Sublinear Algorithms for MAXCUT and Correlation Clustering

Aditya Bhaskara
School of Computing, University of Utah, USA
http://www.cs.utah.edu/~bhaskara
bhaskara@cs.utah.edu

Samira Daruki
Expedia Research, USA
https://sites.google.com/site/samiradaruki
sdaruki@expedia.com

Suresh Venkatasubramanian
School of Computing, University of Utah, USA
http://www.cs.utah.edu/~suresh
suresh@cs.utah.edu

Abstract

We study sublinear algorithms for two fundamental graph problems, MAXCUT and correlation clustering. Our focus is on constructing core-sets as well as developing streaming algorithms for these problems. Constant space algorithms are known for *dense* graphs for these problems, while $\Omega(n)$ lower bounds exist (in the streaming setting) for sparse graphs.

Our goal in this paper is to bridge the gap between these extremes. Our first result is to construct core-sets of size $\widetilde{O}(n^{1-\delta})$ for both the problems, on graphs with average degree n^{δ} (for any $\delta > 0$). This turns out to be optimal, under the exponential time hypothesis (ETH). Our core-set analysis is based on studying random-induced sub-problems of optimization problems. To the best of our knowledge, all the known results in our parameter range rely crucially on near-regularity assumptions. We avoid these by using a biased sampling approach, which we analyze using recent results on concentration of quadratic functions. We then show that our construction yields a 2-pass streaming $(1+\varepsilon)$-approximation for both problems; the algorithm uses $\widetilde{O}(n^{1-\delta})$ space, for graphs of average degree n^{δ}.

2012 ACM Subject Classification Theory of computation → Sketching and sampling

Keywords and phrases Sublinear algorithms, Streaming algorithms, Core-sets, Maximum cut, Correlation clustering

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.16

Related Version A full version of the paper is available at https://arxiv.org/abs/1802.06992.

Funding This research was supported in part by National Science Foundation under grants IIS-1251049, IIS-1633724

1 Introduction

Sublinear algorithms are a powerful tool for dealing with large data problems. The range of questions that can be answered accurately using sublinear (or even polylogarithmic) space or time is enormous, and the underlying techniques of sketching, streaming, sampling and core-sets have been proven to be a rich toolkit.

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 16; pp. 16:1–16:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

When dealing with large graphs, the sublinear paradigm has yielded many powerful results. For many NP-hard problems on graphs, classic results from property testing [20, 7] imply extremely efficient sublinear approximations. In the case of dense graphs, these results (and indeed older ones of [10, 16]) provide constant time/space algorithms. More recently, graph sketching techniques have been used to obtain efficient approximation algorithms for cut problems on graphs [2, 3] in a streaming setting. These algorithms use space that is nearly linear in n (the number of vertices) and are sublinear in the number of edges as long as $|E| = \omega(n)$ (this is called the "semi-streaming" setting).

By way of lower bounds, recent results have improved our understanding of the limits of sketching and streaming. In a sequence of results [22, 23, 25], it was shown that for problems like matching and MaxCut in a streaming setting, $\Omega(n)$ space is necessary in order to obtain any approximation better than a factor 2 in one round. (Note that a factor 2 is trivial by simply counting edges.) Furthermore, Andoni et al. [9] showed that any sketch for all the cuts in a graph must have size $\Omega(n)$.

While these lower bounds show that $O(n)$ space is the best possible for approximating problems like MaxCut in general, the constructions used in these bounds are quite specialized. In particular, the graphs involved are sparse, i.e., have $\Theta(n)$ edges. Meanwhile, as we mentioned above, if a graph is *dense* ($\Omega(n^2)$ edges), random sampling is known to give $O(1)$ space and time algorithms. The question we study in this paper is if there is a middle ground: can we get truly sublinear (i.e., $o(n)$) algorithms for natural graph problems in between (easy) dense graphs and (hard) sparse graphs?

Our main contribution is to answer this in the affirmative. As long as a graph has average degree n^δ for some $\delta > 0$, truly sub-linear space $(1 + \epsilon)$ approximation algorithms are possible for problems such as MaxCut and correlation clustering.[1] Indeed, we show that a biased sample of vertices forms a "core-set" for these problems. A core-set for an optimization problem (see [1]), is a subset of the input with the property that a solution to the subset provides an approximation to the solution on the entire input.

Our arguments rely on understanding the following fundamental question: given a graph G, is the induced subgraph on a random subset of vertices a core-set for problems such as MaxCut? This question of sub-sampling and its effect on the value of an optimization problem is well studied. Results from property testing imply that a uniformly random sample of constant size suffices for many problems on *dense* graphs. [16, 6] generalized these results to the case of arbitrary k-CSPs. More recently, [12], extending a result in [14], studied the setting closest to ours. For graphs, their results imply that when the maximum and minimum degrees are both $\Theta(n^\delta)$, then a random induced subgraph with $\widetilde{O}(n^{1-\delta})$ acts as a core-set for problems such as MaxCut. Moreover, they showed that for certain lifted relaxations, subsampling does *not* preserve the value of the objective. Finally, using more modern techniques, [31] showed that the *cut norm* of a matrix (a quantity related to the MaxCut) is preserved up to a constant under random sampling, improving on [16, 6]. While powerful, we will see that these results are not general enough for our setting. Thus we propose a new, conceptually simple technique to analyze sub-sampling, and present it in the context of MaxCut and correlation clustering.

1.1 Our Results

As outlined above, our main result is to show that there exist core-sets of size $\widetilde{O}(n^{1-\delta})$ for MaxCut and correlation clustering for graphs with $\Omega(n^{1+\delta})$ edges (where $0 < \delta \leq 1$). This

[1] We consider the max-agreement version of correlation clustering (see Section 2).

then leads to a two-pass streaming algorithm for MaxCut and correlation clustering on such graphs, that uses $\widetilde{O}(n^{1-\delta})$ space and produces a $1+\varepsilon$ approximation.

This dependence of the core-set size on δ is optimal up to logarithmic factors, by a result of [15]. Specifically, [15] showed that any $(1+\varepsilon)$ approximation algorithm for MaxCut on graphs of average degree n^{δ} must have running time $2^{\Omega(n^{1-\delta})}$, assuming the exponential time hypothesis (ETH). Since a core-set of size $o(n^{1-\delta})$ would trivially allow such an algorithm (we can perform exhaustive search over the core-set), our construction is optimal up to a logarithmic factor, assuming ETH.

Our streaming algorithm for correlation clustering can be viewed as improving the semi-streaming (space $\widetilde{O}(n)$) result of Ahn et al. [4], while using an additional pass over the data. Also, in the context of the lower bound of Andoni et al. [9], our result for MaxCut can be interpreted as saying that while a sketch that approximately maintains *all* cuts in a graph requires an $\Omega(n)$ size, one that preserves the MaxCut can be significantly smaller, when the graph has a polynomial average degree.

At a technical level, we analyze the effect of sampling on the value of the MaxCut and correlation clustering objectives. As outlined above, several techniques are known for such an analysis, but we give a new and conceptually simple framework that (a) allows one to analyze *non-uniform* sampling for the first time, and (b) gets over the assumptions of near-regularity (crucial for [14, 12]) and density (as in [16, 6]). We expect the ideas from our analysis to be applicable to other settings as well, especially ones for which the 'linearization' framework of [10] is applicable.

The formal statement of results, an outline of our techniques and a comparison with earlier works are presented in Section 3.

1.2 Related Work

MaxCut and correlation clustering are both extremely well-studied problems, and thus we will only mention the results most relevant to our work.

Dense graphs. A graph is said to be dense if its average degree is $\Omega(n)$. Starting with the work of Arora et al. [10], many NP hard optimization problems have been shown to admit a PTAS when the instances are dense. Indeed, a small random induced subgraph is known to be a core-set for problems such as MaxCut, and indeed all k-CSPs [20, 6, 16, 29]. The work of [10] relies on an elegant *linearization* procedure, while [16, 6] give a different (and more unified) approach based on "cut approximations" of a natural tensor associated with a CSP.

Polynomial density. The focus of our work is on graphs that are *in between* sparse (constant average degree) and dense graphs. These are graphs whose density (i.e., average degree) is n^{δ}, for some $0 < \delta < 1$. Fotakis et al. [15] extended the approach of [10] to this setting, and obtained $(1+\varepsilon)$ approximation algorithms with run-time $\exp(\widetilde{O}(n^{1-\delta}))$. They also showed that it was the best possible, under the exponential time hypothesis (ETH). By way of core-sets, in their celebrated work on the optimality of the Goemans-Williamson rounding, Feige and Schechtman [14] showed that a random sample of $\widetilde{O}(n^{1-\delta})$ is a core-set for MaxCut, if the graphs are *almost regular* and have an average degree n^{δ}. This was extended to other CSPs by [12]. These arguments seem to use near-regularity in a crucial way, and are based on restricting the number of possible 'candidates' for the maximum cut.

Streaming algorithms and lower bounds. In the streaming setting, there are several algorithms [2, 27, 18, 3, 19, 26] that produce cut or spectral sparsifiers with $O(\frac{n}{\epsilon^2})$ edges using $\widetilde{O}(\frac{n}{\epsilon^2})$ space. Such algorithms preserve every cut within $(1+\epsilon)$-factor (and therefore also preserve the max cut). Andoni et al. [9] showed that such a space complexity is essential; in fact, [9] show that any sketch for all the cuts in a graph must have bit complexity $\Omega(\frac{n}{\epsilon^2})$ (not necessarily streaming ones). However, this does not rule out the possibility of being able to find a *maximum* cut in much smaller space.

For MaxCut, Kapralov et al. [24] and independently Kogan et al. [28] proved that any streaming algorithm that can approximate the MaxCut value to a factor better than 2 requires $\tilde{\Omega}(\sqrt{n})$ space, even if the edges are presented in random order. For adversarial orders, they showed that for any $\epsilon > 0$, a one-pass $(1+\epsilon)$-approximation to the max cut value must use $n^{1-O(\epsilon)}$ space. Very recently, Kapralov et al. [25] went further, showing that there exists an $\epsilon^* > 0$ such that every randomized single-pass streaming algorithm that yields a $(1+\epsilon^*)$-approximation to the MAXCUT size must use $\Omega(n)$ space.

Correlation clustering. Correlation clustering was formulated by Bansal et al. [11] and has been studied extensively. There are two common variants of the problem – maximizing agreement and minimizing disagreement. While these are equivalent for exact optimization (their sum is a constant), they look very different under an approximation lens. Maximizing agreement typically admits constant factor approximations, but minimizing disagreement is much harder. In this paper, we focus on the maximum-agreement variant of correlation clustering and in particular we focus on $(1+\epsilon)$-approximations. Here, Ailon and Karnin [5] presented an approximation scheme with sublinear query complexity (which also yields a semi-streaming algorithm) for dense instances of correlation clustering. Giotis and Guruswami [17] described a sampling based algorithm combined with a greedy strategy which guarantees a solution within (ϵn^2) additive error. (Their work is similar to the technique of Mathieu and Schudy [29].) Most recently, Ahn et al. [4] gave a single-pass semi-streaming algorithm for max-agreement. For bounded weights, they provide an $(1+\epsilon)$-approximation streaming algorithm and for graphs with arbitrary weights, they present a $0.766(1-\epsilon)$-approximation algorithm. Both algorithms require $(n\epsilon^{-2})$ space. The key idea in their approach was to adapt multiplicative-weight-update methods for solving the natural SDPs for correlation clustering in a streaming setting using linear sketching techniques.

2 Definitions

▶ **Definition 1** (MaxCut). Let $G = (V, E, w)$ be a graph with weights $w\colon E \to \mathbb{R}^+$. Let (A, B) be a partition of V and let $w(A, B)$ denote the sum of weights of edges between A and B. Then $\textsc{MaxCut}(G) = \max_{(A,B) \text{ partition of } V} w(A, B)$.

For ease of exposition, we will assume that the input graph for MaxCut is unweighted. Our techniques apply as long as all the weights are $O(1)$. Also, we denote by Δ the average degree, i.e., $\sum_{i,j} w_{ij}/|V|$.

Moving now to correlation clustering, let $G = (V, E, c^+, c^-)$ be a graph with edge weights c^+_{ij} and c^-_{ij} where for every edge ij we have $c^+_{ij}, c^-_{ij} \geq 0$ and only one of them is nonzero. For every edge $ij \in E$, we define $\eta_{ij} = c^+_{ij} - c^-_{ij}$ and for each vertex, $d_i = \sum_{i \in \Gamma(j)} |\eta_{ij}|$. We will also assume that all the weights are bounded by an absolute constant in magnitude (for simplicity, we assume it is 1). We define the "average degree" Δ (used in the statements that follow) of a correlation clustering instance to be $(\sum_i d_i)/n$.

▶ **Definition 2** (MAX-AGREE correlation clustering). Given $G = (V, E, c^+, c^-)$ as above, consider a partition of V into *clusters* $C_1, C_2, \ldots, C_k$, and let χ_{ij} be an indicator that is 1 if i an j are in the same cluster and 0 otherwise. The MAX-AGREE score of this clustering is given by $\sum_{ij} c^+_{ij}\chi_{ij} + \sum_{ij} c^-_{ij}(1 - \chi_{ij})$. The goal is to find a partition *maximizing* this score. The maximum value of the score over all partitions of V will be denoted by $CC(G)$.

Note that the objective value can be simplified to $\sum_{ij} c^-_{ij} + \eta_{ij}\chi_{ij} = C^- + \sum_{ij} \eta_{ij}\chi_{ij}$, where C^- denotes the sum $\sum_{ij} c^-_{ij}$.

We will also frequently use concentration bounds; these are stated in full version of the paper.

3 Technical overview

We now present an outline of our main ideas. Suppose we have a graph $G = (V, E)$. First, we define a procedure VERTEX SAMPLE. This takes as input probabilities p_i for every vertex, and produces a random weighted induced subgraph.

Procedure vertex sample ($\{p_i\}_{i\in V}$). Sample a set S' of vertices by selecting each vertex v_i with probability p_i independently. Define H to be the induced subgraph of G on the vertex set S'. For $i, j \in S'$, define $w_{ij} = \frac{1}{p_i p_j \Delta^2}$.[2]

Intuitively, the edge weights are chosen so that the total number of edges remains the same, in expectation. Next, we define the notion of an importance score for vertices. Let d_i denote the degree of vertex i.

▶ **Definition 3.** The *importance score* h_i of a vertex i is defined as $h_i = \min\{1, \frac{\max\{d_i, \epsilon\Delta\}}{\Delta^2 \alpha_\epsilon}\}$, where α_ϵ is an appropriately chosen parameter (for MAXCUT, we set it to $\frac{\epsilon^4}{C \log n}$, and for correlation clustering, we set it to $\frac{\epsilon^8}{C \log n}$, where C is an absolute constant).

The main result is now the following:

▶ **Theorem 4** (Core-set). *Let $G = (V, E)$ have an average degree Δ. Suppose we apply* VERTEX SAMPLE *with probabilities $p_i \in [h_i, 2h_i]$ to obtain a weighted graph H. Then H has $\widetilde{O}(\frac{n}{\Delta})$ vertices and the quantities* MAXCUT(H) *and $CC(H)$ are within a $(1+\epsilon)$ factor of the corresponding quantities* MAXCUT(G) *and $CC(G)$, w.p. at least $1 - \frac{1}{n^2}$.*

While the number of vertices output by the vertex sample procedure is small, we would like a core-set of small "total size". This is ensured by the following.

Procedure edge sample (H). Given a weighted graph H with total edge weight W, sample each edge $e \in E(H)$ independently with probability $p_e := \min(1, \frac{8|S'|w_e}{\varepsilon^2 W})$, to obtain a graph H'. Now, assign a weight w_e/p_e to the edge e in H'.

The procedure samples roughly $|S'|/\varepsilon^2$ edges, with probability proportional to the edge weights. The graph is then re-weighted in order to preserve the total edge weight in expectation, yielding:

▶ **Theorem 5** (Sparse core-set). *Let G be a graph with n vertices and average degree $\Delta = n^\delta$. Let H' be the graph obtained by first applying* VERTEX SAMPLE *and then applying* EDGE SAMPLE. *Then H' is a ϵ-core-set for* MAXCUT *and CC, having size $\widetilde{O}(\frac{n}{\Delta}) = \widetilde{O}(n^{1-\delta})$.*

We then show how to implement the above procedures in a streaming setting. This gives:

▶ **Theorem 6** (Streaming algorithm). *Let G be a graph on n vertices and average degree $\Delta = n^\delta$, whose edges arrive in a streaming fashion in adversarial order. There is a two-pass streaming algorithm with space complexity $\widetilde{O}(\frac{n}{\Delta}) = \widetilde{O}(n^{1-\delta})$ for computing a $(1+\epsilon)$-approximation to* MAXCUT(G) *and $CC(G)$.*

[2] In correlation clustering, we have edge weights to start with, so the weight in H will be $w_{ij} \cdot c^+_{ij}$ (or c^-_{ij}).

Of these, Theorem 4 is technically the most challenging. Theorem 5 follows via standard edge sampling methods akin to those in [2] (which show that w.h.p., every cut size is preserved). It is presented in full version of the paper, for completeness. The streaming algorithm, and a proof of Theorem 6, are presented in Section 6. In the following section, we give an outline of the proof of Theorem 4.

3.1 Proof of the sampling result (theorem 4): an outline

In this outline we will restrict ourselves to the case of MaxCut as it illustrates our main ideas. Let G be a graph as in the statement of the theorem, and let H be the output of the procedure vertex sample.

Showing that MaxCut(H) is at least MaxCut(G) up to an $\varepsilon n\Delta$ additive term is easy. We simply look at the projection of the maximum cut in G to H (see, for instance, [14]). Thus, the challenge is to show that a sub-sample cannot have a significantly larger cut, w.h.p. The natural approach of showing that every cut in G is preserved does not work as 2^n cuts is too many for the purposes of a union bound.

There are two known ways to overcome this. The first approach is the one used in [20, 14] and [12]. These works essentially show that in a graph of average degree Δ, we need to consider only roughly $2^{n/\Delta}$ cuts for the union bound. If all the degrees are roughly Δ, then one can show that all these cuts are indeed preserved, w.h.p. There are two limitations of this argument. First, for non-regular graphs, the variance (roughly $\sum_i pd_i^2$, where p is the sampling probability) can be large, and we cannot take a union bound over $\exp(n/\Delta)$ cuts. Second, the argument is combinatorial, and it seems difficult to generalize this to analyze non-uniform sampling.

The second approach is via *cut decompositions*, developed in [16, 6]. Here, the adjacency matrix A is decomposed into poly$(1/\varepsilon)$ rank-1 matrices, plus a matrix that has a small *cut norm*. It turns out that solving many quadratic optimization problems (including MaxCut) on A is equivalent (up to an additive $\varepsilon n\Delta$) to solving them over the sum of rank-1 terms (call this A'). Now, the adjacency matrix of H is an induced square sub-matrix of A, and since we care only about A' (which has a simple structure), [6] could show that MaxCut(H) $\leq$ MaxCut(G) $+ \varepsilon n^2$, w.h.p. To the best of our knowledge, such a result is not known in the "polynomial density" regime (though the cut decomposition still exists).

Our technique. We consider a new approach. While inspired by ideas from the works above, it also allows us to reason about non-uniform sampling in the polynomial density regime. Our starting point is the result of Arora et al. [10], which gives a method to estimate the MaxCut using a collection of linear programs (which are, in turn, derived using a sample of size n/Δ). Now, by a double sampling trick (which is also used in the approaches above), it turns out that showing a sharp concentration bound for the value of an *induced sub-program* of an LP as above, implies Theorem 4. As it goes via a linear programming and not a combinatorial argument, analyzing non-uniform sampling turns out to be quite direct. Let us now elaborate on this high level plan.

Induced sub-programs. First, we point out that an analysis of induced sub-programs is also an essential idea in the work of [6]. The main difference is that in their setting, only the variables are sub-sampled (and the number of constraints remains the same). In our LPs, the constraints correspond to the vertices, and thus there are fewer constraints in the sampled LP. This makes it harder to control the value of the objective. At a technical level, while a duality-based argument using Chernoff bounds for linear forms suffices in the setting of [6], we need the more recent machinery on concentration of quadratic functions.

We start by discussing the estimation technique of [10].

Estimation with Linear Programs. The rough idea is to start with the natural quadratic program for MaxCut: $\max \sum_{(i,j)\in E} x_i(1-x_j)$, subject to $x_i \in \{0,1\}$.[3] This is then "linearized" using a seed set of vertices sampled from G. We refer to Section 4 for details. For now, Est(G) is a procedure that takes a graph G and a set of probabilities $\{\gamma_i\}_{i\in V(G)}$, samples a seed set using γ, and produces an estimate of MaxCut (G).

Now, suppose we have a graph G and a sample H. We can imagine running Est(G) and Est(H) to obtain good estimates of the respective MaxCut values. But now suppose that in both cases, we could use precisely the same seed set. Then, it turns out that the LPs used in Est(H) would be 'induced' sub-programs (in a sense we will detail in Section 5) of those used in Est(G), and thus proving Theorem 4 reduces to showing a sufficiently strong concentration inequality for sub-programs.

The key step above was the ability to use same seed set in the Est procedures. This can be formalized as follows.

Double sampling. Consider the following two strategies for sampling a *pair* of subsets (S, S') of a universe $[n]$ (here, $q_v \le p_v$ for all v):

- Strategy A: choose $S' \subseteq [n]$, by including every v w.p. p_v, independently; then for $v \in S'$, include them in S w.p. q_v/p_v, independently.
- Strategy B: pick $S \subseteq [n]$, by including every v w.p. q_v; then iterate over $[n]$ once again, placing $v \in S'$ with a probability equal to 1 if $v \in S$, and p_v^* if $v \notin S$.

▶ **Lemma 7.** *Suppose $p_v^* = p_v(1 - \frac{q_v}{p_v})/(1-p_v)$. Then the distribution on pairs (S, S') obtained by strategies A and B are identical.*

The proof is by a direct calculation and the details can be found in full version.

Proof of Theorem 4. To show the theorem, we use p_v as in the statement of the theorem, and set q to be the uniform distribution $q_v = \frac{16\log n}{\varepsilon^2\Delta}$. The proof now proceeds as follows. Let S' be a set sampled using the probabilities p_v. These form the vertex set of H. Now, the procedure Est on H (with sampling probabilities q_v/p_v) samples the set S (as in strategy A). By the guarantee of the estimation procedure (Corollary 5.1.2 in full version), we have MaxCut$(H) \approx$ Est(H), w.h.p. Next, consider the procedure Est on G with sampling probabilities q_v. Again, by the guarantee of the estimation procedure (Corollary 5.1.1), we have MaxCut$(G) \approx$ Est(G), w.h.p.

Now, we wish to show that Est$(G) \approx$ Est(H). By the equivalence of the sampling strategies, we can now take the strategy B view above. This allows us to assume that the Est procedures use the same S, and that we pick S' *after* picking S. This reduces our goal to one of analyzing the value of a random induced sub-program of an LP, as mentioned earlier. The details of this step are technically the most involved, and are presented in Section 5. This completes the proof of the theorem. (Note that the statement also includes a bound on the number of vertices of H. This follows immediately from the choice of p_v.) ◀

[3] This is a valid formulation, because for every $x_i \neq x_j$ that is an edge contributes 1 to the objective, and $x_i = x_j$ contribute 0.

4 Estimation via linear programming

We now present the estimation procedure Est used in our proof. It is an extension of [10] to the case of weighted graphs and non-uniform sampling probabilities.

Let $H = (V, E, w)$ be a weighted, undirected graph with edge weights w_{ij}, and let $\gamma : V \to [0, 1]$ denote sampling probabilities. The starting point is the quadratic program for MaxCut: max $\sum_{ij \in E} w_{ij} x_i (1 - x_j)$, subject to $x_i \in \{0, 1\}$. The objective can be re-written as $\sum_{i \in V} x_i (d_i - \sum_{j \in \Gamma(i)} w_{ij} x_j)$, where d_i is the weighted degree, $\sum_{j \in \Gamma(i)} w_{ij}$. The key idea now is to "guess" the value of $\rho_i := \sum_{j \in \Gamma(i)} w_{ij} x_j$, by using a seed set of vertices. Given a guess, the idea is to solve the following linear program, which we denote by $LP_\rho(V)$.

$$\begin{aligned} \text{maximize} \quad & \sum_i x_i (d_i - \rho_i) - s_i - t_i \\ \text{subject to} \quad & \rho_i - t_i \le \sum_{j \in \Gamma(i)} w_{ij} x_j \le \rho_i + s_i \\ 0 \le x_i \le 1, \quad & \quad s_i, t_i \ge 0. \end{aligned}$$

The variables are x_i, s_i, t_i. Note that if we fix the x_i, the optimal s_i, t_i will satisfy $s_i + t_i = |\rho_i - \sum_{j \in \Gamma(i)} w_{ij} x_j|$. Also, note that if we have a perfect guess for ρ_i's (coming from the MaxCut), the objective can be made $\ge$ MaxCut(H).

Estimation procedure. The procedure Est is the following: first sample a set $S \subseteq V$ where each $i \in V$ is included w.p. γ_i independently. For every partition $(A, S \setminus A)$ of S, set $\rho_i = \sum_{j \in \Gamma(i) \cap A} \frac{w_{ij}}{\gamma_j}$, and solve $LP_\rho(V)$ (in what follows, we denote this LP by $LP^\gamma_{A, S \setminus A}(V)$, as this makes the partition and the sampling probabilities clear). Return the maximum of the objective values.

Our result here is a sufficient condition for having Est$(H) \approx$ MaxCut(H).

▶ **Theorem 8.** *Let H be a weighted graph on n vertices, with edge weights w_{ij} that add up to W. Suppose the sampling probabilities γ_i satisfy the condition*

$$w_{ij} \le \frac{W\varepsilon^2}{8 \log n} \frac{\gamma_i \gamma_j}{\sum_u \gamma_u} \quad \text{for all } i, j. \tag{1}$$

Then, we have Est$(H, \gamma) \in$ MaxCut$(H) \pm \varepsilon W$*, with probability at least $1 - 1/n^2$ (where the probability is over the random choice of S).*

The proof consists of claims showing the upper and lower bound separately. The technical details are presented in full version. Finally, to show Theorem 4 (as outlined in Section 3.1), we need to apply Theorem 8 with specific values for γ and w_{ij}. The related corollaries are stated in full version of the paper.

5 Random induced linear programs

We will now show that the Est on H has approximately the same value as the estimate on G (with appropriate γ values). First, note that Est(G) is $\max_{A \subseteq S} LP^\gamma_{A, S \setminus A}(G)$, where $\gamma_i = q_i$. To write the LP, we need the constants ρ_i, defined by the partition $(A, S \setminus A)$ as $\rho_i := \sum_{j \in \Gamma(i) \cap A} \frac{1}{q_j}$. For the graph H, the estimation procedure uses an identical program, but the sampling probabilities are now $\alpha_i := q_i / p_i$, and the estimates ρ, which we now denote by $\widetilde{\rho}_i$, are defined by $\widetilde{\rho}_i := \sum_{j \in \Gamma(i) \cap A} \frac{p_j w_{ij}}{q_j}$. Also, by the way we defined w_{ij}, $\widetilde{\rho}_i = \frac{\rho_i}{p_i \Delta^2}$.

$$\begin{aligned} \max \quad & \sum_{i \in G} [x_i(d_i - \rho_i) - (s_i + t_i)] \\ \text{s.t.} \quad & \sum_{j \in \Gamma(i)} x_j \le \rho_i + s_i, \quad \forall i \in [n] \\ & - \sum_{j \in \Gamma(i)} x_j \le -\rho_i + t_i, \quad \forall i \in [n] \\ & 0 \le x_i \le 1 \quad \forall i \in [n] \end{aligned}$$

(a) The LP on the full graph

$$\begin{aligned} \max \quad & \sum_{i \in S'} [x_i(\widetilde{d}_i - \widetilde{\rho}_i) - (\tilde{s}_i + \tilde{t}_i)] \\ \text{s.t.} \quad & \sum_{j \in \Gamma(i) \cap S'} w_{ij} x_j \le \widetilde{\rho}_i + \tilde{s}_i, \quad \forall i \in S' \\ & - \sum_{j \in \Gamma(i) \cap S'} w_{ij} x_j \le -\widetilde{\rho}_i + \tilde{t}_i, \quad \forall i \in S' \\ & 0 \le x_i \le 1, \quad \tilde{s}_i, \tilde{t}_i \ge 0 \quad \forall i \in S' \end{aligned}$$

(b) The sampled LP

Figure 1 The two LPs.

$$\begin{aligned} & \text{minimize} \quad \sum_{i \in G} u_i + \rho_i z_i \quad \text{s.t.} \\ & u_i + \sum_{j \in \Gamma(i)} z_j \ge d_i - \rho_i \quad \forall i \in V \\ & u_i \ge 0, \quad -1 \le z_i \le 1 \quad \forall i \in V \end{aligned}$$

(a) The dual of $LP^{\gamma}_{A,S\setminus A}(G)$

$$\begin{aligned} & \text{minimize} \quad \sum_{i \in S'} [\tilde{u}_i + \widetilde{\rho}_i \tilde{z}_i] \quad \text{s.t.} \\ & \tilde{u}_i + \sum_{j \in \Gamma(i) \cap S'} w_{ij} \tilde{z}_j \ge \widetilde{d}_i - \widetilde{\rho}_i \quad \forall i \in S' \\ & \tilde{u}_i \ge 0, \quad -1 \le \tilde{z}_i \le 1 \quad \forall i \in S'. \end{aligned}$$

(b) The dual of the induced program $LP^{\alpha}_{A,S\setminus A}(H)$.

Figure 2 The dual LPs.

The degrees are now $\widetilde{d}_i := \sum_{j \in \Gamma(i) \cap S'} w_{ij} = \sum_{j \in \Gamma(i) \cap S'} \frac{1}{p_i p_j \Delta^2}$. The two LPs are shown in Figure 1. Our aim in this section is to show the following:

▶ **Theorem 9.** *Let G be an input graph, and let (S, S') be sampled as described in Section 3.1. Then, with probability $\ge 1 - \frac{1}{n^2}$, we have*

$$\max_{A \subseteq S} LP^{\gamma}_{A,S\setminus A}(G) \ge \Delta^2 \cdot \max_{A \subseteq S} LP^{\alpha}_{A,S\setminus A}(H) - \varepsilon n \Delta.$$

Proof outline. To prove the theorem, the idea is to take the "strategy B" viewpoint of sampling (S, S'), i.e., fix S, and sample S' using the probabilities p^*. Then, we only need to understand the behavior of an "induced sub-program" sampled with the probabilities p^*. This is done by considering the duals of the LPs, and constructing a feasible solution to the induced dual whose cost is not much larger than the dual of the full program, w.h.p. This implies the result, by linear programming duality.

Let us thus start by understanding the dual of $LP^{\gamma}_{A,S\setminus A}(G)$ given A, shown in Figure 2a. We note that for any given z, the optimal choice of u_i is $\max\{0, d_i - \rho_i - \sum_{j \in \Gamma(i)} z_j\}$; thus we can think of the dual solution as being the vector z. The optimal u_i may thus be bounded by $2d_i$, a fact that we will use later. Next, we write down the dual of the induced program, $LP^{\alpha}_{A,S\setminus A}(H)$, as shown in Figure 2b. Following the outline above, we will construct a feasible solution to LP (2b), whose cost is close to the optimal dual solution to LP (2a). The construction we consider is very simple: if z is the optimal dual solution to (2a), we set $\tilde{z}_i = z_i$ for $i \in S'$ as the candidate solution to (2b). This is clearly feasible, and thus we only need to compare the solution costs. The dual objective values are as follows

$$\mathsf{Dual}_G = \sum_{i\in V} \rho_i z_i + \max\{0,\ d_i - \rho_i - \sum_{j\in\Gamma(i)} z_j\} \tag{2}$$

$$\mathsf{Dual}_H \le \sum_{i\in S'} \widetilde{\rho}_i z_i + \max\{0,\ \widetilde{d}_i - \widetilde{\rho}_i - \sum_{j\in\Gamma(i)\cap S'} w_{ij} z_j\} \tag{3}$$

Note that there is a $\le$ in (3), as $\tilde{z}_i = z_i$ is simply one feasible solution to the dual (which is a minimization program). Next, our goal is to prove that w.p. at least $1 - \frac{1}{n^2}$,

$$\max_{A\subseteq S} \mathsf{Dual}_H \le \frac{1}{\Delta^2} \cdot \max_{A\subseteq S} \mathsf{Dual}_G + \frac{\varepsilon n}{\Delta}.$$

Note that here, the probability is over the choice of S' *given* S (as we are taking view-B of the sampling). The first step in proving the above is to move to a slight variant of the quantity Dual_H, which is motivated by the fact that $\mathbf{Pr}[Y_i = 1]$ is not quite p_i, but p_i^* (as we have conditioned on S). Let us define $\widetilde{\rho}_i^* := \frac{\rho_i}{p_i^* \Delta^2}$ (recall that $\widetilde{\rho}_i$ is $\frac{\rho_i}{p_i \Delta^2}$), and $w_{ij}^* := \frac{1}{p_i^* p_j^* \Delta^2}$. So also, let $d_i^* := \sum_{j\in\Gamma(i)} Y_j w_{ij}^*$. Then, define

$$\mathsf{Dual}_H^* := \sum_{i\in S'} \widetilde{\rho}_i^* z_i + \max\{0,\ \widetilde{d}_i^* - \widetilde{\rho}_i^* - \sum_{j\in\Gamma(i)\cap S'} w_{ij}^* z_j\}. \tag{4}$$

A straightforward lemma is as follows. The proof is presented in full version of the paper.

▶ **Lemma 10.** *Let (S, S') be sampled as in Section 3.1. Then w.p. at least $1 - \frac{1}{n^4}$, we have that for all $z \in [-1, 1]^n$ and for all partitions $(A, S \setminus A)$ of S,*[4] *$|\mathsf{Dual}_H - \mathsf{Dual}_H^*| \le \frac{\varepsilon n}{2\Delta}$.*

Thus our goal is to show the following (where condition (b) on S is a technical one needed to bound ρ_i).

▶ **Lemma 11.** *Let S satisfy the conditions (a) $|S| \le \frac{20 n \log n}{\varepsilon^2 \Delta}$, and (b) for all $i \in V$, $\sum_{j\in\Gamma(i)\cap S} \frac{1}{q_j} \le 2(d_i + \varepsilon\Delta)$. Then, w.p. $\ge 1 - \frac{1}{n^4}$ over the choice of S' given S, we have*

$$\max_{A\subseteq S} \mathsf{Dual}_H^* \le \frac{1}{\Delta^2} \cdot \max_{A\subseteq S} \mathsf{Dual}_G + \frac{\varepsilon n}{2\Delta}.$$

Proof of Theorem 9. The conditions we assumed on S in Lemma 11 hold w.p. at least $1 - \frac{1}{n^4}$ (via a simple application of Bernstein's inequality). Thus the conclusion of the lemma holds w.p. at least $1 - \frac{2}{n^4}$. Combining this with Lemma 10, we have that $\max_A \mathsf{Dual}_H \le \frac{1}{\Delta^2} \max_A \mathsf{Dual}_G + \frac{\varepsilon n}{\Delta}$ w.p. at least $1 - \frac{3}{n^4}$. The theorem then follows via LP duality. ◀

It thus suffices to prove Lemma 11 via a concentration bound on a quadratic function that is not quite a quadratic *form*. Details can be found in full version of the paper.

6 A 2-pass streaming algorithm

We now show how our main core set result can be used to design a streaming algorithm for MaxCut. The algorithm works in two passes: the first pass builds a core-set S of size $\widetilde{O}(n/\Delta)$ as prescribed by Theorem 4 and the second pass builds the induced weighted graph $G[S]$ and computes its max cut. This algorithm works under edge insertion/deletion.

[4] Note that the partition defines the ρ_i.

6.1 Pass 1: Building a core set

To construct S, Theorem 4 states that each vertex must be sampled with probability p_i, where $p_i \geq h_i$ and $h_i = \min(1, \frac{\max(d_i, \epsilon\Delta)}{\Delta^2 \alpha_\epsilon})$ is the importance score of a vertex. As the goal is to only choose a small number of vertices, we will also make sure that $p_i \leq 2h_i$. The challenge here is two-fold: we need to sample (roughly) proportional to the degree d_i, which can only be computed *after* the stream has passed, and we also need the actual value of p_i (or a close enough estimate of it) in order to correctly reweight edges in the second pass.

The degree d_i of a vertex i is the "count" of the number of times i appears in the edge stream. To sample with probability proportional to d_i we will therefore make use of streaming algorithms for ℓ_1-sampling [30, 8, 21]. We borrow some notation from [8].

▶ **Definition 12.** Let $\rho > 0, f \in [1, 2]$. A (ρ, f)-approximator to $\tau > 0$ is a quantity $\hat{\tau}$ such that $\tau/f - \rho \leq \hat{\tau} \leq f\tau + \rho$

▶ **Lemma 13** ([21] (rephrased from [8])). *Given a vector $x \in \mathbb{R}^n$ and parameters $\epsilon, \delta > 0, c > 0$ there exists an algorithm A that uses space $O(\log(1/\epsilon)\epsilon^{-1} \log^2 n \log(1/\delta))$ and generates a pair (i, v) from a distribution D_x on $[1 \dots n]$ such that with probability $1 - \delta$*

- *$D_x(i)$ is a $(\frac{1}{n^c}, 1 + \epsilon)$-approximator to $|x_i|/\|x\|_1$*
- *v is a $(0, 1 + \epsilon)$-approximator to x_i*

where c is a fixed constant.

We will also need to maintain heavy hitters: all vertices of degree at least Δ^2 (up to constants). To do this, we will make use of the standard CountMin sketch [13]. For completeness, we state its properties here.

▶ **Lemma 14** ([13]). *Fix parameters $k, \delta > 0$. Then given a stream of m updates to a vector $x \in \mathbb{R}^n$ there is a sketch CM of size $O(k \log \delta^{-1}(\log m + \log n))$ and a reconstruction procedure $f : [n] \to \mathbb{R}$ such that with probability $1 - \delta$, for any x_i, $|x_i - f(i)| \leq \|x\|_1/k$*

Outline. We will have a collection of r, roughly n/Δ ℓ_1-samplers. These samplers will together give a good estimate ($(1+\varepsilon)$-approximation) of the importance h_i for all the vertices that have a *small* degree (which we define to be $< \alpha_\varepsilon \Delta^2$). Then, we use the CM sketch to maintain the degrees of all the 'high degree' vertices, i.e., those with degrees $\geq \alpha_\varepsilon \Delta^2$. Taken together, we obtain the desired sampling in the first pass.

▶ **Definition 15.** Given two sets of pairs of numbers S, S', let

$$S \cup_{\max} S' = \{(x, \max_{(x',y) \in S \cup S', x'=x} y)\}.$$

▶ **Lemma 16.** *Let $S = \{(i, v_i)\}$ be the set returned by Algorithm 1. Then*

- *S has size $\widetilde{O}\left(\frac{n}{\Delta}\right)$.*
- *Each $i \in [n]$ is sampled with probability p_i that is $(0, 1 + \epsilon)$ approximated by v_i and that $(n^{-c}, 1 + \epsilon)$-approximates h_i.*

Proof sketch. Consider any vertex i with $d_i \geq \Delta^2 \alpha_\epsilon$. By Lemma 14, such a vertex will report a count of at least $f(i) = (1 - \zeta)\Delta^2 \alpha_\epsilon$ and thus is guaranteed to be included in S_h. Its reported score $v_i = 1$ satisfies the requirement of the Lemma. Secondly, consider any vertex with degree $d_i < \epsilon\Delta$. For such a vertex, $h_i = \epsilon/\Delta\alpha_\epsilon$ and thus it is included in S_l with the desired probability and v_i.

Finally, consider a vertex i with $\epsilon\Delta \leq d_i < \Delta^2 \alpha_\epsilon$. The probability that none of the ℓ_1-samplers yield i is $(1 - d_i/n\Delta)^r$, and since $d_i/n\Delta \ll 1$, this can be approximated as $(1 - rd_i/n\Delta)$. Thus, the probability of seeing i is $rd_i/n\Delta = d_i/\Delta^2\alpha_\epsilon$ as desired. ◀

Algorithm 1 Given n and average degree Δ

```
Initialize $S_l, S_m, S_h \leftarrow \emptyset$, and $\zeta = \alpha_\varepsilon$.
Sample elements from $[1 \dots n]$ each with probability $\epsilon/\Delta\alpha_\epsilon$. For each sampled $i$ add $(i, \epsilon/\Delta\alpha_\epsilon)$ to $S_l$.
Fix $\zeta > 0$. Initialize a CountMin sketch CM with size parameter $k = n/\Delta\zeta^2$. Let $f$ be the associated reconstruction procedure.
Initialize $r = O(n/\Delta\alpha_\epsilon)$ copies $A_1 \dots A_r$ of the algorithm $A$ from Lemma 13.
for each stream update $(i, w)$ (a vertex to which current edge is incident, and weight) do
    Update each $A_j, 1 \le j \le r$. Update CM.
for $j = 1$ to $r$ do
    Sample $(i, v)$ from $A_j$. $S_m = S_m \cup_{\max} \{(i, v)\}$
$S_h = \{(i, 1) \mid f(i) \ge (1 - \zeta)\Delta^2\alpha_\epsilon\}$
return $S_l \cup_{\max} S_m \cup_{\max} S_h$
```

▶ **Corollary 17.** *For each $(i, v_i) \in S, h_i \le v_i \le 2h_i$.*

6.2 Pass 2: Building the induced weighted graph

The first pass produces a set S of $\widetilde{O}(n/\Delta)$ vertices together with estimates v_i for their importance score h_i. If we weight each edge ij in $G[S]$ by $w_{ij} = 1/v_i v_j \Delta^2$, Theorem 4, along with Corollary 17 guarantee that a MaxCut in the resulting weighted graph is a good approximation of the true max cut. Thus, knowing S, constructing the re-weighted $G[S]$ in the second pass is trivial if we had space proportional to the number of edges in $G[S]$. Unfortunately this can be quadratic in $|S|$, so our goal is to implement the edge sampling of Theorem 5 in the second pass. This is done as follows: we maintain a set of edges E'. Every time we encounter an edge ij with $i, j \in S$, we check to see if it is already in E'. If not, we toss a coin and with probability $p_{ij} = \min(1, w_{ij} \log n/\epsilon^2)$ we insert $(i, j, w_{ij}/p_{ij})$ into E'. By Lemma 7.1 in full version of the paper, the size of E' is $\widetilde{O}(n/\Delta)$, and the resulting graph yields a $(1 + \epsilon)$ approximation to the MaxCut.

7 Correlation Clustering

Our argument for correlation clustering parallels the one for MaxCut. The MAX-AGREE variant of correlation clustering, while not a CSP (as the number of clusters is arbitrary), almost behaves as one. We start with two simple observations. The first is that we can restrict the number of clusters to $1/\varepsilon$, for the purposes of a $(1 + \varepsilon)$ approximation (See full version of the paper). Next, observe that the optimum objective value is at least $\max\{C^+, C^-\} \ge n\Delta/2$. This is simply because placing all the vertices in a single cluster gives a value C^+, while placing them all in different clusters gives C^-. Thus, it suffices to focus on additive approximation of $\varepsilon n\Delta$. Once we fix the number of clusters k, we can write correlation clustering as a quadratic program in a natural way: for each vertex i, have k variables $x_{i\ell}$, which is supposed to indicate if i is given the label ℓ. We then have the constraint that $\sum_\ell x_{i\ell} = 1$ for all i. The objective function then has a clean form:

$$\sum_{ij}[\sum_{\ell=1}^{k} x_{i\ell}(1 - x_{j\ell})c^-_{ij} + x_{i\ell}x_{j\ell}c^+_{ij}] = \sum_{ij}\sum_{\ell} x_{i\ell}c^-_{ij} + x_{i\ell}x_{j\ell}\eta_{ij} = \sum_{i,\ell} x_{i\ell}(\rho_{i\ell} + d^-_i),$$

where $x_{i\ell} = 1$ iff vertex $i \in C_\ell$, and $\rho_{i\ell} = \sum_{j \in \Gamma(i)} x_{j\ell}\eta_{ij}$ and $d^-_i = \sum_j c^-_{ij}$.

Note the similarity with the program for MaxCut. We will show that the framework from Section 3.1 carries over with minor changes. The details of the new Est procedure can be found in full version of the paper (it requires one key change: we now need to consider k-partitions of the seed set in order to find ρ). The duality based proof is slightly more involved; however we can use the same rough outline. The proof is presented in full version of the paper.

References

1 Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

2 Kook Jin Ahn and Sudipto Guha. Graph sparsification in the semi-streaming model. In *International Colloquium on Automata, Languages, and Programming*, pages 328–338. Springer, 2009.

3 Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 5–14. ACM, 2012.

4 KookJin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *International Conference on Machine Learning*, pages 2237–2246, 2015.

5 Nir Ailon and Zohar Karnin. A note on: No need to choose: How to get both a ptas and sublinear query complexity. *arXiv preprint arXiv:1204.6588*, 2012.

6 Noga Alon, W Fernandez De La Vega, Ravi Kannan, and Marek Karpinski. Random sampling and approximation of max-csps. *Journal of computer and system sciences*, 67(2):212–243, 2003.

7 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39(1):143–167, 2009.

8 Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms via precision sampling. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 363–372. IEEE, 2011.

9 Alexandr Andoni, Robert Krauthgamer, and David P Woodruff. The sketching complexity of graph cuts. *arXiv preprint arXiv:1403.7058*, 2014.

10 Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 284–293. ACM, 1995.

11 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

12 Boaz Barak, Moritz Hardt, Thomas Holenstein, and David Steurer. Subsampling mathematical relaxations and average-case complexity. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 512–531, Philadelphia, PA, USA, 2011. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=2133036.2133077`.

13 Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

14 Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for max cut. *Random Structures & Algorithms*, 20(3):403–440, 2002.

15 Dimitris Fotakis, Michael Lampis, and Vangelis Th Paschos. Sub-exponential approximation schemes for csps: from dense to almost sparse. *arXiv preprint arXiv:1507.04391*, 2015.

16 Alan Frieze and Ravi Kannan. The regularity lemma and approximation schemes for dense problems. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 12–20. IEEE, 1996.

17 Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1167–1176. Society for Industrial and Applied Mathematics, 2006.

18 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. Graph sparsification via refinement sampling. *arXiv preprint arXiv:1004.4915*, 2010.

19 Ashish Goel, Michael Kapralov, and Ian Post. Single pass sparsification in the streaming model with edge deletions. *arXiv preprint arXiv:1203.4900*, 2012.

20 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

21 Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 49–58. ACM, 2011.

22 Michael Kapralov. Better bounds for matchings in the streaming model. In *SODA*, pages 1679–1697. SIAM, 2013.

23 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *SODA*, pages 734–751. SIAM, 2014.

24 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating max-cut. In *SODA*, pages 1263–1282. SIAM, 2015.

25 Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. (1+ ω (1))-approximation to max-cut requires linear space. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1703–1722. SIAM, 2017.

26 Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 561–570. IEEE, 2014.

27 Jonathan A Kelner and Alex Levin. Spectral sparsification in the semi-streaming setting. *Theory of Computing Systems*, 53(2):243–262, 2013.

28 Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proc. ITCS*, pages 367–376. ACM, 2015.

29 Claire Mathieu and Warren Schudy. Yet another algorithm for dense max cut: go greedy. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 176–182. Society for Industrial and Applied Mathematics, 2008.

30 Morteza Monemizadeh and David P Woodruff. 1-pass relative-error lp-sampling with applications. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1143–1160. SIAM, 2010.

31 Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4), 2007. doi:10.1145/1255443.1255449.