Expander Decomposition with Fewer Inter-Cluster Edges Using a Spectral Cut Player

Daniel Agassy
Tel Aviv University, Israel

Dani Dorfman
Tel Aviv University, Israel

Haim Kaplan
Tel Aviv University, Israel

Abstract

A (ϕ, ϵ)-expander decomposition of a graph G (with n vertices and m edges) is a partition of V into clusters $V_1, \ldots, V_k$ with conductance $\Phi(G[V_i]) \geq \phi$, such that there are at most ϵm inter-cluster edges. Such a decomposition plays a crucial role in many graph algorithms. We give a randomized $\tilde{O}(m/\phi)$ time algorithm for computing a $(\phi, \phi \log^2 n)$-expander decomposition. This improves upon the $(\phi, \phi \log^3 n)$-expander decomposition also obtained in $\tilde{O}(m/\phi)$ time by [Saranurak and Wang, SODA 2019] (SW) and brings the number of inter-cluster edges within logarithmic factor of optimal.

One crucial component of SW's algorithm is a non-stop version of the cut-matching game of [Khandekar, Rao, Vazirani, JACM 2009] (KRV): The cut player does not stop when it gets from the matching player an unbalanced sparse cut, but continues to play on a trimmed part of the large side. The crux of our improvement is the design of a non-stop version of the cleverer cut player of [Orecchia, Schulman, Vazirani, Vishnoi, STOC 2008] (OSVV). The cut player of OSSV uses a more sophisticated random walk, a subtle potential function, and spectral arguments. Designing and analysing a non-stop version of this game was an explicit open question asked by SW.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Exapander Decomposition, Cut-Matching Game

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.56

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2205.10301 [1]

Funding Israel science foundation (ISF) grant 1595/19, Israel science foundation (ISF) grant 2854/20, and the Blavatnik research foundation.

1 Introduction

The *conductance* of a cut $(S, V \setminus S)$ is $\Phi_G(S, V \setminus S) = \frac{|E(S,V\setminus S)|}{\min(\mathbf{vol}(S),\mathbf{vol}(V\setminus S))}$, where $\mathbf{vol}(S)$ is the sum of the degrees of the vertices of S. The conductance of a graph G is the smallest conductance of a cut in G.

A (ϕ, ϵ)-*expander decomposition* of a graph G is a partition of the vertices of G into clusters $V_1, \ldots, V_k$ with conductance $\Phi(G[V_i]) \geq \phi$ such that there are at most ϵm inter-cluster edges, where $\phi, \epsilon \geq 0$. We consider the problem of computing in almost linear time ($\tilde{O}(m)$ time) a (ϕ, ϵ)-expander decomposition for a given graph G and $\phi > 0$, while minimizing ϵ as a function of ϕ. It is known that a (ϕ, ϵ)-expander decomposition, with $\epsilon = O(\phi \log n)$, always exists and that $\epsilon = \Theta(\phi \log n)$ is optimal [23, 2].

Expander decomposition algorithms have been used in many cutting edge results, such as directed/undirected Laplacian solvers [27, 11], graph sparsification [9, 10], distributed algorithms [6], and maximum flow algorithms [15]. Expander decomposition was also used

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 56; pp. 56:1–56:21
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[10] (in the deterministic case) in order to break the $O(\sqrt{n})$ dynamic connectivity bound and achieve an improved running time of $O(n^{o(1)})$ per operation. It was also used in the recent breakthrough result by Chen et al. [8], who showed algorithms for maximum flow and minimum cost flow in almost linear time.

Given an $f(n)$-approximation algorithm for the problem of finding a minimum conductance cut, one can get a $(\phi, O(f(n) \cdot \phi \log n))$-expander decomposition algorithm by recursively computing approximate cuts (and thus splitting V) until all components are certified as expanders. In particular, using an exact minimum conductance cut algorithm ensures the existence of an expander decomposition with $\epsilon = O(\phi \log n)$ as mentioned above. Using the polynomial algorithms of [20, 4] which provide the best approximation ratios of $O\left(\sqrt{\phi}\right)$ and $O\left(\sqrt{\log n}\right)$, respectively, for conductance, gives polynomial time expander decomposition algorithms with $\epsilon = O\left(\phi^{3/2} \log n\right)$ and $\epsilon = O\left(\phi \log^{\frac{3}{2}} n\right)$. However, these decomposition algorithms might lead to a linear recursion depth, and therefore have superlinear time complexity.

To get a near linear time algorithm using this recursive approach, one must be able to efficiently compute low conductance cuts with additional guarantees. We get such cuts using the cut-matching framework of [16] (abbreviated as KRV). In order to present our results in the appropriate context we now give a brief background on the cut-matching framework.

Cut-matching: Edge-expansion is a connectivity measure related to conductance. The *edge-expansion* of a cut $(S, V \setminus S)$ is $h_G(S, V \setminus S) = \frac{|E(S,V\setminus S)|}{\min(|S|,|V\setminus S|)}$ and the *edge-expansion* of a graph G is the smallest edge-expansion of a cut in G.

The cut-matching game is a technique that reduces the approximation task for sparsest cut (in terms of edge-expansion) to a polylogarithmic number of maximum flow problems. The resulting approximation algorithm for sparsest cut is remarkably simple and robust.

The cut-matching game is played between a *cut player* and a *matching player*, as follows. We start with an empty graph G_0 on n vertices. At round t, the cut player chooses a bisection $(S_t, \overline{S_t})$ of the vertices (we assume n is even). In response, the matching player presents a perfect matching M_t between the vertices of S_t and $\overline{S_t}$ and the game graph is updated to $G_t = G_{t-1} \cup M_t$. Note that this graph may contain parallel edges. The game ends when G_t is a sufficiently good edge-expander. The goal of this game is to devise a strategy for the cut player that maximizes the ratio $r(n) := \phi/T$, where T is the number of rounds and $\phi = h(G_T)$ is the edge-expansion of G_T. KRV showed that one can translate a cut strategy of quality $r(n)$ into a sparsest cut algorithm of approximation ratio $1/r(n)$ by applying a binary search on a sparsity parameter ϕ until we certify that $h(G) \geq \phi$ and $h(G) = O(\phi/r(n))$.

KRV devised a randomized cut-player strategy that finds the bisection using a stochastic matrix that corresponds to a random walk on all previously discovered matchings. Their walk traverses the previous matchings in order and with probability half takes a step according to each matching. They showed that the matrix corresponding to this random walk can actually be embedded (as a flow matrix) into G_t with constant congestion. They terminate when the random walk matrix is close to uniform (i.e. having constant edge-expansion), resulting in G_T for $T = O\left(\log^2 n\right)$, having constant edge-expansion.

Orecchia et al. [21] (abbreviated as OSVV) took the same approach but devised a more sophisticated random walk and used Cheeger's inequality [7] in order to show that G_T, for $T = O\left(\log^2 n\right)$, has $\Omega(\log n)$ edge-expansion. That is, they got a ratio of $r(n) = \Omega\left(\frac{1}{\log n}\right)$.

Equipped with this background we now get back to expander decomposition, and focus on the $\tilde{O}(m/\phi)$ time algorithm by Saranurak and Wang [23] (abbreviated as SW). Their algorithm is randomized, follows the recursive scheme described above, and computes a $(\phi, \phi \log^3 n)$-expander decomposition in $O\left(\frac{m \log^4 n}{\phi}\right)$ time. Its number of inter-cluster edges is off by a factor of $O\left(\log^2 n\right)$ from optimal and off by a factor of $O\left(\log^{\frac{3}{2}} n\right)$ from the aforementioned best achievable polynomial time construction.

One core component of this algorithm is a variation of the cut-matching game (inspired by Räcke et al. [22]). In this variation, the game graph $G_t = (V_t, E_t)$ may lose vertices (*i.e.*, $V_{t+1} \subseteq V_t$) throughout the game and the objective of the cut player is to make V_T a *near expander* in G_T (see Definition 9). The result of each round does not consist of a perfect matching in V_t, but rather a subset to remove from V_t and a matching of the remaining vertices. The game ends either with a balanced cut of low conductance, or with an unbalanced cut of low conductance, such that the larger side is a *near expander*. This allows SW to avoid recurring on the large side of the cut. Indeed, if the cut is balanced, they run recursively on both sides, and if it is unbalanced, they use the fact that the large side is a *near expander* and "trim" it by finding a large subset of this side which is an expander. Then, they run recursively on the smaller side combined with the "trimmed" vertices. SW's analysis of the new cut-matching game is based on the ideas and the potential function of KRV while carefully taking into account of the shrinkage of the game graph.

An open question, raised by SW, was whether one can adapt the technique of the cut-matching strategy of OSVV to improve their decomposition. A major obstacle is how to perform an OSVV-like spectral analysis when we lose vertices throughout the process and need to bound the near-expansion of the final piece. This is challenging as the analysis of OSVV is already somewhat more complicated than that of KRV: It uses a different lazy random walk and a subtle potential to measure progress towards near expansion. Moreover Cheeger's inequality is suitable to show high expansion and the object we are targeting is a near expander.

Our contribution: In this paper we answer this question of SW affirmatively. We present and analyze an expander decomposition algorithm with a new cut-player inspired by OSVV. This improves the result of SW and gives a randomized $\tilde{O}(m/\phi)$ time algorithm for computing an $(\phi, \phi \log^2 n)$-expander decomposition (Theorem 18). This brings the number of inter-cluster edges to be off only by $O(\log n)$ factor from the best possible.

To achieve this we overcome two main technical challenges: (1) We generalize the lazy random walk of the cut player of OSVV and the subtle potential tracking its progress, to the setting in which the vertex set shrinks (by ripping off of it small cuts as in SW). (2) We show that when the generalized potential is small the remaining part of the game graph is a near expander. This required a generalization of Cheeger's inequality appropriate for our purpose (see Lemma 33).

Our techniques may be applied in similar contexts. One concrete such context is the construction of tree-cut sparsifiers. Specifically, one could try to use our technique to improve the $O\left(\log^4 n\right)$-approximate tree-cut sparsifier construction of [22] by a factor of $\log n$. (Note that [22] in fact construct a tree-flow sparsifier, which is a stronger notion.)

The cut-matching framework [16] is formalized for edge-expansion rather than conductance. Consequently, SW and others whose primary objective is conductance had to transform the graph into a *subdivision-graph* in order to use this framework. The subdivision graph is obtained by adding a new vertex (called a *split-node*) in the middle of each edge e, splitting e into a path of length two. Consequently, the analysis has to translate cuts of low expansion in the modified graph (the *subdivision graph*) to cuts of low conductance in the original graph. This transformation complicates the algorithms and their analysis.

To avoid this transformation we revisit the seminal results of KRV and OSVV and redo them directly for conductance. This is not trivial and requires subtle changes to the cut players, and the matching players, and the potentials measuring progress towards a graph with small conductance. In particular the matching player does not produce a matching anymore but rather what we call a d_G-matching, which is a graph with the same degrees as G.

Our new cut-matching algorithm is then described using this natural reformulation of the cut-matching framework directly for conductance, removing the complications that would have followed from using the split graph.

We believe that our clean presentations of the cut-matching framework for conductance would prove useful for other applications of cut-matching that require optimization for conductance rather than expansion.

Further related work: Computing the expansion and the conductance of a graph G is NP-hard [18, 25], and there is a long line of research on approximating these connectivity measures. The best known polynomial algorithms for approximating the minimum conductance cut have either $O\left(\sqrt{\log n}\right)$ [4, 24] or $O\left(\sqrt{\Phi(G)}\right)$ approximation ratios [20]. Approximation algorithms for expansion and conductance play a crucial role in algorithms for expander decomposition [23, 5, 10], expander hierarchies [12, 14], and tree flow sparsifiers [22].

In his thesis, Orecchia [19] elaborates on the two cut-matching strategies described in OSVV, one based on a lazy random walk, called C_{NAT}, and a more sophisticated one based on the *heat-kernel* random walk, called C_{EXP}. Orecchia proves (Theorem 4.1.5 of [19]) that using C_{NAT} or C_{EXP}, after $T = \Theta(\log^2 n)$ iterations, the graph G_T has expansion $\Omega(\log n)$ (and thereby conductance $\Omega\left(\frac{1}{\log n}\right)$, since it is regular with degrees $\Theta(\log^2 n)$). Orecchia also bounds the second largest eigenvalue of the normalized Laplacian of G_T. However, Orecchia does not show how to use cut-matching to get approximation algorithms for the conductance of G.

In a recent paper [3] Ameranis *et al.* use a generalized notion of expansion, also mentioned in [19], where we normalize the number of edges crossing the cut by a general measure (μ) of the smaller side of the cut. They define a corresponding generalized version of the cut-matching game, and show how to use a cut strategy for this game to get an approximation algorithm for two generalized cut problems. They claim that one can construct a cut strategy for this measure using ideas from [19].[1]

Both SW and our result can be implemented in $\tilde{O}(m)$ time using the recent result of [17], by replacing Bounded-Distance-Flow (Lemma 21) and the "Trimming Step" of [23] with the algorithm of [17, Section 8]. This $\tilde{O}(m)$ hides many log factors and requires more complicated machinery.

The structure of this paper is as follows. Section 2 contains additional definitions. In order to provide the appropriate context for our work, Section 3 gives an overview of the cut-matching games in [16] and [21] and highlights the differences between them. In the full version of this paper, we give a complete and self-contained description of these approximation algorithms directly **for conductance**. A reader knowledgeable in the Cut-Matching game can skip directly to Section 4. In Section 4 we present our new non-stop spectral cut player and expander decomposition algorithm. Section 5 contains the analysis of our algorithm. Due to the space constraints some of the proofs are omitted, and are available in the full version of this paper [1].

To be consistent with common terminology we refer to a graph with conductance at least ϕ as a ϕ-expander (rather than ϕ-conductor.) No confusion should arise since in the rest of this paper we focus on conductance and do not use the notion of edge-expansion anymore.

[1] The details of such a cut player do not appear in [3] or [19].

In this paper we only focus on unweighted graphs, although our algorithm can be adapted to the case of integral, polynomially bounded weights.

2 Preliminaries

We denote the transpose of a vector or a matrix x by x'. That is, if v is a column vector then v' is the corresponding row vector. For a vector $v \in \mathbb{R}^n_{\geq 0}$, define $\sqrt{v}$ to be vector whose coordinates are the square roots of those of v. Given $A \in \mathbb{R}^{n\times n}$, we denote by $A(i,j)$ the element at the i'th row and j'th column of A. We denote by $A(i,), A(,i)$ the i'th row and column of A, respectively. We define both $A(i,)$ and $A(,i)$ as column vectors. We use the abbreviation $A(i) := A(i,)$ only with respect to the rows of A. Given a vector $v \in \mathbb{R}^n$, we denote its i'th element by $v(i)$. For disjoint $A, B \subseteq V$, we denote by $E_G(A,B)$ the set of edges connecting A and B. We sometimes omit the subscript when the graph is clear from the context. If $A = V \setminus B$, then we call (A,B) a *cut*.

▶ **Fact 1.** *Let $X, Y \in \mathbb{R}^{n\times n}, m \in \mathbb{N}$, then $\mathrm{Tr}(XY) = \mathrm{Tr}(YX)$.*

▶ **Fact 2.** *Let $X, Y \in \mathbb{R}^{n\times n}$ be symmetric matrices and let $k \in \mathbb{N}$. Then*

$$\mathrm{Tr}\left((XYX)^{2^k}\right) \leq \mathrm{Tr}\left(X^{2^k} Y^{2^k} X^{2^k}\right).$$

▶ **Definition 3** (d_G,$\mathbf{vol}_G(S)$). *Given a graph G, the vector $d_G \in \mathbb{R}^n$ is defined as $d_G(v) = \deg_G(v)$. To simplify the notation, we denote $d := d_G$ whenever the graph G is clear from the context. For $S \subseteq V$, we denote by $\mathbf{vol}_G(S) := \sum_{v\in S} d_G(v)$ the* volume *of S.*

▶ **Definition 4** ($G\{A\}$). *Let $G = (V,E)$ be a graph, and let $A \subseteq V$ be a set of vertices. We define the graph $G\{A\} = (V', E')$ as the graph induced by A with self-loops added to preserve the degrees: $V' = A, E' = \{\{u,v\} \in E : u,v \in A\} \cup \{\{u,u\} : u \in A, v \in V \setminus A, \{u,v\} \in E\}$.*

▶ **Definition 5** (d-Matching). *Given a vector $d \in \mathbb{N}^n$ and a collection of pairs $M = \{(u_i, v_i)\}_{i=1}^m$. We say that M is a* d-matching *if the graph defined by M (i.e., the graph whose edges are M) satisfies $d_M(v) = d(v)$, for every v.*

▶ **Definition 6** (d_G-stochastic). *A matrix $F \in \mathbb{R}^{n\times n}$ is* d_G-stochastic *with respect to a graph G if the following two conditions hold: (1) $F \cdot \mathbb{1}_n = d_G$ and (2) $\mathbb{1}'_n \cdot F = d'_G$.*

▶ **Definition 7** (Laplacian, Normalized Laplacian). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and let $d = A \cdot \mathbb{1}_n$, $D = \mathbf{diag}(d)$. The* Laplacian *of A is defined as $\mathcal{L}(A) = D - A$. The* normalized-Laplacian *of A is defined as $\mathcal{N}(A) = D^{-\frac{1}{2}}\mathcal{L}(A)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. The (normalized) Laplacian of an undirected graph is defined analogously using its adjacency matrix.*

▶ **Definition 8** (Conductance). *Let $G = (V,E)$ and $S \subset V$, $S \neq \emptyset$. The* conductance *of the cut $(S, V \setminus S)$, denoted by $\Phi_G(S, V\setminus S)$, is*

$$\Phi_G(S, V \setminus S) = \frac{|E(S, V\setminus S)|}{\min(\mathbf{vol}(S), \mathbf{vol}(V \setminus S))}.$$

The conductance of G is defined to be $\Phi(G) = \min_{S\subseteq V} \Phi_G(S, V \setminus S)$.

▶ **Definition 9** (Expander, Near-Expander). *Let $G = (V,E)$. We say that G is a* ϕ-expander *if $\Phi(G) \geq \phi$. Let $A \subseteq V$. We say that A is a* near ϕ-expander *in G if*

$$\min_{S\subseteq A} \frac{|E(S, V\setminus S)|}{\min(\mathbf{vol}(S), \mathbf{vol}(A \setminus S))} \geq \phi.$$

That is, a near expander is allowed to use cut edges that go outside of A. Note that the above definition applies to both directed and undirected graphs.

▶ **Definition 10** (Embedding). *Let $G = (V, E)$ be an undirected graph. Let $F \in \mathbb{R}_{\geq 0}^{V \times V}$ be a matrix (not necessarily symmetric). We say that F is* embeddable in G with congestion c, *if there exists a multi-commodity flow f in G, with $|V|$ commodities, one for each vertex (vertex v is the source of its commodity), such that, simultaneously for each $(u, v) \in V \times V$, f routes $F(u, v)$ units of u's commodity from u to v, and the total flow on each edge is at most c.* [2]

If F is the weighted adjacency matrix of a graph H on the same vertex set V, we say that H is embeddable in G with congestion c if F is embeddable in G with congestion c.

▶ **Lemma 11.** *Let G, H be two graphs on the same vertex set V. Let $A \subseteq V$. Let $\alpha > 0$ be a constant such that for each $v \in V$, $d_G(v) = \alpha \cdot d_H(v)$. Assume that H is embeddable in G with congestion c, and that A is a near ϕ-expander in H. Then, A is a near $\frac{\phi}{c\alpha}$-expander in G.*

▶ **Corollary 12.** *Let G, H be two graphs on the same vertex set V. Let $\alpha > 0$ be a constant such that for each $v \in V$, $d_G(v) = \alpha \cdot d_H(v)$. Assume that H is embeddable in G with congestion c, and that H is a ϕ-expander. Then, G is a $\frac{\phi}{c\alpha}$-expander.*

Proof. This follows from Lemma 11 by choosing $A = V$. ◀

3 Approximating conductance via cut-matching

In preparation for our expander decomposition algorithm we give a high level overview of the conductance approximation algorithms of [16] and [21]. [16] and [21] described their results for edge-expansion rather than conductance. In the full version of this paper, we give a complete description and analysis of these algorithms for conductance. This translation from edge-expansion to conductance is not trivial as both the cut player, the matching player, and the analysis have to be carefully modified to take the degrees into account. Here we give a high level overview of the key components of these algorithms and the differences between them so one can better absorb our main algorithm in Section 4.2.

The cut-matching game of [16] (in the conductance setting) works as follows.

The Cut-Matching game for conductance, with parameters T and a degree vector d:
- The game is played on a series of graphs G_i. Initially, $G_0 = \emptyset$.
- In iteration t, the cut player produces two multisets of size m, $L_t, R_t \subseteq V$, such that each $v \in V$ appears in $L_t \cup R_t$ exactly $d(v)$ times.
- The matching player responds with a d-matching M_t that only matches vertices in L_t to vertices in R_t.
- We set $G_{t+1} = G_t \cup M_t$.
- The game ends at iteration T, and the *quality* of the game is $r := \Phi(G_T)$. Note that the volume of G_t increases from one iteration to the next.

Given a strategy for the cut player of quality r, one can create a $\frac{1}{r}$ approximation algorithm for the conductance of a given graph G. To this end, the matching player has to provide matchings that can be embedded in G.

The difference between the results of [16] and [21] is mainly in the cut player. They both run the game for $T = \Theta(\log^2 n)$ iterations but [16]'s cut player achieves quality of $r =$

[2] This definition requires to route $F(u, v) = F(v, u)$ both from u to v and from v to u if F is symmetric.

$\Omega\left(\frac{1}{\log^2 n}\right)$ whereas [21]'s achieves quality of $r = \Omega\left(\frac{1}{\log n}\right)$. Notice that the cut player produces the stated expansion result in G_T regardless of the matchings given by the matching player.

3.1 KRV's Cut-Matching Game for Conductance

The cut player implicitly maintains a d_G-stochastic flow matrix (*i.e.*, representing flow demands) $F_t \in \mathbb{R}^{n\times n}$, and the graph G_t which is the union of the matchings that it obtained so far from the matching player (t is the index of the round). The flow F_t and the graph G_t have two crucial properties. First, we can embed F_t in G_t with $O(1)$ congestion (See Definition 10). Second, after $T = \Theta(\log^2 n)$ rounds, with high probability, F_T will have constant conductance.[3] Since the degrees in G_T are factor of $O(\log^2 n)$ larger than the degrees in F_T (when we think of F_T as a weighted graph) then it follows by Corollary 12 that G_T is $\Omega(1/\log^2 n)$ expander. Note that the cut player is unrelated to the input graph G in which we would like to approximate the conductance. Its goal is to produce the expander G_T.

At the beginning, $F_0 = D = \mathbf{diag}(d)$, and G_0 is the empty graph on $V = [n]$. The cut player updates F_t as follows. It draws a random unit vector $r \in \mathbb{R}^n$ orthogonal to $\sqrt{d}$ and computes the projections $u_i = \frac{1}{d(i)}\langle D^{-\frac{1}{2}}F_t(i), r\rangle$.[4] The cut player computes these projections in $O(m\log^2 n)$ time since the vector of all projections is $u := D^{-1}F_tD^{-\frac{1}{2}} \cdot r$ and F_t is defined (see below) as a multiplication of $\Theta(\log^2 n)$ sparse matrices, each having $O(m)$ non-zero entries. The cut player sorts the projections as $u_{i_1} \leq ... \leq u_{i_n}$. Consider the sequence $Q = (u_{i_1}, u_{i_1}, \ldots, u_{i_1}, u_{i_2}, u_{i_2}, \ldots, u_{i_2}, \ldots, u_{i_n}, \ldots, u_{i_n})$, where each u_{i_j} appears $d(i_j)$ times. Then, $|Q| = 2m$. Take $L_t \subseteq Q$ to be the multi-set containing the first m elements, and $R_t = Q \setminus L_t$ to be the multi-set containing the last m elements. Define $\eta \in \mathbb{R}$ such that $L_t \subseteq \{i_k : u_{i_k} \leq \eta\}$ and $R_t \subseteq \{i_k : u_{i_k} \geq \eta\}$. Note that a vertex can appear both in L_t and in R_t, if $u_{i_j} = \eta$. For a vertex $v \in V$, denote by m_v the number of times v appears in L_t, and by $\bar{m}_v$ the number of times v appears in R_t. That is, except for (maybe) one vertex, for any $v \in V$, either $m_v = 0$ and $\bar{m}_v = d(v)$ or $m_v = d(v)$ and $\bar{m}_v = 0$.

The cut player hands out the partition L_t, R_t to the matching player who sends back a d_G-matching M_t (we think of M_t as an $n \times n$ matrix with at most m non-zero entries that encodes the matching) between L_t and R_t. The cut player updates its flow matrix using M_t and sets $F_{t+1}(v) = \frac{1}{2}F_t(v) + \sum_{(v,u)\in M_t} \frac{1}{2d(u)}F_t(u)$ (in matrix form $F_{t+1} = \frac{1}{2}\left(I + M_t \cdot D^{-1}\right)F_t$).[5] This update keeps F_t a d_G-stochastic matrix for all t. The cut player also defines the graph G_{t+1} as $G_{t+1} = G_t \cup M_t$. This completes the description of the cut player of [16] adapted for conductance.

The matching player constructs an auxiliary flow problem on $G' := G \cup \{s,t\}$, where s is a new vertex which would be the source and t is a new vertex which would be the sink. We add an arc (s,v) for each $v \in L_t$ of capacity m_v and we add an arc (v,t) of capacity $\bar{m}_v$ for each $v \in R_t$. The capacity of each edge $e \in G$ is set to be $c = \Theta\left(\frac{1}{\phi\log^2 n}\right)$, where c is an integer. The matching player computes a maximum flow g from s to t in this network.

If the value of g is less than m, then the matching player uses the minimum cut in G' separating the source from the sink to find a cut in G of conductance $O(\phi\log^2 n)$. Otherwise,

---

3 We think about F_t as a weighted graph on $V = [n]$. The definitions of conductance, expander and near-expander for weighted graphs are the same as Definitions 8-9 where $|E(S, V \setminus S)|$ is the sum of the weights of the edges crossing the cut.

4 Recall that $F_t(i)$ is a column vector.

5 Note that it is possible that some $u \in V$ appears in the sum $\sum_{(v,u)\in M_t} \frac{1}{2d(u)}F_t(u)$ multiple times, if v is matched to u multiple times in M_t.

it decomposes g to a set of paths, each carrying exactly one unit of flow from a vertex $u \in L_t$ to a vertex $v \in R_t$.[6] Then it defines the d_G-matching M_t as $M_t = ((v_j, u_j))_{j=1}^m$, where v_j and u_j are the endpoints of path j. We view M_t as a symmetric $n \times n$ matrix, such that $M_t(v, u)$ is the number of paths between v and u. The matching player connects the game to the input graph G. Indeed, by solving the maximum flow problems in G it guarantees that the expander G_T is embeddable in G with congestion $O(cT) = O(1/\phi)$. Since the degrees of G_T are a factor of $O(\log^2 n)$ larger than the degrees of G and G_T is $\Omega(1/\log^2 n)$ expander, we get that G is a $\Omega(\phi)$-expander (see Corollary 12). The following theorem summarizes the properties of this algorithm.

▶ **Theorem 13** ([16]'s cut-matching game for conductance)**.** *Given a graph G and a parameter $\phi > 0$, there exists a randomized algorithm, whose running time is dominated by computing a polylogarithmic number of maximum flow problems, that either*

1. *Certifies that $\Phi(G) = \Omega(\phi)$ with high probability; or*
2. *Finds a cut $(S, V \setminus S)$ in G whose conductance is $\Phi_G(S, V \setminus S) = O(\phi \log^2 n)$.*

If the matching player finds a sparse cut in any iteration then we terminate with Case (2). On the other hand, if the game continues for $T = O(\log^2 n)$ rounds then since the cut player can embed F_T in G_T and the matching player can embed G_T in G, and since F_t is an expander, then we get Case (1).

The running time of the cut player is $O(m \log^4 n)$. The matching player solves $O(\log^2 n)$ maximum flow problems. By using the most recent maximum flow algorithm of [8], we get the matching player to run in $O\left(m^{1+o(1)}\right)$ time. Alternatively, we can adapt the cut-matching game, and use a version of the Bounded-Distance-Flow algorithm (which was called *Unit-Flow* in [23]; see Lemma 21), to get a running time of $\tilde{O}(\frac{m}{\phi})$ for the matching player. We can also get $\tilde{O}(m)$ running time using the recent result [17].

The key part of the analysis is to show that F_T is indeed an $\Omega(1)$-expander for any choice of d_G-matchings of the matching player. To this end, we keep track of the progress of the cut player using the potential function

$$\psi(t) = \sum_{i \in V} \sum_{j \in V} \frac{1}{d(i) \cdot d(j)} \left(F_t(i,j) - \frac{d(i)d(j)}{2m} \right)^2 = \left\| D^{-\frac{1}{2}} F_t D^{-\frac{1}{2}} - \frac{1}{2m} \sqrt{d} \sqrt{d'} \right\|_F^2$$

where the matrix norm which we use here is the Frobenius norm (sum of the squares of the entries). This potential represents the distance between the normalized flow matrix $\bar{F}_t = D^{-\frac{1}{2}} F_t D^{-\frac{1}{2}}$ and the (normalized) uniform random walk distribution $d_G d'_G / 2m$. Let $P = I - \frac{1}{2m} \sqrt{d} \sqrt{d'}$ be the projection matrix on the orthogonal complement of the span of the vector $\sqrt{d}$, then we can also write this potential as

$$\psi(t) = \left\| \bar{F}_t P \right\|_F^2 = \operatorname{Tr}\left((\bar{F}_t P)(\bar{F}_t P)' \right) = \operatorname{Tr}(\bar{F}_t P^2 \bar{F}_t') = \operatorname{Tr}(P \bar{F}_t' \bar{F}_t).$$

The first equality holds since F_t is d-stochastic and the last equality is due to Fact 1 (and that $P^2 = P$ as a projection matrix).

The crux of the proof is to show that after T rounds this potential is smaller than $1/(16m^2)$ which implies that for every pair of vertices u and v, $F_T(u, v) \geq d(v)d(u)/(4m)$. From this we get a lower bound of $1/4$ on the conductance of every cut.

[6] Note that there can be multiple flow paths between a pair of vertices $u \in L_t$ and $v \in R_t$. Furthermore, if $u \in L_t \cap R_t$ then it is possible that a path starts and ends at u.

3.2 OSVV's Cut-Matching Game for Conductance

The cut player of [21] also maintains (implicitly) a flow matrix F_t and the union G_t of the d_G-matchings it got from the matching player. Let $P = I - \frac{1}{2m}\sqrt{d}\sqrt{d'}$ be the projection to the subspace orthogonal to $\sqrt{d}$ as before (hence $P^2 = P$). Let $\delta = \Theta(\log n)$ be a power of 2. Here the matrix $W_t = (PD^{-\frac{1}{2}}F_tD^{-\frac{1}{2}}P)^\delta$ takes the role of $D^{-\frac{1}{2}}F_tD^{-\frac{1}{2}}$ from the cut player of Section 3.1.

In round t the cut player computes the projections $u_i = \frac{1}{\sqrt{d(i)}}\langle W_t(i), r\rangle$, and defines L_t and R_t based on these projections as in the previous section.[7] Then it gets a d_G-matching M_t between L_t and R_t from the matching player. It defines $N_t = \frac{\delta-1}{\delta}D + \frac{1}{\delta}M_t$ and updates the flow to be $F_{t+1} = N_t \cdot D^{-1}F_tD^{-1}N_t$. If we think of F_t as a random walk then $D^{-1}N_t$ is a lazy step that we add before and after the walk F_t to get F_{t+1}. It holds that F_{t+1} is d_G-stochastic and moreover that for all rounds t, F_t is embeddable in G_t with congestion $\frac{4}{\delta} = O(1/\log n)$. Note that here we embed F_t in G_t with smaller congestion than in Section 3.1. We can still prove, however, that F_T for $T = O(\log^2 n)$ is a $\Omega(1)$ expander and therefore, G_T is a $\Omega(1/\log n)$ expander.

The matching player solves the same flow problem as in Section 3.1 but with an integer capacity value of $c = \Theta(\frac{1}{\phi \log n})$ on the edges of G. If the value of maximum flow is less than m then it finds a cut of conductance $O(\phi \log n)$, and otherwise it returns the matching that it derives from a decomposition of the flow into paths. The matching player guarantees that the expander G_T is embeddable in G with congestion $O(cT) = O(\log n/\phi)$. Since the degrees of G_T are larger by a factor of $O(\log^2 n)$ than the degrees of G and G_T is $\Omega(1/\log n)$-expander, we get that G is a $\Omega(\phi)$-expander (see Lemma 11). The following theorem summarizes the properties of this algorithm.

▶ **Theorem 14** ([21]'s cut-matching game for conductance)**.** *Given a graph G and a parameter $\phi > 0$, there exists a randomized algorithm, whose running time is dominated by computing a polylogarithmic number of maximum flow problems, that either*

1. *Certifies that $\Phi(G) = \Omega(\phi)$ with high probability; or*
2. *Finds a cut $(S, V \setminus S)$ in G whose conductance is $\Phi_G(S, V \setminus S) = O(\phi \log n)$.*

The running time of the cut player is dominated by computing the projections in $O(m \log^3 n)$ time per iteration for a total of $O(m \log^5 n)$ time. The matching player solves $O(\log^2 n)$ maximum flow problems. Again, we can modify the algorithm so that its running time is $\tilde{O}(\frac{m}{\phi})$ or $\tilde{O}(m)$, similarly to the previous subsection.

As in Section 3.1, the key part of the analysis is to show that F_T is indeed an $\Omega(1)$-expander for any choice of d_G-matchings of the matching player. Here we keep track of the progress of the cut player using the potential function

$$\psi(t) = \left\| (D^{-\frac{1}{2}}F_tD^{-\frac{1}{2}})^\delta - \frac{1}{2m}\sqrt{d}\sqrt{d'} \right\|_F^2 .$$

Recall that $W_t = (PD^{-\frac{1}{2}}F_tD^{-\frac{1}{2}}P)^\delta$, so we can rewrite the potential function as

$$\psi(t) = \left\| (D^{-\frac{1}{2}}F_tD^{-\frac{1}{2}})^\delta P \right\|_F^2 = \mathrm{Tr}(P(D^{-\frac{1}{2}}F_tD^{-\frac{1}{2}})^{2\delta}P) \overset{(4)}{=} \mathrm{Tr}((PD^{-\frac{1}{2}}F_tD^{-\frac{1}{2}}P)^{2\delta}) = \mathrm{Tr}(W_t^2) ,$$

[7] Computing these projections takes $O(m \log^3 n)$ time since F_t is a multiplication of $\Theta(\log^2 n)$ sparse matrices, each with $O(m)$ non-zero entries. Therefore W_t is a multiplication of $\Theta(\log^3 n)$ matrices, each of which is either P or a sparse matrix.

where equality (4) follows since F_t is d-stochastic and the fact that $P^2 = P$. A careful argument shows that after $T = O(\log^2 n)$ iterations, $\psi(T) \leq 1/n$. From this we deduce that the second smallest eigenvalue of the normalized Laplacian of F_T is at least 1/2 and then by Cheeger's inequality [7] we get that $\Phi(F_T) = \Omega(1)$.

4 Expander decomposition via spectral Cut-Matching

To put our main result in context we first show how SW [23] modified the cut-matching game of KRV [16] for their expander decomposition algorithm.

4.1 SW's Cut-Matching for expander decomposition

SW [23] take a recursive approach to find an expander decomposition. One can use the cut-matching game to find a sparse cut, but if the cut is unbalanced, we want to avoid recursing on the large side.

In order to refrain from recursing on the large side of the cut, SW changed the cut-matching game as follows. The cut player now maintains a partition of V into a small set R and a large set $A = V \setminus R$, where initially $R = \emptyset$ and $A = V$. In each iteration the cut and the matching player interact as follows.

- The cut player computes two disjoint sets $A^l, A^r \subseteq A$ such that $|A^l| \leq n/8$ and $|A^r| \geq n/2$.
- The matching player returns a partition $(S, A \setminus S)$ of A, which may be empty ($S = \emptyset$), and a matching of $A^l \setminus S$ to a subset of $A^r \setminus S$.

The cut player computes the sets A^l and A^r by projecting the rows of a *flow-matrix* F that it maintains (as in KRV [16]) onto a random unit vector r, and applying a result by [22] to generate the sets A^l and A^r from the values of the projections. For the matching player, SW use a flow-based algorithm which simultaneously gives a cut $(S, A \setminus S)$ of conductance $O(\phi \log^2 n)$ of $G[A]$, and a matching of the vertices left in $A^l \setminus S$ to vertices of $A^r \setminus S$ (S may be empty when $G[A]$ has conductance $\geq \phi$). If the matching player found a sparse cut $(S, A \setminus S)$ then the cut player updates the partition (R, A) of V by moving S from A to R.

The game terminates either when the volume of R gets larger than $\Omega(m/\log^2 n)$ or after $O(\log^2 n)$ rounds. In the latter case, SW proved that the remaining set A (which is large) is a near ϕ-expander in G (see Definition 9).

To prove that after $T = \Theta(\log^2 n)$ iterations, the remaining set A is a near ϕ-expander, SW essentially followed the footsteps of KRV and used a similar potential. The argument is more complicated since they have to take the shrinkage of A into account. SW did not use a version of KRV suitable to conductance as we give in the full version. Therefore, they had to modify the graph by adding a split node for each edge, essentially reducing conductance to edge-expansion, a reduction that made their algorithm and analysis somewhat more complicated. The following theorem summarized the properties of the cut-matching game of [23].

▶ **Theorem 15** (Theorem 2.2 of [23]). *Given a graph $G = (V, E)$ of m edges and a parameter $0 < \phi < 1/\log^2 n$,*[8] *there exists a randomized algorithm, called "the cut-matching step", which takes $O\left((m \log n)/\phi\right)$ time and terminates in one of the following three cases:*

1. *We certify that G has conductance $\Phi(G) = \Omega(\phi)$ with high probability.*

[8] The theorem is trivial if $\phi \geq \frac{1}{\log^2 n}$, because any cut $(A, V \setminus A)$ has conductance $\Phi_G(A, V \setminus A) \leq 1$. We can therefore assume that $\phi < \frac{1}{\log^2 n}$.

2. *We find a cut (R, A) of G of conductance $\Phi_G(R, A) = O(\phi \log^2 n)$, and $\mathbf{vol}(R), \mathbf{vol}(A)$ are both $\Omega(\frac{m}{\log^2 n})$, i.e., we find a relatively balanced low conductance cut.*
3. *We find a cut (R, A) of G with $\Phi_G(R, A) \leq c_0 \phi \log^2 n$ for some constant c_0, and $\mathbf{vol}(R) \leq \frac{m}{10c_0 \log^2 n}$, and with high probability A is a near ϕ-expander in G.*

SW derived an expander decomposition algorithm from this modified cut-matching game by recursing on both sides of the cut only if Case (2) occurs. In Case (3) they find a large subset $B \subseteq A$ which is an expander (in what they called the *trimming step*), add $A \setminus B$ to R and recur only on R. The main result of [23] is as follows.

▶ **Theorem 16** (Theorem 1.2 of [23])**.** *Given a graph $G = (V, E)$ of m edges and a parameter ϕ, there is a randomized algorithm that with high probability finds a partitioning of V into clusters $V_1, \dots, V_k$ such that $\forall i : \Phi_{G\{V_i\}} = \Omega(\phi)$ and there are at most $O(\phi m \log^3 n)$ inter cluster edges.*[9] *The running time of the algorithm is $O(m \log^4 n/\phi)$.*

4.2 Our contribution: Spectral cut player for expander decomposition

SW [23] left open the question if one can improve their expander decomposition algorithm using tools similar to the ones that allowed OSVV [21] to improve the conductance approximation algorithm of KRV [16]. We give a positive answer to this question. Specifically we improve the cut-matching game of SW and derive the following improved version of Theorem 15.

▶ **Theorem 17.** *Given a graph $G = (V, E)$ of m edges and a parameter $0 < \phi < \frac{1}{\log n}$,*[10] *there exists a randomized algorithm which takes $O\left(m \log^5 n + \frac{m \log^2 n}{\phi}\right)$ time and must end in one of the following three cases:*

1. *We certify that G has conductance $\Phi(G) = \Omega(\phi)$ with high probability.*
2. *We find a cut (R, A) in G of conductance $\Phi_G(R, A) = O(\phi \log n)$, and $\mathbf{vol}(R), \mathbf{vol}(A)$ are both $\Omega(\frac{m}{\log n})$, i.e, we find a relatively balanced low conductance cut.*
3. *We find a cut (R, A) with $\Phi_G(R, A) \leq c_0 \phi \log n$ for some constant c_0, and $\mathbf{vol}(R) \leq \frac{m}{10c_0 \log n}$, and with high probability A is a near $\Omega(\phi)$-expander in G.*

The proof of Theorem 17 is given in Section 5. Theorem 17 implies the following theorem

▶ **Theorem 18.** *Given a graph $G = (V, E)$ of m edges and a parameter ϕ, there is a randomized algorithm that with high probability finds a partition of V into clusters $V_1, ..., V_k$ such that $\forall i : \Phi_{G\{V_i\}} = \Omega(\phi)$ and $\sum_i |E(V_i, V \setminus V_i)| = O(\phi m \log^2 n)$. The running time of the algorithm is $O(m \log^7 n + \frac{m \log^4 n}{\phi})$.*[11]

To get Theorem 17 we use the following cut player and matching player.

---

[9] $G\{V_i\}$ is defined in Definition 4.

[10] The theorem is trivial if $\phi \geq \frac{1}{\log n}$, because any cut $(A, V \setminus A)$ has conductance $\Phi_G(A, V \setminus A) \leq 1$. We can therefore assume that $\phi < \frac{1}{\log n}$.

[11] Note that if $\phi \leq \frac{1}{\log^3 n}$, then the running time matches the running time of [23] in Theorem 16. In case that $\phi \geq \frac{1}{\log^3 n}$, we get a slightly worse running time of $O(m \log^7 n)$ instead of $O(\frac{m \log^4 n}{\phi})$.

4.3 Cut player

Like in Section 3, we consider a d-stochastic flow matrix $F_t \in \mathbb{R}^{n\times n}$, and a series of graphs G_t. F_0 is initialized as $F_0 = D := \mathbf{diag}(d)$, and G_0 is initialized as the empty graph on $V = [n]$. Here the cut player also maintains a low conductance cut $A_t \subseteq V, R_t = V \setminus A_t$, such that after $T = \Theta(\log^2 n)$ rounds, with high probability, A_T is a near expander in G_T. At the beginning, $A_0 = V$, $R_0 = \emptyset$,

Since the new cut-matching game consists of iteratively shrinking the domain $A_t \subseteq V$, we start by generalizing our matrices from Section 3 to this context of shrinking domain.

▶ **Definition 19** ($I_t, d_t, D_t, P_t, \mathbf{vol}_t$). *We define the following variables*[12]

1. $I_t \in \mathbb{R}^{n\times n}$ *is the diagonal* $0/1$ *matrix that have* 1*'s on the diagonal entries corresponding to* A_t.
2. $d_t = I_t \cdot d \in \mathbb{R}^n$, *i.e the projection of* d *onto* A_t.
3. $D_t = I_t \cdot D = \mathbf{diag}(d_t) \in \mathbb{R}^{n\times n}$.
4. $\mathbf{vol}_t = \mathbf{vol}_G(A_t)$.
5. $P_t = I_t - \frac{1}{\mathbf{vol}_t}\sqrt{d_t}\sqrt{d_t'} \in \mathbb{R}^{n\times n}$.

We define the matrix $W_t = (P_t D^{-\frac{1}{2}} F_t D^{-\frac{1}{2}} P_t)^\delta$, where $\delta = \Theta(\log n)$ is set in Lemma 33, that plays a crucial role in this section. This definition is similar to the definition of W_t in Section 3.2, but with P_t instead of P. This makes us "focus" only on the remaining vertices A_t, as any row/column of W_t corresponding to a vertex $v \in R_t$ is zero. The matrix W_t is used in this section to define the projections that our algorithm uses to update F_t. It is also used in Section 5.3 to define the potential that measures how far is the remaining part of the graph from a near expander. In particular, we show in Lemma 33 and Corollary 34 that if W_T^2 has small eigenvalues (which will be the case when the potential is small) then A_T is near-expander in G_T.

Let $r \in \mathbb{R}^n$ be a random unit vector. Consider the projections $u_i = \frac{1}{\sqrt{d(i)}}\langle W_t(i), r\rangle$, for $i \in A_t$. Note that because $P_t\sqrt{d_t} = 0$, and W_t is symmetric:

$$\sum_{i\in A_t} d(i)u_i = \sum_{i\in A_t} \sqrt{d(i)}\,\langle W_t(i), r\rangle = \left\langle \sum_{i\in A_t} \sqrt{d(i)}W_t(i), r \right\rangle = \left\langle W_t\sqrt{d_t}, r\right\rangle = 0$$

We use the following lemma to partition (some of) the remaining vertices into two multisets A_t^l and A_t^r.[13] The lemma follows by applying Lemma 3.3 in [22] on the multiset of the u_i's, where each u_i appears with multiplicity of $d(i)$.

▶ **Lemma 20** (Lemma 3.3 in [22]). *Given* $u_i \in \mathbb{R}$ *for all* $i \in A_t$, *such that* $\sum_{i\in A_t} d(i)u_i = 0$, *we can find in time* $O(|A_t| \cdot \log(|A_t|))$ *a multiset of source nodes* $A_t^l \subseteq A_t$, *a multiset of target nodes* $A_t^r \subseteq A_t$, *and a separation value* η *such that each* $i \in A_t$ *appears in* $A_t^l \cup A_t^r$ *at most* $d(i)$ *times, and additionally:*

1. η *separates the sets* A_t^l, A_t^r, *i.e., either* $\max_{i\in A_t^l} u_i \le \eta \le \min_{j\in A_t^r} u_j$, *or* $\min_{i\in A_t^l} u_i \ge \eta \ge \max_{j\in A_t^r} u_j$,
2. $|A_t^r| \ge \frac{\mathbf{vol}_t}{2}$, $|A_t^l| \le \frac{\mathbf{vol}_t}{8}$,
3. $\forall i \in A_t^l : (u_i - \eta)^2 \ge \frac{1}{9}u_i^2$,
4. $\sum_{i\in A_t^l} m_i u_i^2 \ge \frac{1}{80}\sum_{i\in A_t} d(i)u_i^2$, *where* m_i *is the number of times* i *appears in* A_t^l.

---

[12] These variables are the analogs of $I, d, D, \mathbf{vol}(G)$ and P (respectively) from Section 3.2 in $G[A_t]$.
[13] Note that this does not produce a bisection of V.

Note that a vertex could appear both in A_t^l and in A_t^r, if $u_{i_j} = \eta$. The cut player sends A_t^l, A_t^r and A_t to the matching player.

In turn, the matching player (see Subsection 4.4) returns a cut $(S_t, A_t \setminus S_t)$ and a matching M_t of $A_t^l \setminus S_t$ to $A_t^r \setminus S_t$ (each vertex of A_t^l is matched to a vertex of A_t^r). We add self-loops to M_t to preserve the degrees (that is, M_t is d-stochastic). Define $N_t = \frac{\delta-1}{\delta} D + \frac{1}{\delta} M_t$. The cut player then updates F_t similarly to Section 3.2: $F_{t+1} = N_t \cdot D^{-1} F_t D^{-1} N_t$. Like in the previous sections, we also define the graph G_{t+1} as $G_{t+1} = G_t \cup M_t$.[14]. We define $A_{t+1} = A_t \setminus S_t$.

4.4 Matching player

The matching player receives A_t^l and A_t^r and the current A_t. For a vertex $v \in V$, denote by m_v the number times v appears in A_t^l, and by $\bar{m}_v$ the number of times v appears in A_t^r. The matching player solves the flow problem on $G[A_t]$, specified by Lemma 21 below. This lemma is similar to Lemma B.6 in [23] and is proved using the *Bounded-Distance-Flow* algorithm (called *Unit-Flow* by [13, 23]). The details are provided in the full version of this paper [1]. Note that we can get running time of $\tilde{O}(m)$ mentioned in the introduction by replacing this subroutine is with a fair-cut computation as shown in [17, Section 8].

▶ **Lemma 21.** *Let $G = (V, E)$ be a graph with n vertices and m edges, let $A^l, A^r \subseteq V$ be multisets such that $|A^r| \geq \frac{1}{2}m, |A^l| \leq \frac{1}{8}m$, and let $0 < \phi < \frac{1}{\log n}$ be a parameter. For a vertex $v \in V$, denote by m_v the number times v appears in A^l, and by $\bar{m}_v$ the number of times v appears in A^r. Assume that $m_v + \bar{m}_v \leq d(v)$. We define the flow problem $\Pi(G)$, as the problem in which a source s is connected to each vertex $v \in A^l$ with an edge of capacity m_v and each vertex $v \in A^r$ is connected to a sink t with an edge of capacity $\bar{m}_v$. Every edge of G has the same capacity $c = \Theta\left(\frac{1}{\phi \log n}\right)$, which is an integer. A feasible flow for $\Pi(G)$ is a maximum flow that saturates all the edges outgoing from s. Then, in time $O(\frac{m}{\phi})$, we can find either*

1. *A feasible flow f for $\Pi(G)$; or*
2. *A cut S where $\Phi_G(S, V \setminus S) \leq \frac{7}{c} = O(\phi \log n)$, $\mathbf{vol}(V \setminus S) \geq \frac{1}{3}m$ and a feasible flow for the problem $\Pi(G - S)$, where we only consider the sub-graph $G[V \setminus S \cup \{s, t\}]$ (that is, vertices $v \in A^l \setminus S$ are sources of m_v units, and vertices $v \in A^r \setminus S$ are sinks of $\bar{m}_v$ units).*

▶ Remark 22. It is possible that $A^l \subseteq S$, in which case the feasible flow for $\Pi(G - S)$ is trivial (the total source mass is 0).

Let S_t be the cut returned by the lemma. If the lemma terminates with the first case, we denote $S_t = \emptyset$. Since c is an integer, we can decompose the returned flow into a set of paths (using *e.g.* dynamic trees [26]), each carrying exactly one unit of flow from a vertex $u \in A_t^l \setminus S_t$ to a vertex $v \in A_t^r \setminus S_t$. Note that multiple paths can route flow between the same pair of vertices. If $u \in A_t^l \cap A_t^r$ then it is possible that a path starts and ends at u. Each $u \in A_t^l \setminus S_t$ is the endpoint of exactly $m_u \leq d(u)$ paths, and each $v \in A_t^r \setminus S_t$ is the endpoint of *at most* $\bar{m}_v \leq d(v)$ paths. Define the "matching"[15] $\tilde{M}_t$ as $\tilde{M}_t = ((u_i, v_i))_{i=1}^{|A_t^l \setminus S_t|}$, where u_i and v_i are the endpoints of path i. We can view $\tilde{M}_t$ as a symmetric $n \times n$ matrix, such that $\tilde{M}_t(u, v)$ is the number of paths from u to v. We turn $\tilde{M}_t$ into a d-stochastic matrix by increasing its diagonal entries by $d - \tilde{M}_t \mathbb{1}_n$. Formally, we set $M_t := \tilde{M}_t + \mathbf{diag}(d - \tilde{M}_t \mathbb{1}_n)$.

[14] G_{t+1} may have self-loops.

[15] Note that this is **not** a matching or a d-matching, but rather a graph that connects vertices of A_t^ℓ to vertices of A_t^r, whose degrees are bounded by d.

Notice that $d - \tilde{M}_t \mathbb{1}_n$ has only non-negative entries, so M_t also has non-negative entries. Intuitively, we can think of M_t as the response of the matching player to the subsets A_t^l and A_t^r given by the cut player.

5 Analysis

This section is organized as follows. Subsection 5.1 presents in detail the algorithm for Theorem 17. Subsection 5.2 shows that F_t is embeddable in G_t with congestion $\frac{4}{\delta}$ and that G_t is embeddable in G with congestion $c \cdot t$. Subsection 5.3 shows that if we reach round T, then with high probability, A_T is a near $\Omega(\phi)$-expander in G. Finally, in Subsection 5.4 we prove Theorem 17.

5.1 The Algorithm

Similarly to Section 3.2, let $\delta = \Theta(\log n)$ be a power of 2, let $T = \Theta(\log^2 n)$ and $c = \Theta(\frac{1}{\phi \log n})$. We choose c to be an integer. The algorithm follows along the same lines as the algorithm of SW in Section 4.1. The only modifications are the usage of our new cut player and that the algorithm stops if $\mathbf{vol}(R_t) > \frac{m \cdot c \cdot \phi}{70} = \Omega(\frac{m}{\log n})$. In each round t, we implicitly update F_t (see Section 4.3). Like SW, in order to keep the running time near linear, we use the flow routine *Bounded-Distance-Flow* [13, 23] which is mentioned in Subsection 4.4. This routine may also return a cut $S_t \subseteq A_t$ with $\Phi_{G[A]}(S_t, A_t \setminus S_t) \leq \frac{1}{c}$, in which case we "move" S_t to R_{t+1}. After T rounds, F_T certifies that the remaining part of A_T is a near ϕ-expander.

5.2 F_t is embeddable in G

To begin the analysis of the algorithm, we first define a blocked matrix. This notion will be useful when our matrices "operate" only on vertices of A_t.

▶ **Definition 23.** *Let $A \subseteq V$. A matrix $B \in \mathbb{R}^{n \times n}$ is A-*blocked *if $B(i,j) = 0$ for all $i \neq j$ such that $(i,j) \notin A \times A$.*

▶ **Lemma 24.** *The following holds for all t:*
1. *M_t, N_t, F_t and W_t are symmetric.*
2. *M_t, N_t and F_t are d-stochastic.*
3. *M_t and N_t are A_{t+1}-blocked.*

▶ **Lemma 25.** *For all rounds t, F_t is embeddable in G_t with congestion $\frac{4}{\delta}$.*

▶ **Lemma 26.** *For all rounds t, G_t is embeddable in G with congestion ct.*

5.3 A_T is a near expander in F_T

In this section we prove that after $T = \Theta(\log^2 n)$ rounds, with high probability, A_T is a near $\Omega(1)$-expander in F_T, which will imply that it is a near $\Omega(\phi)$-expander in G.

The section is organized as follows. Lemma 27 contains matrix identities and Lemma 28 specifies a spectral property that our proof requires. We then define a potential function and lower bound the decrease in potential in Lemmas 29-32. Finally, in Lemma 33 and Corollary 34 we use the lower bound on the potential at round T, to show that with high probability A_T is a near $\Omega(1)$-expander in F_T and a near $\Omega(\phi)$-expander in G.

▶ **Lemma 27.** *The following relations hold for all t:*

1. *For any A_t-blocked d-stochastic matrix $B \in \mathbb{R}^{n\times n}$ we have $I_t D^{-\frac{1}{2}} B D^{-\frac{1}{2}} = D^{-\frac{1}{2}} B D^{-\frac{1}{2}} I_t$ and $P_t \cdot D^{-\frac{1}{2}} B D^{-\frac{1}{2}} = D^{-\frac{1}{2}} B D^{-\frac{1}{2}} \cdot P_t$.*
2. *$I_t P_t = P_t$, $I_t^2 = I_t$ and $P_t^2 = P_t$.*
3. *$P_t P_{t+1} = P_{t+1} P_t = P_{t+1}$.*
4. *$P_t = D^{-\frac{1}{2}} \mathcal{L}(\frac{1}{\mathbf{vol}_t} d_t d_t') D^{-\frac{1}{2}}$ (recall the Laplacian defined in Definition 7).*
5. *for any $v \in \mathbb{R}^n$, it holds that $v' \mathcal{L}\left(\frac{1}{\mathbf{vol}_t} d_t d_t'\right) v = \left\| D_t^{\frac{1}{2}} v \right\|_2^2 - \frac{1}{\mathbf{vol}_t} \langle v, d_t \rangle^2$.*
6. *For any $B \in \mathbb{R}^{n\times n}$, $\mathrm{Tr}(I_t B B') = \sum_{i\in A_t} \|B(i)\|_2^2$.*

We define the potential $\psi(t) = \mathrm{Tr}[W_t^2] = \sum_{i\in A_t} \|W_t(i)\|_2^2$, where W_t was defined as $W_t = (P_t D^{-\frac{1}{2}} F_t D^{-\frac{1}{2}} P_t)^{\delta}$. This is the same potential from Section 3.2 with the new definition of W_t. Intuitively, by projecting using P_t instead of P, the potential only "cares" about the vertices of A_t. As show in Lemma 33, having small potential will certify that A_T is a near expander in F_t.

Before we bound the decrease in potential, we recall Definition 7 of a normalized Laplacian $\mathcal{N}(A) = D^{-\frac{1}{2}} \mathcal{L}(A) D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where A is a symmetric d-stochastic matrix.

▶ **Lemma 28.** *For any matrix $A \in \mathbb{R}^{n\times n}$, $\mathrm{Tr}(A'(I - (D^{-\frac{1}{2}} N_t D^{-\frac{1}{2}})^{4\delta})A) \geq \frac{1}{3} \mathrm{Tr}(A' \mathcal{N}(M_t) A)$.*

The following lemma bounds the decrease in potential. The bound takes into account both the contribution of the matched vertices and the removal of S_t from A_t.

▶ **Lemma 29.** *For each round t,*

$$\psi(t) - \psi(t+1) \geq \frac{1}{3} \sum_{\{i,k\}\in M_t} \left\| \left(\frac{W_t(i)}{\sqrt{d(i)}} - \frac{W_t(k)}{\sqrt{d(k)}} \right) \right\|_2^2 + \sum_{j\in S_t} d(j) \left\| \frac{W_t(j)}{\sqrt{d(j)}} \right\|_2^2$$

Proof. To simplify the notation, we denote $\bar{N}_t := D^{-\frac{1}{2}} N_t D^{-\frac{1}{2}}$ and $\bar{F}_t := D^{-\frac{1}{2}} F_t D^{-\frac{1}{2}}$. We rewrite the potential in the next iteration as follows:

$$\begin{aligned}
\psi(t+1) = \mathrm{Tr}(W_{t+1}^2) &= \mathrm{Tr}\left(\left(P_{t+1} D^{-\frac{1}{2}} F_{t+1} D^{-\frac{1}{2}} P_{t+1} \right)^{2\delta} \right) \\
&= \mathrm{Tr}\left(\left(P_{t+1} D^{-\frac{1}{2}} (N_t D^{-1} F_t D^{-1} N_t) D^{-\frac{1}{2}} P_{t+1} \right)^{2\delta} \right) \\
&= \mathrm{Tr}\left(\left(P_{t+1} D^{-\frac{1}{2}} (N_t D^{-\frac{1}{2}} D^{-\frac{1}{2}} F_t D^{-\frac{1}{2}} D^{-\frac{1}{2}} N_t) D^{-\frac{1}{2}} P_{t+1} \right)^{2\delta} \right) \\
&= \mathrm{Tr}\left(\left(P_{t+1} \bar{N}_t \bar{F}_t \bar{N}_t P_{t+1} \right)^{2\delta} \right) \overset{(6)}{=} \mathrm{Tr}\left(\left(\bar{N}_t P_{t+1} \bar{F}_t P_{t+1} \bar{N}_t \right)^{2\delta} \right) \\
&\overset{(7)}{=} \mathrm{Tr}\left(\left(\bar{N}_t P_{t+1} P_t \bar{F}_t P_t P_{t+1} \bar{N}_t \right)^{2\delta} \right) = \mathrm{Tr}\left(\left(\bar{N}_t P_{t+1} (P_t \bar{F}_t P_t) P_{t+1} \bar{N}_t \right)^{2\delta} \right) ,
\end{aligned}$$

where equality (6) follows from Lemma 27 (1) for N_t (which is A_{t+1}-blocked d-stochastic by Lemma 24), and equality (7) follows from Lemma 27 (3).

By Properties (1) and (2) of Lemma 27 it holds that $\bar{N}_{t+1} P_{t+1} = P_{t+1} \bar{N}_{t+1} = P_{t+1} \bar{N}_{t+1} P_{t+1}$.

Therefore, the potential can be written in terms of symmetric matrices:

$$\begin{aligned}
\psi(t+1) &= \mathrm{Tr}\left(\left((P_{t+1}\bar{N}_tP_{t+1})(P_t\bar{F}_tP_t)(P_{t+1}\bar{N}_tP_{t+1})\right)^{2\delta}\right)\\
&\le \mathrm{Tr}((P_{t+1}\bar{N}_tP_{t+1})^{2\delta}(P_t\bar{F}_tP_t)^{2\delta}(P_{t+1}\bar{N}_tP_{t+1})^{2\delta})\\
&\overset{(2)}{=} \mathrm{Tr}((P_{t+1}\bar{N}_tP_{t+1})^{4\delta}(P_t\bar{F}_tP_t)^{2\delta}) = \mathrm{Tr}((\bar{N}_tP_{t+1})^{4\delta}W_t^2)\\
&\overset{(4)}{=} \mathrm{Tr}(\bar{N}_t^{4\delta}P_{t+1}W_t^2) \overset{(5)}{=} \mathrm{Tr}(\bar{N}_t^{2\delta}P_{t+1}\bar{N}_t^{2\delta}W_t^2) \overset{(6)}{=} \mathrm{Tr}(W_t\bar{N}_t^{2\delta}P_{t+1}\bar{N}_t^{2\delta}W_t)\\
&\overset{(7)}{=} \mathrm{Tr}(W_t\bar{N}_t^{2\delta}D^{-\frac{1}{2}}\mathcal{L}\left(\frac{1}{\mathbf{vol}_{t+1}}d_{t+1}d'_{t+1}\right)D^{-\frac{1}{2}}\bar{N}_t^{2\delta}W_t)\\
&= \mathrm{Tr}\left(\left(D^{-\frac{1}{2}}\cdot\bar{N}_t^{2\delta}W_t\right)'\cdot\mathcal{L}\left(\frac{1}{\mathbf{vol}_{t+1}}d_{t+1}d'_{t+1}\right)\cdot\left(D^{-\frac{1}{2}}\cdot\bar{N}_t^{2\delta}W_t\right)\right),
\end{aligned}$$

where the inequality follows from Fact 2, equality (2) follows from Fact 1. Equalities (4) and (5) follow from Properties (1) and (2) of Lemma 27 (and from the fact that N_t is A_{t+1}-blocked d-stochastic, by Lemma 24). Equality (6) again uses Fact 1, and equality (7) follows from Lemma 27 (4).

Let $Z_t = D^{-\frac{1}{2}}\cdot\bar{N}_t^{2\delta}W_t$. By applying Lemma 27 (5) we get

$$\begin{aligned}
\psi(t+1) &\le \mathrm{Tr}\left(Z'_t\mathcal{L}\left(\frac{1}{\mathbf{vol}_{t+1}}d_{t+1}d'_{t+1}\right)Z_t\right) = \sum_{i=1}^{n}(Z_t(,i))'\mathcal{L}\left(\frac{1}{\mathbf{vol}_{t+1}}d_{t+1}d'_{t+1}\right)Z_t(,i)\\
&\overset{(2)}{=} \sum_{i=1}^{n}\left(\left\|D_{t+1}^{\frac{1}{2}}Z_t(,i)\right\|_2^2 - \frac{1}{\mathbf{vol}_{t+1}}\langle Z_t(,i),d_{t+1}\rangle^2\right) \le \sum_{i=1}^{n}\left\|D_{t+1}^{\frac{1}{2}}Z_t(,i)\right\|_2^2\\
&= \sum_{i=1}^{n}\sum_{j\in A_{t+1}}\left(\sqrt{d(j)}Z_t(j,i)\right)^2 = \sum_{j\in A_{t+1}}\left\|\left(D_{t+1}^{\frac{1}{2}}Z_t\right)(j)\right\|_2^2 \overset{(5)}{=} \sum_{j\in A_{t+1}}\left\|\left(\bar{N}_t^{2\delta}W_t\right)(j)\right\|_2^2\\
&= \sum_{j\in A_t}\left\|\left(\bar{N}_t^{2\delta}W_t\right)(j)\right\|_2^2 - \sum_{j\in S_t}\left\|\left(\bar{N}_t^{2\delta}W_t\right)(j)\right\|_2^2, \qquad (1)
\end{aligned}$$

where equality (2) holds by Property (5) of Lemma 27 and equality (5) holds since we only sum rows in A_{t+1}. Since $\bar{N}_t$ is diagonal outside A_{t+1} (by the definition of M_t), we have that $\left(\bar{N}_t^{2\delta}W_t\right)(j) = W_t(j)$, for every $j\in S_t$. Thus,

$$\sum_{j\in S_t}\left\|\left(\bar{N}_t^{2\delta}W_t\right)(j)\right\|_2^2 = \sum_{j\in S_t}\|W_t(j)\|_2^2. \qquad (2)$$

By Lemma 27 (6), we get

$$\begin{aligned}
\sum_{j\in A_t}\left\|\left(\bar{N}_t^{2\delta}W_t\right)(j)\right\|_2^2 &= \mathrm{Tr}(I_t\cdot\bar{N}_t^{2\delta}\cdot W_t^2\cdot\bar{N}_t^{2\delta}) = \mathrm{Tr}(\bar{N}_t^{2\delta}\cdot I_t\cdot W_t^2\cdot\bar{N}_t^{2\delta})\\
&= \mathrm{Tr}(\bar{N}_t^{2\delta}\cdot W_t^2\cdot\bar{N}_t^{2\delta}) = \mathrm{Tr}(\bar{N}_t^{4\delta}W_t^2) \qquad (3)
\end{aligned}$$

where second equality holds since N_t is A_{t+1}-blocked d-stochastic (by Lemma 24), so in particular it is A_t-blocked d-stochastic, and we can use Lemma 27 (1). The third equality holds because $I_tW_t = I_t(P_t\bar{F}_tP_t)^{\delta}$ and $I_tP_t = P_t$ (by Lemma 27 (2)), and the last equality follows from Fact 1. Plugging Equations (2) and (3) into (1) we get the following bound on the decrease in potential:

$$
\begin{aligned}
\psi(t) - \psi(t+1) &\geq \mathrm{Tr}((I - \bar{N}_t^{4\delta})W_t^2) + \sum_{j \in S_t} \|W_t(j)\|_2^2 \\
&= \mathrm{Tr}(W_t(I - \bar{N}_t^{4\delta})W_t) + \sum_{j \in S_t} \|W_t(j)\|_2^2 \geq \frac{1}{3}\,\mathrm{Tr}(W_t \mathcal{N}(M_t) W_t) + \sum_{j \in S_t} \|W_t(j)\|_2^2 \\
&= \frac{1}{3}\,\mathrm{Tr}((D^{-\frac{1}{2}}W_t)'\mathcal{L}(M_t)(D^{-\frac{1}{2}}W_t)) + \sum_{j \in S_t} d(j) \left\| \frac{W_t(j)}{\sqrt{d(j)}} \right\|_2^2 \\
&= \frac{1}{3} \sum_{\{i,k\} \in M_t} \left\| \frac{W_t(i)}{\sqrt{d(i)}} - \frac{W_t(k)}{\sqrt{d(k)}} \right\|_2^2 + \sum_{j \in S_t} d(j) \left\| \frac{W_t(j)}{\sqrt{d(j)}} \right\|_2^2
\end{aligned}
$$

where the second inequality follows Lemma 28, and the last equality follows from by Laplacian matrix properties. ◀

The following lemma states that the potential is expected to drop by a factor of $1 - \Omega(1/\log n)$.

▶ **Lemma 30.** *For each round t,*

$$
\mathbb{E}\left[\frac{1}{3} \sum_{\{i,k\} \in M_t} \left\| \frac{W_t(i)}{\sqrt{d(i)}} - \frac{W_t(k)}{\sqrt{d(k)}} \right\|_2^2 + \sum_{j \in S_t} d(j) \left\| \frac{W_t(j)}{\sqrt{d(j)}} \right\|_2^2\right] \geq \frac{1}{3000\alpha \log n}\psi(t) - \frac{3}{n^{\alpha/16}}
$$

for every $\alpha > 48$, where the expectation is over the unit vector $r \in \mathbb{R}^n$.

The following two corollaries follow by Lemmas 29 and 30.

▶ **Corollary 31.** *For each round t, $\mathbb{E}[\psi(t+1)] \leq \left(1 - \frac{1}{3000\alpha \log n}\right)\psi(t) + \frac{3}{n^{\alpha/16}}$, where the expectation is over the unit vector $r \in \mathbb{R}^n$.*

▶ **Corollary 32** (Total Decrease in Potential)**.** *With high probability over the choices of r, $\psi(T) \leq \frac{1}{n}$.*

The following lemma uses the low potential to derive the near-expansion of A_T in F_T.

▶ **Lemma 33** (Variation of Cheeger's inequality)**.** *Let $H = (V, \bar{E})$ be a graph on n vertices, such that F_T is its weighted adjacency matrix. Assume that $\psi(T) \leq \frac{1}{n}$. Then, A_T is a near $\frac{1}{5}$-expander in H.*

Proof. Recall that F_T is symmetric and d-stochastic. Let $k = \mathbf{vol}(A_T)$. Let $S \subseteq A_T$ be a cut, and denote $d_S \in \mathbb{R}^n$ to be the vector where $d_S(u) = \begin{cases} d(u) & \text{if } u \in S, \\ 0 & \text{otherwise.} \end{cases}$ Additionally, denote $\ell = \mathbf{vol}(S) \leq \frac{1}{2}k$. Note that $\left\|\sqrt{d_S}\right\|_2^2 = \ell$.

Denote by $\bar{\lambda} \geq 0$ the largest singular value of $X_T := P_T D^{-\frac{1}{2}} F_T D^{-\frac{1}{2}} P_T$ (square root of the largest eigenvalue of $(P_T D^{-\frac{1}{2}} F_T D^{-\frac{1}{2}} P_T)^2$). Because $\mathrm{Tr}(X_T^{2\delta}) = \psi(T) \leq \frac{1}{n}$, we have in particular that the largest eigenvalue of $X_T^{2\delta}$ is at most $\frac{1}{n}$, so we have $\bar{\lambda} \leq \frac{1}{n^{\frac{1}{\delta}}}$. We choose $\delta = \Theta(\log n)$ such that $\frac{1}{n^{\frac{1}{\delta}}} \leq \frac{1}{20}$, so $\bar{\lambda} \leq \frac{1}{20}$.

In order to prove near-expansion we need to lower bound $|E_{F_T}(S, V \setminus S)|$. We do so by upper bounding $|E_{F_T}(S, S)| = \mathbb{1}'_S F_T \mathbb{1}_S$. Note that $\mathbb{1}'_S F_T \mathbb{1}_S = \mathbb{1}'_S (I_T F_T I_T) \mathbb{1}_S$. Observe the following relation between X_T and $I_T F_T I_T$:

$$\begin{aligned} D^{\frac{1}{2}} X_T D^{\frac{1}{2}} &= D^{\frac{1}{2}}(P_T D^{-\frac{1}{2}} F_T D^{-\frac{1}{2}} P_T) D^{\frac{1}{2}} \\ &= D^{\frac{1}{2}}(I_T - \frac{1}{k}\sqrt{d_T}\sqrt{d'_T}) D^{-\frac{1}{2}} F_T D^{-\frac{1}{2}} (I_T - \frac{1}{k}\sqrt{d_T}\sqrt{d'_T}) D^{\frac{1}{2}} \\ &= (I_T - \frac{1}{k} d_T \mathbb{1}'_T) F_T (I_T - \frac{1}{k}\mathbb{1}_T d'_T) \\ &= I_T F_T I_T - \frac{1}{k} d_T \mathbb{1}'_T F_T I_T - \frac{1}{k} I_T F_T \mathbb{1}_T d'_T + \frac{1}{k^2} d_T \mathbb{1}'_T F_T \mathbb{1}_T d'_T. \end{aligned}$$

Rearranging the terms, we get

$$I_T F_T I_T = D^{\frac{1}{2}} X_T D^{\frac{1}{2}} + \frac{1}{k} d_T \mathbb{1}'_T F_T I_T + \frac{1}{k} I_T F_T \mathbb{1}_T d'_T - \frac{1}{k^2} d_T \mathbb{1}'_T F_T \mathbb{1}_T d'_T \ .$$

Therefore

$$|E_{F_T}(S,S)| = \mathbb{1}'_S F_T \mathbb{1}_S = \mathbb{1}'_S \left(D^{\frac{1}{2}} X_T D^{\frac{1}{2}} + \frac{1}{k} d_T \mathbb{1}'_T F_T I_T + \frac{1}{k} I_T F_T \mathbb{1}_T d'_T - \frac{1}{k^2} d_T \mathbb{1}'_T F_T \mathbb{1}_T d'_T \right) \mathbb{1}_S.$$

We analyze the summands separately. The first summand can be bounded using $\bar{\lambda}$, the largest singular value of X_T:

$$\mathbb{1}'_S D^{\frac{1}{2}} X_T D^{\frac{1}{2}} \mathbb{1}_S = \sqrt{d'_S} X \sqrt{d_S} = \left\langle \sqrt{d_S}, X\sqrt{d_S} \right\rangle \le \left\| \sqrt{d_S} \right\|_2 \left\| X_T \sqrt{d_S} \right\|_2 \le \left\| \sqrt{d_S} \right\|_2^2 \bar{\lambda} \le \frac{\ell}{20},$$

where the first inequality is the Cauchy-Schwartz inequality. Observe that the second and third summands are equal:

$$\frac{1}{k} \mathbb{1}'_S d_T \mathbb{1}'_T F_T I_T \mathbb{1}_S = \frac{\ell}{k} \mathbb{1}'_T F_T \mathbb{1}_S = \frac{\ell}{k} \mathbb{1}'_S F_T \mathbb{1}_T = \frac{1}{k} \mathbb{1}'_S I_T F_T \mathbb{1}_T d'_T \mathbb{1}_S,$$

where the second equality follows by transposing and since F_T is symmetric. We now bound the sum of the second, third and fourth summands:

$$\begin{aligned} &\mathbb{1}'_S \left(\frac{1}{k} d_T \mathbb{1}'_T F_T I_T + \frac{1}{k} I_T F_T \mathbb{1}_T d'_T - \frac{1}{k^2} d_T \mathbb{1}'_T F_T \mathbb{1}_T d'_T \right) \mathbb{1}_S = \frac{2\ell}{k} \mathbb{1}'_T F_T \mathbb{1}_S - \frac{\ell^2}{k^2} \mathbb{1}'_T F_T \mathbb{1}_T \\ &\le \left(\frac{2\ell}{k} - \frac{\ell^2}{k^2} \right) \mathbb{1}'_T F_T \mathbb{1}_S \le \left(\frac{2\ell}{k} - \frac{\ell^2}{k^2} \right) \mathbb{1}' F_T \mathbb{1}_S = \left(\frac{2\ell}{k} - \frac{\ell^2}{k^2} \right) d' \mathbb{1}_S = \frac{\ell}{k} \left(2 - \frac{\ell}{k} \right) \ell, \end{aligned}$$

where the first inequality follows since $S \subseteq A_t$. Note that $\frac{\ell}{k} \in [0, \frac{1}{2}]$. The last inequality is true because for $\frac{\ell}{k}$ in this range, $\left(\frac{2\ell}{k} - \frac{\ell^2}{k^2} \right) \ge 0$. Moreover, because $\frac{\ell}{k} \in \left[0, \frac{1}{2}\right]$, we have $\frac{\ell}{k}\left(2 - \frac{\ell}{k}\right) \le \frac{3}{4}$. Therefore, $|E_{F_T}(S,S)| \le \frac{1}{20}\ell + \frac{3}{4}\ell = \frac{4}{5}\ell$, and

$$\begin{aligned} |E(S, V \setminus S)| &= \sum_{u \in S} \sum_{v \in V \setminus S} F_T(u,v) = \sum_{u \in S} \sum_{v \in V} F_T(u,v) - \sum_{u \in S} \sum_{v \in S} F_T(u,v) \\ &= \sum_{u \in S} d(u) - \sum_{u \in S} \sum_{v \in S} F_T(u,v) \ge \ell - \frac{4}{5}\ell = \frac{\ell}{5} \ . \end{aligned}$$

So $\Phi_G(S, V \setminus S) = \frac{|E(S,V\setminus S)|}{\mathbf{vol}(S)} \ge \frac{1}{5}$, and this is true for all cuts $S \subseteq A$ with $\frac{\mathbf{vol}(S)}{\mathbf{vol}(A)} \le \frac{1}{2}$. ◀

▶ **Corollary 34.** *If we reach round T, then with high probability, A_T is a near $\Omega(\phi)$-expander in G.*

Proof. Assume we reach round T. By Corollary 32 and Lemma 33, with high probability, A_T is a near $\Omega(1)$-expander in F_T. By Lemma 25, F_T is embeddable in G_T with congestion $O(\frac{1}{\delta})$. Note that G_T is a union of T d_G-matchings $\{M_t\}_{t=1}^{T}$, each having $d_{M_t} = d_G = d_{F_T}$. Therefore, $d_{G_T} = T \cdot d_{F_T}$. So by Lemma 11, A_T is a near $\Omega(\frac{\delta}{T})$-expander in G_T. By Lemma 26, G_T is embeddable in G with congestion cT. Together with the fact that $d_G = \frac{1}{T} \cdot d_{G_T}$, we get by Lemma 11 again, that A is a near $\Omega(\frac{\delta}{cT})$-expander in G. Recall that $c = O\left(\frac{1}{\phi \log n}\right)$, $\delta = \Theta(\log n)$, and $T = O(\log^2 n)$. Therefore, A is an near $\Omega(\phi)$-expander in G. ◀

5.4 Proof of Theorem 17

We are now ready to prove Theorem 17.

Proof of Theorem 17. Recall that S_t denotes the cut returned by Lemma 21 at iteration t, so that $A_{t+1} = A_t \setminus S_t$.

Observe first that in any round t, we have $\Phi_G(A_t, R_t) \leq \frac{7}{c} = O(\phi \log n)$. This is because $R_t = \bigcup_{0 \leq t' < t} S_{t'}$ and by Lemma 21, for each t', $\Phi_{G[A_{t'}]}(S_{t'}, V \setminus S_{t'}) \leq \frac{7}{c} = O(\phi \log n)$.

Assume the algorithm terminates because $\mathbf{vol}(R_t) > \frac{m \cdot c \cdot \phi}{70} = \Omega(\frac{m}{\log n})$. We also have, by Lemma 21, that $\mathbf{vol}(A_t) = \Omega(m) = \Omega(\frac{m}{\log n})$. Then (A_t, R_t) is a balanced cut where $\Phi_G(A_t, R_t) = O(\phi \log n)$. We end in Case (2) of Theorem 17.

Otherwise, the algorithm reached round T and we apply Corollary 34. If $R = \emptyset$, then we obtain the first case of Theorem 17 because the whole vertex set V is, with high probability, a near $\Omega(\phi)$-expander, which means that G is an $\Omega(\phi)$-expander. Otherwise, we write $c = \frac{c_1}{\phi \log n}$ for some constant c_1, and let $c_0 := \frac{7}{c_1}$. We have $\Phi_G(A_T, R_T) \leq \frac{7}{c} = \frac{7}{c_1} \phi \log n = c_0 \phi \log n$. Additionally, $\mathbf{vol}(R_T) \leq \frac{m \cdot c \cdot \phi}{70} = \frac{m \cdot c_1}{70 \log n} = \frac{m}{10 c_0 \log n}$, and, with high probability, A_T is a near $\Omega(\phi)$-expander in G, which means we obtain the third case of Theorem 17.

To bound the running time, note that the algorithm performs at most $T = \Theta(\log^2 n)$ iterations and each iteration's running time is dominated by computing $W_t \cdot r$ in $O(t \cdot \delta \cdot m)$ and by running the matching player (Lemma 21) in $O(\frac{m}{\phi})$. ◀

References

1 Daniel Agassy, Dani Dorfman, and Haim Kaplan. Expander decomposition with fewer inter-cluster edges using a spectral cut player. *arXiv preprint arXiv:2205.10301*, 2022.

2 Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph clustering using effective resistance. *arXiv preprint arXiv:1711.06530*, 2017.

3 Konstantinos Ameranis, Lorenzo Orecchia, Kunal Talwar, and Charalampos Tsourakakis. Practical almost-linear-time approximation algorithms for hybrid and overlapping graph clustering. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 17071–17093, 2022.

4 Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009.

5 Yi-Jun Chang and Thatchaphol Saranurak. Improved distributed expander decomposition and nearly optimal triangle enumeration. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODS)*, pages 66–73, 2019.

6 Yi-Jun Chang and Thatchaphol Saranurak. Deterministic distributed expander decomposition and routing with applications in distributed derandomization. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 377–388, 2020.

7 Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems in analysis*, 625(195-199):110, 1970.

8 Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *arXiv preprint arXiv:2203.00671*, 2022.

9 Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junxing Wang. Graph sparsification, spectral sketches, and faster resistance computation via short cycle decompositions. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 18–85, 2020.

10 Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1158–1167, 2020.

11 Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual Symposium on Theory of Computing (STOC)*, pages 410–419, 2017.

12 Gramoz Goranci, Harald Räcke, Thatchaphol Saranurak, and Zihan Tan. The expander hierarchy and its applications to dynamic graph algorithms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2212–2228. SIAM, 2021.

13 Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1919—1938, 2017.

14 Yiding Hua, Rasmus Kyng, Maximilian Probst Gutenberg, and Zihang Wu. Maintaining expander decompositions via sparse cuts. *arXiv preprint arXiv:2204.02519*, 2022.

15 Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 217–226, 2014.

16 Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM*, 56(4):1–15, 2009.

17 Jason Li, Danupon Nanongkai, Debmalya Panigrahi, and Thatchaphol Saranurak. Near-linear time approximations for cut problems via fair cuts. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 240–275, 2023.

18 David W. Matula and Farhad Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discrete Applied Mathematics*, 27(1-2):113–123, 1990.

19 Lorenzo Orecchia. *Fast approximation algorithms for graph partitioning using spectral and semidefinite-programming techniques.* PhD thesis, Berkeley, 2011.

20 Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K. Vishnoi. Approximating the exponential, the lanczos method and an $\tilde{O}(m)$-time spectral algorithm for balanced separator. In *Proceedings of the 44th Annual Symposium on Theory of Computing (STOC)*, pages 1141–1160, 2012.

21 Lorenzo Orecchia, Leonard J. Schulman, Umesh V. Vazirani, and Nisheeth K. Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the 40th Annual Symposium on Theory of Computing (STOC)*, pages 461–470, 2008.

22 Harald Räcke, Chintan Shah, and Hanjo Täubig. Computing cut-based hierarchical decompositions in almost linear time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 227–238, 2014.

23 Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2616–2635, 2019.

24 Jonah Sherman. Breaking the multicommodity flow barrier for $O(\sqrt{\log n})$-approximations to sparsest cut. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 363–372. IEEE, 2009.

25 Jiří Šíma and Satu Elisa Schaeffer. On the NP-completeness of some graph cluster measures. In *International Conference on Current Trends in Theory and Practice of Computer Science (SOFTSEM)*, pages 530–537. Springer, 2006.

26 Daniel D. Sleator and Robert Endre Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983.

27 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual Symposium on Theory of Computing (FOCS)*, pages 81–90, 2004.