A Local Approach for Identifying Clusters in Networks.

Sumit Singh *
Indian Institute of Technology Kanpur,
Kanpur, India.
singhsumitbit@gmail.com

Abstract

Graph clustering is a fundamental problem that has been extensively studied both in theory and practice. The problem has been defined in several ways in the literature and most of them have been proven to be NP-Hard. Due to their high practical relevancy, several heuristics for graph clustering have been introduced which constitute a central tool for coping with NP-completeness, and are used in applications of clustering ranging from computer vision, to data analysis, to learning. There exist many methodologies for this problem, however most of them are global in nature and are unlikely to scale well for very large networks. In this paper, we propose two scalable local approaches for identifying the clusters in any network. We further extend one of these approaches for discovering the overlapping clusters in these networks. Some experimentation results obtained for the proposed approaches are also presented.

1 Introduction

Identifying clusters in complex networks is difficult and has attracted a lot of attention in recent years as well as in the past, especially in computer science community, statistical physics community [31, 28], biological community [39] and applied mathematics community. Popular examples include the food webs, metabolic networks [4, 39], protein interactions [37], Internet and world wide web [7, 10], communication and distributed networks, and the social networks [12, 29]. To extract clusters in such networks, one typically chooses an objective function that captures the above intuition of a cluster/community as a set of nodes with better internal connectivity than external connectivity. Generally, the objective is typically NP-hard to optimize exactly [24, 12], one employs heuristics [12, 8] or approximation algorithms [3] to find sets of nodes that approximately optimize the objective function and that can be understood or interpreted as ‘real’ communities.

In this work, we have explored random walk based techniques and tried to improve their performance by some heuristics which are very popular in statistical physics community. The heuristics seems to help algorithm converge faster than usual on several standard datasets containing upto 30 million nodes.

1.1 Related Work

The earliest attempts dating back to the seventies were based on local search, with the Kernighan-Lin heuristic [17] achieving the best results in this class. Spectral methods [14, 16, 36], which found widespread use in the eighties, compute eigenvectors, which are embeddings of the graph onto the real line. In the early nineties, the linear programming methods were shown to find better cuts [22] than Kernighan-Lin and the spectral method but were too slow to be used. In the mid-nineties, a multilevel heuristic embodied in the package METIS was introduced [16]. This method is very fast and produces cuts that are almost as good as those obtained by linear programming based methods in practice.

An optimization version of clustering, Sparsest Cut Problem has been extensively studied. The Sparsest cut problem is to bipartition the vertices so as to minimize the ratio of the number of edges across the cut divided by the number of vertices in the smaller half of the partition. This problem has been

*Supported by MHRD, Govt. of India. PS: The work was under-progress but is stopped due to some reasons.
proven to be NP-Complete [24, 11]. Leighton and Rao [24] used linear programming to obtain $O(\log n)$ approximation of the sparsest cut. Arora, Rao and Vazirani [3] improved this to $O(\sqrt{\log n})$ through semi-definite programming. Faster algorithms obtaining similar guarantees have been constructed by Arora, Hazan and Kale [1], Khandekar, Rao and Vazirani [18], Arora and Kale [2], and Orecchia, Schulman, Vazirani, and Vishnoi [33]. Other related problems which have been extensively studied by theoreticians are Balanced Min-cut problem and Densest Subgraph problem.

In early 2000, some min-cut based techniques were suggested that essentially iteratively finds the min-cut in the graph and removes it to form partitions [11, 13]. Few years back, modularity based approach suggested by Newman [12] from statistical physics community gained a lot of popularity mainly because of being practical and being able to quantify the goodness of a cluster. Later it was shown that maximizing modularity is an NP-Hard problem [6] however the naive greedy approach has been shown to work well in practice.

In many networks such as social networks or author collaborative network, one expects a node to be in more than cluster. Hence we further look onto to unveil this overlapping structure in the networks. One of the first attempts to obtain the overlapping community structure of a graph appears in [21, 35]. The approach in [35] is based on retrieving all cliques of the graph; however, this operation turns out to be prohibitive for large graphs. A more efficient algorithm is given in [21], which finds communities by maximizing a certain fitness function. Recently, a very comprehensive study is carried out by Kovcs, Palotai, Szalay and Csermely [19] where they propose and evaluate the efficiencies of several methods that aim to discover the overlapping community structure in both directed and undirected graphs.

### 1.2 Local Clustering

A graph algorithm is a local algorithm if given a particular vertex as input, at each step only examines the vertices connected to those it has seen before. The use of a local algorithm naturally leads to the question of in which order one should explore the vertices of a graph. While it may be natural to explore vertices in order of shortest-path distance from the input vertex, such an ordering is a poor choice in graphs of low-diameter, such as social network graphs [25]. A natural inclination is then to choose neighbours randomly or the vertices which come first in the random walk. Our proposed approaches are based on this inclination which has been used for clustering in the past [35].

Our choice of choosing vertices randomly or considering probability distributions is further strengthened by the following results. Earlier, Meila and Shi [28, 27] had shown that the normalized cut of a graph, one of the variants of graph conductance, can be expressed in terms of the transition probabilities and the stationary distribution of a random walk in the graph, thus linking the mathematics of random walks to those of cut-based clustering. Orponen and Schaeffer [34] in turn expressed the absorption times of a random walk in a graph in terms of the eigenvectors of the graph’s Laplacian and use their locally computable approximation of the Fiedler vector [32] to obtain an approximation of the absorption times. This links the random walks to spectral clustering, which relates to cut-based methods. Recently, Teng [20] introduced a Laplacian Paradigm for designing nearly linear time and scalable algorithms which discusses a lot of theoretical aspects of these approaches.

### 2 Problem Definition

Let $G = (V, E)$ be any undirected network, then a cluster $S$ of $G$ is a subset of $V$ that is richly intra-connected but sparsely connected with the rest of the graph. The quality of a cluster can be measured by conductance which is defined as the ratio of the number of external connections to the total number of connections.

Let $E(S, V - S)$ be the set of edges where an edge $e = (u, v)$ belongs to this set if $u \in S$ and $v \in V - S$ or vice versa. Let $d_i$ be the degree of $i^{th}$ vertex, then the volume $\mu(S)$ of any set $S \subseteq V$ is defined as sum of degrees of every vertex in $S$. The conductance is formally defined as:

$$\alpha(S) = \frac{|E(S, V - S)|}{\min\{\mu(S), \mu(V - S)\}}$$
The conductance of the graph $G$ is defined as the minimum conductance over the conductance of all subsets $S$ of $V$. The subset $S$ defines the cut and the partitions defined are $S$ and $V - S$. Clustering is thus presented as an optimization problem: Given an undirected graph $G$ and a conductance parameter, find a cluster $C$ such that $\Phi(C) \leq \phi$, or determine no such cluster exists.

**Graph Preprocessing**

We have introduced self-loops in the transition probability of the given graph $G$ and the graph is therefore assumed to be ergodic. A graph is said to be ergodic if aperiodic and irreducible. Introducing self-loops ensures aperiodicity and the fact that graph is undirected and connected ensures that it is irreducible.

## 3 Proposed Approach

The paper suggests two approaches for the clustering problem and extends one of these approaches to discover overlapping clusters. The underlying observation behind these approaches is that a random walk started at some vertex $v$ will initially be trapped in the cluster to which $v$ belongs. Thus carrying out this process again and again for walks of some length, we will revisit the nodes in the cluster of $v$ more frequently. One of the primary motivation of such random walk based algorithms comes from the fact that was implicit in the analysis of the volume estimation algorithm of Lovasz and Simonovits [26] that one can find a cut with small conductance from the distributions of the steps of the random walk starting at any vertex from which the walk does not mix rapidly. However when the algorithm based on above intuitions were carried by us on various networks, it was observed that a large number of iterations are required to find good clusters and to ensure convergence. We therefore extend this naive approach so as to give good results in less time.

### 3.1 Distribution Based Approach

Informally, conductance of any set $S$ defines the probability of going out of set $S$ given that we are in set $S$. Since we do not know the cluster $S$ apriori, we try to estimate set $S$ by trapping ourself heuristically within some set $X$. The set $X$ can be thought of as an estimation of $S$ that is likely to improve as the algorithm progresses. The approach considers a probability distribution vector which is initially concentrated on the vertex $v$ whose cluster we are trying to find. This distribution is then evolved based on the transition probabilities of the network. The transition probability from any node $x$ to any node $y$ are strongly aperiodic and is defined as follows for any network:

$$P[x \rightarrow y] = \begin{cases} \frac{1}{2d_x} & \text{if } x = y \\ \frac{1}{2d_x} & \text{if } y \text{ is a neighbour of } x \\ 0 & \text{otherwise} \end{cases}$$

For the exposition of the ideas the approach is formally given below in a more readable manner without ignoring the implementative issues.

**Algorithm 1 Distribution-Based Approach (vertex v)**

1. Construct the probability matrix $P$.
2. Initialize distribution[] vector to 0 except at index $v$ which is set to 1.
3. while(Condition) do
   a) For each vertex $w$ such that distribution[$w$] > 0, do
      i) For each vertex $u$ neighbouring to $w$, increase distribution[$u$] by $\frac{1}{2d_w}$
   b) Truncate(distribution,$v$,\alpha).

The initialization step in the algorithm is basically capturing the fact that the random walk starts at vertex $v$ and as the walk progresses the probability of walker to be at different states is captured in distribution vector. Note that in each step a truncation step is performed. In absence of truncation, the distribution vector will converge to stationary distribution vector of the network in a long run. The truncation ensures two things: one is that the experiment remains local i.e. the random walk is allowed to diffuse in the network in a constrained manner and the other is that instead of starting a new walk as was done in previous approaches, we reinforce the elements that seems going out of cluster/community.
Algorithm 2 Truncate(distribution[], vertex v, parameter \(\alpha\))

1: tmp = 0;
2: For each vertex w
   if distribution[w] \(\geq\) \(\alpha\) * distribution[v]
      a) tmp += distribution[w].
      b) distribution[w] = 0.
3: distribution[v] += tmp.

back into it. Thus this reduces the number and length of run(s) of the experiments. The parameters in truncation plays a dominating role for controlling the cluster size of the network. If the parameter is small then vertices having less belonging to that cluster will also be included in the cluster. We also define a belongingness measure here for each vertex \(v\), which is the ratio of distribution of the node \(v\) to the distribution of the node which is presently the centre of cluster. Note that since the probability distribution is changing abruptly due to truncation step, the process described above doesn’t seem to be associated with a single/fixed random walk. However, we observe experimentally that the values of distribution vector converge for a fixed value of parameter \(\alpha\).

3.2 Adaptive Walk Based Approach

This approach is inspired by Wang-Landau algorithm \[40\] from statistical physics community. The problem that we faced in random walk based naive approach is that the walk may escape through the cluster and so we need to perform it again from starting. These things are bound to happen in ergodic networks as any markov chain tries to mix such that the corresponding distribution achieve the stationary distribution. Therefore for a network having small or moderate mixing time, the naive approach may require high number of iterations. We observe that if we increase the incoming probability for those vertices which we have visited earlier then we may be able to trap the random walk into some local minima which would ideally be the cluster itself. Therefore the task is to redefine the probabilities at certain interval of walk, so that the random walk is trapped inside the cluster of the starting vertex. The Wang-Landau algorithm in some sense does the reverse of this by defining transition probabilities such that the random walk does not trap in local minima. For theoretical reasons, the Wang-Landau algorithm is not a markov process but has been adapted to a markov process recently. There are some converging issues with this algorithm for various networks which has been resolved in the physics community \[43\]. We have not yet studied the implication of similar modifications theoretically in our approach and is an interesting problem to look at.

In the adapted approach, each vertex has an associated energy value. The probability of transition from node \(u\) to node \(v\) is defined as:

\[
P(u \rightarrow v) = \min\left\{ \frac{\text{energy}[v]}{\text{energy}[u]}, 1 \right\}
\]

The new adapted approach is described as follows:

Algorithm 3 Adaptive_Walk_Based_Approach(vertex v, parameter f, parameter \(\alpha\), parameter \(\beta\))

1: For each vertex w: energy[w] = \(\frac{\alpha}{d_w}\).
2: energy[v] = \(\frac{\beta}{d_v}\).
3: current_vertex = v.
4: while(Condition) do
   a) next_vertex = random_walk(current_vertex, P).
   b) energy[current_vertex] = energy[current_vertex] * f.
   c) current_vertex = next_vertex.

The random_walk function above chooses the next vertex randomly based on the current vertex and the transition probability. The value of \(f\) in the Wang Landau algorithm was kept equal to \(e = 2.71828\ldots\).
and was then slowly decreased with further iterations. In the experiments, we performed we started with a low value of $f$ around 1.3 and then increased it to higher values. One natural question that emerges out of this is when should the value of $f$ be changed. Though there is no correct answer for this, it depends on the size of community we want to find. The algorithm is actually defining some loose boundary for lower values of $f$ then it is filtering again with higher values of $f$. After readjusting $f$, the current vertex is reset to the original vertex for which the algorithm was called. The values of parameter $\alpha$ and $\beta$ is chosen low (small constant) and high respectively. There are still some convergence issues in this approach for any arbitrary network(s) both experimentally and theoretically.

4 Overlapping Clusters

A node in a network may belong to more than one cluster in a network. These overlapping structures can easily be observed in our social lifes. Thus a natural extension to the clustering problem is to find the overlapping clusters in a network. We used Fuzzy C Means Clustering to resolve the problem and obtained good results. Fuzzy c-means is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1973 [9] and improved by Bezdek in 1981 [5]) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$F_m = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^m ||x_i - c_j||^2$$

over variables $u_{ij}$ and $c$ with $\sum_j u_{ij} = 1$. The degree of fuzzification is controlled by parameter $m \in [1, \infty)$. $u_{ij}$ is the degree of membership of $x_i$ in the cluster $j$. $x_i$ is the $i$th $d$-dimensional measured data point. $c_j$ is the dimension centre of the cluster $j$. $||*||$ denotes the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function. The first approach suggested in this paper defines distribution of each vertex $u$ when any vertex $v$ is chosen as its center. Several distribution value(s) are obtained for each node by varying the centres. These values are then treated as vector $x$ which is required in the objective function mentioned above. The cluster centre $c$ is also well defined in the above process. We then use $l_2$ norm as the similarity measure for our purpose. The degree of fuzzification was varied through some values till it ended in a local minima for the objective function defined above. The number of dimensions of vector $x$ can be varied here and for small networks, the best one of them can be chosen as a solution based on some objective function.

5 Experiments

In this section, we are not referring to the overlapping clustering scheme unless specifically mentioned. There are numerous evaluation criterion analogous to different objective functions which have been used in literature. When we talk of evaluation of any clustering algorithm, the following two issues are expected to be addressed:

1. How good is the quality of the clusters?

2. How quickly can these clusters be computed?

An inherent question in second issue is how much parallelizable/distributive/scalable is the algorithm?

5.1 Quality of Clusters

We use modularity to measure the goodness of the cluster(s). Given a graph $G = (V, E)$, where $m = |E|$, let $S \subseteq V$ and the number of edges whose both end-points are in $S$ be $m_S$. We define the modularity $f(S)$ of a set $S$ as $\frac{1}{2m_S}(m - E(m_S))$, where $E(m_S)$ is the expected number of edges in the random graph with same node degree sequence such that edges have both end-points in $S$.

We carried out experiments on some of the existing datasets to give an idea of how these approaches score against one of the best existing approach due to Newman. We carried out experiments on various
datasets and have listed some of these in table 1. The data sets are not very large datasets because the greedy algorithm based on maximizing modularity is an $O(n^2 \log n)$ time algorithm for sparse graphs. It is to be noted that there are some controlling parameters in the approaches suggested by us. These parameters become important when we are not sure of community sizes. Also in approach 2 there are some scaling factor decisions to be made whose timings we are ignorant of. This turns out to be the bottleneck problem for the performance of the approach 2.

5.2 Time Required to form Clusters

The datasets chosen for these experiments are from [30]. The results presented in this section are for orkut named dataset which has over 3 million nodes and around 118 million edges. All the results, we present have been obtained from a single core processor. The time required by the adaptive random walk based approach has not been discussed due to the uncertain parameters in this approach. This is because though the method is very local but it’s convergence is still an issue for massive networks.

The timing results for the distribution based approach is shown in figures 1, 2, 3, 4 followed by the same examples of convergence in figures 5, 6. All the figures 1, 2, 3, 4 show the average relation between the number of iterations and the time taken. The first iteration usually requires around 24 seconds due to file read operation. It is to be noted that the computation time increases with lower parameter value but later stabilize. Also if the community size is less than 2000, then the computational time required for finding one cluster is around 10 seconds. Thus it take few hours at most to naively compute all the clusters. Using architectures like CUDA or other parallel environments, we expect the algorithm to terminate within some minutes.

There is another and more important issue of convergence of these approaches. We note that for very large networks, the distribution based scheme does seem to converge well however the same is not true for the adaptive walk based scheme mainly due to the reason of being heavily parametrized. The figures 5, 6 show the convergence of values in distribution vector for a particular node in distribution based approach.

5.3 Overlapping Clusters

We consider the famous network from the social science literature, the 'karate club' of Zachary [42] (figure 7). The network is of particular interest because, shortly after the observation and construction of the network, the club in question split in two as a result of an internal dispute. We apply the fuzzy c-means based algorithm and discover a third overlapping community apart from these two which are mostly the vertices lying on boundaries of the two original sets (will update soon with coloured pictures).

6 Conclusion

We have proposed two approaches for the graph clustering problem in undirected graphs. There are still some improvements that can be expected in adaptive walk based approach which might improve its convergence. We also suggested an approach for finding overlapping clusters in these graphs. It would be interesting to see how these approaches with appropriate modifications can applied to directed graphs.

| Network Name  | Size | Newman | Approach 1 | Approach 2 |
|---------------|------|--------|------------|------------|
| karate club   | 34   | 0.42   | 0.42       | 0.42       |
| jazz musicians| 198  | 0.44   | 0.43       | 0.40       |
| email         | 1133 | 0.57   | 0.51       | 0.46       |
| physicist     | 27519| 0.72   | 0.68       | 0.39       |
Figure 1: Time vs Iterations (parameter value = $10^{-3}$)

Figure 2: Time vs Iterations (parameter value = $10^{-5}$)
Figure 3: Time vs Iterations (parameter value = $10^{-6}$)

Figure 4: Time vs Iterations (parameter value = $10^{-7}$)
Figure 5: Convergence vs Iterations (parameter value $= 10^{-5}$)

Figure 6: Convergence vs Iterations (parameter value $= 10^{-7}$)
Figure 7: Zachary Network

References

[1] S. Arora, E. Hazan, and S. Kale. $O(\sqrt{\log n})$ approximation to sparsest cut in $O(n^2)$ time. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pages 238–247, Washington, DC, USA, 2004. IEEE Computer Society.

[2] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, STOC ’07, pages 227–236, New York, NY, USA, 2007. ACM.

[3] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. J. ACM, 56:5:1–5:37, April 2009.

[4] A. Barabasi, B. Tombor, H. Jeong, R. Albert, and Z. Oltvai. The large-scale organization of metabolic networks. Nature, 407:651–654, 2000.

[5] J. C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[6] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. IEEE Trans. Knowl. Data Eng., 20(2):172–188, 2008.

[7] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. Comput. Netw., 33:309–320, June 2000.

[8] K. B. Dhillon I.S., Yuqiang Guan. Weighted graph cuts without eigenvectors a multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29:1944–1957, November 2007.

[9] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics, 3(3):32–57, 1973.

[10] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. SIGCOMM Comput. Commun. Rev., 29:251–262, August 1999.

[11] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. Computer, 35:66–71, March 2002.

[12] M. Girvan and M. Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, 19:7821–7826, 2001.
[13] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letter*, 76(4-6):175–181, 2000.

[14] B. Hendrickson and R. Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Sci. Comput.*, 16:452–469, March 1995.

[15] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: application in vlsi domain. In *Proceedings of the 34th annual Design Automation Conference*, DAC ’97, pages 526–529, New York, NY, USA, 1997. ACM.

[16] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20:359–392, December 1998.

[17] L. S. Kernighan B. W. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–307, 1970.

[18] R. Khandekar, S. Rao, and U. Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56:19:1–19:15, July 2009.

[19] I. Kovcs, R. Palotai, M. Szalay, and P. Csermely. Community landscapes: a novel, integrative approach for the determination of overlapping network modules. *PLoS ONE* 5, e12528, 2010.

[20] J. Kratochvíl, A. Li, J. Fiala, and P. Kolman, editors. *Theory and Applications of Models of Computation, 7th Annual Conference, TAMC 2010, Prague, Czech Republic, June 7-11, 2010. Proceedings*, volume 6108 of *Lecture Notes in Computer Science*. Springer, 2010.

[21] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure of complex networks. *NEW JOURNAL OF PHYSICS*, 11:033015, 2009.

[22] K. Lang and S. Rao. Finding near-optimal cuts: an empirical evaluation. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, SODA ’93, pages 212–221, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[23] T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA. MIT Press*, 2001.

[24] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46:787–832, November 1999.

[25] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *Proceedings of the 17th international conference on World Wide Web*, WWW ’08, pages 915–924, New York, NY, USA, 2008. ACM.

[26] L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Struct. Algorithms*, 4(4):359–412, 1993.

[27] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, San Francisco, CA, USA, 2001. Morgan Kauffman.

[28] J. Mendes and S. Dorogovtsev. Evolution of networks: From biological nets to the internet. *Advances in Physics*, 51:1079–1187, 2002.

[29] A. Mislove. *Online Social Networks: Measurement, Analysis, and Applications to Distributed Information Systems*. PhD thesis, Rice University, Department of Computer Science, May 2009.

[30] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC’07)*, San Diego, CA, October 2007.

[31] M. Newman. The structure and function of a complex networks. *SIAM Review*, 45:167–256, 2003.
[32] S. E. Nikoletseas, editor. *Experimental and Efficient Algorithms, 4th International Workshop, WEA 2005, Santorini Island, Greece, May 10-13, 2005, Proceedings*, volume 3503 of *Lecture Notes in Computer Science*. Springer, 2005.

[33] L. Orecchia, L. J. Schulman, U. V. Vazirani, and N. K. Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC ’08, pages 461–470, New York, NY, USA, 2008. ACM.

[34] P. Orponen, S. E. Schaeffer, and V. A. Gaytán. Locally computable approximations for spectral clustering and absorption times of random walks. *CoRR*, abs/0810.4061, 2008.

[35] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[36] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11:430–452, May 1990.

[37] J. Ruan and W. Zhang. Identifying hierarchical community structures in biological networks. *Algorithmic Biology*, 2006.

[38] S. van Dongen. *Graph clustering by flow simulation*. PhD thesis, Universiteit Utrecht, Utrecht, The Netherlands, 2000.

[39] A. Wagner and D. Fell. The small world inside large metabolic networks. *Proc Biol Sci.*, 268:1803–1810, 2001.

[40] F. Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Phys. Rev. Lett.*, 86:2050–2053, Mar 2001.

[41] J. Wiedermann, G. Tel, J. Pokorný, M. Biediková, and J. Stuller, editors. *SOFSEM 2006: Theory and Practice of Computer Science, 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merín, Czech Republic, January 21-27, 2006, Proceedings*, volume 3831 of *Lecture Notes in Computer Science*. Springer, 2006.

[42] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[43] C. Zhou and J. Su. Optimal modification factor and convergence of the wang-landau algorithm. *Phys. Rev. E*, 78:046705, Oct 2008.