An Efficient Sparse Regularity Concept

Amin Coja-Oghlan*
School of Informatics,
University of Edinburgh,
Edinburgh, UK,
email:acoghlan@inf.ed.ac.uk,

Colin Cooper
Department of Computer Science,
King's College,
University of London,
London, UK,
email:ccooper@dcs.kcl.ac.uk, †

Alan Frieze
Department of Mathematical Sciences,
Carnegie Mellon University,
Pittsburgh, USA,
email:alan@random.math.cmu.edu, ‡

June 19, 2008

Abstract

Let $\mathbf{A}$ be a $0/1$ matrix of size $m \times n$, and let p be the density of $\mathbf{A}$ (i.e., the number of ones divided by $m \cdot n$). We show that $\mathbf{A}$ can be approximated in the cut norm within $\varepsilon \cdot mnp$ by a sum of cut matrices (of rank 1), where the number of summands is independent of the size $m \cdot n$ of $\mathbf{A}$, provided that $\mathbf{A}$ satisfies a certain boundedness condition. This decomposition can be computed in polynomial time. This result extends the work of Frieze and Kannan [16] to *sparse* matrices. As an application, we obtain efficient $1 - \varepsilon$ approximation algorithms for "bounded" instances of MAX CSP problems.
Key words: approximation algorithms, regularity lemma, matrix decomposition, cut norm, random discrete structures.

1 Introduction and Results

For many fundamental optimization problems there are *NP-hardness of approximation* results known, showing that not only is it NP-hard to compute the optimum exactly, but even to approximate the optimum within a factor bounded away from 1. For instance, in the MAX k-SAT problem it is NP-hard to achieve an approximation ratio better than $1 - 2^{-k}$ [20]. Furthermore, it is NP-hard to approximate MAX CUT within better than $16/17 \approx 0.94118$ [20, 25] (which can be tightened to ≈ 0.87856 under a stronger hypothesis [21]).

Frieze and Kannan [16] showed that the situation is much better for *dense* problem instances. For example, if $G = (V, E)$ is a graph on n vertices of density $p = 2n^{-2}|E|$, then its MAX CUT can be approximated within a factor of $1 - \varepsilon$ in time $\text{poly}(\exp((\varepsilon p)^{-2}) \cdot n)$. Hence, if $p > \delta$ for some fixed number $\delta > 0$, then this algorithm has a polynomial running time. Similarly, if F is a k-SAT formula with at least $\delta 2^k \binom{n}{k}$ clauses (i.e., at least a constant fraction of all possible clauses is present), then the maximum number of simultaneously satisfiable clauses can be approximated within $1 - \varepsilon$ in polynomial time for any fixed $\varepsilon > 0$.

The key ingredient in [16] is an algorithm for approximating a dense matrix $\mathbf{A}$ by a sum of a bounded number of "cut matrices". Applied to the adjacency matrix of a graph, this yields the aforementioned algorithm for MAX CUT.

*Supported by DFG CO 646. Research done while visiting Carnegie Mellon University.
†Supported by Royal Society Grant 2006/R2-IJP.
‡Supported in part by NSF grant CCF0502793.

Moreover, an extension of this matrix algorithm to k-dimensional tensors yields the approximation algorithms for dense instances of MAX CSP problems. To explain the matrix decomposition, let us consider a 0/1-matrix $\mathbf{A}$ of size $m \times n$, and let $0 \le p \le 1$ be the *density* of $\mathbf{A}$, i.e., the number of ones in $\mathbf{A}$ divided by $m \cdot n$. A *cut matrix* is a matrix $\mathbf{D}$ such that there are sets $S \subset [m]$, $T \subset [n]$ and a number d such that the entry $\mathbf{D}_{ij}$ is equal to d if $(i, j) \in S \times T$ and 0 otherwise. We denote such a matrix by $\mathbf{D} = \mathrm{CUT}(d, S, T)$, and observe that cut matrices have rank one. The *cut norm* of a $m \times n$ matrix $\mathbf{M} = (\mathbf{M}_{ij})_{i\in[m],j\in[n]}$ is

$$\|\mathbf{M}\|_\square = \max_{S\subset[m],T\subset[n]} |\mathbf{M}(S,T)|, \quad \text{where } \mathbf{M}(S,T) = \sum_{(s,t)\in S\times T} \mathbf{M}_{st}.$$

Frieze and Kannan proved that for any $\mathbf{A}$ and any $\varepsilon > 0$ there exist cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_s$ such that

$$\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_s)\|_\square < \varepsilon \cdot mn,$$

where $s \le c\varepsilon^{-2}$ for a constant $c > 0$. Indeed, such a decomposition can be computed in time $\varepsilon^{-2} \cdot \mathrm{poly}(mn)$ (or even in "constant" expected time $O(\varepsilon^{-2} \cdot \mathrm{polylog}(1/\varepsilon))$ by sampling). Hence, if $p \ge \delta$ for some fixed $\delta > 0$, i.e., if $\mathbf{A}$ is a *dense* matrix, then setting $\varepsilon' = \varepsilon p$ we can use this algorithm to find a decomposition of $\mathbf{A}$ within $\varepsilon \|A\|_\square = \varepsilon \cdot mnp$ efficiently by a sum of at most $c\varepsilon'^{-2} = c(\varepsilon p)^{-2} \le c(\varepsilon\delta)^{-2}$ cut matrices. The crucial point here is that the number of cut matrices is bounded *independently* of the size $m \cdot n$ of $\mathbf{A}$.

The goal of the present paper is to extend this result to *sparse* matrices, where the density p of $\mathbf{A}$ is no longer bounded below by a fixed number. Thus, in asymptotic terms, we are interested in $p = o(1)$ as $m, n \to \infty$. Clearly, in this case the bound $c(\varepsilon p)^{-2}$ on the number of cut matrices in the decomosition guaranteed by [16] is no longer "constant", but grows with the size $m \cdot n$ of $\mathbf{A}$. Of course, we cannot expect to obtain the same results in the sparse as in the dense case for *arbitrary* sparse matrices; for in the light of the aforementioned hardness results this would imply P=NP. Hence, our main result is that even in the sparse case a $0/1$ matrix $\mathbf{A}$ (or, more generally, a k-dimensional tensor) can be approximated in the cut norm by a sum of cut matrices with a number of summands independent of m, n, and p, *provided that* $\mathbf{A}$ satisfies a certain boundedness condition. This condition basically requires that $\mathbf{A}$ does not feature relatively large, extraordinarily dense spots. In addition, we shall use these decomposition results to obtain $(1 - \varepsilon)$-approximation algorithms for instances of MAX CSP problems that have a suitable boundedness property. As we shall see, in a sense these results mediate between the "average" and the worst case analysis of algorithms.

1.0.1 Outline.

In this section we state our results and discuss related work. Section 2 contains a few preliminaries, and in Section 3 we present the algorithms and their analyses for decomposing matrices and graphs. Further, in Section 4 we deal with k-dimensional tensors. Then, in Section 5 we apply the tensor algorithm to approximate MAX CSP problems. Finally, Section 6 contains a few examples, which link our results to the "average case" analysis of algorithms.

1.1 Approximating $0/1$ matrices

Let $\mathbf{A}$ be a $0/1$ matrix of size $m \times n$ and density p. Given $C, \gamma > 0$, we say that $\mathbf{A}$ is (C, γ)-*bounded* if for any two sets $S \subset [m]$ and $T \subset [n]$ of sizes $|S| \ge \gamma m$, $|T| \ge \gamma n$ we have

$$\mathbf{A}(S,T) = \sum_{(s,t)\in S\times T} \mathbf{A}_{st} \le C \cdot |S| \cdot |T| \cdot p. \tag{1}$$

In words, for any two sufficiently large sets S, T the number $\mathbf{A}(S, T)$ of ones in the square $S \times T$ must not exceed the number $|S| \cdot |T| \cdot p$ that we would expect if S, T were *random* sets by more than a factor of C.

Theorem 1 *There is an algorithm* `ApxMatrix`*, absolute constants $\zeta, \zeta' > 0$, and a polynomial Π such that the following holds. Suppose that $0 < \varepsilon < \frac{1}{2}, C > 1$. Let*

$$\kappa = \frac{\zeta C^2}{\varepsilon^2} \quad \textit{and} \quad \gamma = \gamma(\epsilon, C) = \frac{\zeta'\epsilon}{2^{10\kappa} C}. \tag{2}$$

If $\mathbf{A}$ is a (C,γ)-bounded 0/1 matrix, then in time $\kappa \cdot \Pi(m \cdot n)$, `ApxMatrix`$(\mathbf{A}, C, \varepsilon)$ outputs cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_s$ such that $s \leq \kappa$ and $\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_s)\|_\square \leq \epsilon \|\mathbf{A}\|_\square$.

We emphasize that the upper bound κ on the number of cut matrices depends *only* on C and ε, but not on the size of $\mathbf{A}$ or the density p. Moreover, while for the sake of similicity we assume that `ApxMatrix` is given the boundedness parameter C as an input, this can easily be avoided by performing a binary search (details omitted).

Given the $0/1$ matrix $\mathbf{A}$ and partitions $\mathcal{S}$ of $[m]$ and $\mathcal{T}$ of $[n]$, we define a matrix $\mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ as follows. If $s \in S \in \mathcal{S}$ and $t \in T \in \mathcal{T}$, then the corresponding entry $(\mathbf{A}_{\mathcal{S}\times\mathcal{T}})_{s,t}$ equals $|S|^{-1}|T|^{-1}\mathbf{A}(S,T)$. Hence, on each square $S \times T$ the matrix $\mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ is constant, and the value it takes is just the average of $\mathbf{A}$ over that square.

Corollary 1 *There is an algorithm `PartMatrix` and a polynomial Π that satisfy the following. Suppose that $\varepsilon, C > 0$, let κ, γ be as in (2), and assume that $\mathbf{A}$ is a (C,γ)-bounded $0/1$ matrix of size $m \times n$. Then in time $2^\kappa \cdot \Pi(m \cdot n)$ `PartMatrix`$(\mathbf{A}, C, \varepsilon)$ computes partitions $\mathcal{S}$ of $[m]$ and $\mathcal{T}$ of $[n]$ such that $\|\mathbf{A} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq 2\varepsilon \|\mathbf{A}\|_\square$. The number of classes in each partition $\mathcal{S}$, $\mathcal{T}$ is at most 2^κ.*

1.2 Weak regular partitions of graphs

Let $G = (V, E)$ be a graph on n vertices, and let $0 \leq p \leq 1$ be such that $|E| = n^2 p/2$; we refer to p as the *density* of G. Moreover, we assume that $V = [n]$. In addition, let $\mathbf{A} = \mathbf{A}(G)$ be the adjacency matrix of G. Then we say that G is (C,γ)*-bounded* if $\mathbf{A}$ has this property. Thus, if G is (C,γ)-bounded, then for any two sets $S, T \subset V$ of size at least γn we have $e_G(S,T) \leq C\gamma|S||T|p$, where $e_G(S,T)$ is the number of S-T-edges in G.

We call a partition $\mathcal{V}$ of V a *weak ε-regular partition* of G if $\|\mathbf{A} - \mathbf{A}_{\mathcal{V}\times\mathcal{V}}\|_\square \leq \varepsilon \|\mathbf{A}\|_\square = 2\varepsilon|E|$. Hence, if, for instance, $S, T \subset V$ are disjoint sets of vertices, then the number $\mathbf{A}(S,T)$ of S-T-edges is within $2\varepsilon|E|$ of $\mathbf{A}_{\mathcal{V}\times\mathcal{V}}(S,T)$. As we shall see below, this definition is related to the notion of regular partitions introduced by Szemerédi.

Corollary 2 *There is an algorithm `WeakPartition` and a polynomial Π that satisfy the following. Suppose that $C > 1$, $0 < \varepsilon < \frac{1}{2}$, and let $\kappa, \gamma > 0$ be as in (2), and let $G = (V, E)$ be a (C,γ)-bounded graph on n vertices. Then `WeakPartition`(G, C, ε) computes a weak 4ε-regular partition of G in time $2^{2\kappa} \cdot \Pi(n)$. This partition has at most $2^{2\kappa}$ classes.*

1.3 Approximating k-dimensional $0/1$ tensors

A *k-dimensional tensor* is a map $\mathbf{M}: R_1 \times R_2 \times \cdots \times R_k \to \mathbf{R}$, where $R_1, \ldots, R_k$ are finite index sets. Moreover, extending the matrix case in the obvious way to k dimensions, we say that a tensor $\mathbf{C}: R_1 \times R_2 \times \cdots \times R_k \to \mathbf{R}$ is a *cut tensor* if there exist sets $S_i \subseteq R_i$ for $i = 1, 2, \ldots, k$ and a real number d such that

$$\mathbf{C}(i_1, i_2, \ldots, i_k) = \begin{cases} d & \text{if } (i_1, i_2, \ldots, i_k) \in S_1 \times S_2 \times \cdots \times S_k \\ 0 & \text{otherwise.} \end{cases}$$

In this case we write $\mathbf{C} = \mathrm{CUT}(d, S_1, \ldots, S_k)$. Further, we define the cut norm of a tensor as

$$\|\mathbf{M}\|_\square = \max_{S_i \subseteq R_i} |\mathbf{M}(S_1, S_2, \ldots, S_k)|, \quad \text{where} \quad \mathbf{M}(S_1, \ldots, S_k) = \sum_{(s_1,\ldots,s_k) \in S_1 \times \cdots \times S_k} \mathbf{M}(s_1, \ldots, s_k).$$

Let $\mathbf{A}: R_1 \times R_2 \times \cdots \times R_k \to \{0,1\}$ be a $0/1$ tensor. Set $k_1 = \lfloor k/2 \rfloor$. Then letting $\mathcal{R} = R_1 \times R_2 \times \cdots \times R_{k_1}$ and $\mathcal{C} = R_{k_1+1} \times R_{k_1+2} \times \cdots \times R_k$, we define a (2-dimensional) matrix $\mathbf{B} = \mathbf{B}(\mathbf{A}) : \mathcal{R} \times \mathcal{C} \to \{0,1\}$ by

$$\mathbf{B}((i_1, i_2, \ldots, i_{k_1}), (i_{k_1+1}, i_{k_1+2}, \ldots, i_k)) = \mathbf{A}(i_1, i_2, \ldots, i_k). \tag{3}$$

We say that $\mathbf{A}$ is (C,γ)-bounded if $\mathbf{B}(\mathbf{A})$ has this property.

Theorem 2 *There are an algorithm* `ApxTensor`*, a polynomial* Π *and a constant* $\Gamma > 0$ *such that the following is true. Suppose that* $C > 1$ *and* $0 < \varepsilon < \frac{1}{2}$*. Let*

$$\gamma = \exp(-\Gamma(C/\varepsilon)^2).$$

If $\mathbf{A}: R_1 \times R_2 \times \cdots \times R_k \to \{0,1\}$ *is a* (C,γ)*-bounded* $0/1$ *tensor, then* `ApxTensor`$(\mathbf{A}, C, \varepsilon)$ *outputs cut tensors*

$$\mathbf{D}_i = CUT(d_i, S_i^1, \ldots, S_i^k) \qquad (S_i^1 \subset R_1, \ldots, S_i^k \subset R_k)$$

for $i = 1, \ldots, s$ *with* $s \leq (\Gamma C/\epsilon)^{2(k-1)}$ *such that*

$$\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_s)\|_\square \leq \epsilon \|\mathbf{A}\|_\square .$$

Moreover, $\sum_{i=1}^s d_i^2 \leq (Cp)^2 \Gamma^{2k}$*. The running time is* $(2^{(C/\varepsilon)^2} + (C/\varepsilon)^{3k}) \cdot \Pi(|R_1 \times \cdots \times R_k|)$*.*

If $\mathcal{R}_1, \ldots, \mathcal{R}_k$ are partitions of $R_1, \ldots, R_k$, then we define a tensor $\mathbf{A}_{\mathcal{R}_1 \times \cdots \times \mathcal{R}_k} : R_1 \times \cdots \times R_k \to [0,1]$ as follows: if $t_i \in \rho_i \in \mathcal{R}_i$ for $i = 1, \ldots, k$, then we set

$$\mathbf{A}_{\mathcal{R}_1 \times \cdots \times \mathcal{R}_k}(t_1, \ldots, t_k) = \frac{\mathbf{A}(\rho_1, \ldots, \rho_k)}{\prod_{i=1}^k |\rho_i|} = \frac{\sum_{(v_1,\ldots,v_k) \in \rho_1 \times \cdots \times \rho_k} \mathbf{A}(v_1, \ldots, v_k)}{\prod_{i=1}^k |\rho_i|}.$$

In words, on every rectangle $\rho_1 \times \cdots \times \rho_k$ made up of partition classes $\rho_i \in \mathcal{R}_i$ the entry of $\mathbf{A}_{\mathcal{R}_1 \times \cdots \times \mathcal{R}_k}$ is the average of $\mathbf{A}$ over that rectangle.

Corollary 3 *There are an algorithm* `PartTensor`*, a polynomial* Π *and a constant* $\Gamma > 0$ *such that the following is true. Suppose that* $C > 0$ *and* $0 < \varepsilon < \frac{1}{2}$*. Let* $\gamma = \exp(-\Gamma(C/\varepsilon)^2)$*. If* $\mathbf{A}: R_1 \times R_2 \times \cdots \times R_k \to \{0,1\}$ *is a* (C,γ)*-bounded* $0/1$ *tensor, then* `PartTensor`$(\mathbf{A}, C, \varepsilon)$ *computes partitions* $\mathcal{R}_1, \ldots, \mathcal{R}_k$ *of* $R_1, \ldots, R_k$ *such that*

$$\|\mathbf{A} - \mathbf{A}_{\mathcal{R}_1 \times \cdots \times \mathcal{R}_k}\|_\square < \varepsilon \|\mathbf{A}\|_\square .$$

Each of the partitions $\mathcal{R}_i$ *consists of at most* $\exp((\Gamma C/\varepsilon)^{2(k-1)})$ *classes. The running time is bounded by*

$$\left[\exp((\Gamma C/\varepsilon)^{2(k-1)}) + (C/\varepsilon)^{3k}\right] \Pi(n^k).$$

1.4 More General Coefficient Values

It is not absolutely necessary to assume that our matrices and tensors are 0/1 valued. For convenience, we will only describe the case where **A** is an $m \times n$ matrix (i.e., 2-dimensional). It will be apparent how to generalise the results to higher dimensions.

We can in fact assume that our coefficient entries are in $\{0, 1, 2, \ldots, d\}$ for some positive integer $d = O(1)$. We write $\mathbf{A} = \mathbf{A}^{(1)} + \cdots + \mathbf{A}^{(d)}$ where $\mathbf{A}^{(r)}(i,j) = \vec{1}_{\mathbf{A}_{i,j} \geq r/d}$ for $r = 1, \ldots, d$. Let p be the density of $\mathbf{A}^{(1)}$ i.e. the proportion of non-zero values. We assume that $\mathbf{A}^{(1)}$ is (C,γ)-bounded, with $\gamma = \gamma(\epsilon/d, C)$ as in Theorem 1. Then $\mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(d)}$ have the following property: if $S \subset [m]$ and $T \subset [n]$ are sets such that $|S| \geq \gamma m$ and $|T| \geq \gamma n$, then $A^{(j)}(S,T) \leq C|S \times T|p$ $(j = 2, \ldots, d)$. Therefore, we can apply the algorithm `ApxMatrix` from Theorem 1 to each of these matrices and replace each $\mathbf{A}^{(i)}$ by a sum of cut-matrices that is within cut-norm $\epsilon mnp/d$ of its cut-norm. The sum of these cut-matrix approximations is then within ϵmnp cut-norm of **A**.

1.5 An approximation algorithm for bounded MAX CSPs

Let $V = \{x_1, \ldots, x_n\}$ be a set of n Boolean variables. A *(binary) k-constraint over* V is a map $\phi : \{0,1\}^{V_\phi} \to \{0,1\}$ that is not identically zero, where $V_\phi \subset V$ is a set of size k. For an assignment $\sigma \in \{0,1\}^V$ we let $\phi(\sigma) = \phi(\sigma(x))_{x \in V_\phi}$. Further, a *k-CSP instance over* V is a set $\mathcal{F}$ of k-constraints over V, and we define

$$\text{OPT}(\mathcal{F}) = \max_{\sigma \in \{0,1\}^V} \sum_{\phi \in \mathcal{F}} \phi(\sigma).$$

We let $\Psi = \Psi_k$ be the set of all $2^{2^k} - 1$ non-zero maps $\{0,1\}^k \to \{0,1\}$. Let $\psi \in \Psi$ and let $\phi : \{0,1\}^{V_\phi} \to \{0,1\}$ be a k-constraint, where $V_\phi = \{x_{i_1}, \ldots, x_{i_k}\}$ with $1 \le i_1 < \cdots < i_k \le n$. Then we say that ϕ *is of type* ψ if for any $\sigma : V_\phi \to \{0,1\}$ we have $\psi(\sigma(x_{i_1}), \ldots, \sigma(x_{i_k})) = \phi(\sigma)$. With this notion we can represent a k-CSP instance $\mathcal{F}$ by a family $(\mathbf{A}^\psi_\mathcal{F})_{\psi\in\Psi}$ of $2^{2^k} - 1$ k-tensors as follows. We let

$$\mathbf{A}^\psi_\mathcal{F}(i_1, \ldots, i_k) = \begin{cases} 1 & \text{if there is a } \phi \in \mathcal{F} \text{ of type } \psi \text{ with } V_\phi = \{x_{i_1}, \ldots, x_{i_k}\} \\ 0 & \text{otherwise.} \end{cases}$$

Further, we say that $\mathcal{F}$ is (C,γ)*-bounded* if all tensors $\mathbf{A}^\psi_\mathcal{F}$ are (C,γ)-bounded ($\psi \in \Psi$).

Theorem 3 *There are an algorithm* `ApxCSP`*, a constant* $\Gamma > 0$*, and a polynomial* Π *such that for any* $k, C > 1$*,* $0 < \varepsilon < \frac{1}{2}$ *there is a number* $n_0 = n_0(C,\varepsilon,k)$ *such that the following is true. Let*

$$\gamma = \exp(-\Gamma 2^{-2^k-2k-2}(C/\varepsilon)^2).$$

If $\mathcal{F}$ *is* (C,γ)*-bounded* k*-CSP instance over* $V = \{x_1, \ldots, x_n\}$ *for some* $n \ge n_0$*, then* `ApxCSP`$(\mathcal{F}, C, \varepsilon)$ *outputs an assignment* $\sigma : V \to \{0,1\}$ *such that*

$$\sum_{\phi\in\mathcal{F}} \phi(\sigma) \ge (1-\varepsilon)\mathit{OPT}(\mathcal{F}).$$

The running time is at most $\Pi\left[\exp(k 2^k 2^{2^k} (C/\varepsilon)^{2k} \ln(C/\varepsilon)) n^k\right]$.

1.6 Related work

1.6.1 Approximating dense matrices and tensors.

As mentioned earlier, Frieze and Kannan [16] dealt with *dense* matrices and tensors. More precisely, they showed that for a tensor $\mathbf{A} : R_1 \times \cdots \times R_k \to [0,1]$ and an $\varepsilon > 0$ one can compute cut tensors $\mathbf{D}_1, \ldots, \mathbf{D}_s$ such that $\|\mathbf{A} - \sum_{i=1}^s \mathbf{D}_i\|_\square < \varepsilon|R_1 \times \cdots \times R_k|$ in time $O(\varepsilon^{2(1-k)}\text{polylog}(1/\varepsilon))$ with $s \le O(\varepsilon)^{2(1-k)}$ as $\varepsilon \to 0$. Let us point out two things.

1. The running time of their algorithm depends *only* on ε, and not on the size of $\mathbf{A}$. This is achieved by randomization. Basically the algorithm just works with a bounded (by a function of ε only) size sample of the input data, and produces an implicit representation of the desired decomposition. (Further results of this type can be found in Arora, Karger and Karpinski [5], Fernandez de la Vega [12], Goldwasser, Goldreich and Ron [18], Alon, de la Vega, Kannan and Karpinski [2] and de la Vega, Kannan, Karpinski and Vempala [13]).) Of course, if $\mathbf{A} : R_1 \times \cdots \times R_k \to \{0,1\}$ is a sparse $0/1$ tensor with density $p = \|\mathbf{A}\|_\square / |R_1 \times \cdots \times R_k| = o(1)$ as the problem size $N = |R_1 \times \cdots \times R_k| \to \infty$, then this sampling approach cannot yield an approximation within εNp. For any constant sized sample of $\mathbf{A}$ is likely to be just identically 0. Therefore, in the present work we do not aim for a sublinear running time.

2. The error term $\varepsilon|R_1 \times \cdots \times R_k|$ does not account for the density of $\mathbf{A}$. For example, suppose that $\mathbf{A}$ is the adjacency matrix of a graph $G = (V,E)$ on n vertices with density $p = 2n^{-2}|E|$. Then the algorithm from [16] can be used to compute a cut norm approximation of $\mathbf{A}$ to within εn^2 for any $\varepsilon > 0$. Hence, we can use this approximation to solve graph partitioning problems such as MAX CUT within an additive error of εn^2 (edges). This is why this approach is limited to *dense* problem instances: if the total number of edges is of lower order than n^2, then an approximation within an additive εn^2 for a fixed $\varepsilon > 0$ is of little value. For similar reasons the techniques of [16] only apply to dense problem instances of k-ary MAX CSP problems, i.e., instances with at least $\Omega(n^k)$ constraints, where n is the number of variables.

In spite of these differences, some of the algorithms that we present are very similar to those from [16]. Thus, our main contribution is to *analyze* these algorithms on sparse matrices/graphs/tensors. For instance, the matrix approximation algorithm for Theorem 1 is almost identical to the procedure described in [16, Section 4.1]. The only difference is that [16] employs as a subroutine a combinatorial procedure for approximating the cut norm of a given $m \times n$ matrix within an *additive* error of εmn, whereas here we need to approximate the cut norm within a constant *multiplicative* factor. To this end, we rely on an algorithm of Alon and Naor [4] (which is based on semidefinite programming). Nonetheless, as we shall see in Section 3 new ideas are necessary to analyze, e.g., the number of cut matrices that are necessary to approximate the input matrix $\mathbf{A}$ within the desired $\varepsilon \|A\|_\square$ in the cut norm (rather than within εmn).

1.6.2 Szemerédi's regularity lemma.

Theorem 2 and the concept of weak regular partitions is related to Szemerédi's well-known regularity lemma [24]. While [24] only deals with "dense" graphs, Kohayakawa [22] and Rödl [23] independently extended the regularity lemma to the sparse case; for a comprehensive survey on the subject see Gerke and Steger [17]. They showed that for any $\varepsilon > 0$ and any $C > 0$ there is a number γ such any (C, γ)-bounded graph has a regular partition $(V_1, \ldots, V_s)$ in the following sense.

- We have $|V_i - n/s| \leq 1$ for all i.
- All but εs^2 pairs (V_i, V_j) satisfy the following. For any two sets $S \subset V_i, T \subset V_j$ of size $|S| \geq \varepsilon |V_i|$, $|T| \geq \varepsilon |V_j|$ we have
$$\left| e_G(S,T) - \frac{|S \times T|}{|V_i \times V_j|} \cdot e_G(V_i, V_j) \right| \leq \varepsilon e_G(V_i, V_j). \tag{4}$$

The number s of classes is bounded by a function $\mathcal{T}(C/\varepsilon)$, i.e., it is *independent* of n. This is the key fact that makes Szemerédi's lemma so useful in extremal combinatorics. However, from an algorithmic perspective the bound $\mathcal{T}(C/\varepsilon)$ is somewhat disappointing, because it is a tower function of height $(C/\varepsilon)^3$:

$$2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} (C/\varepsilon)^3.$$

In fact, there is an infinite family of graphs for which the number of classes in the smallest ε-regular Szemerédi partition is a tower of height C/ε [19]. Moreover, the number γ required in the boundedness condition is as tiny as $\mathcal{T}((C/\varepsilon)^3)^{-1}$.

While [22, 23, 24] focus on proving that a regular partition exists, [1, 3] deal with algorithmic versions of the regularity lemma. In the dense case (i.e., $|E| = \Omega(n^2)$) there is a purely combinatorial algorithm [3] with running time $\mathcal{T}(\varepsilon^{-3}) \cdot \mathrm{poly}(n)$. Moreover, an algorithm for the sparse case was presented in [1]; the running time is $\mathcal{T}((C/\varepsilon)^{-3}) \cdot \mathrm{poly}(n)$ for (C, γ)-bounded graphs, and the algorithm is based on the semidefinite programming algorithm for approximating the cut norm from [4]. For instance, this yields an algorithm for approximating the MAX CUT on (C, γ)-bounded graphs within $1 - \varepsilon$ in time $\mathcal{T}((C/\varepsilon)^3) \cdot \mathrm{poly}(n)$.

Corollary 2 relates to [1] as follows. While the "strong" regularity condition (4) takes into account the "microscopic" edge distribution within (almost) each pair (V_i, V_j), the "weak" regularity concept from Corollary 2 just provides a "macroscopic" approximation w.r.t. the cut norm. This approximation is sufficiently strong for algorithmic applications such as MAX CUT (but it would not suffice for applications in extremal combinatorics that rely on the "counting lemma"). In effect, the algorithm is more efficient. Indeed, instead of scaling as a tower function $\mathcal{T}((C/\varepsilon)^5)$, the running time of the algorithm `WeakPartition` from Corollary 2 is bounded by $\exp(O(C/\varepsilon)^2)$ in terms of C, ε. Although this may still seem impractical, this is just a worst-case upper bound, and it is quite conceivable that it is practically much easier to find a good approximation in the cut norm than a good regular partition. Besides, as Theorem 1 shows, one can approximate a (C, γ)-bounded adjacency matrix by a sum of $O(C/\varepsilon)^2$ cut matrices (if the actual partition of the vertex set is not needed), thus avoiding the exponential dependence on C/ε. Similarly, the parameter γ required in the boundedness condition is just $\gamma = \exp(-O(C/\varepsilon)^2)$, rather than $\gamma = 1/\mathcal{T}((C/\varepsilon)^3)$ as in [1]. Consequently, Corollary 2 applies to a larger class of graphs.

A further novel aspect here is that we extend our results to k-dimensional tensors (or k-uniform hypergraphs). This point is not addressed in [1].

2 Preliminaries

An important ingredient to the algorithm `ApxMatrix` for Theorem 1 is the the following algorithmic version of Grothendieck's inequality from Alon and Naor [4].

Theorem 4 *There is a polynomial time algorithm and a number $\alpha_0 > 0$ that has the following property. Given a $m \times n$ matrix $\mathbf{M}$, the algorithm outputs sets $S \subset [m]$ and $T \subset [n]$ such that $|\mathbf{M}(S,T)| \geq \alpha_0 \|\mathbf{M}\|_\square$.*

Alon and Naor present a randomized algorithm with $\alpha_0 > 0.56$, and a deterministic one with $\alpha_0 \geq 0.03$.

The algorithm `ApxTensor` for Theorem 2 employs an algorithm `FKTensor` from [16] as a subroutine.

Theorem 5 *There are a polynomial Π_{FK}, an algorithm `FKTensor` and a number $\Gamma_{FK} > 0$ such that the following is true. Suppose that $\mathbf{M} : R_1 \times \cdots \times R_k \to [0,1]$ is a tensor and let $0 < \delta < 1$. Then `FKTensor`$(\mathbf{M}, \delta)$ outputs cut tensors $\mathbf{D}_1, \ldots, \mathbf{D}_s$ such that $\|\mathbf{M} - \mathbf{D}_1 - \cdots - \mathbf{D}_s\|_\square \leq \delta \prod_{i=1}^k |R_i|$ and $s \leq (\Gamma_{FK}/\delta)^{2(k-1)}$. Moreover, $\sum_{i=1}^s \|\mathbf{D}_i\|_\infty^2 \leq \Gamma_{FK}^k$, and the running time is at most $\delta^{-3k}\Pi_{FK}(\prod_{i=1}^k |R_i|)$.*

Actually Frieze and Kannan have a slightly stronger statement [16, Section 6] (better running time), but the above is sufficient for our purposes and easier to state.

If $\mathbf{M}$ is a real $m \times n$ matrix, then we let

$$\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n \mathbf{M}_{ij}^2}$$

signify the Frobenius norm of $\mathbf{M}$. Moreover, if G is a graph, then we denote the vertex set of G by $V(G)$ and the edge set by $E(G)$. For sets $S, T \subset V(G)$ we let $e_G(S,T)$ signify the number of S-T-edges of G, and $e_G(S)$ signifies the number of edges spanned by S.

Suppose that X is a set and that $\mathcal{P}_1, \mathcal{P}_2$ are partitions of X. We say that $\mathcal{P}_1$ is *coarser* than $\mathcal{P}_2$ if each class of $\mathcal{P}_2$ is contained in a class of $\mathcal{P}_1$. If $\mathcal{S}$ is an arbitrary set of subsets of X, then there is a unique partition $\mathcal{P}$ of X such that

1. each set in X is a union of classes of $\mathcal{P}$,
2. $\mathcal{P}$ is coarser than any other partition that satisfies 1.

This partition $\mathcal{P}$ has at most $2^{|\mathcal{S}|}$ classes.

3 Approximating and partitioning $0/1$ matrices and graphs

3.1 Proof of Theorem 1

Let $C > 1$ and $0 < \varepsilon < \frac{1}{2}$. Moreover, let α_0 be the constant from Theorem 4 and set

$$\kappa = \frac{513C^2}{\varepsilon^2 \alpha_0}, \quad \gamma = \frac{\varepsilon \alpha_0}{2^{10\kappa} C}, \quad \gamma' = 2^\kappa \gamma.$$

Throughout this section we assume that $\mathbf{A}$ is a (C, γ) bounded 0/1 matrix of size $m \times n$.

Algorithm 6 `ApxMatrix`$(\mathbf{A}, C, \varepsilon)$
Input: A 0/1 matrix $\mathbf{A}$ of size $m \times n$, numbers $C, \varepsilon > 0$.
Output: A sequence of cut matrices.

1. Set $\mathbf{A}_0 = \mathbf{A}$.
2. For $j = 0, 1, 2, \ldots, \kappa$ do
3. Compute sets S_{j+1}, T_{j+1} of sizes $|S_{j+1}| \geq m/2, |T_{j+1}| \geq n/2$ s.t. $|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_\square /4$.
4. If $|\mathbf{A}_j(S_{j+1}, T_{j+1})| < \alpha_0 \varepsilon mnp/4$ and $j \geq 1$, then output the cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_j$ and halt. Else,
5. Compute
$$d_{j+1} = \frac{\mathbf{A}_j(S_{j+1}, T_{j+1})}{|S_{j+1}||T_{j+1}|},$$
set $\mathbf{D}_{j+1} = \text{CUT}(d_{j+1}, S_{j+1}, T_{j+1})$, and let $\mathbf{A}_{j+1} = \mathbf{A}_j - \mathbf{D}_{j+1}$.
6. Output "failure".

In order to approximate the given $0/1$ matrix $\mathbf{A}$ by a sum $\mathbf{D}_1 + \cdots + \mathbf{D}_j$ of cut matrices, ApxMatrix proceeds as follows. After j iterations, $\mathbf{A}_j = \mathbf{A} - \sum_{i=1}^{j} \mathbf{D}_j$ is the "error term" that results from approximating $\mathbf{A}$ by $\sum_{i=1}^{j} \mathbf{D}_j$. Thus, the goal is to eventually achieve an error term $\mathbf{A}_j$ that has a small cut norm. Therefore, Step 3 computes sets S_{j+1}, T_{j+1} of rows and columns such that $|\mathbf{A}_j(S_{j+1}, T_{j+1})|$ is a good approximation of the cut norm of $\mathbf{A}_j$. If the term $|\mathbf{A}_j(S_{j+1}, T_{j+1})|$ (and hence the cut norm of $\mathbf{A}_j$) is small, then Step 4 terminates and outputs the cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_j$. Otherwise, S_{j+1}, T_{j+1} witness a set of rows/columns on which $\sum_{i=1}^{j} \mathbf{D}_j$ does not provide a good enough approximation. Therefore, Step 5 adds a further "patch" $\mathbf{D}_{j+1}$, which is a cut matrix whose value on $S_{j+1} \times T_{j+1}$ is just the average d_{j+1} of $\mathbf{A}_j$ over that square (note that d_{j+1} may be negative). This ensures that $\mathbf{A}_{j+1}(S_{j+1}, T_{j+1}) = 0$, and thus takes care of the discrepancy witnessed by S_{j+1}, T_{j+1}.

If the algorithm outputs cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_j$, then clearly
$$\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_j)\|_\square = \|\mathbf{A}_j\|_\square \leq \varepsilon mnp = \varepsilon \|\mathbf{A}\|_\square ,$$
because of the halting condition in Step 4. Hence, in order to establish Theorem 1, we need to prove that

(a) Step 3 of ApxMatrix can be implemented by a polynomial time algorithm,

(b) the halting condition in Step 4 is satisfied for some $1 \leq j \leq \kappa$.

Proposition 1 *In Step 3 the sets S_{j+1}, T_{j+1} can be computed in time poly(mn).*

Proof To obtain S_{j+1}, T_{j+1}, we use the polynomial time algorithm from Theorem 4, which yields sets S'_{j+1}, T'_{j+1} such that $|\mathbf{A}_j(S'_{j+1}, T_{j+1})'| \geq \alpha_0 \|\mathbf{A}_j\|_\square$. If $|S'_{j+1}| \geq n/2$ then we take $S_{j+1} = S'_{j+1}$. If $|S'_{j+1}| < n/2$ then since
$$\mathbf{A}(R, T'_{j+1}) = \mathbf{A}(S'_{j+1}, T'_{j+1}) + \mathbf{A}(R \setminus S'_{j+1}, T'_{j+1})$$
we get $\max\{|\mathbf{A}(R, T'_{j+1})|, \ \mathbf{A}(R \setminus S'_{j+1}, T'_{j+1})|\} \geq \alpha_0 \|\mathbf{A}_0\|_\square /2$. We can therefore take either R or $R \setminus S'_{j+1}$ as our set S_{j+1} and note it is at least $n/2$ in size. We perform the same operation to get T_{j+1}, losing (at most) another factor 2 in the approximation. □

With respect to (b), we will study the Frobenius norm of $\mathbf{A}_j$. Namely, it is not difficult to show that $\|\mathbf{A}_j\|_F^2 \leq \|\mathbf{A}\|_F^2(1 - j \cdot \alpha_0^2\varepsilon^2 p/4)$. Since trivially $\|\mathbf{A}_j\|_F \geq 0$, this implies that the total number of iterations is at most $4/(\alpha_0^2\varepsilon^2 p)$. Hence, if p is bounded from below by a constant, then this argument shows that the total number of iterations is bounded by a number that does not depend on n, m. In fact, this is the basic argument used to establish the matrix decomposition theorem in [16, Section 4.1].

But in the present work we do *not* assume that p remains bounded away from 0 by a number independent of n, m. In effect, the aforementioned argument does not apply. As it turns out, the problem is that the above argument just uses the trivial lower bound $\|\mathbf{A}_j\|_F^2 \geq 0$. By contrast, the basic idea here is to use the boundedness condition to establish $\|\mathbf{A}_F\|^2(1 - C^2 p)$ as a lower bound. Indeed, if we could show that $\|\mathbf{A}_j\|_F^2 \geq \|\mathbf{A}_F\|^2(1 - C^2 p)$ for all j, then the bound $\|\mathbf{A}_j\|_F^2 \leq \|\mathbf{A}\|_F^2(1 - j \cdot \alpha_0^2\varepsilon^2 p/4)$ would imply that the number of iterations is at most $4C^2/(\alpha_0^2\varepsilon^2)$, and thus independent of m, n, p.

However, we can't quite use the boundedness condition to prove that $\|\mathbf{A}_j\|_F^2 \geq \|\mathbf{A}_F\|^2(1 - C^2 p)$. The reason is that the boundedness condition only applies to "sufficiently large" sets, i.e., sets of size at least γn. Therefore, to show that ApxMatrix stops after at most κ iterations, we will consider slightly different sequences of matrices $\mathbf{D}'_j$, $\mathbf{A}'_j$, to which the boundedness condition applies. The matrices $\mathbf{D}'_j$, $\mathbf{A}'_j$ will be "close" to $\mathbf{D}_j$, $\mathbf{A}_j$ in cut norm, and to bound the number of iterations we are going to investigate the Frobenius norm of $\mathbf{A}'_j$.

We construct the matrices $\mathbf{D}'_j$, $\mathbf{A}'_j$ as follows. *Let us assume (for contradiction) that* `ApxMatrix` *outputs "failure", i.e., the number of iterations performed by Steps 2–5 is κ.* Then during these κ iterations the algorithm constructed sets $S_1, \ldots, S_\kappa$ of rows and $T_1, \ldots, T_\kappa$ of columns. Let $\mathcal{S}$ be the coarsest partition of the set $[m]$ of row indices such that each S_i is a union of classes of $\mathcal{S}$ (thus, $\mathcal{S}$ consists of the classes of the Venn diagram of the sets $S_1, \ldots, S_\kappa$). Similarly, let $\mathcal{T}$ be the coarsest partition of the columns set $[n]$ such that every T_i is a union of classes of $\mathcal{T}$. Clearly, both $\mathcal{S}$ and $\mathcal{T}$ have at most 2^κ classes. The reason why the boundedness condition does not imply directly that $\|\mathbf{A}_j\|_F^2 \geq \|\mathbf{A}_F\|^2(1 - C^2 p)$ is that some classes of $\mathcal{S}$ and $\mathcal{T}$ may have size less than γm or γn. Therefore, we let

$$R_0 = \bigcup_{S \in \mathcal{S} : |S| < \gamma m} S, \quad C_0 = \bigcup_{T \in \mathcal{T} : |T| < \gamma n} T$$

comprise the "small" classes of the partitions $\mathcal{S}, \mathcal{T}$. Setting $\gamma' = 2^\kappa \gamma$, we have

$$|R_0| \leq \gamma' m, \quad |C_0| \leq \gamma' n. \tag{5}$$

Further, let $\mathbf{A}'_0 = \mathbf{A}'$ be the matrix obtained from $\mathbf{A}$ by replacing all rows in R_0 and all columns in C_0 by 0. In addition, define inductively sets $S'_j = S_j \setminus R_0$ and $T'_j = T_j \setminus T_0$ and

$$d'_{j+1} = \frac{\mathbf{A}'_j(S'_{j+1}, T'_{j+1})}{|S'_{j+1}||T'_{j+1}|}, \quad \mathbf{D}'_{j+1} = \mathrm{CUT}(S'_{j+1}, T'_{j+1}, d'_{j+1}), \quad \mathbf{A}'_{j+1} = \mathbf{A}'_j - \mathbf{D}'_{j+1}.$$

Let $\mathcal{S}'$ be the coarsest partition of $[m] \setminus R_0$ such that each S'_j is a union of classes of $\mathcal{S}'$, and define a partition $\mathcal{T}'$ of $[n] \setminus C_0$ analogously w.r.t. the sets T'_j. Then the construction of the sets S'_j, T'_j readily implies:

Fact 7 *All classes of $\mathcal{S}'$ (resp. $\mathcal{T}'$) have size at least γm (resp. γn).*

Consequently, we can use the boundedness condition to infer the following.

Lemma 1 *For all $1 \leq j \leq \kappa$ we have $\|\mathbf{A}'_j\|_F^2 \geq \|\mathbf{A}'\|_F^2 (1 - 2C^2 p)$.*

Proof Let $\mathbf{M} = \sum_{i=1}^{j} \mathbf{D}'_j$. Then for any two sets $S \in \mathcal{S}'$, $T \in \mathcal{T}'$ the matrix $\mathbf{M}$ is constant on the square $S \times T$, because every $\mathbf{D}'_j$ is a cut matrix on the square $S'_j \times T'_j$, and S'_j, T'_j are unions of classes of $\mathcal{S}'$, $\mathcal{T}'$. Thus, letting $m_{S \times T}$ signify the value that $\mathbf{M}$ takes on $S \times T$, we obtain

$$\|\mathbf{A}'_j\|_F^2 = \|\mathbf{A}' - \mathbf{M}\|_F^2 = \sum_{S \in \mathcal{S}', T \in \mathcal{T}'} \sum_{(v,w) \in S \times T} (\mathbf{A}'(v,w) - m_{S \times T})^2.$$

For any $S \in \mathcal{S}'$, $T \in \mathcal{T}'$ the sum $\sum_{(v,w) \in S \times T} (\mathbf{A}'(v,w) - m_{S \times T})^2$ is minimized iff

$$m_{S \times T} = m^*_{S \times T} = \mathbf{A}'(S,T)/(|S| \cdot |T|).$$

Therefore,

$$\begin{aligned}
\sum_{(v,w) \in S \times T} (\mathbf{A}'(v,w) - m_{S \times T})^2 &\geq \sum_{(v,w) \in S \times T} (\mathbf{A}'(v,w) - m^*_{S \times T})^2 \\
&= \sum_{(v,w) \in S \times T} \mathbf{A}'(v,w)^2 - 2\mathbf{A}'(S,T) m^*_{S \times T} + m^{*2}_{S \times T} |S| \cdot |T| \\
&= \sum_{(v,w) \in S \times T} \mathbf{A}'(v,w)^2 - m^{*2}_{S \times T} |S| \cdot |T|.
\end{aligned}$$

Since $\mathbf{A}'$ is (C, γ) bounded and because $|S| \geq \gamma m$, $|T| \geq \gamma n$ by Lemma 7, we get $m^*_{S \times T} \leq Cp$. Hence,

$$\|\mathbf{A}'_j\|_F^2 \geq \|\mathbf{A}'\|_F^2 - (Cp)^2 mn. \tag{6}$$

Finally, using the fact that $\mathbf{A}$ and $\mathbf{A}'$ are $0,1$ matrices, we have

$$\begin{aligned}\|\mathbf{A}\|_F^2 - \|\mathbf{A}'\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij}^2 - {\mathbf{A}'_{ij}}^2 = \mathbf{A}(R_0,[n]) + \mathbf{A}([m],C_0) - \mathbf{A}(R_0,C_0) \\ &\leq \mathbf{A}(R_0,[n]) + \mathbf{A}([m],C_0) \overset{(5)}{\leq} 2C\gamma' mnp < mnp/2,\end{aligned}$$

whence $\|\mathbf{A}'\|_F^2 \geq \|\mathbf{A}\|_F^2/2 = mnp/2$. Thus, the assertion follows from (6). □

To show that our assumption that ApxMatrix performs at least κ iterations yields a contradiction, we shall derive the following upper bound on $\|\mathbf{A}'_j\|_F^2$.

Lemma 2 *For all $1 \leq j \leq \kappa$ we have $\|\mathbf{A}'_j\|_F^2 \leq \|\mathbf{A}'\|_F^2 (1 - j \cdot \alpha_0^2 \varepsilon^2 p/256)$.*

Combining Lemmas 1 and 2 and setting $j = \kappa$, we conclude $2C^2 \geq \kappa \cdot \alpha_0^2\varepsilon^2/256$, which contradicts our choice of κ (cf. (2)). This completes the proof of Theorem 1.

In the rest of this section we prove Lemma 2. The following lemma shows that $\mathbf{A}'$ is close to $\mathbf{A}$ in cut norm. Recall that $\gamma' = 2^\kappa \gamma$.

Lemma 3 *We have $\|\mathbf{A} - \mathbf{A}'\|_\square \leq 2C\gamma' mnp$.*

Proof For any sets $S \subset [m]$, $T \subset [n]$ we have

$$|\mathbf{A}(S,T) - \mathbf{A}'(S,T)| \leq \mathbf{A}(R_0,[n]) + \mathbf{A}([m],C_0).$$

If $|R_0| \geq \gamma m$, then we let $R'_0 = R_0$; otherwise, let $R'_0 \supset R_0$ be any superset of size γm. Then the fact that $\mathbf{A}$ is (C,γ)-bounded implies that for any $T \subseteq [n]$, $|T| \geq \gamma n$,

$$\mathbf{A}(R_0,T) \leq \mathbf{A}(R'_0,T) \leq C|R'_0|np \leq C\gamma' mnp, \qquad \text{from (5).} \tag{7}$$

Applying the same argument to $\mathbf{A}([m],C_0)$ gives the result. □

In addition, we also need to show that the matrices $\mathbf{D}_j, \mathbf{D}'_j$ are close in cut norm. To this end, we derive a bound on the coefficients d_j from Step 5 of ApxMatrix.

Lemma 4 *$|d_j| \leq 2^j Cp$ for all $1 \leq j \leq \kappa$.*

Proof The proof is by induction on j. For $j = 1$ we have

$$d_1 = \frac{\mathbf{A}_0(S_1,T_1)}{|S_1||T_1|} \leq Cp,$$

because $\mathbf{A}_0(S_1,T_1) \leq Cp|S_1||T_1|$ by the boundedness condition. Furthermore, assuming that $|d_i| \leq 2^i Cp$ for all $i \leq j$, we can bound d_{j+1} as follows.

$$\begin{aligned}|\mathbf{A}_j(S_{j+1},T_{j+1})| &= \left|\mathbf{A}_0(S_{j+1},T_{j+1}) + \sum_{i=1}^{j} d_i |S_{j+1}\cap S_i|\,|T_{j+1}\cap T_i|\right| \\ &\leq |S_{j+1}||T_{j+1}| \cdot \left[Cp + \sum_{i=1}^{j}|d_i|\right] \leq |S_{j+1}||T_{j+1}|Cp\sum_{i=0}^{j} 2^i \leq 2^{j+1}|S_{j+1}||T_{j+1}|Cp.\end{aligned}$$

Thus, $|d_{j+1}| = |\mathbf{A}_j(S_{j+1},T_{j+1})|/(|S_{j+1}||T_{j+1}|) \leq 2^{j+1}Cp$. □

Lemma 4 implies the following bound on the cut norm of $\mathbf{D}_j - \mathbf{D}'_j$.

Corollary 4 *For all $1 \leq j \leq \kappa$ we have $\left\|\mathbf{D}_j - \mathbf{D}'_j\right\|_\square \leq 2^{8j} C\gamma' mnp$.*

Proof We proceed by induction. The definitions of d_j and d'_j imply that

$$\begin{aligned}
\left|d'_j - d_j\right| &= \left|\frac{\mathbf{A}'_{j-1}(S'_j,T'_j)}{|S'_j||T'_j|} - \frac{\mathbf{A}_{j-1}(S_j,T_j)}{|S_j||T_j|}\right| = \frac{\left||S_j||T_j|\mathbf{A}'_{j-1}(S'_j,T'_j) - |S'_j||T'_j|\mathbf{A}_{j-1}(S_j,T_j)\right|}{|S_j||S'_j||T_j||T'_j|} \\
&\le \frac{\left|\mathbf{A}_{j-1}(S'_j,T'_j) - \mathbf{A}_{j-1}(S_j,T_j)\right|}{|S'_j||T'_j|} + \frac{\left|(|S_j||T_j| - |S'_j||T'_j|)\mathbf{A}_{j-1}(S_j,T_j)\right|}{|S_j||S'_j||T_j||T'_j|} \\
&\qquad + \frac{\left|\mathbf{A}'_{j-1}(S'_j,T'_j) - \mathbf{A}_{j-1}(S'_j,T'_j)\right|}{|S'_j||T'_j|} \\
&\le 16(mn)^{-1}(|\mathbf{A}_{j-1}(R_0,T_j)| + |\mathbf{A}_{j-1}(S_j,C_0)|) \\
&\quad +64(mn)^{-2}|\mathbf{A}_{j-1}(S_j,T_j)|(|R_0||C_0| + |R_0||T_j| + |S_j||C_0|) \\
&\quad +16(mn)^{-1}\left|\mathbf{A}'_{j-1}(S'_j,T'_j) - \mathbf{A}_{j-1}(S'_j,T'_j)\right|, \qquad (8)
\end{aligned}$$

because $|S_j| \ge m/2$, $|T_j| \ge n/2$ by construction and $|S'_j| \ge |S_j| - |R_0| \ge m/4$, $|T_j| \ge |T_j| - |C_0| \ge n/4$ by (5).

Observe that for $j = 1$ the third term in (8) equals 0. Furthermore, the boundedness condition and (5) imply that $\mathbf{A}(R_0,[n]) + \mathbf{A}([n],C_0) \le 2C\gamma' mnp$, (see (7)) and so we conclude that $|d'_1 - d_1| \le C\gamma' p(32+192) = 224C\gamma' p$, and thus

$$\|\mathbf{D}_1 - \mathbf{D}'_1\|_\square \le |d_1 - d'_1|mn + 2C\gamma' mnp \le 226C\gamma' mnp.$$

Now assume inductively that for $1 \le i \le j-1$ we have

$$\|\mathbf{D}_i - \mathbf{D}'_i\|_\square \le 2^{8j}C\gamma' mnp. \qquad (9)$$

Then, for $j > 1$ Lemma 4 implies that

$$|\mathbf{A}_{j-1}(R_0,T_j)| \le |\mathbf{A}(R_0,T_j)| + \sum_{i=1}^{j-1} |d_i||R_0|n \le 2^j C\gamma' mnp.$$

Similarly, $|\mathbf{A}_{j-1}(S_j,C_0)| \le 2^j C\gamma' mnp$. Thus the first term in (8) is at most $2^{j+5}C\gamma' p$.

Moreover, once more by Lemma 4 we have

$$|\mathbf{A}_{j-1}(S_j,T_j)| \le |\mathbf{A}(S_j,T_j)| + mn\sum_{i=1}^{j-1}|d_i| \le 2^j Cmnp.$$

Consequently, the second term in (8) is at most $3 \cdot 2^{j+6}C\gamma' p$. Finally, by induction we obtain

$$\begin{aligned}
\left|\mathbf{A}'_{j-1}(S'_j,T'_j) - \mathbf{A}_{j-1}(S'_j,T'_j)\right| &\le \sum_{i=1}^{j-1}|\mathbf{D}'_i(S'_j,T'_j) - \mathbf{D}_i(S'_j,T'_j)| \le \sum_{i=1}^{j-1}\|\mathbf{D}'_i - \mathbf{D}_i\|_\square \\
&\le C\gamma' mnp\sum_{i=1}^{j-1}2^{8i} < 2^{8j-7}C\gamma' mnp.
\end{aligned}$$

Hence, the third term in (8) is at most $2^{8j-3}C\gamma' p$. Plugging these bounds into (8) we get $|d'_j - d_j| \le 2^{8j-1}C\gamma' p$. Finally, (5) and Lemma 4 yield

$$\left\|\mathbf{D}_j - \mathbf{D}'_j\right\|_\square \le |d_j - d'_j|mn + (|R_0|n + m|C_0|)|d_j| \le 2^{8j-1}C\gamma' mnp + 2^{j+1}C\gamma' mnp \le 2^{8j}C\gamma' mnp,$$

as desired. □

Combining Lemma 3 with Corollary 4, we conclude that the two matrices $\mathbf{A}_j = \mathbf{A} - \sum_{i=1}^j \mathbf{D}_j$ and $\mathbf{A}'_j = \mathbf{A}' - \sum_{i=1}^j \mathbf{D}'_j$ are close in cut norm.

Corollary 5 *For all $1 \le j \le \kappa$ we have $\left\|\mathbf{A}'_j - \mathbf{A}_j\right\|_\square \le 2^{8j+1}C\gamma' mnp$.*

Proof of Lemma 2. We are going to show that

$$\|\mathbf{A}'_{j+1}\|_F^2 \leq \|\mathbf{A}'_j\|_F^2 - \alpha_0^2\varepsilon^2 mnp^2/256 \tag{10}$$

for any $1 \leq j < \kappa$; this bound immediately implies the assertion. Remember that $|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_\square /4$ by the construction of S_{j+1}, T_{j+1} in Step 3 of `ApxMatrix`. Therefore, combining Corollary 5 with Lemmas 3 and 4, we obtain

$$\begin{aligned} |\mathbf{A}'_j(S'_{j+1}, T'_{j+1})| &\geq |\mathbf{A}_j(S_{j+1}, T_{j+1})| - |\mathbf{A}_j(S_{j+1}, T_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})| \\ &\quad - |\mathbf{A}'_j(S'_{j+1}, T'_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})| \\ &\geq \alpha_0\varepsilon mnp/4 - \left(2C\gamma' mnp + \sum_{i=1}^{j} 2^i C\gamma' mnp\right) - 2^{8j+1}C\gamma' mnp \\ &\geq \alpha_0 \varepsilon mnp/8, \end{aligned} \tag{11}$$

where the last inequality follows from our choice of γ and the fact that $\gamma' = 2^\kappa \gamma$. Further, as

$$d'_{j+1} = \frac{\mathbf{A}'_j(S'_{j+1}, T'_{j+1})}{|S'_{j+1}||T'_{j+1}|}$$

by construction, (11) implies that

$$\begin{aligned} \|\mathbf{A}'_j\|_F^2 - \|\mathbf{A}'_{j+1}\|_F^2 &= \sum_{(s,t)\in S'_{j+1}\times T'_{j+1}} \mathbf{A}'_j(s,t)^2 - (\mathbf{A}'_j(s,t) - d'_{j+1})^2 \\ &= {d'_{j+1}}^2 \mathbf{A}'(S'_{j+1}, T'_{j+1}) = \frac{\mathbf{A}'_j(S'_{j+1}, T'_{j+1})^2}{|S'_{j+1}||T'_{j+1}|} \geq \frac{(\alpha_0\varepsilon mnp)^2}{256mn}, \end{aligned}$$

whence (10) follows. □

3.2 Proof of Corollary 1

Let $0 < \varepsilon < \frac{1}{2}$ and $C > 1$, and let κ, γ be as in (2). Given a (C, γ)-bounded matrix $\mathbf{A}$ of size $m \times n$ and the numbers C, ε, `PartMatrix` calls `ApxMatrix`$(\mathbf{A}, C, \varepsilon)$ to obtain cut matrices

$$\mathbf{D}_i = \text{CUT}(d_i, S_i, T_i) \qquad (i = 1, \ldots, s)$$

for some $1 \leq s \leq \kappa$. Then, it computes the coarsest partition $\mathcal{S}$ of $[m]$ such that each class S_i is a union of classes of $\mathcal{S}$ $(1 \leq i \leq s)$. Similarly, $\mathcal{T}$ is the coarsest partition of $[n]$ such that each class T_i is a union of classes of $\mathcal{T}$ $(1 \leq i \leq s)$.

This construction ensures that $|\mathcal{S}|, |\mathcal{T}| \leq 2^s \leq 2^\kappa$. Hence, the running time of `PartMatrix` is $(\kappa + 2^\kappa)\Pi(mn)$ for a fixed polynomial Π. Thus, to complete the proof of Corollary 1 we just need to show that $\|\mathbf{A} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq 2\varepsilon \|\mathbf{A}\|_\square$. Since the matrix $\mathbf{D} = \sum_{i=1}^s \mathbf{D}_i$ satisfies $\|\mathbf{A} - \mathbf{D}\|_\square \leq \varepsilon \|\mathbf{A}\|_\square$ by Theorem 1, it suffices to prove that

$$\|\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq \|\mathbf{D} - \mathbf{A}\|_\square . \tag{12}$$

To prove (12), we use the same argument as in [16, Section 5]. Let $X \subset [m]$ and $Y \subset [n]$ be sets such that $|(\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}})(X, Y)| = \|\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square$. On each square $S \times T$ with $S \in \mathcal{S}$ and $T \in \mathcal{T}$ the matrix $\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ is constant. We may therefore assume that X is a union of classes of $\mathcal{S}$ and Y is a union of classes of $\mathcal{T}$. Furthermore, as $A(S,T) = A_{\mathcal{S}\times\mathcal{T}}(S,T)$ for any $S \in \mathcal{S}$ and $T \in \mathcal{T}$ by the definition of $A_{\mathcal{S}\times\mathcal{T}}(S,T)$, we conclude that

$$\begin{aligned} \|\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square &= |\mathbf{D}(X,Y) - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}(X,Y)| = \left|\mathbf{D}(X,Y) - \sum_{S\in\mathcal{S}:S\subset X}\sum_{T\in\mathcal{T}:T\subset Y} \mathbf{A}_{\mathcal{S}\times\mathcal{T}}(S,T)\right| \\ &= \left|\mathbf{D}(X,Y) - \sum_{S\in\mathcal{S}:S\subset X}\sum_{T\in\mathcal{T}:T\subset Y} \mathbf{A}(S,T)\right| = |\mathbf{D}(X,Y) - \mathbf{A}(X,Y)| \leq \|\mathbf{D} - \mathbf{A}\|_\square , \end{aligned}$$

thereby proving (12). □

3.3 Proof of Corollary 2

Let $C > 1$ and $0 < \varepsilon < \frac{1}{2}$, let κ, γ be as in (2) and suppose that $G = (V, E)$ is a (C, γ)-bounded graph on n vertices $V = \{1, \ldots, n\}$ with adjacency matrix $\mathbf{A}$. The algorithm `WeakPartition`(G, C, ε) calls `PartMatrix`$(\mathbf{A}, C, \varepsilon)$ to obtain two partitions $\mathcal{S}, \mathcal{T}$ of V such that $\|\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}\|_\square \leq 2\varepsilon \|A\|_\square$. By Corollary 1 both $\mathcal{S}, \mathcal{T}$ have at most 2^κ classes. Then, the algorithm constructs the coarsest partition $\mathcal{V}$ of V that is a refinement of both $\mathcal{S}$ and $\mathcal{T}$. Clearly, $|\mathcal{V}| \leq 2^{2\kappa}$, and the running time of the algorithm is at most $2^{2\kappa}\Pi(n)$ for some fixed polynomial Π.

To complete the proof, we need to show that $\|\mathbf{A}_{\mathcal{V}\times\mathcal{V}} - \mathbf{A}\|_\square \leq 4\varepsilon \|\mathbf{A}\|_\square$. Since $\|\mathbf{A} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq 2\varepsilon \|\mathbf{A}\|_\square$ by Corollary 1, we just need to prove that $\|\mathbf{A}_{\mathcal{V}\times\mathcal{V}} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq \|\mathbf{A} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square$. To show this, we use a similar argument to that given in the proof of Corollary 1. Namely, let $X, Y \subset V$ be such that $|(\mathbf{A}_{\mathcal{V}\times\mathcal{V}} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}})(X, Y)| = \|\mathbf{A}_{\mathcal{V}\times\mathcal{V}} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square$. Since both $\mathbf{A}_{\mathcal{V}\times\mathcal{V}}$ and $\mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ are constant on each square $S \times T$ with $S, T \in \mathcal{V}$, we may assume that X, Y are unions of classes of $\mathcal{V}$. Therefore, the definition of $\mathbf{A}_{\mathcal{V}\times\mathcal{V}}$ entails that

$$\begin{aligned}
\|\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}_{\mathcal{V}\times\mathcal{V}}\|_\square &= |(\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}_{\mathcal{V}\times\mathcal{V}})(X, Y)| \\
&= \left|\mathbf{A}_{\mathcal{S}\times\mathcal{T}}(X, Y) - \sum_{S,T\in\mathcal{V}: S\times T\subset X\times Y} \mathbf{A}_{\mathcal{V}\times\mathcal{V}}(S, T)\right| \\
&= \left|\mathbf{A}_{\mathcal{S}\times\mathcal{T}}(X, Y) - \sum_{S,T\in\mathcal{V}: S\times T\subset X\times Y} \mathbf{A}(S, T)\right| \\
&\leq |(\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A})(X, Y)| \leq \|\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}\|_\square,
\end{aligned}$$

as claimed. □

4 Approximating and partitioning k-dimensional tensors

4.1 Proof of Theorem 2

Let $0 < \varepsilon < \frac{1}{2}$ and $C > 1$. Let $\kappa = 64(\zeta + 100)(C/\varepsilon)^2$, where ζ is the constant from Theorem 1, and set $\gamma = 2^{-3\kappa}$. Throughout this section we assume that $\mathbf{A} : R_1 \times \cdots \times R_k \to \{0, 1\}$ is a (C, γ)-bounded tensor. Let $k_1 = \lfloor k/2 \rfloor$.

Algorithm 8 `ApxTensor`$(\mathbf{A}, C, \varepsilon)$
Input: A tensor $\mathbf{A} : R_1 \times \cdots \times R_k \to \{0, 1\}$, numbers $C, \varepsilon > 0$.
Output: A sequence of cut tensors.

1. Set up the matrix $\mathbf{B} = \mathbf{B}(\mathbf{A})$ as in (3) and let p be the density of $\mathbf{B}$.
2. Call `PartMatrix`$(\mathbf{B}, C, \varepsilon/8)$ to obtain partitions $\mathcal{S}$ of $R_1 \times \cdots \times R_{k_1}$ and $\mathcal{T}$ of $R_{k_1+1} \times \cdots \times R_k$ (cf. Corollary 1).
3. Let $\hat{\mathbf{A}} : R_1 \times \cdots \times R_k \to [0, 1]$ be the tensor defined by
$$\hat{\mathbf{A}}(i_1, \ldots, i_k) = \min\left\{1, \frac{\mathbf{B}_{\mathcal{S}\times\mathcal{T}}((i_1, \ldots, i_{k_1}), (i_{k_1+1}, \ldots, i_k))}{Cp}\right\},$$
where $\mathbf{B}_{\mathcal{S}\times\mathcal{T}}$ is the approximation of $\mathbf{B}$ corresponding to the parition $\mathcal{S} \times \mathcal{T}$ (cf. Corollary 1).
4. Call `FKTensor`$(\hat{\mathbf{A}}, \varepsilon/(2C))$ to obtain cut tensors $\mathbf{D}_1, \ldots, \mathbf{D}_s$.
Output the cut tensors $Cp \cdot \mathbf{D}_1, \ldots, Cp \cdot \mathbf{D}_s$.

The basic idea behind the algorithm `ApxTensor` for Theorem 2 is to transform the given sparse tensor $\mathbf{A}$ into a dense tensor $\hat{\mathbf{A}}$ and to apply the algorithm `FKTensor` from Theorem 5 to the latter. To obtain $\hat{\mathbf{A}}$, `ApxTensor` sets up the $|R_1 \times \cdots \times R_{k_1}|$ by $|R_{k_1+1} \times \cdots \times R_k|$ matrix $\mathbf{B}(\mathbf{A})$ as in (3). As this matrix is (C, γ)-bounded by assumption, we can apply `PartMatrix` to obtain a cut norm approximation $\mathbf{B}_{\mathcal{S}\times\mathcal{T}}$ that is constant on rectangles $S \times T$ with $S \in \mathcal{S}, T \in \mathcal{T}$ and whose entries are in $[0, 1]$. Then, $\hat{\mathbf{A}}$ is (basically) obtained by dividing $\mathbf{B}_{\mathcal{S}\times\mathcal{T}}$ by Cp.

Finally, ApxTensor applies FKTensor to $\hat{\mathbf{A}}$ to obtain cut tensors $\mathbf{D}_1, \ldots, \mathbf{D}_s$, which of course need to get scaled by a factor Cp to get the desired approximation of $\mathbf{A}$.

The key step in the analysis is to show that $Cp\hat{\mathbf{A}}$ is close to $\mathbf{A}$.

Lemma 5 *We have* $\left\|\mathbf{A} - Cp\hat{\mathbf{A}}\right\|_\square < \varepsilon \|\mathbf{A}\|_\square /2$.

Proof Let $m = \prod_{1 \le i \le k_1} |R_i|$ and $n = \prod_{k_1 < i \le k} |R_i|$. Moreover, let $\hat{\mathbf{B}}$ be the matrix defined by

$$\hat{\mathbf{B}}((i_1, \ldots, i_{k_1}), (i_{k_1+1}, \ldots, i_k)) = Cp\hat{\mathbf{A}}(i_1, \ldots, i_k).$$

Then

$$\begin{aligned}
\left\|\mathbf{A} - Cp\hat{\mathbf{A}}\right\|_\square &\le \left\|\mathbf{B} - \hat{\mathbf{B}}\right\|_\square && (13)\\
&\le \|\mathbf{B} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}\|_\square + \left\|\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square \\
&\le \frac{\varepsilon}{8}\|\mathbf{B}\|_\square + \left\|\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square && (14)\\
&= \frac{\varepsilon}{8}\|\mathbf{A}\|_\square + \left\|\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square . && (15)
\end{aligned}$$

Here (13) follows from the fact that the LHS is effectively the maximum over a subset of choices of the RHS. Equation (14) follows from our choice of $\mathcal{S}, \mathcal{T}$ and (15) follows because $\|\mathbf{A}\|_\square = \|\mathbf{B}\|_\square =$ the number of 1's in both matrices/tensors.

To bound $\left\|\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square$, observe that

$$0 \le \hat{\mathbf{B}}(i,j) \le \mathbf{B}_{\mathcal{S}\times\mathcal{T}}(i,j) \text{ for all } i \in \prod_{1\le a\le k_1} R_a,\ j \in \prod_{k_1 < b \le k} R_b. \tag{16}$$

Moreover, let S_0 be the union of all classes $S \in \mathcal{S}$ such that $|S| < \gamma m$. Similarly, let T_0 be the union of all $T \in \mathcal{T}$ of size $|T| < \gamma n$. Then

$$|S_0| \le 2^\kappa \gamma m < \varepsilon m/100, \quad |T_0| \le 2^\kappa \gamma n < \varepsilon n/100 \tag{17}$$

due to the upper bound on the number of classes in $\mathcal{S}, \mathcal{T}$ from Corollary 1 and our choice of γ. Further, we claim that

$$\hat{\mathbf{B}}(i,j) \quad = \quad \mathbf{B}_{\mathcal{S}\times\mathcal{T}}(i,j) \text{ for all } i \in \prod_{1\le a\le k_1} R_a \setminus S_0,\ j \in \prod_{k_1 < b \le k} R_b \setminus T_0. \tag{18}$$

To see this, consider $i \notin S_0$, $j \notin T_0$, and let $S \in \mathcal{S}$, $T \in \mathcal{T}$ be the classes such that $i \in S$, $j \in T$. Then by the construction of S_0, T_0 we have $|S| \ge \gamma m$, $|T| \ge \gamma n$. Therefore, the fact that $\mathbf{B}$ is (C,γ)-bounded implies that $\mathbf{B}(S,T) \le C \cdot |S \times T| \cdot p$. Hence, $\mathbf{B}_{\mathcal{S}\times\mathcal{T}}(i,j) = \mathbf{B}(S,T)/|S\times T| \le Cp$. The definiton of $\hat{\mathbf{A}}$ in Step 3 of ApxTensor yields (18).

Finally, for any two sets $X \subset \prod_{1\le ai \le k_1} R_a$, $Y \subset \prod_{k_1 < b \le k} R_b$ we obtain

$$\begin{aligned}
&\left|(\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}))(X,Y)\right| \\
&\le \left|(\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X \setminus S_0, Y \setminus T_0)\right| + \left|(\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X \cap S_0, Y)\right| + \left|(\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X, Y \cap T_0)\right| \\
&\overset{(18)}{=} \left|(\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X \cap S_0, Y)\right| + \left|(\hat{\mathbf{B}} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X, Y \cap T_0)\right| \\
&\overset{(16)}{\le} |\mathbf{B}_{\mathcal{S}\times\mathcal{T}}(X \cap S_0, Y)| + |\mathbf{B}_{\mathcal{S}\times\mathcal{T}}(X, Y \cap T_0)| \\
&\le |\mathbf{B}(X \cap S_0, Y)| + |\mathbf{B}(X, Y \cap T_0)| + 2\|\mathbf{B} - \mathbf{B}_{\mathcal{S}\times\mathcal{T}}\|_\square \\
&\le |\mathbf{B}(X \cap S_0, Y)| + |\mathbf{B}(X, Y \cap T_0)| + \frac{\varepsilon}{4}\|\mathbf{B}\|_\square . && (19)
\end{aligned}$$

As $\mathbf{B}$ is (C,γ)-bounded and $|X \cap S_0| \le |S_0| \le \varepsilon m/100$, $|Y \cap T_0| \le |T_0| \le \varepsilon n/100$ by (17), we have (see (7))

$$|\mathbf{B}(X \cap S_0, Y)|, |\mathbf{B}(X, Y \cap T_0)| \le \varepsilon \|B\|_\square /100.$$

Consequently, (19) yields $\left|(\hat{\mathbf{B}} - \mathbf{B})(X,Y)\right| \le 27\varepsilon \|\mathbf{B}\|_\square /100$. Since this is true for any X, Y, we conclude that $\left\|\hat{\mathbf{B}} - \mathbf{B}\right\|_\square < \varepsilon \|\mathbf{B}\|_\square /3$. Plugging this bound into (15) completes the proof. □

Proof of Theorem 2. The desired bound on the running time follows directly from Corollary 1 and Theorem 5. Moreover, since `FKTensor` is applied with $\delta = \varepsilon/(2C)$, Theorem 5 implies that the number of cut tensors is at most $s \le (\Gamma C/\varepsilon)^{2(k-1)}$ for a certain constant $\Gamma > 0$. Furthermore, Lemma 5 and Theorem 5 yield

$$\begin{aligned}\left\|\mathbf{A} - Cp\sum_{i=1}^{s}\mathbf{D}_i\right\|_\square &\le \left\|\mathbf{A} - Cp\hat{\mathbf{A}}\right\|_\square + Cp\cdot\left\|\hat{\mathbf{A}} - \sum_{i=1}^{s}\mathbf{D}_i\right\|_\square < \frac{\varepsilon}{2}\|\mathbf{A}\|_\square + Cp\cdot\left\|\hat{\mathbf{A}} - \sum_{i=1}^{s}\mathbf{D}_i\right\|_\square \\ &\le \frac{\varepsilon}{2}\|\mathbf{A}\|_\square + Cp\cdot\frac{\varepsilon}{2C}\prod_{i=1}^{k}|R_i| = \varepsilon\|\mathbf{A}\|_\square,\end{aligned}$$

as desired. □

4.2 Proof of Corollary 3

Let $0 < \varepsilon < \frac{1}{2}$ and $C > 1$. Moreover, let Γ be a sufficiently large constant. Suppose that $\mathbf{A} : R_1 \times \cdots \times R_k \to \{0,1\}$ is a (C,γ)-bounded $0/1$ tensor. The algorithm `PartTensor` calls `ApxTensor` to obtain cut tensors

$$\mathbf{D}_i = \mathrm{CUT}(d_i, S_{1i}, \ldots, S_{ki}) \qquad (1 \le i \le s)$$

such that $\|\mathbf{A} - \sum_{i=1}^{s}\mathbf{D}_i\|_\square \le \varepsilon\|A\|_\square /2$. Then, for each $1 \le j \le k$ `ApxTensor` outputs the coarsest partition $\mathcal{R}_j$ of R_j such that each of the sets S_{ji} is a union of classes of $\mathcal{S}_j$ ($i = 1, \ldots, s$).

Each partition $\mathcal{R}_j$ has at most 2^s classes. Hence, the upper bound on s from Theorem 2 entails the upper bound on $|\mathcal{R}_j|$ stated in Corollary 3. Moreover, bound on the running time follows from Theorem 2 as well. Hence, we finally need to show that

$$\|\mathbf{A} - \mathbf{A}_{\mathcal{R}_1\times\cdots\times\mathcal{R}_k}\|_\square \le \varepsilon\|A\|_\square.$$

To simplify the notation we let $\mathbf{D} = \sum_{i=1}^{s}\mathbf{D}_i,\ \mathbf{B} = \mathbf{A}_{\mathcal{R}_1\times\cdots\times\mathcal{R}_k}$. We know that

$$\|\mathbf{D} - \mathbf{A}\|_\square \le \varepsilon\|A\|_\square /2. \tag{20}$$

To complete the proof, we are going to show that $\|\mathbf{D} - \mathbf{B}\|_\square \le \varepsilon\|A\|_\square /2$ as well. Thus, let $X_1 \subset R_1, \ldots, X_k \subset R_k$ be sets such that

$$\|\mathbf{D} - \mathbf{B}\|_\square = |(\mathbf{D} - \mathbf{B})(X_1, \ldots, X_k)|.$$

Since both $\mathbf{D}$ and $\mathbf{B}$ are constant on any rectangle $S_1 \times \cdots \times S_k$ with $S_i \in \mathcal{R}_i$, we may assume that X_i is a union of classes of $\mathcal{R}_i$ for all $1 \le i \le k$. Furthermore, if $S_i \in \mathcal{R}_i$ for $1 \le i \le k$, then $\mathbf{B}(S_1, \ldots, S_k) = \mathbf{A}(S_1, \ldots, S_k)$. Therefore,

$$\begin{aligned}\|\mathbf{D} - \mathbf{B}\|_\square &= |(\mathbf{D} - \mathbf{B})(X_1, \ldots, X_k)| = \left|\mathbf{D}(X_1, \ldots, X_k) - \sum_{S_1\in\mathcal{R}_1 : S_1\subset X_1}\cdots\sum_{S_k\in\mathcal{R}_k : S_k\subset X_k}\mathbf{B}(S_1, \ldots, S_k)\right| \\ &= \left|\mathbf{D}(X_1, \ldots, X_k) - \sum_{S_1\in\mathcal{R}_1 : S_1\subset X_1}\cdots\sum_{S_k\in\mathcal{R}_k : S_k\subset X_k}\mathbf{A}(S_1, \ldots, S_k)\right| \\ &= |(\mathbf{D} - \mathbf{A})(X_1, \ldots, X_k)| \le \|\mathbf{D} - \mathbf{A}\|_\square \le \varepsilon\|\mathbf{A}\|_\square /2,\end{aligned}$$

by (20). Hence, $\|\mathbf{A} - \mathbf{B}\|_\square \le \|\mathbf{A} - \mathbf{D}\|_\square + \|\mathbf{D} - \mathbf{B}\|_\square \le \varepsilon\|A\|_\square$, as desired. □

5 Approximating MAX CSP problems

Throughout this section we keep the notation from Section 1.5. Given $0 < \varepsilon < \frac{1}{2}, C > 1$, we set $\gamma = \exp(-\Gamma(C/\varepsilon)^2)$, where Γ is the constant from Theorem 2. Moreover, we assume that $\mathcal{F}$ is a (C, γ)-bounded k-CSP instance on n variables $V = \{1, \dots, n\}$, where $n > n_0$ for some sufficiently large number $n_0 = n_0(C, \varepsilon, k)$. Let $m = |\mathcal{F}|$ be the number of constraints.

5.1 The algorithm **ApxCSP**

Algorithm 9 `ApxCSP`$(\mathcal{F}, C, \varepsilon)$
Input: A k-CSP instance $\mathcal{F}$ over $V = \{x_1, \dots, x_n\}$, numbers $C, \varepsilon > 0$.
Output: An assignment $\hat{\sigma} : V \to \{0, 1\}$.

1. Set up the tensors $\mathbf{A}^{\psi}_{\mathcal{F}}$ for all $\psi \in \Psi$.
   Let $\alpha = \varepsilon 2^{-2^k - 2k - 2}$.
   Call `ApxTensor`$(\mathbf{A}^{\psi}_{\mathcal{F}}, C, \alpha)$ for each $\psi \in \Psi$ to obtain tensors
   $$\mathbf{B}^{\psi} = \sum_{i=1}^{s} \mathbf{D}^{\psi}_i, \text{ where } \mathbf{D}^{\psi}_i = \text{CUT}(d^{\psi}_i, S^{\psi}_{i1}, \dots, S^{\psi}_{ik}).$$
   Let $\mathcal{P}$ be the coarsest partition of V such that each set S^{ψ}_{ih} is a union of classes of $\mathcal{P}$ ($1 \le i \le s$, $1 \le h \le k$, $\psi \in \Psi$).
2. Let $\delta = C^{-1}\Gamma^{-k}s^{-1}2^{-2^k-4k-4}$ and $\nu = \lceil \delta n \rceil$.
   Compute an optimal solution $(\hat{\tau}^{\psi}_{ih}(1), \hat{\tau}^{\psi}_{ih}(0), \hat{z}_P)_{i \in [s], h \in [k], \psi \in \Psi, P \in \mathcal{P}}$ to the following optimization problem.
   $$\begin{aligned} \text{OPT}'' = \quad \max \quad & \sum_{\psi \in \Psi} \sum_{i=1}^{s} \sum_{y \in \{0,1\}^k} d^{\psi}_i \psi(y) \nu^k \prod_{h=1}^{k} \tau^{\psi}_{ih}(y_h) \\ \text{s.t.} \quad & 0 \le \tau^{\psi}_{ih}(1) \le \lfloor |S^{\psi}_{ih}|/\nu \rfloor \quad \text{is an integer for all } 1 \le i \le s,\ 1 \le h \le k,\ \psi \in \Psi, \\ & \tau^{\psi}_{ih}(0) = \nu^{-1} S^{\psi}_{ih} - \tau^{\psi}_{ih}(1) \quad \text{for all } 1 \le i \le s,\ 1 \le h \le k,\ \psi \in \Psi \\ & \tau^{\psi}_{ih}(1)\nu \le \sum_{P \in \mathcal{P} : P \subset S^{\psi}_{ih}} z_P \le (\tau^{\psi}_{ih}(1) + 1)\nu \quad \text{for all } 1 \le i \le s,\ 1 \le h \le k,\ \psi \in \Psi \\ & 0 \le z_P \le |P| \quad \text{for all } P \in \mathcal{P}. \end{aligned}$$
   (The numbers z_P are *not* required to be integers.)
   Output an assignment $\hat{\sigma} : V \to \{0, 1\}$ such that $||\hat{\sigma}^{-1}(1) \cap P| - \hat{z}_P| \le 1$ for all $P \in \mathcal{P}$.

The first step of `ApxCSP` relies on the procedure `ApxTensor` from Theorem 2. Since we assume that all the tensors $\mathbf{A}^{\psi}_{\mathcal{F}}$ are (C, γ)-bounded, we can apply `ApxTensor` to each of them to obtain an approximation $\mathbf{B}^{\psi}$ consisting of a bounded number of cut tensors $\mathbf{D}^{\psi}_i$. The basic idea is to approximate the MAX CSP problem, i.e., the optimization problem

$$\text{OPT} = \max_{\sigma \in \{0,1\}^V} \sum_{\phi \in \mathcal{F}} \phi(\sigma) = \max_{\sigma \in \{0,1\}^V} \sum_{\psi \in \Psi} \sum_{(z_1, \dots, z_k) \in V^k} \mathbf{A}^{\psi}(z_1, \dots, z_k) \psi(\sigma(z_1), \dots, \sigma(z_k))$$

by the optimization problem

$$\text{OPT}' = \max_{\sigma \in \{0,1\}^V} \sum_{\psi \in \Psi} \sum_{(z_1, \dots, z_k) \in V^k} \mathbf{B}^{\psi}(z_1, \dots, z_k) \psi(\sigma(z_1), \dots, \sigma(z_k)).$$

The following lemma, whose proof we defer to Section 5.2, shows that any assignment σ that approximates OPT′ well also provides a good approximation for OPT.

Lemma 6 *Let $\sigma \in \{0,1\}^V$ be such that*

$$\sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^\psi(z)\psi(\sigma(z)) \geq \mathit{OPT}' - 2^{-k-1}\varepsilon m. \tag{21}$$

Then $\sum_{\phi \in \mathcal{F}} \phi(\sigma) \geq (1-\varepsilon)\mathit{OPT}(\mathcal{F})$.

It is worth pointing out that OPT′ can be solved exactly in "polynomial" time. This is because the tensors $\mathbf{B}^\psi$ consist of only a bounded (w.r.t. n) number of cut tensors. More precisely, the partition $\mathcal{P}$ constructed in Step 1 of the algorithm has the following property: if $S_1, \ldots, S_k \in \mathcal{P}$, then all the tensors $\mathbf{B}^\psi$, $\psi \in \Psi$, are constant on the rectangle $S_1 \times \cdots \times S_k$. Therefore, as far as OPT′ is concerned, the individual variables in each set $S \in \mathcal{P}$ are completely indistinguishable. More precisely, consider an assignment $\sigma : V \to \{0,1\}$ and let

$$\mathcal{Z}_P = |\{v \in P : \sigma(v) = 1\}| \text{ for each } P \in \mathcal{P}, \tag{22}$$

$$\mathcal{T}_{ih}^\psi(1) = \sum_{P \in \mathcal{P}: P \subset S_{ih}^\psi} \mathcal{Z}_P,\ \mathcal{T}_{ih}^\psi(0) = |S_{ih}^\psi| - \mathcal{T}_{ih}^\psi(1) \qquad \text{for } 1 \leq i \leq s, 1 \leq h \leq k, \psi \in \Psi. \tag{23}$$

In words, $\mathcal{T}_{ih}^\psi(y)$ is the number of variables in S_{ih}^ψ that attain the value y under σ ($y = 0,1$). Let us further define

$$\sigma(y) = (\sigma(y_1), \ldots, \sigma(y_k)) \qquad \text{for } y \in V^k.$$

Then

$$\sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^\psi(z)\psi(\sigma(z)) = \sum_{\psi \in \Psi} \sum_{i=1}^{s} \sum_{z \in \prod_{h=1}^k S_{ih}^\psi} d_i^\psi \psi(\sigma(z)) = \sum_{\psi \in \Psi} \sum_{i=1}^{s} \sum_{y \in \{0,1\}^k} d_i^\psi \psi(y) \prod_{h=1}^{k} \mathcal{T}_{ih}^\psi(y_h).$$

Hence, to solve OPT′ optimally, we could just try all possible tuples $(\mathcal{Z}_P)_{P \in \mathcal{P}}$ such that $0 \leq \mathcal{Z}_P \leq |P|$ is an integer. Since the number of such tuples is at most $n^{|\mathcal{P}|}$ and the number $|\mathcal{P}|$ of classes is independent of n, this yields a polynomial time algorithm for any fixed ε, k, C.

To speed things up, we use an idea developed in [16] for dense MAX CSP problems; this will eventually lead to the problem OPT″ detailed in Step 2 of `ApxCSP`. The basic idea is the following. Instead of optimizing over all possible $(\mathcal{Z}_P)_{P \in \mathcal{P}}$, we could just enumerate all tuples $(\mathcal{T}_{ih}^\psi(1))_{i,h,\psi}$ with $0 \leq \mathcal{T}_{ih}^\psi(1) \leq |S_{ih}^\psi|$. The issue is that not all such tuples correspond to an assignment $\sigma : V \to \{0,1\}$ as in (22) and (23). Hence, for each tuple $(\mathcal{T}_{ih}^\psi(1))_{i,h,\psi}$ we will have to check feasbility, i.e., if there is a tuple $(\mathcal{Z}_P)_P$ such that (23) holds. Since we are just aiming to solve OPT′ approximately, we can drop the requirement that all $\mathcal{Z}_P$ must be integeral. Thus, checking (23) turns into a linear programming problem. In effect, we can reduce the running time from $\exp(|\mathcal{P}| \cdot \ln n)$ to $\exp(sk2^{2^k} \cdot \ln n)$. (Remember that in general $|\mathcal{P}|$ is exponential in $sk2^{2^k}$.)

Finally, to remove the $\ln n$ factor, we chop each set S_{ih}^ψ into chunks of size $\nu = \lceil \delta n \rceil$, where $\delta > 0$ is bounded by a function of C, ε, k only. Hence, instead of optimizing over the number $0 \leq \mathcal{T}_{ih}^\psi(1) \leq |S_{ih}^\psi|$ of variables to be set to 1 in each S_{ih}^ψ, we optimize over the number $0 \leq \tau_{ih}^\psi(1) = \lfloor \mathcal{T}_{ih}^\psi(1)/\nu \rfloor \leq \lfloor S_{ih}^\psi/\nu \rfloor$ of chunks set to 1. This is sufficient because we just need to solve OPT′ within an additive $\varepsilon 2^{-k-1} m$ (cf. (21)). Of course, for each $\tau_{ih}^\psi(1)$ the number of possible values is at most $1 + \delta^{-1}$, i.e., independent of n. To check feasibility, we then have to verify that there are $0 \leq z_P \leq 1$ ($P \in \mathcal{P}$) such that $\tau_{ih}^\psi(1)\nu \leq \sum_{P \in \mathcal{P}: P \subset S_{ih}^\psi} z_P \leq (\tau_{ih}^\psi(1) + 1)\nu$ for all i, h, ψ, which is again an LP problem. This leaves us with the optimization problem OPT″ quoted in Step 2 of `ApxCSP`. After finding an optimal solution to OPT″, the algorithm sets up the assignment $\hat{\sigma}$ that mirrors the resulting z_P values. We defer the proof of the following proposition to Section 5.3

Proposition 2 *The assigment $\hat{\sigma}$ satisfies (21).*

Proof of Theorem 3. The fact that the assignment $\hat{\sigma}$ computed by `ApxCSP` satisfies at least $(1-\varepsilon)$OPT constraints follows from Lemma 6 and Proposition 2. Thus, we finally need to analyze the running time. By Theorem 2 the running time of Step 1 is at most

$$2^{2^k} 2^{(C/\alpha)^2} (C/\alpha)^{3k} \Pi'(n^k)$$

for some polynomial Π'. Moreover, for each $\psi \in \Psi$ the resulting decomposition of $\mathbf{A}^{\psi}_{\mathcal{F}}$ consists of $s \leq (\Gamma C/\alpha)^{2(k-1)}$ cut matrices for some constant $\Gamma > 0$. Hence,

$$|\mathcal{P}| \leq \Lambda = 2^{k2^{2^k}s}.$$

Step 2 solves OPT$''$ by enumerating all possible values for the integer variables $\tau^{\psi}_{ih}(1)$. The number of these integer variables is $sk2^{2^k}$. Futhermore, for each $\tau^{\psi}_{ih}(1)$ there are at most $1 + |S^{\psi}_{ih}|/\nu \leq 1 + \delta^{-1} \leq \delta^{-2}$ values to consider. Therefore, the total number of possibilities that Step 2 enumerates over is at most

$$\Lambda' = \exp(-2sk2^{2^k} \ln \delta).$$

For each of these choices we need to check the feasibility of the system of linear inequalities

$$\tau^{\psi}_{ih}(1)\nu \leq \sum_{P \in \mathcal{P}: P \subset S^{\psi}_{ih}} z_P \leq (\tau^{\psi}_{ih}(1) + 1)\nu, \quad 0 \leq z_P \leq |P| \qquad (1 \leq i \leq s,\, 1 \leq h \leq k,\, \psi \in \Psi,\, P \in \mathcal{P}).$$

This can be performed in time polynomial in the encoding length of these linear equations. There are $|\mathcal{P}| + k2^{2^k}s \leq 2\Lambda$ constraints and $|\mathcal{P}| \leq \Lambda$ variables. The encoding length of the numbers involved is at most $\ln(n/\delta)$. Therefore, the running time is

$$\mathrm{poly}\left(|\Lambda| \ln(n/\delta)\right).$$

Consequently, the total running time is at most

$$\Pi(\exp(-sk2^{2^k} \ln \delta) \cdot n^k) \leq \Pi\left(\exp(k\Gamma^k 2^{2^k}(C/\varepsilon)^{2k} \ln(C/\varepsilon)) \cdot n^k\right)$$

for some fixed polynomial Π and a constant $\Gamma > 0$. □

5.2 Proof of Lemma 6

We shall prove below that for any $\tau \in \{0,1\}^V$

$$\left| \sum_{\phi \in \mathcal{F}} \phi(\tau) - \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\tau(z)) \right| \leq 2^{-k-2}\varepsilon m. \tag{24}$$

This implies the assertion as follows. Since (24) holds for any $\tau \in \{0,1\}^V$, we have OPT$' \geq$ OPT$(\mathcal{F}) - 2^{-k-2}\varepsilon m$. Hence, if $\sigma \in \{0,1\}^V$ satisfies (21), then (24) yields

$$\sum_{\phi \in \mathcal{F}} \phi(\sigma) \quad \geq \quad \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\sigma(z)) - 2^{-k-2}\varepsilon m \geq \mathrm{OPT}' - \frac{3}{4} \cdot 2^{-k}\varepsilon m \geq \mathrm{OPT}(\mathcal{F}) - 2^{-k}\varepsilon m. \tag{25}$$

Finally, as for a random assignment $\tau \in \{0,1\}^V$ we have

$$\mathrm{E}\left[\sum_{\phi \in \mathcal{F}} \phi(\tau)\right] \geq 2^{-k}m,$$

we conclude that OPT$(\mathcal{F}) \geq 2^{-k}m$. Hence, the assertion follows from (25).

To prove (24), we fix $\tau \in \{0,1\}^V$ and let $D = \sum_{\psi \in \Psi} \left|\sum_{z \in V^k} (\mathbf{A}^{\psi} - \mathbf{B}^{\psi})(z)\psi(\tau(z))\right|$, so that

$$\left| \sum_{\phi \in \mathcal{F}} \phi(\tau) - \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\tau(z)) \right| = \left| \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{A}^{\psi}(z)\psi(\tau(z)) - \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\tau(z)) \right| \leq D. \tag{26}$$

In order to estimate D, let $T_\psi = \psi^{-1}(1) \subset \{0,1\}^k$ for any $\psi \in \Psi$. Moreover, for any $t = (t_1, \ldots, t_k) \in \{0,1\}^k$ define $R(t) = \prod_{i=1}^k \sigma^{-1}(t_i) \subset V^k$. Then

$$\left| \sum_{z \in R(t)} (\mathbf{A}^\psi - \mathbf{B}^\psi)(z) \right| \le \left\| \mathbf{A}^\psi - \mathbf{B}^\psi \right\|_\square .$$

Therefore, the bound $\left\| \mathbf{A}^\psi - \mathbf{B}^\psi \right\|_\square \le \alpha m$ entails

$$D = \sum_{\psi \in \Psi} \left| \sum_{t \in T_\psi} \sum_{z \in R(t)} (\mathbf{A}^\psi - \mathbf{B}^\psi)(z) \right| \le \sum_{\psi \in \Psi} \sum_{t \in T_\psi} \left\| \mathbf{A}^\psi - \mathbf{B}^\psi \right\|_\square \le 2^{2^k + k} \alpha m \le 2^{-k-2} \varepsilon m.$$

Thus, (24) follows from (26).

5.3 Proof of Proposition 2

Lemma 7 *We have OPT′ − OPT″* $\le 2^{-k-2}\varepsilon m$.

Proof Given an assignment $\sigma \in \{0,1\}^V$, we let $\theta_{ih}^\psi(y) = |S_{ih}^\psi \cap \sigma^{-1}(y)|$ for all ψ, i, h and $y = 0, 1$. Then

$$\begin{aligned} \sum_{\psi \in \Psi} \sum_{v \in V^k} \mathbf{B}^\psi(z) \psi(\sigma(z)) &= \sum_{\psi \in \Psi} \sum_{i=1}^s \sum_{z \in V^k} \mathbf{D}_i^\psi(z) \psi(\sigma(z)) \\ &= \sum_{\psi \in \Psi} \sum_{i=1}^s \sum_{z \in \prod_{h=1}^k S_{ih}^\psi} d_i^\psi \psi(\sigma(z)) = \sum_{\psi \in \Psi} \sum_{i=1}^s \sum_{y \in \{0,1\}^k} d_i^\psi \psi(y) \prod_{h=1}^k \theta_{ih}^\psi(y_h). \quad (27) \end{aligned}$$

We obtain a feasible solution to OPT″ by letting $z_P = |\sigma^{-1}(1) \cap P|$ for all $P \in \mathcal{P}$, $\tau_{ih}^\psi(1) = \lceil \theta_{ih}^\psi(1)/\nu \rceil$, and $\tau_{ih}^\psi(0) = \nu^{-1}|S_{ih}^\psi| - \tau_{ih}^\psi(1)$ ($1 \le i \le s$, $1 \le h \le k$, $\psi \in \Psi$). To complete the proof, we shall compare the objective function value attained by this solution with (27). To this end, observe that $|\theta_{ih}^\psi(y) - \tau_{ih}^\psi(y)\nu| \le \nu$ for all i, h, ψ, y. Therefore,

$$\left| \prod_{h=1}^k \theta_{ih}^\psi(y_h) - \prod_{h=1}^k \tau_{ih}^\psi(y_h)\nu \right| \le 2^k \nu n^{k-1} \qquad \text{for all } i, y, \psi. \quad (28)$$

Since by Theorem 2 we have $|d_i^\psi| \le Cp\Gamma^k$ for all i, ψ, (28) yields

$$\begin{aligned} \left| \sum_{\psi \in \Psi} \sum_{i=1}^s \sum_{y \in \{0,1\}^k} d_i^\psi \psi(y) \left[\prod_{h=1}^k \theta_{ih}^\psi(y_h) - \nu^k \prod_{h=1}^k \tau_{ih}^\psi(y_h) \right] \right| &\le Cp\Gamma^k \cdot s 2^{2^k + 2k} \nu n^{k-1} \\ &\le Cp\Gamma^k \cdot \delta s 2^{2^k + 2k + 1} n^k \le 2^{-k-2} \varepsilon m \quad (29) \end{aligned}$$

by our choice of δ. Finally, combining (27) and (29), we conclude that OPT″ $\ge$ OPT′ $- 2^{-k-2}\varepsilon m$, as desired. □

Proof of Proposition 2. Letting $\theta_{ih}^\psi(y) = |S_{ih}^\psi \cap \sigma^{-1}(y)|$ for all ψ, i, h and $y = 0, 1$, we have

$$\sum_{\psi \in \Psi} \sum_{v \in V^k} \mathbf{B}^\psi(z) \psi(\sigma(z)) = \sum_{\psi \in \Psi} \sum_{i=1}^s \sum_{y \in \{0,1\}^k} d_i^\psi \psi(y) \prod_{h=1}^k \theta_{ih}^\psi(y_h). \qquad \text{(cf. (27))}.$$

Furthermore, since $|z_P - |\sigma^{-1}(1) \cap P|| \le 1$, we have

$$\tau_{ih}^\psi(1)\nu \le \sum_{P \in \mathcal{P}: P \subset S_{ih}^\psi} z_P \le \sum_{P \in \mathcal{P}: P \subset S_{ih}^\psi} 1 + |\sigma^{-1}(1) \cap P| \le |\sigma^{-1}(1) \cap S_{ih}^\psi| + s \le |S_{ih}^\psi| + \nu;$$

the last inequality follows from our assumption that $n > n_0 = 2s/\delta$. Similarly,

$$(1+\tau_{ih}^{\psi}(1))\nu \quad \geq \quad \sum_{P\in\mathcal{P}:P\subset S_{ih}^{\psi}} z_P \geq \sum_{P\in\mathcal{P}:P\subset S_{ih}^{\psi}} |\sigma^{-1}(1)\cap P| - 1 \geq |\sigma^{-1}(1)\cap S_{ih}^{\psi}| - s \geq |S_{ih}^{\psi}| - \nu,$$

Hence, $|\tau_{ih}^{\psi}(y)\nu - \theta_{ih}^{\psi}(y)| \leq 2\nu$. Consequently,

$$\left|\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) - \prod_{h=1}^{k}\tau_{ih}^{\psi}(y_h)\nu\right| \leq 2^{2k}\nu n^{k-1} \qquad \text{for all } i, y, \psi. \tag{30}$$

As $|d_i^{\psi}| \leq Cp\Gamma^k$ for all i, ψ by Theorem 2, (30) yields

$$\left|\sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{y\in\{0,1\}^k} d_i^{\psi}\psi(y)\left[\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) - \nu^k\prod_{h=1}^{k}\tau_{ih}^{\psi}(y_h)\right]\right| \quad \leq \quad Cp\Gamma^k\cdot s2^{2^k+3k}\nu n^{k-1} \leq 2^{-k-2}\varepsilon m \tag{31}$$

by the definition of δ. Thus, (30) and (31) yield the assertion □

6 Examples

We present a few examples of bounded problem instances of MAX CUT and (MAX) k-SAT. The present techniques provide a unified approach to problems that were previously studied by individually tailored methods. The first two examples demonstrate how our results can be used to generalize average case analyses of algorithms. In the third instance we show how our techniques complement a prior result on "planted" random 3-SAT.

6.1 MAX CUT

Let $0 \leq p = p(n) \leq 1$ be a sequence of edge probabilities, and let $G_{n,p}$ be a random graph on n vertices $V = \{1,\ldots,n\}$ obtained by including each of the $\binom{n}{2}$ possible edges with probability p independently. We say that $G_{n,p}$ has some property $\mathcal{E}$ *with high probability* ("w.h.p.") if the probability that $\mathcal{E}$ holds tends to 1 as $n \to \infty$. For any graph G we let $\mathcal{I}(G)$ denote the set of all subgraphs H of G such that $|E(H)| \geq 0.01|E(G)|$. Furthermore, for a fixed $\varepsilon > 0$ we say that an algorithm $\mathcal{A}$ *approximates MAX CUT within* $1-\varepsilon$ *on* $G_{n,p}$*-bounded graphs* if the following two conditions are satisfied:

1. For *any* input graph G the algorithm $\mathcal{A}$ either outputs a cut that is within a $1-\varepsilon$ factor of the maximum cut, or just outputs "fail". In the first case we say that the algorithm *succeeds*, in the second case it *fails*.
2. If $G = G_{n,p}$ is a random graph, then with high probability $\mathcal{A}$ succeeds for all graphs in $\mathcal{I}(G)$.

Thus, the algorithm *never* outputs a solution that is off by more than $1-\varepsilon$, and for almost all outcomes $G = G_{n,p}$ it succeeds on all subgraphs $G_* \subset G$ that contain at least 1% of the edges of G. One can think of G_* being constructed by a malicious adversary, starting from the random graph G.

Theorem 10 *Suppose that $np \geq c_0(\varepsilon)$ for a number $c_0(\varepsilon)$ that only depends on $\varepsilon > 0$. The polynomial time algorithm* `ApxCSP` *from Theorem 3 approximates MAX CUT within $1-\varepsilon$ on $G_{n,p}$-bounded graphs.*

Proof MAX CUT fits into the general CSP framework discussed in Section 1.5 as follows. The set of variables is the vertex set of the input graph $G_* = (V, E_*)$. Moreover, each edge $e = \{v, w\} \in E_*$ yields the (binary) contraint

$$\sigma \in \{0,1\}^V \mapsto \begin{cases} 1 & \text{if } \sigma(v) \neq \sigma(w), \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the objective function value of the resulting CSP $\mathcal{F}$ is just the number of crossing edges of a maximum cut of G_*.

Let $\varepsilon > 0$, and let γ be as in Theorem 3 with $C = 360$. We claim that if $np \geq c_0(\varepsilon)$ for a sufficiently large $c_0(\varepsilon) > 0$, then **whp** $G = G_{n,p}$ has the property that for any $G_* \in \mathcal{I}(G)$ the CSP instance $\mathcal{F}$ is $(360, \gamma)$-bounded. By the construction of $\mathcal{F}$, it is sufficient to show that the adjacency matrix $A = A(G_*)$ is $(180, \gamma)$-bounded. To see this, consider any two sets $S, T \subset V$ of sizes $|S|, |T| \geq \gamma n$. Then

$$A(S,T) \leq 2e_{G_*}(S,T) \leq 2e_G(S,T).$$

Since $G = G(n,p)$ is a random graph, we have $\mathrm{E}(2e_G(S,T)) \leq 2|S \times T|p$. Moreover, as $e_G(S,T)$ is binomially distributed, Chernoff bounds entail that

$$\mathrm{P}\left[2e_G(S,T) > 3|S \times T|p\right] \leq \exp(-0.01|S \times T|p) \leq \exp(-0.01 \cdot \gamma^2 n^2 p) \leq \exp(-0.01\gamma^2 c_0(\varepsilon) \cdot n).$$

Hence, if $c_0(\varepsilon)$ is sufficiently large, then $A(S,T) \leq 3|S \times T|p$ with probability at least $1 - \exp(-2n)$. Since there are at most 2^n ways to choose S, T, the union bound entails that **whp** for all pairs of sets S, T of size at least γn we have

$$A(S,T) \leq 3|S \times T|p. \tag{32}$$

Finally, let q be the density of A. Since the number of edges of $G(n,p)$ is $(1 + o(1))\binom{n}{2}p$ **whp** (by Chernoff bounds), and since $G_* \in \mathcal{I}(G)$, we have $0.009p \leq q$ **whp**. Hence, (32) entails that $A(S,T) \leq 180|S \times T|q$ for all S, T of size at least γn **whp**, i.e., A is $(180, \gamma)$-bounded. □

Theorem 10 readily yields a result on the "planted model" for MAX CUT. In this model a random graph $G = G_{n,p,q}$ is generated by partitioning the vertex set $V = \{1, \ldots, n\}$ randomly into two parts V_1, V_2, inserting each possible V_1-V_2-edge with probability p, and each possible edge inside V_1, V_2 with probability $q < p$ independently. Improving upon prior work by Boppana [6], Coja-Oghlan [7] showed that a MAX CUT of $G_{n,p,q}$ can be computed in polynomial time **whp**, provided that $n(p - q) \geq \zeta\sqrt{np\ln(np)}$ for a certain constant $\zeta > 0$ (actually [6, 7] are stated in terms of MIN BISECTION, but things carries over to MAX CUT easily). Since the random graph $G_{n,p,q}$ can be obtained by first choosing $G_{n,p}$, then choosing a random partition (V_1, V_2), and finally removing random edges inside of V_1, V_2, Theorem 10 encompasses this model. In fact, Theorem 10 comprises various generalizations of the "planted cut" model (e.g., instead of planting a single cut, we could plant an arbitrary number of cuts, etc.).

6.2 MAX k-SAT

Let $V = \{x_1, \ldots, x_n\}$ be a set of n propositional variables, and let $F_k(n,p)$ signify a k-SAT formula obtained by including each of the $(2n)^k$ possible k-clauses with probability $0 \leq p \leq 1$ independently (hence, we think of each clause as an order k-tuple of literals). Let $m = (2n)^k p$ denote the expected number of clauses. We say that $F_k(n,p)$ has some property $\mathcal{E}$ *with high probability* if the probability that $\mathcal{E}$ holds tends to one as $n \to \infty$. Moreover, for any k-SAT formula F we let $\mathcal{I}(F)$ denote the set of all sub-formulas F_* of F that contain at least $0.01m$ clauses. Furthermore, for a fixed $\varepsilon > 0$ we say that an algorithm $\mathcal{A}$ *approximates MAX k-SAT within* $1-\varepsilon$ *on* $F_k(n,p)$*-bounded formulas* if the following two conditions are satisfied:

1. For *any* input F the algorithm $\mathcal{A}$ either outputs an assignment such that the number of satisfied clauses is within a $1 - \varepsilon$ factor of the optimum for MAX k-SAT or just outputs "fail".
2. If $F = F_k(n,p)$, then **whp** $\mathcal{A}$ succeeds on all formulas in $\mathcal{I}(F)$.

Theorem 11 *Suppose that* $k \geq 2$ *is fixed and that* $c_0(\varepsilon)n^{\lceil k/2 \rceil} \leq m = o(n^k)$ *for a number* $c_0(\varepsilon)$ *that only depends on* ε*. The polynomial time algorithm* `ApxCSP` *from Theorem 3 approximates MAX k-SAT within* $1 - \varepsilon$ *on* $F_k(n,p)$*-bounded formulas.*

Proof Let $\mathcal{F} = F_k(n,p)$ be a random k-SAT formula. Then the problem of finding an assignment that maximizes the number of simultaneously satisfied clauses can be stated as a MAX CSP problem in the sense of Section 1.5 as follows. Each clause $l_1 \vee \cdots \vee l_k$ of $\mathcal{F}$ yields Boolean function as follows. Let $s_i = 1$ if l_i is just a variable y_i, and $s_i = -1$ if l_i is the negation of a variable y_i. Then the clause yields the function

$$\sigma \in \{0,1\}^V \mapsto \max_{i=1,\ldots,k} \frac{1 + 2\sigma(y_i)s_i - s_i}{2} \in \{0,1\}.$$

Hence, for at most 2^k functions $\psi \in \Psi$ the tensor $\mathbf{A}_{\mathcal{F}}^{\psi}$ is non-zero, and each of these 2^k functions corresponds to one way of choosing the signs $(s_1, \ldots, s_k)$.

Furthermore, the tensors $\mathbf{A}_{\mathcal{F}}^{\psi}$ corresponding to a sequence $(s_1, \ldots, s_k)$ of signs are random. More precisely, for any tuple of indices $1 \le i_1 < \cdots < i_k \le n$ the entry of $\mathbf{A}_{\mathcal{F}}^{\psi}(i_1, \ldots, i_k)$ is one iff one of the $k!$ clauses corresponding to any permutation of the variables $x_{i_1} \vee \cdots \vee x_{i_k}$ occurs in $\mathcal{F}$; since we are assuming that $m = o(n^k)$, the probability that more than one of these clauses occurs is $o(1)$. Hence, the entries of $\mathbf{A}_{\mathcal{F}}^{\psi}$ are mutually independent random variables. Therefore, similarly as in the proof of Theorem 10 Chernoff bounds show that $\mathbf{A}_{\mathcal{F}}^{\psi}$ is $(1000, \gamma)$-bounded for any fixed $\gamma > 0$ **whp**. □

In particular, Theorem 11 applies to plainly random formula $F_k(n, p)$, in which case the algorithm yields a lower *and* an upper bound on the number of simultaneously satisfiable clauses. If $k \ge 3$, then for $m \ge c_0(\varepsilon) n^{\lceil k/2 \rceil}$ the optimal assignment of $F_k(n, p)$ satisfies a $1 - 2^{-k} + o(1)$ fraction of the clauses **whp** (by a standard first moment argument). Hence, w.h.p. the polynomial time algorithm `ApxSAT` yields a *proof* that there is no assignment satisfying more than a $1 - 2^{-k} + \varepsilon$ fraction of all clauses. The problem of deriving such a proof in polynomial time is known as the "strong refutation problem" for random k-SAT (cf. Feige [9]), and a number of authors have tailored algorithms specifically for this problem [8, 10, 15]. For even values of k, Theorem 10 matches the best known result [8].

6.3 Planted 3-SAT

Througout this section we let $\delta > 0$ be a sufficiently small and $\zeta > 0$ a sufficiently large constant; their precise values will be specified implicitly in due course.

Consider the following model of random 3-SAT. Let $V = \{x_1, \ldots, x_n\}$ be a set of Boolean variables, and let $L = \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ be the set of literals. Let $\vec{p} = (p_1, p_2, p_3)$ be a triple of numbers between 0 and 1. Then the random formula $F(n, \vec{p})$ is the outcome of the following experiment.

- Choose an assignment $\sigma : V \to \{0, 1\}$ uniformly at random.
- For any triple $(l_1, l_2, l_3) \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ of literals such that $i = |\{j \in \{1, 2, 3\} : \sigma(l_j) = 1\}| \ge 1$ include the clause $l_1 \vee l_2 \vee l_3$ with probaility p_i independently.

In words, $F(n, \vec{p})$ has a "planted" assignment σ, and each possible clause containing $i \ge 1$ satisfied literals under σ gets included with probability p_i independently. Considering the clauses as *ordered* triples of literals, we see that the expected total number of clauses is $n^3 p_3 + 3n^2 p_2 + 3n^2 p_1$. The following result concerning this model is due to Flaxman [14].

Theorem 12 *There is a polynomial time algorithm `SpecSAT` that satisfies the following. Assume that either $p_2 \ge \delta(p_1 + p_3)$ or $p_1 \ge (1 + \delta)p_3$ or $p_1 \le (1 - \delta)p_3$. Moreover, assume that $n^2(p_1 + 3p_2 + 3p_3) \ge \zeta$. Then `SpecSAT` applied to a random formula $\mathcal{F} = F(n, \vec{p})$ finds a satisfying assingment* **whp**.

`SpecSAT` exploits spectral properties of the "projection graph" $G(\mathcal{F})$ of a random formula $\mathcal{F} = F(n, \vec{p})$. The vertex set of the projection graph is the set of literals, and two literals l, l' are adjacent iff they occur together in a clause of $\mathcal{F}$. Hence, each clause corresponds to a triangle in $G(\mathcal{F})$. If the triple $\vec{p}$ satisfies the assumptions of Theorem 12, then the assignment σ yields a partition of $G(F)$ with one of the following properties; let T be the set of literals set to true under σ and $F = L \setminus T$.

1. The number of T–F-edges is at least $(\frac{1}{2} + \delta')|E(G(\mathcal{F}))|$.
2. The number of edges within the set T is at least $(\frac{1}{4} + \delta')|E(G(\mathcal{F}))|$.
3. The number of edges within the set F is at least $(\frac{1}{4} + \delta')|E(G(\mathcal{F}))|$.

Here $\delta' > 0$ is a number that depends only on δ. In each of the three cases, the partition $T \cup F$ of the vertex set L is reflected in spectral properties of $G(\mathcal{F})$. The algorithm `SpecSAT` exploits this spectral information to recover (a very good approximation to) the partition (T, F) and hence a satisfying assignment.

However, if $p_2 \le \delta(p_1 + p_3)$ and $(1 - \delta)p_3 \le p_1 \le (1 + \delta)p_3$, then the partition (T, F) does not stand out anymore. In fact, if $p_2 = 0$ and $p_1 = p_3$, then $G(\mathcal{F})$ is a quasi-random graph (i.e., the global edge distribution is identical to that of a uniformly random graph with the same number of edges). Hence, in this case it is not possible to recover the partition (T, F) from $G(\mathcal{F})$. Nonetheless, in the case $p_2 = 0$ and $p_1 = p_3$ it is easy to find a satisfying assignment, because then $\mathcal{F}$ is a random 3-XOR formula and thus a satisfying assignment can be found by Gaussian elimination. Of course, this trick only applies if p_2 is identically zero; if $p_2 > 0$ but $p_2 < \delta(p_1 + p_3)$, then the resulting problem is a perturbed 3-XOR formula, in which case Gaussian elimination fails. Our contribution here is an algorithm for solving $F(n, \vec{p})$ that also applies to this case.

Theorem 13 *There is a polynomial time algorithm* `Find3SAT` *that satisfies the following. Suppose that* $n^2(p_1 + 3p_2 + 3p_3) \geq \sqrt{n}\ln^{10} n$. *Then applied to a random formula* $\mathcal{F} = F(n, \vec{p})$ `Find3SAT` *yields a satisfying assignment* **whp**.

Note that Theorem 13 requires the expected number of clauses to be at least $n^{3/2}\ln^{10} n$, whereas Theorem 12 just requires ζn clauses for some constant $\zeta > 0$. The reason for this is that random "perturbed" 3-XOR formulas seem more difficult to deal with than other types of random formulas. Indeed, perturbed 3-XOR formulas play a distinguished role in the context of *refuting* the existence of a satisfying assignment for a random 3-SAT formula $F_3(n, p)$. Here p is chosen so that $F_3(n, p)$ is unsatisfiable **whp** and the goal is to *certify* in polynomial time that no satisfying assignment exists (cf. Section 6.2). Given a random formula $\mathcal{F} = F_3(n, m)$ with $m \geq \zeta n$ clauses for some large enough constant ζ, it is easy to certify in polynomial time that if $\mathcal{F}$ has a satisfying assignment τ, then τ satisfies all but δm clauses in a 3-XOR fashion (i.e., either all or exactly one literal is satisfied) [9]. But in order to refute the existence of a satisfying assignment of this type (and thus to certify that the formula has no satisfying assignment at all), the best current polynomial time algorithm requires $m \geq \zeta n^{3/2}$ [11]. In fact, techniques that allow to improve the bound in Theorem 13 to $n^{3/2-\Omega(1)}$ may very well yield improved refutation algorithms (and vice versa).

Algorithm 14 `Find3SAT`$(\mathcal{F})$
Input: A 3-SAT formula $\mathcal{F}$ over the variables $V = \{x_1, \ldots, x_n\}$ with m clauses.
Output: An assignment $\tau : V \to \{0, 1\}$.

1. If `SpecSAT`$(\mathcal{F})$ finds a satisfying assignment of $\mathcal{F}$, output this assignment and terminate.
2. Let $\mathcal{R}$ be the 4-SAT formula obtained from $\mathcal{F}$ as follows.

   If l_1, l_2, l_3, l_4 are four literals such that there is a variable z such that the clauses $l_1 \vee l_2 \vee z$ and $l_3 \vee l_4 \vee \bar{z}$ occur in $\mathcal{F}$, then include the clause $l_1 \vee l_3 \vee l_2 \vee l_4$ into $\mathcal{R}$.

   Call `ApxCSP`$(\mathcal{R}, 100, \delta)$ and let τ' be the resulting assignment. Let τ'' be the inverse of τ'. Let τ be the assignment among τ', τ'' that satisfies the larger number of clauses of $\mathcal{F}$.
3. Repeat the following $\lceil \ln n \rceil$ times.
4. For any literal λ set to false under τ compute the number ν_λ of clauses $\lambda \vee l \vee l'$ such that l, l' are literals set to true under τ. Let $\Lambda = \{\lambda : 8n\nu_\lambda > m\}$. Modify τ so that all literals in Λ get set to true.
5. Output τ if it is a satisfying assignment. Otherwise, output "fail".

In its first step `Find3SAT` calls `SpecSAT`, which finds a satisfying assignment **whp** unless $p_2 \leq \delta(p_1 + p_3)$ and $(1-\delta)p_3 \leq p_1 \leq (1+\delta)p_3$. If `SpecSAT` fails, `Find3SAT` sets up the 4-SAT formula $\mathcal{R}$ via the resolution principle. Hence, $\mathcal{R}$ is a satisfiable 4-SAT formula. Then, `Find3SAT` applies `ApxCSP` to $\mathcal{R}$. The following lemma shows that calling `ApxCSP` is feasible.

Lemma 8 *Suppose that* $p_2 \leq \delta(p_1 + p_3)$ *and* $(1-\delta)p_3 \leq p_1 \leq (1+\delta)p_3$. *For any fixed number* $\gamma > 0$ *the formula* $\mathcal{R}$ *is* $(100, \gamma)$*-bounded* **whp**

Hence, `ApxCSP` outputs an assignment τ that satisfies at least a $1 - \delta$ fraction of all clauses of $\mathcal{R}$.

Lemma 9 *Suppose that* $p_2 \leq \delta(p_1 + p_3)$ *and* $(1-\delta)p_3 \leq p_1 \leq (1+\delta)p_3$. *The assignment* τ *computed in Step 2 is within Hamming distance at most* $0.01n$ *of the planted assignment* σ **whp**

Hence, Step 2 yields an assignment that is "close" to the planted assignment **whp** Then, Steps 3 and 4 perform a local improvement operation.

Lemma 10 *Suppose that $p_2 \leq \delta(p_1+p_3)$ and $(1-\delta)p_3 \leq p_1 \leq (1+\delta)p_3$. After i iterations of Step 4 the assignment τ is at Hamming distance at most $5^{-i}n$ from the planted assignment σ.*

Finally, Theorem 13 is an immediate consequence of Theorem 12 and Lemmas 8–10.

6.3.1 Proof of Lemma 8.

Let $\mathcal{F}$ be a 3-SAT formula over V. We set up a 4-tensor $\mathbf{A}(\mathcal{F}) : L^4 \to \{0,1\}$ with entries

$$\mathbf{A}(l_1,l_2,l_3,l_4) = \begin{cases} 1 & \text{if the clause } l_1 \vee l_2 \vee l_3 \vee l_4 \text{ occurs in } \mathcal{R} \\ 0 & \text{otherwise.} \end{cases}$$

In order to show that $\mathcal{F}$ is $(100,\gamma)$-bounded **whp**, it suffices to show that

$$\mathbf{A}(\mathcal{F}(n,\vec{p})) \text{ is } (100,\gamma)\text{-bounded } \textbf{whp} \text{ for any fixed } \gamma > 0. \tag{33}$$

Let $q = p_1+p_2+p_3$, set $\vec{q} = (q,q,q)$, and let $\mathcal{F}^* = F(n,\vec{q})$. We can think of $\mathcal{F}^*$ as a random formula obtained by first choosing $\mathcal{F} = F(n,\vec{p})$, and then adding each possible clause with i satisfied literals that is not present in $\mathcal{F}$ with probability $(q-p_i)/(1-p_i)$ independently. Since the expected number of clauses in $\mathcal{F}(n,\vec{q})$ is at most three times the expected number of clauses in $F(n,\vec{p})$, the following implies (33).

$$\mathbf{A}(F(n,\vec{q})) \text{ is } (33,\gamma)\text{-bounded } \textbf{whp} \text{ for any fixed } \gamma > 0. \tag{34}$$

To show (34), we employ the following result from [8, Lemma 3.3]. We let $\vec{J}$ signify a matrix with all entries equal to one.

Lemma 11 *Let $\mathcal{F} = F(n,\vec{q})$, Let $\mathbf{B}(\mathcal{F})$ be the $(2n)^2 \times (2n)^2$ matrix constructed from $\mathbf{A}(\mathcal{F})$ as in (3). Let $Q = nq^2$. Then $\|Q\vec{J} - \mathbf{B}(\mathcal{F})\| = o(n^2 Q)$.*

Proof of (34). Let $\mathbf{B} = \mathbf{B}(F(n,\vec{q}))$. The expected number $\mathrm{E}\,\|\mathbf{B}\|_\square$ of ones in $\mathbf{B}$ is $(2n)^4 Q$. For $\mathbf{B}((l_1,l_2),(l_3,l_4))$ equals one iff there is a variable z such that both $l_1 \vee l_3 \vee z$ and $l_2 \vee l_4 \vee \bar{z}$ occur in $F(n,\vec{q})$, and since the probability of this event is q^2, and there are n ways to choose z, we have $\mathrm{P}\left[\mathbf{B}((l_1,l_2),(l_3,l_4)) = 1\right] = nq^2 = Q$. Hence, Chernoff bounds entail that $\|\mathbf{B}\|_\square \sim (2n)^4 Q$ **whp**. Consequently, the density $\hat{Q} = (2n)^{-4}\|\mathbf{B}\|_\square$ satisfies $\hat{Q} \sim Q$ **whp**.

Let $S,T \subset L \times L$ be sets of size at least γn^2. Let $\vec{1}_S \in \{0,1\}^{L\times L}$ be the indicator of S, and let $\vec{1}_T$ be the indicator of T. Then by Lemma 11

$$\begin{aligned} |Q|S\times T| - \mathbf{B}(S,T)| &= \left|\left\langle (Q\vec{J} - \mathbf{B})\vec{1}_T, \vec{1}_S \right\rangle\right| \leq \|Q\vec{J} - \mathbf{B}\| \cdot \|\vec{1}_T\| \cdot \|\vec{1}_S\| \\ &= o(n^2 Q) \cdot \sqrt{|S\times T|} = o(Q|S\times T|), \end{aligned}$$

where the last step follows from the assumption $|S|,|T| \geq \gamma n^2$. Hence, $\mathbf{B}(S,T) \sim Q|S\times T|$ **whp**. As $Q \sim \hat{Q}$, this implies that $\mathbf{B}(S,T) \sim \hat{Q}|S\times T|$, and thus $\mathbf{B}(S,T) \leq 1.01\hat{Q}|S\times T|$ **whp**. Consequently, $\mathbf{B}$ is $(1.01,\gamma)$-bounded **whp**, whence (34) follows (with room to spare). □

6.3.2 Proof of Lemma 9.

Let $\alpha = 0.01$.

Lemma 12 *W.h.p. either τ' or τ'' is within Hamming distance $\leq \alpha n$ of σ.*

Proof Without loss of generality we may assume that $\sigma(x) = 1$ for all $x \in V$. We will use a first moment argument to show that **whp** any assignment τ' that satisfies a $(1-\delta)$-fraction of the clauses of $\mathcal{R}$ has the property that either τ' or its inverse τ'' is at Hamming distance at most αn from σ. Then the assertion follows from Theorem 3.

Thus, consider any assignment τ' such that both τ' and τ'' are at Hamming distance more than αn from σ. Then there are at least αn literals l such that $\sigma(l) = 1$ and $\tau'(l) = 0$ and at least αn literals l' such that $\sigma(l') = 0$ and

$\tau'(l') = 1$. For a variable z we let ℓ_z be the set of all clauses $l_1 \vee l_2 \vee z$ in $\mathcal{F} = F(n, \vec{p})$ such that $\sigma(l_1) = \sigma(l_2) = 1$ and $\tau'(l_1) = \tau'(l_2) = 0$ $(l_1, l_2 \in L)$. Moreover, $\bar{\ell}_z$ denotes the set of all clauses $l_3 \vee l_4 \vee \bar{z}$ in $\mathcal{F}$ such that $\sigma(l_3) = 1$ and $\sigma(l_4) = \tau'(l_3) = \tau'(l_4) = 0$ $(l_3, l_4 \in L)$. Then $(|\ell_z|, |\bar{\ell}_z|)_{z \in Z}$ is a family of mutually independent binomial random variables. Moreover, for all $z \in V$

$$\mathrm{E}(|\ell_z|) \geq \alpha^2 n^2 p_3 \geq \sqrt{n} \ln^9 n, \qquad \mathrm{E}(|\bar{\ell}_z|) \geq \alpha^2 n^2 p_1 \geq \sqrt{n} \ln^9 n. \tag{35}$$

Let us call z *bad* for τ' if either $\ell_z < \mathrm{E}(|\ell_z|)/2$ or $\bar{\ell}_z < \mathrm{E}(|\bar{\ell}_z|)/2$. Then (35) implies in combination with Chernoff bounds that z is bad with probability at most $\exp(-\sqrt{n})$. Since the numbers $|\ell_z|, |\bar{\ell}_z|$ are independent for all z, this entails that with probability at least $1 - 4^{-n}$ there are at most $n/2$ bad variables. Hence, by the union bound there is no assignment τ' such that both τ' and τ'' are at Hamming distance more than αn from σ and τ' has more than $n/2$ bad variables.

Thus, suppose that there are less than $n/2$ bad variables for τ'. Consider a $z \in V$ that is not bad. Then every pair of clauses $(l_1, l_2, z) \in \ell_z$, $(l_3, l_4, z) \in \bar{\ell}_z$ yields a clause (l_1, l_3, l_2, l_4) of $\mathcal{R}$ that τ' does not satisfy. Consequently, τ' fails to satisfy at least

$$\sum_{z \in V} \ell_z \bar{\ell}_z \geq \frac{n}{2} \cdot \frac{\alpha^4}{4} n^4 p_1 p_3 = \alpha^4 n^5 p_1 p_3 / 8 \tag{36}$$

clauses. On the other hand, the expected number of clauses in $\mathcal{R}$ is at most $2n^5 p_1 p_3$ by our assumption that $p_2 < \delta(p_1 + p_3)$ and $(1 - \delta)p_3 \leq p_1 \leq (1 + \delta)p_3$. Hence, by Chernoff bounds the total number of clauses in $\mathcal{R}$ is at most $3n^5 p_1 p_3$ **whp**. Therefore, the assertion follows from the lower bound (36) on the number of clauses that are unsatisfied under τ', provided that δ is sufficiently small. □

To complete the proof of Lemma 9, we establish the following two facts.

Fact 15 *W.h.p. any assignment $\chi : V \to \{0, 1\}$ at Hamming distance at least $(1 - \alpha)n$ from the planted assignment σ fails to satisfy at least $(1 - \alpha)^4 n^3 p_3$ clauses of $\mathcal{F} = F(n, \vec{p})$.*

Proof We use a first moment argument. If χ is at Hamming distance at least $(1 - \alpha)n$ from σ, then there are at least $(1 - \alpha)n$ literals l such that $\sigma(l) = 1 - \chi(l) = 1$. Hence, the expected number of clauses $l_1 \vee l_2 \vee l_3$ occurring in $F(n, \vec{p})$ such that $\sigma(l_i) = 1 - \chi(l_i) = 1$ for $i = 1, 2, 3$ is at least $(1 - \alpha)^3 n^3 p_3 \geq (1 - \alpha)^3 n^{3/2} \ln^9 n$. Clearly, each of these clauses is unsatisfied under χ. Moreover, the number of such clauses is binomially distributed, whence Chernoff bounds entail that with probability at least $1 - 4^{-n}$ χ fails to satisfy at least $(1 - \alpha)^4 n^{3/2} \ln^9 n$ clauses. As there are only 2^n assignments $V \to \{0, 1\}$ in total, the assertion follows from the union bound. □

Fact 16 *W.h.p. any assignment $\chi : V \to \{0, 1\}$ at Hamming distance at most αn from the planted assignment σ fails to satisfy at most $6\alpha(n^3 p_3 + 3n^3 p_1 + 3n^3 p_2)$ clauses of $\mathcal{F} = F(n, \vec{p})$.*

Proof Any clause of $\mathcal{F} = F(n, \vec{p})$ that is unsatisfied under χ contains a literal l such that $\sigma(l) \neq \chi(l)$. The probability that a randomly chosen literal has this property is at most α. Since each clause contains three literals, the expected number of clauses that contain a literal l such that $\sigma(l) \neq \chi(l)$ is at most $3\alpha(n^3 p_3 + 3n^3 p_1 + 3n^3 p_2)$. As the number of such clauses is binomially distributed, with probability at least $1 - 4^{-n}$ there are at most $6\alpha(n^3 p_3 + 3n^3 p_1 + 3n^3 p_2)$ of them. Hence, the assertion follows from the union bound. □

Finally, Lemma 9 is an immediate consequence of Lemma 12 and Facts 15 and 16. □

6.3.3 Proof of Lemma 10.

We shall establish the following fact via a first moment argument. Let $\alpha = 0.01$.

Fact 17 *W.h.p. any assignment $\tau : V \to \{0, 1\}$ that is at Hamming distance $\Delta \leq \alpha n$ from σ satisfies the following.*

1. *The number of literals λ such that $\tau(\lambda) = 1 - \sigma(\lambda) = 0$ and $\nu_\lambda \leq m/(8n)$ is less than $\Delta/10$.*
2. *The number of literals λ such that $\tau(\lambda) = \sigma(\lambda) = 0$ and $\nu_\lambda > m/(8n)$ is less than $\Delta/10$.*

Since the initial assignment τ has Hamming distance at most αn from σ by Lemma 9, Fact 17 shows that the Hamming distance of τ and σ decreases by a factor of 5 in each iteration. Hence, after at most $\ln n$ iterations we have $\sigma = \tau$, as desired.

Thus, the remaining task is to establish Fact 17. Let $0 < \Delta \leq \alpha n$ and let $\tau : V \to \{0,1\}$ be any assignment such that τ and σ have Hamming distance Δ. With respect to the first item, suppose that $\tau(\lambda) = 0$ and $\sigma(\lambda) = 1$. There are at least $(1-\alpha)^2 n^2$ pairs (l, l') of literals such that $\sigma(l) = \sigma(l') = \tau(l) = \tau(l') = 1$, and for each such pair the clause $\lambda \vee l \vee l'$ is present in the random formula $\mathcal{F} = F(n, \vec{p})$ with probability p_3 independently. Hence, letting ν'_λ signify the number of such clauses, we have $\nu_\lambda \geq \nu'_\lambda$ and ν'_λ is binomially distributed with mean at least $(1-\alpha)^2 n^2 p_3 \geq \sqrt{n} \ln^9 n$. Hence, by Chernoff bounds

$$\mathrm{P}\left[\nu_\lambda < 0.99 \cdot (1-\alpha)^2 n^2 p_3\right] \leq \mathrm{P}\left[\nu'_z < 0.99 \cdot (1-\alpha)^2 n^2 p_3\right] \leq \exp(-\sqrt{n}).$$

Since the random variables ν_λ are mutually independent for all λ, the number of literals $\lambda \in \tau^{-1}(0) \cap \sigma^{-1}(1)$ such that $\nu_\lambda < 0.99 \cdot (1-\alpha)^2 n^2 p_3$ is dominated by a binomially distributed random variable with mean $\exp(-\sqrt{n})\Delta$. Hence, with probability at least $1 - n^{-3\Delta}$ the number of λ with $\tau(\lambda) = 1 - \sigma(\lambda) = 0$ and $\nu_\lambda < 0.99 \cdot (1-\alpha)^2 n^2 p_3$ is less than $\Delta/10$. Since there are $\binom{n}{\Delta}$ assignments τ at Hamming distance Δ from σ, we thus conclude that with probability at least $1 - \binom{n}{\Delta} n^{-3\Delta} \geq 1 - n^{-2}$ all of them have the property that there are at most $\Delta/10$ such λ. Finally, as m is binomially distributed with mean $n^3 p_3 + 3n^3 p_2 + 3n^3 p_1$, Chernoff bounds yield that $m \sim n^3 p_3 + 3n^3 p_2 + 3n^3 p_1$ **whp**. If this is so, then any literal λ with $\tau(\lambda)) = 1 - \sigma(\lambda) = 0$ that satisfies $\nu_\lambda \geq 0.99 \cdot (1-\alpha)^2 n^2 p_3$ also satisfies $\nu_\lambda \geq m/(8n)$, because we are assuming that $p_2 \leq \delta(p_1 + p_2)$ and $p_1 \leq (1+\delta)p_3$. This complete the proof of the first item.

Regarding the second item, we consider a literal λ such that $\tau(\lambda) = \sigma(\lambda) = 0$. If $l, l' \in \tau^{-1}(1)$ are literals such that $\lambda \vee l \vee l'$ occurs as a clause in $\mathcal{F} = F(n, \vec{p})$, then either $\sigma(l) = \sigma(l') = 1$ and $\lambda \vee l \vee l'$ has exactly two satisfied literals under σ, or σ and τ differ on exactly one of l, l'. Hence, the expected number of such clauses is at most

$$\mathrm{E}(\nu_\lambda) \leq n^2 p_2 + 2\alpha n^2 p_1 \leq 3\alpha n^2 p_1,$$

because we are assuming that $p_2 \leq \delta(p_1 + p_3)$ and $p_1 \geq (1-\delta)p_3$. Since $3\alpha n^2 p_1 \geq \sqrt{n} \ln^9 n$ and ν_λ is binomially distributed, Chernoff bounds imply that

$$\mathrm{P}\left[\nu_\lambda > 6\alpha n^2 p_1\right] \leq \exp(-\sqrt{n}).$$

Furthermore, the numbers ν_λ are mutually independent. Hence, the number of all λ with $\tau(\lambda) = \sigma(\lambda) = 0$ such that $\nu_\lambda > 6\alpha n^2 p_1$ is binomially distributed with mean $\leq n \exp(-\sqrt{n})$. Therefore, Chernoff bounds entail that with probability at least $1 - n^{-3\Delta}$ there are at most $\Delta/10$ such λ. Since the total number of assignments τ at Hamming distance Δ from σ is $\binom{n}{\Delta}$, we conclude that with probability at least $1 - n^{-2}$ for all of them there are at most $\Delta/10$ literals λ with $\tau(\lambda) = \sigma(\lambda) = 0$ such that $\nu_\lambda > 6\alpha n^2 p_1$. Finally, as $m \sim n^3 p_3 + 3n^3 p_2 + 3n^3 p_1$ **whp** and because we are assuming that $p_2 \leq \delta(p_1 + p_3)$ and $(1+\delta)p_3 \geq p_1 \geq (1-\delta)p_3$, we have $m/(8n) > 6\alpha n^2 p_1$ **whp**. Hence, there are at most $\Delta/10$ literals λ with $\tau(\lambda) = \sigma(\lambda) = 0$ and $\nu_\lambda > m/(8n)$ **whp**. □

References

[1] Alon, N., Coja-Oghlan, A., Han, H., Kang, M., Rödl, V., Schacht, M.: Quasi-randomness and algorithmic regularity for graphs with general degree distributions. Proc. 34th ICALP (2007) 789–800

[2] Alon, N., Fernadez de la Vega, W., Kannan, R., and Karpinski, M.: Random Sampling and approximation of MAX-CSPs, *J. Computer and System Sciences* **67** (2003) 212–243

[3] Alon, N., Duke, R.A., Lefmann, H., Rödl, V., Yuster, R.: The algorithmic aspects of the regularity lemma. J. of Algorithms **16** (1994) 80–109

[4] Alon, N., Naor, A.: Approximating the cut-norm via Grothendieck's inequality. Proc. 36th STOC (2004) 72–80

[5] Arora, S., Karger, D., and Karpinski, M.: Polynomial time approximation schemes for dense instances of NP-hard problems, *Journal of Computer and System Sciences* 58 (1999) 193-210.

[6] Boppana, R.: Eigenvalues and graph bisection: an average-case analysis. Proc 28th FOCS (1987) 280–285

[7] Coja-Oghlan, A.: A spectral heuristic for bisecting random graphs. Random Strucutres and Algorithms **29** (2006) 351–398

[8] Coja-Oghlan, A., Goerdt, A., Lanka, A.: Strong refutation heuristics for random k-SAT. *Combinatorics, Probability, and Computing* 16 (2007) 5–28

[9] Feige, U.: Relations between average case complexity and approximation complexity. Proc. 34th STOC (2002) 534–543

[10] Feige, U., Kim, J.H., Ofek, E.: Witnesses for non-satisfiability of dense random 3CNF formulas. Proc. 47th FOCS (2006) 497–508

[11] Feige, U., Ofek, E.,: Easily refutable subformulas of large random 3CNF formulas. Proc. 31st ICALP (2004) 519–530.

[12] Fernandez de la Vega, W.: *MAX-CUT has a Randomized Approximation Scheme in Dense Graphs*, Random Structures and Algorithms 8 (1996) 187-199.

[13] Fernandez de la Vega, W., Kannan, R., Karpinski, M., and Vempala, S.: Tensor Decomposition and Approximation Algorithms for Constraint Satisfaction Problems. Proc. 37th STOC (2005) 747–754

[14] Flaxman, A.: A spectral technique for random satisfiable 3CNF formulas. Proc. 14th SODA (2003) 357–363

[15] Friedman, J., Goerdt, A., Krivelevich, M.: Recognizing more unsatisfiable random k-SAT instances efficiently. *SIAM Journal on Computing* 35 (2006) 408–430

[16] Frieze, A.M., and Kannan, R.: Quick approximation to matrices and applications *Combinatorica* **19** (1999) 175-220.

[17] Gerke, S., Steger, A.: The sparse regularity lemma and its applications *In: Surveys in Combinatorics, 2005, B. Webb, ed.*, Cambridge University Press, 2005, 227-258.

[18] Goldreich, O., Goldwasser, S., and Ron, D.: Property testing and its connection to learning and approximation, *Journal of the ACM* 45 (1998) 653–750

[19] Gowers, T.: Lower bounds of tower type for Szemerédi's uniformity lemma. *Geometric and functional analysis* 7 (1997) 322–337

[20] Hastad, J.: Some optimal inapproximability results. Journal of the ACM 48 (2001) 798–859

[21] Khot, S., Kindler, G., Mossel, E., O'Donnell, R.: Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? Proc. 45th FOCS (2004) 146–154

[22] Kohayakawa, Y.: Szemerédi's regularity lemma for sparse graphs. In Cucker, F., Shub, M. (eds.): Foundations of computational mathematics (1997) 216–230.

[23] Rödl, V.: unpublished.

[24] Szemerédi, E.: Regular partitions of graphs, *Problémes Combinatoires et Théorie des Graphes Colloques Internationaux CNRS* 260 (1978) 399-401

[25] Trevisan, L., Sorkin, G., Sudan, M., Williamson, D.: Gadgets, approximation, and linear programming. *SIAM Journal on Computing* 29 (2000) 2074–2097