

Brief Announcement: Distributed Sparsest Cut via Eigenvalue Estimation

Yannic Maus¹  

TU Graz, Austria

Tijn de Vos  

TU Graz, Austria

Abstract

We give new, improved bounds for approximating the sparsest cut value or in other words the conductance ϕ of a graph in the CONGEST model. As our main result, we present an algorithm running in $O(\log^2 n / \phi)$ rounds in which every vertex outputs a value $\tilde{\phi}$ satisfying $\phi \leq \tilde{\phi} \leq \sqrt{2.01\phi}$. In most regimes, our algorithm improves significantly over the previously fastest algorithm for the problem [Chen, Meierhans, Probst Gutenberg, Saranurak; SODA 25]. Additionally, our result generalizes to k -way conductance.

We obtain these results, by approximating the eigenvalues of the normalized Laplacian matrix $L := I - \text{Deg}^{-1/2} A \text{Deg}^{-1/2}$, where, A is the adjacency matrix and Deg is the diagonal matrix with the weighted degrees on the diagonal. We show our algorithms are near-optimal by proving a lower bound for computing the smallest non-trivial eigenvalue of L , even in the stronger LOCAL model

The previous state of the art sparsest cut algorithm is in the technical realm of expander decompositions. Our algorithms, on the other hand, are relatively simple and easy to implement. At the core, they rely on the well-known *power method*, which comes down to repeatedly multiplying the Laplacian with a vector. This operation can be performed in a single round in the CONGEST model.

All our algorithms apply to weighted, undirected graphs. Our lower bounds apply even in unweighted graphs.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Distributed algorithms

Keywords and phrases CONGEST, Sparsest Cut, Laplacian, Eigenvalues, Spectral Graph Theory

Digital Object Identifier 10.4230/LIPIcs.DISC.2025.60

Related Version *Full Version*: <http://arxiv.org/abs/2508.19898>

Funding This research was funded in whole or in part by the Austrian Science Fund (FWF) <https://doi.org/10.55776/P36280> and <https://doi.org/10.55776/I6915>. For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission.

Acknowledgements We would like to thank Leo Wennmann, for the many hours of discussion on this topic. We would like Joachim Orthaber and Malte Baumecker, for thinking along. We would like to thank Faith Ellen, Sebastian Brandt, Alexandre Nolin, and Eva Rotenberg, for the initial discussions that were the inspiration for this research. We would like to thank Schloss Dagstuhl, since the research was initiated during the Dagstuhl seminar *Graph Algorithms: Distributed Meets Dynamic*.

¹ The author ordering was randomized using <https://www.aeaweb.org/journals/policies/random-author-order/> generator. It is requested that citations of this work list the authors separated by  instead of commas.



© Yannic Maus and Tijn de Vos;

licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Distributed Computing (DISC 2025).

Editor: Dariusz R. Kowalski; Article No. 60; pp. 60:1–60:7

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A graph cut is a partition of the vertices into two disjoint subsets, where the size of the cut is the total number of edges crossing between them, and the sparsity of the cut measures how small this cut is relative to total edge weight of the smaller side, i.e., in comparison to the sum of its vertex degrees. Graph cuts are a central object in theoretical computer science and algorithm design. Cuts are especially important in communication networks because they represent the capacity for information to flow between different parts of the system. A sparse cut – where only a few edges connect two large subsets – can become a bottleneck, severely limiting bandwidth or throughput. A central question for guaranteeing the performance of network algorithms is to determine the minimum sparsity of cuts in a given communication network. For example the runtime of the famous PUSH-PULL protocol depends on the sparsest cut value [11]. More generally, graphs without sparse cuts support fast mixing of random walks [20], and enable efficient routing due to their strong connectivity [9, 10]. This motivates the use of expander decompositions, which partition a graph into *high-conductance* clusters (i.e, no sparse internal cuts), and sparse inter-cluster connections. Such decompositions are powerful algorithmic tools: thanks to efficient routing, many algorithms run significantly faster on high-conductance subgraphs, allowing problems to be solved locally within clusters before handling the sparse inter-cluster parts.

In this work, we present simple, time and bandwidth efficient distributed algorithms to approximate the sparsest cut value. Our randomized algorithms run in the classic CONGEST model of distributed computing where a communication network is abstracted as an n -vertex graph with vertices being computing entities and edges serving as bandwidth-limited communication links. Communication happens in synchronous rounds, in each of which each vertex can send a $O(\log n)$ -bit message to each of its neighbors and the complexity measure is the number of communication rounds until the vertices have computed their outputs, e.g., until they have output an approximate value of the sparsest cut.

Formally, the *sparsity* of a set (or cut) $S \subseteq V$ in a graph $G = (V, E)$ is defined as as

$$\phi(S) := \frac{|E(S, V \setminus S)|}{\min\{\text{Vol}(S), \text{Vol}(V \setminus S)\}} ,$$

where the volume $\text{Vol}(S)$ of a set of vertices S is the sum of of the degrees of the vertices in the set $\sum_{v \in S} \deg(v)$. The value $\phi(S)$ is also known as the *conductance* of the set S . The *sparsest cut* $\phi(G) \in [0, 1]$ of a graph G is the minimum sparsity of a subset $S \subseteq V$. This is also known as the *conductance* of the graph.

The relative notion $\phi(G)$ can be viewed as a normalized quantification of cut size. In contrast to computing minimum cuts, which quantify the absolute size of a cut, determining the sparsest cut or the conductance is not only NP-complete [17] but also APX-hard under the unique games conjecture [6]. Consequently, although no polynomial-time algorithm can approximate the optimum arbitrarily well (unless $P = NP$), a variety of approximation algorithms with provable – albeit weaker – guarantees have been developed across different computational models. Next, we present our results and compare them with the state of the art in the field. As the sparsest cut and the conductance have been extensively studied we provide further background and related work in the full version.

2 Our Contribution on Sparsest Cut Approximation

Our first result is an efficient CONGEST algorithm to approximate the sparsest cut.

► **Theorem 1.** *There is a randomized CONGEST algorithm that, given an undirected weighted n -node graph with sparsest cut ϕ , w.h.p. gives an approximation of the sparsest cut in $O(\log^2 n/\phi)$ rounds. In particular, every vertex outputs $\tilde{\phi}$ such that $\phi \leq \tilde{\phi} \leq \sqrt{2.01\phi}$.*

In general, there are two approaches for developing approximation algorithms for the sparsest cut, the *flow approach*, that we detail in the full version, and the *spectral approach*, which we use to prove Theorem 1. On a high-level, one computes the spectrum (or parts of it) of the graph, i.e., its eigenvalues, and then uses Cheeger's Inequality² to derive an approximation of $\phi(G)$ from the eigenvalues. Due to the limitations imposed by Cheeger's Inequality, the best possible approximation achievable through spectral methods is $\sqrt{2\phi}$. Our technical result is slightly stronger than Theorem 1. In fact, we can attain this theoretical bound – up to an arbitrarily small additive error – in $O(\log^2 n/\varepsilon^2 + \log n/\varepsilon^4)$ rounds. See the proof of Theorem 1 for the details.

For unweighted graphs, we compare our result against the previously best algorithm by Chen, Meierhans, Probst Gutenberg, and Saranurak that provides a $O(\phi \log^2 n)$ -approximation of ϕ [7]³. Our approximation is more precise than [7] when $\phi = \Omega(1/\log^4 n)$, and is faster in all cases: [7] takes $O(\text{poly log } n/\phi^4)$ rounds. Upon inspection of their $\text{poly log } n$, they have a total round complexity of at least $O(\log^{20} n)$ in the regime where their approximation is better.

Prior to [7], Das Sarma, Molla, and Pandurangan [19] gave a different algorithm to approximate the sparsest cut, which also works for weighted graphs. They computed an $O(\sqrt{\phi} \text{poly log } n)$ -approximation in $O(\frac{1}{b}(n + \frac{1}{\phi}) \log^2 n)$ rounds if the graph has a sparse cut of *balance* b , i.e., both sides S of the cut satisfy $\text{Vol}(S) \geq 2b|E|$. Building on the methods of [19], Kuhn and Molla improved the approximation ratio to $O(\sqrt{\phi \log n})$ and the runtime to $O(D + \frac{\log^2 n}{b\phi})$ [14]. For balanced cuts $b = \Omega(1)$, we improve their approximation ratio by a factor $\sqrt{\log n}$, while matching their running time. For unbalanced cuts, also their algorithm can take as much as $\Omega(n^2)$ time, so we do not only improve the approximation ratio, but also the running time exponentially.

k -Way Sparsest Cut. There exists a natural generalization of the sparsest cut to *multi-cuts* [15]: the k -way sparsest cut or k -way conductance ϕ_k is defined follows:

$$\phi_k := \min_{V_1, V_2, \dots, V_k \subset V} \max_{i=1, \dots, k} \frac{|E(V_i, V \setminus V_i)|}{\text{Vol}(V_i)},$$

where the minimum is taken over disjoint, non-empty V_i . Note that $\phi_2 = \phi$. The k -way conductance captures how well-connected a graph is across multiple parts; it can describe multi-partition bottlenecks, which standard conductance may miss. This is in particular relevant for real-world networks that usually consists of multiple clusters.

We prove the following theorem for approximating the k -way sparsest cut.

² Cheeger's Inequality [2, 1, 21] relates the second-smallest eigenvalue λ_2 of the normalized Laplacian to the graph's conductance ϕ , showing that $\frac{1}{2}\lambda_2 \leq \phi \leq \sqrt{2\lambda_2}$.

³ They do not state this result explicitly, but they implement the cut-matching game from [18] (see Appendix A) in the distributed setting. As stated in [18], this cut-matching game gives such an approximation.

► **Theorem 2.** *There is a randomized CONGEST algorithm that, given a constant $k \geq 2$ and undirected weighted graph with k -way sparsest cut ϕ_k , w.h.p. gives an approximation of the k -way sparsest cut in $O(\log^2 n \text{poly}(\phi_k^{-1}))$ rounds. In particular, every vertex outputs $\tilde{\phi}_k$ such that $\phi_k \leq \tilde{\phi}_k \leq O(\sqrt{\phi_k})$.*

To the best of our knowledge, we are the first to study the k -way conductance in the distributed setting.

Lower bounds on approximating sparse cuts. Generally, it is known that approximating the sparsest cut takes $\Omega(D)$ rounds. In particular, Das Sarma, Molla, and Pandurangan [19] show a $\tilde{\Omega}(D + \sqrt{n})$ CONGEST lower bound for any multiplicative approximation in weighted graphs. They reduce the spanning connected subgraph problem to a sparsest cut instance with edge weights that are either 0 or 1. This means that the lower bound shows that it is hard to decide whether $\phi = 0$ holds or not. For ϕ bounded away from zero, there is no such lower bound. Our upper bound of $O(\log^2 n/\phi)$ rounds, Theorem 1, indeed takes D rounds, since $D = O(\log n/\phi)$. However, for $\phi = \omega(1/\sqrt{n})$, the upper bound does not have a $\sqrt{n}$ -term.

We prove a $\Omega(D)$ lower bound that holds even for unweighted graphs and in the LOCAL model. LOCAL is identical to CONGEST, except that there is no bound on the message size. Even for a class of fixed diameter, any algorithm needs $\Omega(D)$ time for a $O(n)$ -approximation.

► **Theorem 3.** *Let $D \leq n/2$ be a parameter and consider the class of graphs with diameter D . An algorithm that with probability $> \frac{1}{2}$ satisfies that every vertex v of a graph G outputs a value $\tilde{\phi}_v \geq \phi(G)$, for sparsest cut $\phi(G)$, and at least one vertex has $\phi_v \leq \Theta(n \cdot \phi)$ needs $\Omega(D)$ rounds in LOCAL.*

Note that the $\Omega(D)$ lower bound in Theorem 3 holds even in the case where only a single vertex needs to witness the cut.

The above proof is about discerning graphs with sparsity $1/n$ and $1/n^2$. If every vertex needs to output a correct estimation, a diameter lower bound exists for larger values of ϕ .

► **Theorem 4.** *Let $0 < \varepsilon < 1$ be a constant and let $\phi > 1/n^{1-\varepsilon}$. An algorithm that with probability $> \frac{1}{2}$ satisfies that every vertex v outputs a value $\tilde{\phi}_v$ such that $\phi \leq \tilde{\phi} \leq O(n^{1+\varepsilon} \cdot \phi)$, where ϕ is the sparsest cut value of the graph needs at least $\Omega(D)$ rounds in the LOCAL model.*

Our CONGEST upper bound for approximating the sparsest cut w.h.p. is $O(\log^2 n/\phi)$ – only a factor $\log n$ away from optimal, since the graphs from the above observation have sparsest cut ϕ and diameter $\Theta(\log n/\phi)$ [8]. We note that with constant probability, we match the lower bound, see the full version for details.

3 Our Contributions on Spectrum Approximation

Our core technical contributions are efficient CONGEST algorithms to compute the approximate (partial) spectrum of a graph. In essence, Theorems 1 and 2 then follow as direct consequences via Cheeger’s Inequality. Nevertheless, we chose to present these theorems first, as they are more accessible to a broader audience and can be compared with prior work.

Eigenvalues of the Normalized Laplacian. The normalized Laplacian of a graph is the matrix defined as $L := I - \text{Deg}^{-1/2} A \text{Deg}^{-1/2}$, where A is the adjacency matrix and Deg is the (weighted) degree matrix. Its spectrum captures many central properties of the graph,

e.g., various types of its connectedness like k -way conductance [15] and small set expansion [3], or how close it is to being bipartite [4, 16]. We provide the following theorem to compute an approximation of that spectrum.

► **Theorem 5** (Eigenvalue estimation). *Let $\varepsilon > 0$ be a parameter and $k \in \mathbb{N}$ be a constant. There are randomized CONGEST algorithms that, given an undirected, weighted graph $G = (V, E)$, with high probability approximate the eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ of the graph's normalized Laplacian as follows:*

1. $\lambda_1, \dots, \lambda_k$: an additive ε -approximations in $O(\log^2 n \cdot \text{poly}(\varepsilon^{-1}))$ rounds,
2. λ_2 : an additive ε -approximation in $O\left(\frac{\log^2 n}{\varepsilon} + \frac{\log n}{\varepsilon} \cdot \lceil \frac{\lambda_2}{\varepsilon} \rceil\right) = O\left(\frac{\log^2 n}{\varepsilon} + \frac{\log n}{\varepsilon^2}\right)$ rounds,
3. λ_n : a multiplicative $(1 \pm \varepsilon)$ -approximation in $O\left(D + \frac{\log^2 n}{\varepsilon}\right)$ rounds.

Eigenvalue λ_2 is the eigenvalue approximating the sparsest cut via Cheeger's Inequality. It could either be approximated via the second algorithm, or slightly slower via the first.

For the smaller eigenvalues, Theorem 5 states additive approximations avoiding the diameter dependence, e.g., when ε is an arbitrarily small constant Theorem 5 approximates λ_2 in $O(\log^2 n)$ rounds even though the diameter may be up to linear in n .

As ε can be arbitrary in Theorem 5, we can obtain multiplicative approximations by setting $\varepsilon = \varepsilon' \lambda_2$ for any $\varepsilon' > 0$. However, this does not circumvent the diameter lower bound of Theorem 6 presented below. $D = O(\log n / \lambda_2)$ holds for any graph, so the diameter term is then always asymptotically smaller than the overall runtime of the algorithm.

To the best of our knowledge, the only prior distributed eigenvalue computation is via computing a spectral sparsifier, i.e., a sparse subgraph H with $\tilde{O}(n/\varepsilon^2)$ edges that approximately preserve the graph's spectrum. Next, the sparsifier H can be gathered at one vertex of the graph that can locally compute the eigenvalues of H and hence obtain approximations of all eigenvalues of G . Computing H is actually relatively fast and only requires $\text{poly}(\log n, \varepsilon^{-1})$ rounds [13]. Collecting H at a single vertex is the expensive part, requiring $\tilde{O}(n/\varepsilon^2)$ rounds. This is exponentially slower than Theorem 5, though this more general method immediately computes the full spectrum while we focus on single eigenvalues.

We note that all our upper bounds even hold in the more restricted Broadcast CONGEST model in which a vertex needs to send the same message to all of its neighbors in each round. All our algorithms become a $\log n$ -factor faster if we only ask for constant probability as typical in property testing, see the discussion in the full version.

Lower Bound. All of our multiplicative upper bounds take diameter time. This means that at the end of the algorithm, every vertex can easily output the same eigenvalue or sparsest cut estimate. One may ask whether one can obtain faster algorithms if the output is more local. Perhaps every vertex outputs the sparsest cut it sees in some neighborhood, and there is a guarantee that at least one vertex will see an approximately sparsest cut. In Theorem 3, we showed that this is not true for sparsest cut approximation. We also show that approximating λ_2 at even one node requires $\Omega(D)$ rounds.

► **Theorem 6.** *Any algorithm that with probability $> \frac{1}{2}$ ensures that all vertices output a value $\tilde{\lambda}_2 \geq \lambda_2$, and at least one vertex additionally satisfies $\tilde{\lambda}_2 \leq 2\lambda_2$ requires $\Omega(D)$ rounds.*

An $\Omega(D)$ lower bound could also follow from our sparsest cut lower bounds via Cheeger's Inequality, but such an approach incurs an inherent quadratic loss due to the approximation gap in Cheeger's.

4 Techniques: The Power Method

As mentioned, Theorems 1 and 2 follow as direct consequences from our eigenvalue estimation results via Cheeger’s Inequality. Here, we focus on our core technical contribution: efficient CONGEST algorithms to compute the approximate (partial) spectrum of a graph. The main ingredient of our algorithms is the well-known *power method* for approximating the largest eigenvalue of a matrix. Let M be any matrix. The power method, also known as *power iteration* or *Von Mises iteration*, starts with a random vector and converges to the eigenvector corresponding to the largest eigenvalue of M . It is a standard technique in numerical linear algebra, see, e.g., [12]. The procedure is as follows:

Power Method

- 1: **Input:** Matrix $M \in \mathbb{R}^{n \times n}$, number of iterations k .
- 2: **Initialize:** A random vector $x_0 \in \mathbb{R}^n$.
- 3: **for** $i = 1$ to k **do**
- 4: $x_i \leftarrow Mx_{i-1}$.
- 5: **Output:**
 - Approximate dominant eigenvalue: $\lambda \approx \frac{x_k^T M x_k}{x_k^T x_k}$;
 - Approximate dominant eigenvector: x_k .

With constant probability over the randomness of the start vector x_0 , the power method outputs a vector x_k whose *Rayleigh coefficient* $\frac{x_k^T M x_k}{x_k^T x_k}$ is close to the *largest* eigenvalue of the matrix M . In the full version, we provide both intuition and a formal proof.

In community detection, [5] used an algorithm that is in the same spirit as the power method but not entirely equal. In case of a sufficiently small balanced cut, and some additional technical assumptions, they can provide an approximate sparsest cut. Since the approximation factor depends on the technical assumptions, we will not state their exact guarantees here.

In the full version, we describe the power method in detail, provide a proof, and discuss the challenges of implementing it in a distributed setting.

References

- 1 Noga Alon. Eigenvalues and expanders. *Comb.*, 6(2):83–96, 1986. doi:10.1007/BF02579166.
- 2 Noga Alon and V. D. Milman. λ_1 , isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory B*, 38(1):73–88, 1985. doi:10.1016/0095-8956(85)90092-9.
- 3 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *J. ACM*, 62(5):42:1–42:25, 2015. Announced at FOCS ’10. doi:10.1145/2775105.
- 4 Frank Bauer and Jürgen Jost. Bipartite and neighborhood graphs and the spectrum of the normalized graph laplacian. *Communications in Analysis and Geometry*, 21, October 2009. doi:10.4310/CAG.2013.v21.n4.a2.
- 5 Luca Becchetti, Andrea E. F. Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. Find your place: Simple distributed algorithms for community detection. *SIAM J. Comput.*, 49(4):821–864, 2020. Announced at SODA’17. doi:10.1137/19M1243026.
- 6 Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Comput. Complex.*, 15(2):94–114, 2006. Announced at CCC’05. doi:10.1007/S00037-006-0210-9.

- 7 Daoyuan Chen, Simon Meierhans, Maximilian Probst Gutenberg, and Thatchaphol Saranurak. Parallel and distributed expander decomposition: Simple, fast, and near-optimal. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 1705–1719. SIAM, 2025. doi:10.1137/1.9781611978322.53.
- 8 Flavio Chierichetti, George Giakkoupis, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading and conductance. *J. ACM*, 65(4):17:1–17:21, 2018. doi:10.1145/3173043.
- 9 Mohsen Ghaffari, Fabian Kuhn, and Hsin-Hao Su. Distributed MST and routing in almost mixing time. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 131–140. ACM, 2017. doi:10.1145/3087801.3087827.
- 10 Mohsen Ghaffari and Jason Li. New distributed algorithms in almost mixing time via transformations from parallel algorithms. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.DISC.2018.31.
- 11 George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In Thomas Schwentick and Christoph Dürr, editors, *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*, volume 9 of *LIPIcs*, pages 57–68. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2011. doi:10.4230/LIPICS.STACS.2011.57.
- 12 Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- 13 Ioannis Koutis and Shen Chen Xu. Simple parallel and distributed algorithms for spectral graph sparsification. *ACM Trans. Parallel Comput.*, 3(2):14:1–14:14, 2016. doi:10.1145/2948062.
- 14 Fabian Kuhn and Anisur Rahaman Molla. Distributed sparse cut approximation. In Emmanuelle Anceaume, Christian Cachin, and Maria Gradinariu Potop-Butucaru, editors, *19th International Conference on Principles of Distributed Systems, OPODIS 2015, December 14-17, 2015, Rennes, France*, volume 46 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.OPODIS.2015.10.
- 15 James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1117–1130. ACM, 2012. doi:10.1145/2213977.2214078.
- 16 Shiping Liu. Multi-way dual cheeger constants and spectral bounds of graphs. *Advances in Mathematics*, 268:306–338, 2015. arXiv:1401.3147.
- 17 David W. Matula and Farhad Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discret. Appl. Math.*, 27(1-2):113–123, 1990. doi:10.1016/0166-218X(90)90133-W.
- 18 Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *SODA*, pages 2616–2635. SIAM, 2019. doi:10.1137/1.9781611975482.162.
- 19 Atish Das Sarma, Anisur Rahaman Molla, and Gopal Pandurangan. Distributed computation of sparse cuts via random walks. In Sajal K. Das, Dilip Krishnaswamy, Santonu Karkar, Amos Korman, Mohan J. Kumar, Marius Portmann, and Srikanth Sastry, editors, *Proceedings of the 2015 International Conference on Distributed Computing and Networking, ICDCN 2015, Goa, India, January 4-7, 2015*, pages 6:1–6:10. ACM, 2015. doi:10.1145/2684464.2684474.
- 20 Atish Das Sarma, Danupon Nanongkai, Gopal Pandurangan, and Prasad Tetali. Distributed random walks. *J. ACM*, 60(1):2:1–2:31, 2013. doi:10.1145/2432622.2432624.
- 21 Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Inf. Comput.*, 82(1):93–133, 1989. doi:10.1016/0890-5401(89)90067-9.