 18, 1997.
46. C. Swamy. *Correlation clustering: maximizing agreements via semidefinite programming*, proceedings of the 15th annual ACM-SIAM symposium on Discrete algorithms, 526-527, 2004.
47. L. Trevisan. *Max cut and the smallest eigenvalue*, proceedings of the 41st ACM symposium on Theory of computing, 263-272, 2009.
48. V. Vazirani. *Approximation Algorithms*, Springer-Verlag, 2001.