THE UNIVERSITY OF WARWICK

Original citation:
Coja-Oghlan, Amin, Cooper, Colin and Frieze, Alan. (2010) An efficient sparse regularity concept. SIAM Journal on Discrete Mathematics, Vol.23 (No.4). pp. 2000-2034. ISSN 0895-4801

Permanent WRAP url:
http://wrap.warwick.ac.uk/5351

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Copyright statement:
"First Published in SIAM Journal on Discrete Mathematics in Volume 23 and Number 4, published by the Society of Industrial and Applied Mathematics (SIAM) http://dx.doi.org/10.1137/080730160"
"Copyright © by SIAM. Unauthorized reproduction of this article is prohibited."

A note on versions:
The version presented in WRAP is the published version or, version of record, and may be cited as it appears here.

For more information, please contact the WRAP Team at: publications@warwick.ac.uk

warwickpublicationswrap

highlight your research

http://wrap.warwick.ac.uk/

SIAM J. DISCRETE MATH.
Vol. 23, No. 4, pp. 2000–2034
© 2010 Society for Industrial and Applied Mathematics

AN EFFICIENT SPARSE REGULARITY CONCEPT*

AMIN COJA-OGHLAN†, COLIN COOPER‡, AND ALAN FRIEZE§

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 [*Combinatorica*, 19 (1999), pp. 175–220] 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

AMS subject classifications. 05C85, 05C65

DOI. 10.1137/080730160

1. Introduction and results.

For many fundamental optimization problems there are known *NP-hardness of approximation* results, 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 $1 - 2^{-k}$ [21]. Furthermore, it is NP-hard to approximate MAX CUT within better than $16/17 \approx 0.94118$ [21, 26] (which can be tightened to ≈ 0.87856 under a stronger hypothesis [22]).

Frieze and Kannan [17] 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 $\mathrm{poly}(\exp((\varepsilon p)^{-2}) \cdot n)$. 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 [17] 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. 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

*Received by the editors July 14, 2008; accepted for publication (in revised form) October 12, 2009; published electronically January 6, 2010. An extended abstract of this paper appeared in *Proceedings of the Twentieth ACM-SIAM Symposium on Discrete Algorithms*, 2009, pp. 207–216.
http://www.siam.org/journals/sidma/23-4/73016.html

†University of Edinburgh, School of Informatics, 10 Crichton Street, Edinburgh EH8 9AB, UK (acoghlan@inf.ed.ac.uk). This author's research was supported by grant DFG CO 646 and was done while visiting Carnegie Mellon.

‡Department of Computer Science, King's College, University of London, London WC2R 2LS, UK (ccooper@dcs.kcl.ac.uk). This author's research was supported by Royal Society grant 2006/R2-IJP.

§Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (alan@random.math.cmu.edu). This author's research was supported in part by NSF grant CCF0502793.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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 $s \leq c\varepsilon^{-2}$ 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 $p \geq \delta$ 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 $c\varepsilon'^{-2} = c(\varepsilon p)^{-2} \leq c(\varepsilon\delta)^{-2}$ 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 decomposition guaranteed by [17] 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 as in the dense case for *arbitrary* sparse matrices. This is because in 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.

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 $|S| \geq \gamma m$, $|T| \geq \gamma n$ we have

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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

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

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* $\mathbf{D}_1, \dots, \mathbf{D}_s$ *such that* $s \leq \kappa$ *and* $\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_s)\|_\square \leq \varepsilon \|\mathbf{A}\|_\square$.

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. Also observe that, as $\mathbf{A}$ is a $0/1$ matrix, $\|\mathbf{A}\|_\square = mnp$ is just the "number of ones" in $\mathbf{A}$.

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 2. *There exist 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 $V = [n]$. Let $\mathbf{A} = \mathbf{A}(G)$ be the adjacency matrix of G. 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 $e_G(S,T) \leq C\gamma|S \times T|p$, 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 3. *There exist an algorithm* `WeakPartition` *and a polynomial* Π *that satisfy the following. Suppose that* $C > 1$ *and* $0 < \varepsilon < \frac{1}{2}$*, 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 $\mathbf{M}: R_1 \times R_2 \times \cdots \times R_k \to \mathbb{R}$, where $R_1, \dots, R_k$ are finite index sets. Moreover, extending the matrix case to k dimensions, we say that $\mathbf{C}: R_1 \times R_2 \times \cdots \times R_k \to \mathbb{R}$ is a *cut tensor* if there exist sets $S_i \subseteq R_i$ for $i = 1, 2, \dots, k$ and a real number d such that

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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

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

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 4. *There exist an algorithm* `ApxTensor`*, a polynomial* Π*, and a constant* $\Gamma > 1$ *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,* `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* $s \le (\Gamma C/\varepsilon)^{2(k-1)}$ *such that*

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

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

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 other 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 5. *There exist an algorithm* `PartTensor`*, a polynomial* Π*, and a constant* $\tilde{\Gamma} > 0$ *such that the following is true. Suppose that* $C > 0$ *and* $0 < \varepsilon < \frac{1}{2}$*. Let* $\gamma = \exp(-\tilde{\Gamma}(C/\varepsilon)^2)$*. 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* $\exp((\tilde{\Gamma}C/\varepsilon)^{2(k-1)})$ *classes. The running time is bounded by*

$$\left[\exp((\tilde{\Gamma}C/\varepsilon)^{2(k-1)}) + (\tilde{\Gamma}C/\varepsilon)^{3k}\right] \Pi\left(|R_1 \times \cdots \times R_k|\right).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

1.4. An approximation algorithm for bounded MAX CSPs. Let $V = \{x_1, \dots, x_n\}$ 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

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

We let $\Psi = \Psi_k$ be the set of all $2^{2^k} - 1$ nonzero 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 $V_\phi = \{x_{i_1}, \dots, x_{i_k}\}$ 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 $\psi(\sigma(x_{i_1}), \dots, \sigma(x_{i_k})) = \phi(\sigma)$. With this notion we can represent a k-CSP instance $\mathcal{F}$ by a family $(\mathbf{A}_{\mathcal{F}}^{\psi})_{\psi \in \Psi}$ of $2^{2^k} - 1$ k-tensors as follows. For any tuple $(i_1, \dots, i_k) \in [n]^k$ we let

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

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

THEOREM 6. *There exist 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 a* (C, γ)*-bounded* k*-CSP instance over* $V = \{x_1, \dots, x_n\}$ *for some* $n \ge n_0$, *then* **ApxCSP**$(\mathcal{F}, C, \varepsilon)$ *outputs an assignment* $\sigma : V \to \{0,1\}$ *such that*

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

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

1.5. Related work.

1.5.1. Approximating dense matrices and tensors. As mentioned earlier, Frieze and Kannan [17] dealt with *dense* matrices and tensors. More precisely, they showed that given a tensor $\mathbf{A} : R_1 \times \cdots \times R_k \to [0,1]$ and $\varepsilon > 0$ one can compute cut tensors $\mathbf{D}_1, \dots, \mathbf{D}_s$ such that $\|\mathbf{A} - \sum_{i=1}^{s} \mathbf{D}_i\|_\square < \varepsilon |R_1 \times \cdots \times R_k|$ in time $O(\varepsilon^{2(1-k)} \mathrm{polylog}(1/\varepsilon))$ 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 [13], Goldreich, Goldwasser, and Ron [19], Alon et al. [3], and de la Vega et al. [14]. Of course, if $\mathbf{A} : R_1 \times \cdots \times R_k \to \{0,1\}$ is a sparse 0/1 tensor with density $p = \|\mathbf{A}\|_\square / |R_1 \times \cdots \times R_k| = o(1)$ asymptotically as the problem size $N = |R_1 \times \cdots \times R_k|$ grows, then this sampling approach cannot yield an approximation within $\varepsilon N p$. This is because 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.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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 [17] 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 [17] apply only 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 [17]. 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 [17, section 4.1]. The only difference is that [17] 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 in the sparse case the analysis requires new ideas. For instance, additional arguments are necessary in order to bound the number of cut matrices that are needed to approximate the input matrix $\mathbf{A}$ within the desired $\varepsilon \|\mathbf{A}\|_\square$ in the cut norm.

1.5.2. Szemerédi's regularity lemma. Corollary 3 and the concept of weak regular partitions are related to Szemerédi's well-known regularity lemma [25]. While the original version [25] deals only with "dense" graphs, Kohayakawa [23] and Rödl [24] independently extended the regularity lemma to the sparse case; for a comprehensive survey on the subject see Gerke and Steger [18]. The papers [23, 24] establish that for any $\varepsilon > 0$ and any $C > 0$ there is a number γ such that 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

$$\left|\frac{e_G(S,T)}{|S \times T|} - \frac{e_G(V_i,V_j)}{|V_i \times V_j|}\right| \leq \varepsilon p, \tag{4}$$

  where p is the density of G.

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 $(C/\varepsilon)^5$:

$$\left.2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\right\} (C/\varepsilon)^5.$$

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 [20]. Moreover, the number γ required in the boundedness condition is as tiny as $\mathcal{T}((C/\varepsilon)^5)^{-1}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

While [23, 24, 25] focus on proving that a regular partition exists, [1, 2, 9] deal with algorithmic versions of the regularity lemma. In the dense case (i.e., $|E| = \Omega(n^2)$) there is a purely combinatorial algorithm [2] with running time $\mathcal{T}(\varepsilon^{-5}) \cdot \text{poly}(n)$. In addition, the paper [9] contributes an algorithm for computing a regular partition of a dense k-partite graph, provided that $k + \ln(\varepsilon) < 0$. The number of classes is bounded by $4^{\binom{k}{2}\varepsilon^{-5}}$.

An algorithm for computing a regular partition of a sparse graph was presented in [1]. The running time is $\mathcal{T}((C/\varepsilon)^9) \cdot \text{poly}(n)$ 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 $\mathcal{T}((C/\varepsilon)^9) \cdot \text{poly}(n)$.

Corollary 3 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 3 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 $\mathcal{T}((C/\varepsilon)^9)$, the running time of the algorithm `WeakPartition` from Corollary 3 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 $\gamma = 1/\mathcal{T}((C/\varepsilon)^9)$ as in [1]. Consequently, Corollary 3 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 and notation. If $\mathbf{M} = (\mathbf{M}_{ij})_{i \in [m], j \in [n]}$ 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, we set

$$\|\mathbf{M}\|_\infty = \max_{(i,j) \in [m] \times [n]} |\mathbf{M}_{ij}|.$$

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

We call $\mathcal{P}$ the partition *generated* by $\mathcal{S}$. Clearly, $\mathcal{P}$ has at most $2^{|\mathcal{S}|}$ classes. (Intuitively, $\mathcal{P}$ consists of the classes of the Venn diagram of the sets in $\mathcal{S}$.)

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

THEOREM 7. *There exist a polynomial time algorithm and a number $\alpha_0 > 0$ such that the following is true. Given an $m \times n$ matrix $\mathbf{M}$, the algorithm outputs sets $S \subset [m]$ and $T \subset [n]$ such that $|\mathbf{M}(S,T)| \geq \alpha_0 \|\mathbf{M}\|_\Box$.*

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 4 employs an algorithm `FKTensor` from Frieze and Kannan [17] as a subroutine.

THEOREM 8. *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 $\|\mathbf{M} - \mathbf{D}_1 - \cdots - \mathbf{D}_s\|_\Box \leq \delta \prod_{i=1}^k |R_i|$ 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 $(\Gamma_{FK}/\delta)^{3k} \Pi_{FK}(|R_1 \times \cdots \times R_k|)$.*

Actually Frieze and Kannan have a slightly stronger statement [17, section 6] (better running time), but the above is sufficient for our purposes and easier to state.

The following simple observation will prove useful.

LEMMA 9. *Let $\mathbf{A} : R_1 \times \cdots \times R_k \to \mathbb{R}$ be a tensor, and let $\mathcal{R}_1, \ldots, \mathcal{R}_k$ be partitions of $R_1, \ldots, R_k$. Suppose that $\mathbf{A}$ is constant on each rectangle $S_1 \times \cdots \times S_k$ with $S_1 \in \mathcal{R}_1, \ldots, S_k \in \mathcal{R}_k$, i.e.,*

$$\mathbf{A}(x) = \mathbf{A}(x') \qquad \text{for any } x, x' \in S_1 \times \cdots \times S_k. \tag{5}$$

Then there exist sets $X_1 \subset R_1, \ldots, X_k \subset R_k$ such that $|\mathbf{A}(X_1, \ldots, X_k)| = \|\mathbf{A}\|_\Box$ and each X_j is a union of classes of $\mathcal{R}_j$ $(j = 1, \ldots, k)$.

Proof. Let $X_1' \subset R_1, \ldots, X_k' \subset R_k$ be sets such that $|\mathbf{A}(X_1', \ldots, X_k')| = \|\mathbf{A}\|_\Box$. Replacing $\mathbf{A}$ by $-\mathbf{A}$ if necessary, we may assume that $\mathbf{A}(X_1', \ldots, X_k') \geq 0$. Let $S \in \mathcal{R}_1$ be a set such that $X_1' \cap S \neq \emptyset$. The assumption (5) implies that $\mathbf{A}(\{x\}, X_2', \ldots, X_k') = \mathbf{A}(\{x'\}, X_2', \ldots, X_k')$ for all $x, x' \in S$. Hence, if there were $x \in S$ such that

$$\mathbf{A}(\{x\}, X_2', \ldots, X_k') < 0,$$

then

$$\begin{aligned} \mathbf{A}(X_1' \setminus S, X_2', \ldots, X_k') &= \mathbf{A}(X_1', \ldots, X_k') - \sum_{x \in X_1' \cap S} \mathbf{A}(\{x\}, X_2', \ldots, X_k') \\ &> \mathbf{A}(X_1', \ldots, X_k') = \|\mathbf{A}\|_\Box, \end{aligned}$$

which is a contradiction. Thus, $\mathbf{A}(\{x\}, X_2', \ldots, X_k') \geq 0$ for all $x \in S$. Consequently,

$$\begin{aligned} \mathbf{A}(X_1' \cup S, X_2', \ldots, X_k') &= \mathbf{A}(X_1', \ldots, X_k') + \sum_{x \in S \setminus X_1'} \mathbf{A}(\{x\}, X_2', \ldots, X_k') \\ &\geq \mathbf{A}(X_1', \ldots, X_k') = \|\mathbf{A}\|_\Box. \end{aligned}$$

Since this holds for all $S \in \mathcal{R}_1$ such that $X' \cap S \neq \emptyset$, we see that the set $X_1 = \bigcup_{S' \in \mathcal{R}_j : X_1' \cap S \neq \emptyset} S$ satisfies

$$\mathbf{A}(X_1, X_2', \ldots, X_k') \geq \mathbf{A}(X_1', \ldots, X_k') = \|\mathbf{A}\|_\Box.$$

Clearly, this entails that actually $\mathbf{A}(X_1, X_2', \ldots, X_k') = \|\mathbf{A}\|_\Box$. Proceeding inductively, we conclude that the sets

$$X_j = \bigcup_{S \in \mathcal{R}_j : X_j' \cap S \neq \emptyset} S$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

ALGORITHM 10. 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 $|S_{j+1}| \geq m/2, |T_{j+1}| \geq n/2$ such that

$$|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_\square /4.$$

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.
5. else compute

$$d_{j+1} = \frac{\mathbf{A}_j(S_{j+1}, T_{j+1})}{|S_{j+1}||T_{j+1}|},$$

set $\mathbf{D}_{j+1} = \mathsf{CUT}(d_{j+1}, S_{j+1}, T_{j+1})$, and let $\mathbf{A}_{j+1} = \mathbf{A}_j - \mathbf{D}_{j+1}$.
6. Output "failure."

FIG. 1. *Pseudocode for ApxMatrix.*

satisfy $\mathbf{A}(X_1, X_2, \ldots, X_k) = \|\mathbf{A}\|_\square$. This yields the assertion, as $X_1, \ldots, X_k$ are unions of classes of $\mathcal{R}_1, \ldots, \mathcal{R}_k$. □

3. Approximating and partitioning 0/1 matrices and graphs. This section contains the proofs of Theorem 1 and Corollaries 2 and 3. In section 3.1 we discuss the algorithm ApxMatrix in Figure 1 and outline the proof of Theorem 1. Section 3.2 contains the proof of a proposition that is needed to establish Theorem 1. Furthermore, section 3.3 deals with the proof of Corollary 2, and section 3.4 features the proof of Corollary 3.

3.1. The algorithm ApxMatrix for Theorem 1. Let $C > 1$ and $0 < \varepsilon < \frac{1}{2}$. Moreover, let α_0 be the constant from Theorem 7, and set

$$\kappa = \frac{513C^2}{\varepsilon^2 \alpha_0^2}, \qquad \gamma = \frac{\varepsilon \alpha_0}{2^{10(\kappa+1)} C}. \tag{6}$$

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

In order to approximate $\mathbf{A}$ by a sum of cut matrices, ApxMatrix proceeds in up to $\kappa + 1$ rounds $j = 0, 1, \ldots, \kappa$, each time generating a new cut matrix $\mathbf{D}_{j+1}$. Hence, in iteration j $\mathbf{A}_j = \mathbf{A} - \sum_{i=1}^{j} \mathbf{D}_j$ is the remaining "error term" that results from approximating $\mathbf{A}$ by $\sum_{i=1}^{j} \mathbf{D}_j$. If $j = 0$, then of course $\mathbf{A}_0 = \mathbf{A}$ is just the input matrix. Thus, the goal is to eventually achieve an approximation $\mathbf{D}_1 + \cdots + \mathbf{D}_j$ such that the "error term" $\mathbf{A}_j$ has a sufficiently small cut norm (namely, cut norm less than $\varepsilon \|\mathbf{A}\|_\square$).

To this end, 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 $\mathbf{A}_j$. More precisely, we have $|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_\square /4$, where α_0 is the constant from Theorem 7. Hence, if $|\mathbf{A}_j(S_{j+1}, T_{j+1})| < \alpha_0 \varepsilon mnp/4$, then we are done because

$$\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_j)\|_\square = \|\mathbf{A}_j\|_\square \leq \varepsilon mnp = \varepsilon \|\mathbf{A}\|_\square. \tag{7}$$

Thus, in this case, step 4 terminates and outputs the cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_j$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

By contrast, if $|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \varepsilon mnp/4$, then S_{j+1}, T_{j+1} witness a set of rows/columns on which $\mathbf{D}_1 + \cdots + \mathbf{D}_j$ does not yet 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 construction ensures that $\mathbf{A}_{j+1}(S_{j+1}, T_{j+1}) = 0$ and thus remedies the discrepancy witnessed by S_{j+1}, T_{j+1}.

If the algorithm outputs cut matrices $\mathbf{D}_1, \ldots, \mathbf{D}_j$, then (7) guarantees that $\mathbf{D}_1 + \cdots + \mathbf{D}_j$ approximates $\mathbf{A}$ sufficiently well. Hence, in order to establish Theorem 1, we need to prove the following:

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

(b) If $\mathbf{A}$ is (C, γ)-bounded, then the halting condition in step 4 will be satisfied for some $1 \leq j \leq \kappa$.

The following proposition takes care of (a).

PROPOSITION 11. *In step* 3, S_{j+1}, T_{j+1} *can be computed in time* $\text{poly}(mn)$.

Proof. We apply the algorithm from Theorem 7 to the $m \times n$ matrix $\mathbf{A}_j$. The algorithm has running time $\text{poly}(nm)$ and outputs sets S'_{j+1}, T'_{j+1} such that $|\mathbf{A}_j(S'_{j+1}, T'_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_{\square}$. The problem is that Theorem 7 does not guarantee a lower bound on the sizes of these sets, while it is required that $|S_{j+1}| \geq m/2$ and $|T_{j+1}| \geq n/2$. To resolve this issue we proceed as follows.

Case 1: $|S'_{j+1}| \geq m/2$. We just let $S_{j+1} = S'_{j+1}$.

Case 2: $|S'_{j+1}| < m/2$. Since

$$\mathbf{A}_j([m], T'_{j+1}) = \mathbf{A}_j(S'_{j+1}, T'_{j+1}) + \mathbf{A}_j([m] \setminus S'_{j+1}, T'_{j+1}),$$

we have

$$\begin{aligned} \max\{|\mathbf{A}_j([m], T'_{j+1})|, |\mathbf{A}_j([m] \setminus S'_{j+1}, T'_{j+1})|\} &\geq |\mathbf{A}_j(S'_{j+1}, T'_{j+1})|/2 \\ &\geq \alpha_0 \|\mathbf{A}_j\|_{\square}/2. \end{aligned} \tag{8}$$

Let $S_{j+1} = [m]$ if $|\mathbf{A}_j([m], T'_{j+1})| \geq |\mathbf{A}_j([m] \setminus S'_{j+1}, T'_{j+1})|$, and set $S_{j+1} = [m] \setminus S'_{j+1}$ otherwise. Then (8) ensures that $|\mathbf{A}_j(S_{j+1}, T'_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_{\square}/2$ and the assumption $|S'_{j+1}| < m/2$ implies $|S_{j+1}| \geq m/2$.

In order to obtain T_{j+1} we proceed similarly.

Case 1: $|T'_{j+1}| \geq n/2$. Let $T_{j+1} = T'_{j+1}$.

Case 2: $|T'_{j+1}| < n/2$. We have

$$\begin{aligned} \max\{|\mathbf{A}_j(S_{j+1}, [n])|, |\mathbf{A}_j(S_{j+1}, [n] \setminus T'_{j+1})|\} &\geq |\mathbf{A}_j(S_{j+1}, T'_{j+1})|/2 \\ &\geq \alpha_0 \|\mathbf{A}_j\|_{\square}/4. \end{aligned}$$

Setting either $T_{j+1} = [n]$ or $T_{j+1} = [n] \setminus T'_{j+1}$ thus yields a set of size at least $n/2$ such that $|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \|\mathbf{A}_j\|_{\square}/4$.

The overall running time is clearly polynomial in $m \cdot n$. ∎

The following proposition establishes (b) above. We defer its proof to section 3.2.

PROPOSITION 12. *If* $\mathbf{A}$ *is* (C, γ)*-bounded, then there is* $1 \leq j \leq \kappa$ *such that* $|\mathbf{A}_j(S_{j+1}, T_{j+1})| < \alpha_0 \varepsilon mnp/4$.

Proof of Theorem 1. Proposition 11 ensures that each iteration of steps 3–5 runs in time $\Pi(mn)$ for some polynomial Π. Hence, the total running time of `ApxMatrix` is bounded by $\kappa \cdot \Pi(mn)$, as claimed. Furthermore, Proposition 12 ensures that on a (C, γ)-bounded input $\mathbf{A}$ `ApxMatrix` will output a sequence $\mathbf{D}_1, \ldots, \mathbf{D}_j$ of cut matrices for some $1 \leq j \leq \kappa$. Finally, (7) entails that this sequence satisfy $\|\mathbf{A} - (\mathbf{D}_1 + \cdots + \mathbf{D}_j)\|_{\square} \leq \varepsilon \|\mathbf{A}\|_{\square}$, as desired. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

3.2. Proof of Proposition 12. Throughout this section we assume that $\mathbf{A}$ is a (C,γ)-bounded $m \times n$ matrix. We let κ, γ be as in (6) and set $\gamma' = 2^\kappa \gamma$. The proof is by contradiction. That is, we assume that

$$|\mathbf{A}_j(S_{j+1}, T_{j+1})| \geq \alpha_0 \varepsilon mnp/4 \qquad \text{for all } 0 \leq j \leq \kappa. \tag{9}$$

We are going to construct families $(\mathbf{A}'_j)_{1\leq j \leq \kappa}$ and $(\mathbf{D}'_j)_{1\leq j\leq\kappa}$ of matrices such that the matrices $\mathbf{D}'_j$, $\mathbf{A}'_j$ are "close" to $\mathbf{D}_j$, $\mathbf{A}_j$ in the cut norm and such that we can use the boundedness condition to derive upper and lower bounds on the Frobenius norms of $\mathbf{A}'_j$ for $1 \leq j \leq \kappa$. These bounds on the Frobenius norm will then yield a contradiction to (9).

The matrices $\mathbf{D}'_j$, $\mathbf{A}'_j$ are defined as follows. Due to assumption (9), step 4 of `ApxMatrix` does not terminate the algorithm for any $j \leq \kappa$. Hence, steps 2–5 construct sets $S_1, \ldots, S_\kappa$ of rows and $T_1, \ldots, T_\kappa$ of columns. Let $\mathcal{S}$ be the partition of the set $[m]$ of row indices generated by $S_1, \ldots, S_\kappa$. Similarly, let $\mathcal{T}$ be the partition of the column set $[n]$ generated by $T_1, \ldots, T_\kappa$. Then both $\mathcal{S}$ and $\mathcal{T}$ have at most 2^κ classes. We define

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

to be the sets that comprise the "small" classes of the partitions $\mathcal{S}, \mathcal{T}$.

FACT 13. *We have $|R_0| \leq \gamma' m$ and $|C_0| \leq \gamma' n$.*

Proof. The definition of R_0 ensures that $|R_0| \leq |\mathcal{S}| \cdot \gamma m$. Since $\mathcal{S}$ has at most 2^κ classes, we obtain $|R_0| \leq 2^\kappa \gamma m = \gamma' m$. Similarly, $|C_0| \leq 2^\kappa \gamma n = \gamma' n$. ∎

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 $S'_j = S_j \setminus R_0$ and $T'_j = T_j \setminus C_0$ 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 partition of $[m] \setminus R_0$ generated by $S'_1, \ldots, S'_\kappa$, and let $\mathcal{T}'$ be the partition of $[n] \setminus C_0$ generated by $T'_1, \ldots, T'_\kappa$.

FACT 14. *All classes of $\mathcal{S}'$ (resp., $\mathcal{T}'$) have size at least γm (resp., γn).*

Proof. Let $\mathcal{S}''$ be the partition of $[m] \setminus R_0$ that consists of all classes $S \in \mathcal{S}$ such that $S \subset [m] \setminus R_0$. Then each class of $\mathcal{S}''$ has size at least γm, because R_0 contains all classes of $\mathcal{S}$ that are smaller than γm. Moreover, each of the sets $S'_1, \ldots, S'_\kappa$ is a union of classes of $\mathcal{S}''$. Hence, $\mathcal{S}'$ is coarser than $\mathcal{S}''$, and thus each class of $\mathcal{S}'$ contains a class of $\mathcal{S}''$. Therefore, each class of $\mathcal{S}'$ has size at least γm. The same argument applies to $\mathcal{T}'$. ∎

The key step is to derive the following bound on the Frobenius norm of $\mathbf{A}'_j$.

LEMMA 15. *For all $1 \leq j \leq \kappa$ we have $\|\mathbf{A}'_j\|_F^2 \leq \|\mathbf{A}'\|_F^2 (1 - j \cdot \alpha_0^2 \varepsilon^2 p/256)$.*

The proof of Lemma 15 requires some preparations: we need to bound the cut norms $\|\mathbf{A} - \mathbf{A}'\|_\square$ (Lemma 16), $\|\mathbf{D}_j - \mathbf{D}'_j\|_\square$ (Corollary 18), and $\|\mathbf{A}_j - \mathbf{A}'_j\|_\square$ (Corollary 19).

LEMMA 16. *We have $\|\mathbf{A} - \mathbf{A}'\|_\square \leq \mathbf{A}(R_0, [n]) + \mathbf{A}([m], C_0) \leq 2C\gamma' mnp$.*

Proof. Both $\mathbf{A}$ and $\mathbf{A}'$ are 0/1 matrices, and $\mathbf{A}'$ is obtained from $\mathbf{A}$ by replacing

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the rows R_0 and the columns C_0 by 0. Therefore, $\mathbf{A}-\mathbf{A}'$ is a 0/1 matrix, whose cut norm equals the number of ones it contains. Hence,

$$\begin{aligned}\|\mathbf{A}-\mathbf{A}'\|_{\square} &= (\mathbf{A}-\mathbf{A}')([m],[n]) = \mathbf{A}(R_0,[n]) + \mathbf{A}([m],C_0) - \mathbf{A}(R_0,C_0)\\ &\le \mathbf{A}(R_0,[n]) + \mathbf{A}([m],C_0).\end{aligned}$$

To show that $\mathbf{A}(R_0,[n]) \le C\gamma' mnp$ we consider two cases. Recall that we are assuming that $\mathbf{A}$ is (C,γ)-bounded.

Case 1: $|R_0| \ge \gamma m$. Because the boundedness condition implies

$$\mathbf{A}(R_0,[n]) \le C|R_0|np,$$

Fact 13 entails $\mathbf{A}(R_0,[n]) \le C\gamma' mnp$.

Case 2: $|R_0| < \gamma m$. Let $R_0 \subset R_0' \subset [m]$ be a superset of R_0 of size $\lceil \gamma m \rceil$. Since $\mathbf{A}$ is a 0/1 matrix, we have $\mathbf{A}(R_0,[n]) \le \mathbf{A}(R_0',[n])$. Moreover, as $|R_0'| \ge \gamma m$, we can apply the boundedness condition to get $\mathbf{A}(R_0',[n]) \le C|R_0'|np \le C\gamma' mnp$, as desired.

The same argument yields $\mathbf{A}([m],C_0) \le C\gamma' mnp$ and thus the assertion. ∎

To show that the matrices $\mathbf{D}_j, \mathbf{D}_j'$ are close in the cut norm, we need a bound on the coefficients d_j from step 5 of `ApxMatrix`.

LEMMA 17. *$|d_j| \le 2^j Cp$ for all $1 \le j \le \kappa$.*

Proof. The proof is by induction on j. Since the matrix $\mathbf{A} = \mathbf{A}_0$ is (C,γ)-bounded and $|S_0| \ge \frac{m}{2}$ and $|T_0| \ge \frac{n}{2}$ (cf. step 3), we have $\mathbf{A}_0(S_1,T_1) \le C|S_0||T_0|p$. Hence,

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

Furthermore, assuming that $|d_i| \le 2^i Cp$ for all $i \le j$, we obtain

$$\begin{aligned}|\mathbf{A}_j(S_{j+1},T_{j+1})| &= \left|\mathbf{A}_0(S_{j+1},T_{j+1}) - \sum_{i=1}^{j} \mathbf{D}_i(S_{j+1},T_{j+1})\right|\\ &= \left|\mathbf{A}_0(S_{j+1},T_{j+1}) - \sum_{i=1}^{j} d_i |S_{j+1}\cap S_i|\,|T_{j+1}\cap T_i|\right|\\ &\le |\mathbf{A}_0(S_{j+1},T_{j+1})| + |S_{j+1}||T_{j+1}|\sum_{i=1}^{j}|d_i| \qquad \text{[triangle inequality]}\\ &\le |\mathbf{A}_0(S_{j+1},T_{j+1})| + |S_{j+1}||T_{j+1}|\sum_{i=1}^{j}2^i Cp \qquad \text{[by induction]}\\ &\le |\mathbf{A}_0(S_{j+1},T_{j+1})| + (2^{j+1}-1)|S_{j+1}||T_{j+1}|Cp. \end{aligned} \tag{10}$$

As $\mathbf{A}_0$ is (C,γ)-bounded and $|S_{j+1}| \ge \frac{m}{2}$ and $|T_{j+1}| \ge \frac{n}{2}$ by the construction in step 3, we have the bound $|\mathbf{A}_0(S_{j+1},T_{j+1})| \le Cp|S_{j+1}||T_{j+1}|$. Thus, (10) yields $|d_{j+1}| = |\mathbf{A}_j(S_{j+1},T_{j+1})|/(|S_{j+1}||T_{j+1}|) \le 2^{j+1}Cp$. ∎

COROLLARY 18.

1. *For all $1 \le j \le \kappa$ we have $\left\|\mathbf{D}_j - \mathbf{D}_j'\right\|_{\square} \le 2^{8j} C\gamma' mnp$.*
2. *For any $2 \le j \le \kappa$ we have $\left|\mathbf{A}_{j-1}(S_j',T_j') - \mathbf{A}_{j-1}(S_j,T_j)\right| \le 2^{j+1} C\gamma' mnp$.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Proof. We prove the first assertion by induction on j. The definitions of d_j and d'_j imply that

$$
\begin{aligned}
|d'_j - d_j| &= \left| \frac{\mathbf{A}'_{j-1}(S'_j, T'_j)}{|S'_j||T'_j|} - \frac{\mathbf{A}_{j-1}(S_j, T_j)}{|S_j||T_j|} \right| \\
&= \frac{\big| |S_j||T_j|\mathbf{A}'_{j-1}(S'_j, T'_j) - |S'_j||T'_j|\mathbf{A}_{j-1}(S_j, T_j) \big|}{|S_j||S'_j||T_j||T'_j|} \\
&\le \frac{\big|\mathbf{A}_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S_j, T_j)\big|}{|S'_j||T'_j|} + \frac{\big|(|S_j||T_j| - |S'_j||T'_j|)\mathbf{A}_{j-1}(S_j, T_j)\big|}{|S_j||S'_j||T_j||T'_j|} \\
&\qquad + \frac{\big|\mathbf{A}'_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S'_j, T'_j)\big|}{|S'_j||T'_j|}. \qquad (11)
\end{aligned}
$$

To bound the denominators, remember from step 3 that $|S_j| \ge \frac{m}{2}$ and $|T_j| \ge \frac{n}{2}$. Furthermore, as $S'_j = S_j \setminus R_0$ and $|R_0| \le \gamma' m < m/4$ by Fact 13, we have $|S'_j| \ge \frac{m}{4}$. Similarly, $|T'_j| \ge \frac{n}{4}$. Hence, (11) yields

$$
\begin{aligned}
|d'_j - d_j| \le\, & 16(mn)^{-1}(\big|\mathbf{A}_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S_j, T_j)\big|) \\
& + 64(mn)^{-2}|\mathbf{A}_{j-1}(S_j, T_j)|(|R_0||C_0| + |R_0||T_j| + |S_j||C_0|) \\
& + 16(mn)^{-1}\left|\mathbf{A}'_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S'_j, T'_j)\right|. \qquad (12)
\end{aligned}
$$

To start the induction, we evaluate the term on the right-hand side (r.h.s.) of (12) for $j = 1$. Since $S'_1 = S_1 \setminus R_0$ and $T'_1 = T_1 \setminus C_0$, and as $\mathbf{A}'_0$ is obtained from $\mathbf{A}' = \mathbf{A}$ by replacing the rows R_0 and the columns C_0 by 0, we have $\mathbf{A}'_0(S'_j, T'_j) = \mathbf{A}_0(S'_j, T'_j)$. Hence, the third term on the r.h.s. of (12) vanishes. Moreover, as $\mathbf{A}_0$ is a 0/1 matrix, we have

$$
\begin{aligned}
\mathbf{A}_0(S_1, T_1) - \mathbf{A}_0(S'_1, T'_1) &= \mathbf{A}_0(S_1 \cap R_0, T_1) + \mathbf{A}_0(S_1, T_1 \cap C_0) \\
&\quad - \mathbf{A}_0(S_1 \cap R_0, T_1 \cap C_0) \\
&\le \mathbf{A}([m], C_0) + \mathbf{A}(R_0, [n]) \le 2C\gamma' mnp \quad \text{[by Lemma 16].}
\end{aligned}
$$

Hence,

$$16(mn)^{-1}(\big|\mathbf{A}_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S_j, T_j)\big|) \le 32\gamma' Cp. \qquad (13)$$

Further, $\mathbf{A}_0(S_1, T_1) \le \mathbf{A}_0([m], [n]) = mnp$. As $|R_0| \le \gamma' m$ and $|C_0| \le \gamma' n$ by Fact 13, we get

$$64(mn)^{-2}|\mathbf{A}_{j-1}(S_j, T_j)|(|R_0||C_0| + |R_0||T_j| + |S_j||C_0|) \le 64\gamma'(2+\gamma')p \le 192\gamma' p. \qquad (14)$$

Plugging (13) and (14) into (12), we get $|d'_1 - d_1| \le C\gamma' p(32 + 192) = 224C\gamma' p$. Consequently,

$$\|\mathbf{D}_1 - \mathbf{D}'_1\|_\square \le |d_1 - d'_1| mn \le 2^8 C\gamma' mnp,$$

as claimed.

Now let $2 \le j \le \kappa$, and assume that

$$\|\mathbf{D}_i - \mathbf{D}'_i\|_\square \le 2^{8i} C\gamma' mnp \qquad \text{for } 1 \le i \le j-1. \qquad (15)$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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

$$\begin{aligned}
|\mathbf{A}_{j-1}(S,T)| &= \left|\mathbf{A}(S,T) + \sum_{i=1}^{j-1} \mathbf{D}_i(S,T)\right| \\
&\le \mathbf{A}(S,T) + \sum_{i=1}^{j-1} |\mathbf{D}_i(S,T)| \qquad \text{[triangle inequality]} \\
&\le \mathbf{A}(S,T) + \sum_{i=1}^{j-1} |d_i||S||T| \\
&\le \mathbf{A}(S,T) + \sum_{i=1}^{j-1} 2^i Cp \cdot |S||T| \qquad \text{[by Lemma 17]} \\
&\le \mathbf{A}(S,T) + (2^j - 1)Cp|S||T|. \qquad (16)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\left|\mathbf{A}_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S_j, T_j)\right| \\
&\quad = |\mathbf{A}_{j-1}(S_j, T_j \cap C_0) + \mathbf{A}_{j-1}(S_j \cap R_0, T_j) - \mathbf{A}_{j-1}(S_j \cap R_0, T_j \cap C_0)| \\
&\quad \le |\mathbf{A}_{j-1}(S_j, T_j \cap C_0)| + |\mathbf{A}_{j-1}(S_j \cap R_0, T_j)| + |\mathbf{A}_{j-1}(S_j \cap R_0, T_j \cap C_0)| \\
&\quad \le \mathbf{A}(S_j, T_j \cap C_0) + \mathbf{A}(S_j \cap R_0, T_j) + \mathbf{A}(S_j \cap R_0, T_j \cap C_0) \\
&\qquad\quad + (2^j - 1)Cp(|S_j||T_j \cap C_0| + |S_j \cap R_0||T_j| + |S_j \cap R_0||T_j \cap C_0|) \\
&\quad \le \mathbf{A}(R_0, [n]) + \mathbf{A}([m], C_0) + (2^j - 1)Cp(|R_0|n + m|C_0|). \qquad (17)
\end{aligned}$$

Since $\mathbf{A}(R_0, [n]) + \mathbf{A}([m], C_0) \le 2C\gamma' mnp$ by Lemma 16 and $|R_0| \le \gamma' m$, $|C_0| \le \gamma' n$ by Fact 13, (17) yields

$$\left|\mathbf{A}_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S_j, T_j)\right| \le 2^{j+1}\gamma' Cmnp. \qquad (18)$$

Moreover, as $\mathbf{A}$ is a 0/1 matrix, (16) yields

$$\begin{aligned}
|\mathbf{A}_{j-1}(S_j, T_j)| &\le \mathbf{A}(S_j, T_j) + (2^j - 1)Cp|S_j||T_j| \\
&\le \mathbf{A}([m], [n]) + (2^j - 1)Cmnp \le 2^j Cmnp. \qquad (19)
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\left|\mathbf{A}'_{j-1}(S'_j, T'_j) - \mathbf{A}_{j-1}(S'_j, T'_j)\right| &= \left|\sum_{i=1}^{j-1} \mathbf{D}'_i(S'_j, T'_j) - \mathbf{D}_i(S'_j, T'_j)\right| \\
&\le \sum_{i=1}^{j-1} \left|\mathbf{D}'_i(S'_j, T'_j) - \mathbf{D}_i(S'_j, T'_j)\right| \\
&\le \sum_{i=1}^{j-1} \|\mathbf{D}'_i - \mathbf{D}_i\|_{\square} \\
&\le C\gamma' mnp \sum_{i=1}^{j-1} 2^{8j} \quad \text{[by (15)]} \\
&\le 2^{8j-7} C\gamma' mnp. \qquad (20)
\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Plugging the bounds (18)–(20) into (12), we obtain $|d'_j - d_j| \le 2^{8j} C\gamma' p$. Hence, $\left\|\mathbf{D}'_j - \mathbf{D}_j\right\|_\square \le 2^{8j} C\gamma' mnp$. This completes the proof of the first assertion. The second one follows directly from (18). ∎

COROLLARY 19. *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. Using the triangle inequality, Lemma 16, and Corollary 18, we obtain

$$\begin{aligned}\left\|\mathbf{A}'_j - \mathbf{A}_j\right\|_\square &= \left\|\left(\mathbf{A}_0 - \sum_{i=1}^{j}\mathbf{D}_i\right) - \left(\mathbf{A}'_0 - \sum_{i=1}^{j}\mathbf{D}'_i\right)\right\|_\square \\ &\le \|\mathbf{A}_0 - \mathbf{A}'_0\|_\square + \sum_{i=1}^{j}\|\mathbf{D}_i - \mathbf{D}'_i\|_\square \\ &\le 2C\gamma' mnp + \sum_{i=1}^{j} 2^{8j} C\gamma' mnp \le 2^{8j+1} C\gamma' mnp,\end{aligned}$$

as claimed. ∎

Proof of Lemma 15. We are going to show that

$$\|\mathbf{A}'_{j+1}\|_F^2 \le \|\mathbf{A}'_j\|_F^2 - \alpha_0^2\varepsilon^2 mnp^2/256 \tag{21}$$

for any $1 \le j < \kappa$. This bound readily implies the assertion. Remember that we are assuming that $|\mathbf{A}_j(S_{j+1}, T_{j+1})| \ge \alpha_0\varepsilon mnp/4$ for all $1 \le j \le \kappa$. Therefore, by Corollary 19

$$\begin{aligned}|\mathbf{A}'_j(S'_{j+1}, T'_{j+1})| &\ge |\mathbf{A}_j(S_{j+1}, T_{j+1})| - |\mathbf{A}_j(S_{j+1}, T_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})| \\ &\quad - |\mathbf{A}'_j(S'_{j+1}, T'_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})| \\ &\ge \alpha_0\varepsilon mnp/4 - |\mathbf{A}_j(S_{j+1}, T_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})| - \left\|\mathbf{A}'_j - \mathbf{A}_j\right\|_\square \\ &\ge \left(\varepsilon\alpha_0/4 - 2^{8j+1}C\gamma'\right) mnp \\ &\quad - |\mathbf{A}_j(S_{j+1}, T_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})|.\end{aligned} \tag{22}$$

The second part of Corollary 18 yields the bound $|\mathbf{A}_j(S_{j+1}, T_{j+1}) - \mathbf{A}_j(S'_{j+1}, T'_{j+1})| \le 2^{j+2} C\gamma' mnp$. Plugging this into (22), we obtain

$$\begin{aligned}|\mathbf{A}'_j(S'_{j+1}, T'_{j+1})| &\ge \left(\varepsilon\alpha_0/4 - 2^{8j+1}C\gamma' - 2^{j+2}C\gamma'\right) mnp \\ &\ge \left(\varepsilon\alpha_0/4 - 2^{8j+3}C\gamma'\right) mnp \\ &\ge \left(\varepsilon\alpha_0/4 - 2^{9\kappa+3}C\gamma\right) mnp \quad [\text{as } \gamma' = 2^\kappa\gamma] \\ &\ge \varepsilon\alpha_0 mnp/8 \quad \left[\text{because } \gamma = \tfrac{\varepsilon\alpha_0}{2^{10(\kappa+1)}C}\text{; cf. (6)}\right].\end{aligned} \tag{23}$$

As $d'_{j+1} = \frac{\mathbf{A}'_j(S'_{j+1}, T'_{j+1})}{|S'_{j+1}||T'_{j+1}|}$ by the construction in step 5, (23) implies that

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

whence (21) follows. ∎

While Lemma 15 provides an upper bound on $\|\mathbf{A}'_j\|_F$, the following lemma yields a lower bound.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

LEMMA 20. *For all* $1 \le j \le \kappa$ *we have* $\|\mathbf{A}'_j\|_F^2 \ge \|\mathbf{A}'\|_F^2 (1 - 2C^2 p)$.

Proof. Let $\mathbf{M} = \sum_{i=1}^{j} \mathbf{D}'_j$. Recall that $\mathcal{S}'$ is the partition generated by $S_1, \ldots, S_\kappa$ and that $\mathcal{T}'$ is the partition generated by $T_1, \ldots, T_\kappa$. 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,

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

Since $\mathbf{A}'$ is (C,γ) bounded and because $|S| \ge \gamma m$, $|T| \ge \gamma n$ by Fact 14, we get $m^*_{S\times T} \le Cp$. Hence,

$$\|\mathbf{A}'_j\|_F^2 \ge \|\mathbf{A}'\|_F^2 - (Cp)^2 mn. \tag{24}$$

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

$$\begin{aligned}\|\mathbf{A}\|_F^2 - \|\mathbf{A}'\|_F^2 &= \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{A}_{ij}^2 - {\mathbf{A}'_{ij}}^2) = \mathbf{A}(R_0, [n]) + \mathbf{A}([m], C_0) - \mathbf{A}(R_0, C_0) \\ &\le \mathbf{A}(R_0, [n]) + \mathbf{A}([m], C_0) \overset{\text{Fact 13}}{\le} 2C\gamma' mnp < mnp/2,\end{aligned}$$

whence $\|\mathbf{A}'\|_F^2 \ge \|\mathbf{A}\|_F^2/2 = mnp/2$. Thus, the assertion follows from (24). ◻

Proof of Proposition 12. Under assumption (9) we have derived the bounds

$$\|\mathbf{A}'\|_F^2 (1 - 2C^2 p) \le \|\mathbf{A}'_j\|_F^2 \le \|\mathbf{A}'\|_F^2 (1 - j \cdot \alpha_0^2 \varepsilon^2 p/256)$$

for all $1 \le j \le \kappa$. (cf. Lemmas 20 and 15). But for $j = \kappa = \frac{513C^2}{\varepsilon^2 \alpha_0^2}$ the upper bound is smaller than the lower bound. This contradiction refutes (9). Thus, there is $0 \le j < \kappa$ such that $|\mathbf{A}_j(S_{j+1}, T_{j+1})| < \alpha_0 \varepsilon mnp/4$, as claimed. ◻

3.3. Proof of Corollary 2. 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, the algorithm `PartMatrix` is supposed to produce partitions $\mathcal{S}$ of $[m]$ and $\mathcal{T}$ of $[n]$ such that the matrix $\mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ approximates $\mathbf{A}$ well in the cut norm. To this end, `PartMatrix` calls `ApxMatrix`$(\mathbf{A}, C, \varepsilon)$ to obtain cut matrices

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

for some $1 \leq s \leq \kappa$. Then, it computes the partition $\mathcal{S}$ of $[m]$ generated by $S_1, \ldots, S_s$ and the partition $\mathcal{T}$ of $[n]$ generated by $T_1, \ldots, T_s$.

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. To complete the proof of Corollary 2 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

$$\|\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq \|\mathbf{D} - \mathbf{A}\|_\square . \tag{25}$$

To prove (25), we use the same argument as in [17, section 5]. Note that on each rectangle $S \times T$ with $S \in \mathcal{S}$ and $T \in \mathcal{T}$ the matrix $\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ is constant. Therefore, by Lemma 9 there exist a set $X \subset [m]$ that is a union of classes of $\mathcal{S}$ and a set $Y \subset [n]$ that is a union of classes of $\mathcal{T}$ such that

$$\|\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square = |(\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}})(X, Y)| .$$

Hence, we obtain

$$\begin{aligned} \|\mathbf{D} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square &= |\mathbf{D}(X, Y) - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}(X, Y)| \\ &= \left| \sum_{S \in \mathcal{S}: S \subset X} \sum_{T \in \mathcal{T}: T \subset Y} \mathbf{D}(S, T) - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}(S, T) \right| . \end{aligned} \tag{26}$$

The definition of $\mathbf{A}_{\mathcal{S}\times\mathcal{T}}$ ensures that $\mathbf{A}(S, T) = \mathbf{A}_{\mathcal{S}\times\mathcal{T}}(S, T)$ for any $S \in \mathcal{S}$ and $T \in \mathcal{T}$. Therefore, (26) yields

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

as desired. □

3.4. Proof of Corollary 3. 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 $\|\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}\|_\square \leq 2\varepsilon \|\mathbf{A}\|_\square$. By Corollary 2 both $\mathcal{S}, \mathcal{T}$ have at most 2^κ classes. Suppose the classes of $\mathcal{S}$ are $S_1, \ldots, S_s$ and those of $\mathcal{T}$ are $T_1, \ldots, T_t$. The algorithm outputs the partition $\mathcal{V}$ of V generated by $S_1, \ldots, S_s, T_1, \ldots, T_t$. This partition $\mathcal{V}$ consists of all nonempty intersections $S_i \cap T_j$ $(1 \leq i \leq s,\ 1 \leq j \leq t)$. Hence, $|\mathcal{V}| \leq 2^{2\kappa}$, and the running time 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 2, we just need to prove that

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

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 $S, T \in \mathcal{V}$, Lemma 9 entails that there exist sets $X, Y \subset V$ that are unions of classes of $\mathcal{V}$ such that $|(\mathbf{A}_{\mathcal{V}\times\mathcal{V}} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}})(X, Y)| = \|\mathbf{A}_{\mathcal{V}\times\mathcal{V}} - \mathbf{A}_{\mathcal{S}\times\mathcal{T}}\|_\square$. Hence,

$$\begin{aligned} \|\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}_{\mathcal{V}\times\mathcal{V}}\|_\square &= |(\mathbf{A}_{\mathcal{S}\times\mathcal{T}} - \mathbf{A}_{\mathcal{V}\times\mathcal{V}})(X, Y)| \\ &= \left| \sum_{S, T \in \mathcal{V}: S \times T \subset X \times Y} \mathbf{A}_{\mathcal{S}\times\mathcal{T}}(S, T) - \mathbf{A}_{\mathcal{V}\times\mathcal{V}}(S, T) \right| . \end{aligned} \tag{27}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

ALGORITHM 21. $\texttt{ApxTensor}(\mathbf{A}, C, \varepsilon)$
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 $\texttt{PartMatrix}(\mathbf{B}, C, \varepsilon/16)$ 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 2).
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 partition $\mathcal{S} \times \mathcal{T}$ (cf. Corollary 2).
4. Call the algorithm $\texttt{FKTensor}(\hat{\mathbf{A}}, \varepsilon/(2C))$ from Theorem 8 to obtain cut tensors $\mathbf{D}_1, \ldots, \mathbf{D}_s$.
Output the cut tensors $Cp \cdot \mathbf{D}_1, \ldots, Cp \cdot \mathbf{D}_s$.

FIG. 2. *Pseudocode for* *ApxTensor*.

The definition of $\mathbf{A}_{\mathcal{V}\times\mathcal{V}}$ implies that $\mathbf{A}_{\mathcal{V}\times\mathcal{V}}(S,T) = \mathbf{A}(S,T)$ for all $S, T \in \mathcal{V}$. Thus, (27) yields

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

as claimed. ∎

4. Approximating and partitioning *k*-dimensional tensors.

4.1. Proof of Theorem 4. Let $0 < \varepsilon < \frac{1}{2}$ and $C > 1$. Let ζ, ζ' be the constants from Theorem 1, and set $\kappa = 256(\zeta + 100)(C/\varepsilon)^2$ and $\gamma = \zeta'\varepsilon/(C \cdot 2^{10\kappa+16})$. 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 $k_1 = \lfloor k/2 \rfloor$.

The basic idea behind the algorithm `ApxTensor` in Figure 2 for Theorem 4 is to transform the given sparse tensor $\mathbf{A}$ into a dense one $\hat{\mathbf{A}}$ and to apply the algorithm `FKTensor` from Theorem 8 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). Recall that its entries are

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

for $i_1 \in R_1, \ldots, i_k \in R_k$. Thus, $\mathbf{B}(\mathbf{A})$ is obtained by "flattening" the tensor $\mathbf{A}$. 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 22. *We have* $\|\mathbf{A} - Cp\hat{\mathbf{A}}\|_\square < \varepsilon \|\mathbf{A}\|_\square /2$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Proof. Let $m = \prod_{1\leq i\leq k_1}|R_i|$ and $n = \prod_{k_1<i\leq k}|R_i|$. Moreover, let $\hat{\mathbf{B}}$ be the $m\times n$ 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 by the definition of the cut norm for k-dimensional tensors we have

$$\begin{aligned}\left\|\mathbf{A}-Cp\hat{\mathbf{A}}\right\|_\square &= \max_{S_1\subset R_1,\ldots,S_k\subset R_k}\left|(\mathbf{A}-Cp\hat{\mathbf{A}})(S_1,\ldots,S_k)\right|\\ &= \max_{S_1\subset R_1,\ldots,S_k\subset R_k}\left|(\mathbf{B}-Cp\hat{\mathbf{B}})\left(\prod_{j=1}^{k_1}S_j,\prod_{j=k_1+1}^{k}S_j\right)\right|\\ &\leq \max_{\substack{S\subset R_1\times\cdots\times R_{k_1}\\ S'\subset R_{k_1+1}\times\cdots\times R_k}}\left|(\mathbf{B}-Cp\hat{\mathbf{B}})(S,S')\right|\\ &\leq \left\|\mathbf{B}-\hat{\mathbf{B}}\right\|_\square \leq \|\mathbf{B}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\|_\square + \left\|\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square. \end{aligned}\tag{28}$$

Furthermore, Corollary 2 ensures that $\|\mathbf{B}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\|_\square \leq \frac{\varepsilon}{8}\|\mathbf{B}\|_\square$. As both $\|\mathbf{B}\|_\square$ and $\|\mathbf{A}\|_\square$ equal the number of ones occurring in the tensor $\mathbf{A}$, (28) yields

$$\left\|\mathbf{A}-Cp\hat{\mathbf{A}}\right\|_\square \leq \frac{\varepsilon}{8}\|\mathbf{B}\|_\square + \left\|\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square = \frac{\varepsilon}{8}\|\mathbf{A}\|_\square + \left\|\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_\square.\tag{29}$$

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

$$0\leq \hat{\mathbf{B}}(i,j)\leq \mathbf{B}_{\mathcal{S}\times\mathcal{T}}(i,j)\qquad \text{for all } i\in\prod_{1\leq a\leq k_1}R_a,\ j\in\prod_{k_1<b\leq k}R_b.\tag{30}$$

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 $|T|<\gamma n$. Since by Corollary 2 both $\mathcal{S}$ and $\mathcal{T}$ consist of at most 2^κ classes, we have

$$|S_0|\leq 2^\kappa\gamma m < \varepsilon m/(100C),\qquad |T_0|\leq 2^\kappa\gamma n<\varepsilon n/(100C).\tag{31}$$

We claim that

$$\hat{\mathbf{B}}(i,j) = \mathbf{B}_{\mathcal{S}\times\mathcal{T}}(i,j)\qquad \text{for all } i\in\prod_{1\leq a\leq k_1}R_a\setminus S_0,\qquad j\in\prod_{k_1<b\leq k}R_b\setminus T_0.\tag{32}$$

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 $|S|\geq\gamma m$, $|T|\geq\gamma n$. Therefore, the fact that $\mathbf{B}$ is (C,γ)-bounded implies that $\mathbf{B}(S,T)\leq C\cdot|S\times T|\cdot p$. Hence, $\mathbf{B}_{\mathcal{S}\times\mathcal{T}}(i,j) = \mathbf{B}(S,T)/|S\times T|\leq Cp$, and thus the definition of $\hat{\mathbf{A}}$ in step 3 of `ApxTensor` yields (32).

For any two sets $X\subset\prod_{1\leq a\leq k_1}R_a$, $Y\subset\prod_{k_1<b\leq k}R_b$ we have

$$\begin{aligned}\left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}))(X,Y)\right| &\leq \left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X\setminus S_0,Y\setminus T_0)\right|\\ &\quad+\left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X\cap S_0,Y)\right| + \left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X,Y\cap T_0)\right|.\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

The first summand on the r.h.s. vanishes because of (32). Hence, (30) yields

$$\begin{aligned}\left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}))(X,Y)\right| &\leq \mathbf{B}_{\mathcal{S}\times\mathcal{T}}(X\cap S_0,Y)+\mathbf{B}_{\mathcal{S}\times\mathcal{T}}(X,Y\cap T_0)\\ &\leq \mathbf{B}(X\cap S_0,Y)+\mathbf{B}(X,Y\cap T_0)+2\left\|\mathbf{B}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\right\|_{\square}.\end{aligned}\tag{33}$$

Since $\|\mathbf{B}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}\|_{\square}\leq\frac{\varepsilon}{8}\|\mathbf{B}\|_{\square}$ by Corollary 2, (33) yields

$$\left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}}))(X,Y)\right| \leq |\mathbf{B}(X\cap S_0,Y)|+|\mathbf{B}(X,Y\cap T_0)|+\frac{\varepsilon}{4}\|\mathbf{B}\|_{\square}.\tag{34}$$

We are going to show that

$$|\mathbf{B}(X\cap S_0,Y)|,|\mathbf{B}(X,Y\cap T_0)|\leq\varepsilon\|\mathbf{B}\|_{\square}/100.\tag{35}$$

Case 1: $|X\cap S_0|\geq\gamma n$. Since $\mathbf{B}$ is (C,γ)-bounded, we have

$$\mathbf{B}(X\cap S_0,Y)\leq\frac{C|X\cap S_0||Y|}{mn}\cdot\|\mathbf{B}\|_{\square}\overset{(31)}{\leq}\varepsilon\|\mathbf{B}\|_{\square}/100.$$

Case 2: $|X\cap S_0|<\gamma n$. Let $X^*\subset R_1\times\cdots\times R_{k_1}$ be an arbitrary superset of $X\cap S_0$ of size $\lceil\gamma m\rceil$. Then we can apply the boundedness condition to obtain

$$\mathbf{B}(X\cap S_0,Y)\leq\mathbf{B}(X^*,Y)\leq\frac{C|X^*||Y|}{mn}\cdot\|\mathbf{B}\|_{\square}\leq 2C\gamma\|\mathbf{B}\|_{\square}\leq\varepsilon\|\mathbf{B}\|_{\square}/100.$$

In either case we have $\mathbf{B}(X\cap S_0,Y)\leq\varepsilon\|\mathbf{B}\|_{\square}/100$. The same argument shows that $|\mathbf{B}(X,Y\cap T_0)|\leq\varepsilon\|\mathbf{B}\|_{\square}/100$ as well. Thus, we have established (35). Finally, the bounds (34) and (35) imply that

$$\left|(\hat{\mathbf{B}}-\mathbf{B}_{\mathcal{S}\times\mathcal{T}})(X,Y)\right|\leq\frac{\varepsilon}{3}\|\mathbf{B}\|_{\square}=\frac{\varepsilon}{3}\|\mathbf{A}\|_{\square},$$

and thus the assertion follows by plugging this bound into (29). ∎

Proof of Theorem 4. Lemma 22 and Theorem 8 yield

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

as desired. Since step 4 calls `FKTensor` with parameter $\delta=\varepsilon/(2C)$, Theorem 8 implies that the number of cut tensors is at most $s\leq(2\Gamma_{FK}C/\varepsilon)^{2(k-1)}$. By Corollary 2 the running time of step 1 is bounded by $2^\kappa\cdot\Pi_*(|R_1|\cdots|R_k|)$ for a certain polynomial Π_*. Letting $\Gamma_*=256(\zeta+100)$, where ζ is the constant from Theorem 1, we have $2^\kappa\leq\exp(\Gamma_*(C/\varepsilon)^2)$ (recall that $\kappa=256(\zeta+100)(C/\varepsilon)^2$). Furthermore, by Theorem 8 the running time of step 4 is bounded by $(2\Gamma_{FK}C/\varepsilon)^{3k}\Pi^*(|R_1|\cdots|R_k|)$ for a fixed polynomial Π^*. Hence, letting $\Gamma=\Gamma_*+2\Gamma_{FK}$ and $\Pi=\Pi_*+\Pi^*$, we obtain the bound on the running time as stated in the theorem. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

4.2. Proof of Corollary 5. Let $\Gamma > 1$ be the constant from Theorem 4, and set $\tilde{\Gamma} = 4\Gamma$. Suppose that $0 < \varepsilon < \frac{1}{2}$, $C \geq 1$, and let $\gamma = \exp(-\tilde{\Gamma}(C/\varepsilon)^2)$. Moreover, 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`$(\mathbf{A}, C, \varepsilon/2)$ to obtain cut tensors

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

such that $\|\mathbf{A} - \sum_{i=1}^{s} \mathbf{D}_i\|_\square \leq \varepsilon \|\mathbf{A}\|_\square /2$. Then, for each $1 \leq j \leq k$ `ApxTensor` outputs the partition $\mathcal{R}_j$ of R_j generated by $S_{j1}, \ldots, S_{js}$.

Each partition $\mathcal{R}_j$ has at most 2^s classes. Hence, the upper bound

$$s \leq (2\Gamma C/\varepsilon)^{2(k-1)}$$

from Theorem 4 entails the upper bound on $|\mathcal{R}_j|$ stated in Corollary 5. The bound on the running time follows from Theorem 4 as well. Hence, we finally need to show that

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

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

$$\|\mathbf{D} - \mathbf{A}\|_\square \leq \varepsilon \|\mathbf{A}\|_\square /2. \tag{36}$$

To complete the proof, we are going to show that $\|\mathbf{D} - \mathbf{B}\|_\square \leq \varepsilon \|\mathbf{A}\|_\square /2$ as well. Since both $\mathbf{D}$ and $\mathbf{B}$ are constant on any rectangle $S_1 \times \cdots \times S_k$ with $S_i \in \mathcal{R}_i$, Lemma 9 entails that there exist sets $X_1 \subset R_1, \ldots, X_k \subset R_k$ such that

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

and X_i is a union of classes of $\mathcal{R}_i$ for $i = 1, \ldots, k$. We obtain

$$\begin{aligned} \|\mathbf{D} - \mathbf{B}\|_\square &= |(\mathbf{D} - \mathbf{B})(X_1, \ldots, X_k)| \\ &= \left| \sum_{S_1 \in \mathcal{R}_1 : S_1 \subset X_1} \cdots \sum_{S_k \in \mathcal{R}_k : S_k \subset X_k} \mathbf{D}(S_1, \ldots, S_k) - \mathbf{B}(S_1, \ldots, S_k) \right| . \end{aligned} \tag{37}$$

Furthermore, if $S_i \in \mathcal{R}_i$ for $1 \leq i \leq k$, then $\mathbf{B}(S_1, \ldots, S_k) = \mathbf{A}(S_1, \ldots, S_k)$. Therefore, (37) yields

$$\begin{aligned} \|\mathbf{D} - \mathbf{B}\|_\square &= \left| \sum_{S_1 \in \mathcal{R}_1 : S_1 \subset X_1} \cdots \sum_{S_k \in \mathcal{R}_k : S_k \subset X_k} \mathbf{D}(S_1, \ldots, S_k) - \mathbf{A}(S_1, \ldots, S_k) \right| \\ &= |(\mathbf{D} - \mathbf{A})(X_1, \ldots, X_k)| \leq \|\mathbf{D} - \mathbf{A}\|_\square \leq \varepsilon \|\mathbf{A}\|_\square /2 \qquad \text{[due to (36)]}. \end{aligned}$$

Hence, $\|\mathbf{A} - \mathbf{B}\|_\square \leq \|\mathbf{A} - \mathbf{D}\|_\square + \|\mathbf{D} - \mathbf{B}\|_\square \leq \varepsilon \|\mathbf{A}\|_\square$, as desired. ∎

5. Approximating MAX CSP problems. Throughout this section we keep the notation from section 1.4. Given $0 < \varepsilon < \frac{1}{2}$, $C > 1$, $k \geq 2$ we let

$$\alpha = \varepsilon 2^{-2^k - 2k - 2} \quad \text{and} \quad \gamma = \exp(-\Gamma(C/\alpha)^2),$$

where Γ is the constant from Theorem 4. We assume that $\mathcal{F}$ is a (C, γ)-bounded k-CSP instance on n variables $V = \{x_1, \ldots, x_n\}$, 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.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

ALGORITHM 23. ApxCSP($\mathcal{F}, C, \varepsilon$)
Input: A k-CSP instance $\mathcal{F}$ over $V = \{x_1, \ldots, x_n\}$, 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
   $$\mathbf{B}^{\psi} = \sum_{i=1}^{s} \mathbf{D}^{\psi}_i, \text{ where } \mathbf{D}^{\psi}_i = \mathsf{CUT}(d^{\psi}_i, S^{\psi}_{i1}, \ldots, S^{\psi}_{ik}).$$
   Let $\mathcal{P}$ be the partition of V generated by the sets S^{ψ}_{ih} ($1 \le i \le s$, $1 \le h \le k$, $\psi \in \Psi$).
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:
   $$\begin{aligned}
   \mathsf{OPT}'' = \max \sum_{\psi \in \Psi} \sum_{i=1}^{s} \sum_{y \in \{0,1\}^k} d^{\psi}_i \psi(y) \nu^k \prod_{h=1}^{k} \tau^{\psi}_{ih}(y_h) \\
   \text{s.t. } 0 \le \tau^{\psi}_{ih}(1) \le \lfloor |S^{\psi}_{ih}|/\nu \rfloor \quad & \text{is an integer for all} \\
   & 1 \le i \le s,\ 1 \le h \le k,\ \psi \in \Psi, \\
   \tau^{\psi}_{ih}(0) = \nu^{-1} S^{\psi}_{ih} - \tau^{\psi}_{ih}(1) \quad & \text{for all } 1 \le i \le s,\ 1 \le h \le k,\ \psi \in \Psi, \\
   \tau^{\psi}_{ih}(1)\nu \le \sum_{P \in \mathcal{P} : P \subset S^{\psi}_{ih}} z_P \le (\tau^{\psi}_{ih}(1) + 1)\nu & \\
   & \text{for all } 1 \le i \le s, 1 \le h \le k, \psi \in \Psi, \\
   0 \le z_P \le |P| \quad & \text{for all } P \in \mathcal{P}.
   \end{aligned}$$
   (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 $P \in \mathcal{P}$.

FIG. 3. *Pseudocode for ApxCSP.*

5.1. The algorithm ApxCSP. The first step of the algorithm ApxCSP in Figure 3 relies on the procedure ApxTensor from Theorem 4. 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 $\mathbf{D}^{\psi}_1, \ldots, \mathbf{D}^{\psi}_s$. Let $\mathbf{B}^{\psi} = \sum_{i=1}^{s} \mathbf{D}^{\psi}_i$. The basic idea is to approximate the actual MAX CSP problem, i.e., the optimization problem

$$\begin{aligned}
\mathrm{OPT} &= \max_{\sigma \in \{0,1\}^V} \sum_{\phi \in \mathcal{F}} \phi(\sigma) \\
&= \max_{\sigma \in \{0,1\}^V} \sum_{\psi \in \Psi} \sum_{(z_1, \ldots, z_k) \in V^k} \mathbf{A}^{\psi}(z_1, \ldots, z_k) \cdot \psi(\sigma(z_1), \ldots, \sigma(z_k)),
\end{aligned}$$

by the (weighted) optimization problem

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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

$$\sum_{\psi\in\Psi}\sum_{z\in V^k} \mathbf{B}^{\psi}(z)\psi(\sigma(z)) \geq OPT' - 2^{-k-1}\varepsilon m. \tag{38}$$

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

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

$$\mathcal{Z}_P = |\{v \in P : \sigma(v) = 1\}| \qquad \text{for each } P \in \mathcal{P}, \tag{39}$$

$$\mathcal{T}^{\psi}_{ih}(1) = \sum_{P\in\mathcal{P}:P\subset S^{\psi}_{ih}} \mathcal{Z}_P,\ \mathcal{T}^{\psi}_{ih}(0) = |S^{\psi}_{ih}| - \mathcal{T}^{\psi}_{ih}(1) \tag{40}$$

for $1 \leq i \leq s$, $1 \leq h \leq k$, $\psi \in \Psi$. In other words, $\mathcal{T}^{\psi}_{ih}(y)$ is the number of variables in S^{ψ}_{ih} that attain the value y under σ $(y = 0,1)$. Let us further define

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

Since $\mathbf{B}^{\psi} = \sum_{i=1}^{s}\mathbf{D}^{\psi}_i$ with $\mathbf{D}^{\psi}_i = \text{CUT}(d^{\psi}_i, S^{\psi}_{i1},\ldots,S^{\psi}_{ik})$, we have

$$\begin{aligned}\sum_{\psi\in\Psi}\sum_{z\in V^k}\mathbf{B}^{\psi}(z)\psi(\sigma(z)) &= \sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{z\in\prod_{h=1}^{k}S^{\psi}_{ih}} d^{\psi}_i\psi(\sigma(z))\\ &= \sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{y\in\{0,1\}^k} d^{\psi}_i\psi(y)\prod_{h=1}^{k}\mathcal{T}^{\psi}_{ih}(y_h).\end{aligned}$$

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 [17] 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 $(\mathcal{T}^{\psi}_{ih}(1))_{i,h,\psi}$ with $0 \leq \mathcal{T}^{\psi}_{ih}(1) \leq |S^{\psi}_{ih}|$. The issue is that not all such tuples correspond to an assignment $\sigma : V \to \{0,1\}$ as in (39) and (40). Hence, for each tuple $(\mathcal{T}^{\psi}_{ih}(1))_{i,h,\psi}$ we will have to check feasibility, i.e., if there is a tuple $(\mathcal{Z}_P)_P$ such that (40) holds. Since we are just aiming to solve OPT′ approximately, we can drop the requirement that all $\mathcal{Z}_P$ must be integral. Then checking the existence of a tuple $(\mathcal{Z}_P)_P$ such that (40) holds turns into a linear programming (LP) problem. In effect, we can reduce the running time from $\exp(|\mathcal{P}| \cdot \ln n)$ to $\exp(sk|\Psi| \cdot \ln n) \leq \exp(sk2^{2^k} \cdot \ln n)$. Remember that in general $|\mathcal{P}|$ is exponential in $sk2^{2^k}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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 $0 \le \mathcal{T}_{ih}^{\psi}(1) \le |S_{ih}^{\psi}|$ of variables to be set to 1 in each S_{ih}^{ψ}, we optimize over the number $0 \le \tau_{ih}^{\psi}(1) = \lfloor \mathcal{T}_{ih}^{\psi}(1)/\nu \rfloor \le \lfloor S_{ih}^{\psi}/\nu \rfloor$ of chunks set to 1. This is sufficient because we just need to solve OPT$'$ within an additive $\varepsilon 2^{-k-1} m$ (cf. (38)). 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 $0 \le z_P \le 1$ $(P \in \mathcal{P})$ such that $\tau_{ih}^{\psi}(1)\nu \le \sum_{P \in \mathcal{P} : P \subset S_{ih}^{\psi}} z_P \le (\tau_{ih}^{\psi}(1) + 1)\nu$ 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 25. *The assignment $\hat{\sigma}$ satisfies* (38).

Proof of Theorem 6. The fact that the assignment $\hat{\sigma}$ computed by `ApxCSP` satisfies at least $(1-\varepsilon)$OPT constraints follows from Lemma 24 and Proposition 25. Thus, we finally need to analyze the running time. By Theorem 4 the running time of step 1 is at most

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

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

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

Step 2 solves OPT$''$ by enumerating all possible values for the integer variables $\tau_{ih}^{\psi}(1)$. The number of these integer variables is $sk2^{2^k}$. Furthermore, for each $\tau_{ih}^{\psi}(1)$ there are at most $1 + |S_{ih}^{\psi}|/\nu \le 1 + \delta^{-1} \le \delta^{-2}$ 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

$$\tau_{ih}^{\psi}(1)\nu \le \sum_{P \in \mathcal{P} : P \subset S_{ih}^{\psi}} z_P \le (\tau_{ih}^{\psi}(1) + 1)\nu, \qquad 0 \le z_P \le |P|$$

for $1 \le i \le s$, $1 \le h \le k$, $\psi \in \Psi$, $P \in \mathcal{P}$. This can be performed in time polynomial in the encoding length of these linear equations. There are $|\mathcal{P}| + k2^{2^k} s \le 2\Lambda$ constraints and $|\mathcal{P}| \le \Lambda$ variables. The encoding length of the numbers involved is at most $\ln(n/\delta)$. Therefore, the running time is

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

Consequently, the total running time is at most

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

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5.2. Proof of Lemma 24. We shall prove below that for any $\tau \in \{0,1\}^V$

$$\left|\sum_{\phi \in \mathcal{F}} \phi(\tau) - \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\tau(z))\right| \leq 2^{-k-2}\varepsilon m. \tag{41}$$

This implies the assertion as follows. Since (41) holds for any $\tau \in \{0,1\}^V$, we have $\mathrm{OPT}' \geq \mathrm{OPT}(\mathcal{F}) - 2^{-k-2}\varepsilon m$. Hence, if $\sigma \in \{0,1\}^V$ satisfies (38), then (41) yields

$$\begin{aligned}\sum_{\phi \in \mathcal{F}} \phi(\sigma) &\geq \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\sigma(z)) - 2^{-k-2}\varepsilon m \\ &\geq \mathrm{OPT}' - \frac{3}{4} \cdot 2^{-k}\varepsilon m \geq \mathrm{OPT}(\mathcal{F}) - 2^{-k}\varepsilon m. \end{aligned} \tag{42}$$

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 $\mathrm{OPT}(\mathcal{F}) \geq 2^{-k}m$. Hence, the assertion follows from (42).

To prove (41), we fix $\tau \in \{0,1\}^V$ and let

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

Then by the triangle inequality

$$\begin{aligned}&\left|\sum_{\phi \in \mathcal{F}} \phi(\tau) - \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\tau(z))\right| \\ &= \left|\sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{A}^{\psi}(z)\psi(\tau(z)) - \sum_{\psi \in \Psi} \sum_{z \in V^k} \mathbf{B}^{\psi}(z)\psi(\tau(z))\right| \leq D. \end{aligned} \tag{43}$$

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 $R(t) = \prod_{i=1}^k \tau^{-1}(t_i) \subset V^k$. Then

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

Therefore, the bound $\left\|\mathbf{A}^{\psi} - \mathbf{B}^{\psi}\right\|_{\Box} \leq \alpha m$ (cf. step 1) entails

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

Thus, (41) follows from (43).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5.3. Proof of Proposition 25. The assignment $\hat{\sigma}$ results from solving the optimization problem OPT$''$ (cf. step 2 of the algorithm). The goal is to show that $\hat{\sigma}$ provides a good solution to OPT$'$ as well. To establish this we need to compare the optimal values of OPT$'$ and OPT$''$.

LEMMA 26. *We have* $OPT' - OPT'' \leq 2^{-k-2}\varepsilon m$.

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

$$\begin{aligned}\sum_{\psi\in\Psi}\sum_{v\in V^k}\mathbf{B}^{\psi}(z)\psi(\sigma(z)) &= \sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{z\in V^k}\mathbf{D}_i^{\psi}(z)\psi(\sigma(z))\\ &= \sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{z\in\prod_{h=1}^{k}S_{ih}^{\psi}} d_i^{\psi}\psi(\sigma(z))\\ &= \sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{y\in\{0,1\}^k} d_i^{\psi}\psi(y)\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h). \end{aligned}\tag{44}$$

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 $\tau_{ih}^{\psi}(0) = \nu^{-1}|S_{ih}^{\psi}| - \tau_{ih}^{\psi}(1)$ ($1 \leq i \leq s$, $1 \leq h \leq k$, $\psi \in \Psi$). To complete the proof, we shall compare the objective function value attained by this solution with (44). To this end, observe that $|\theta_{ih}^{\psi}(y) - \tau_{ih}^{\psi}(y)\nu| \leq \nu$ for all i, h, ψ, y. Therefore,

$$\left|\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) - \prod_{h=1}^{k}\tau_{ih}^{\psi}(y_h)\nu\right| \leq 2^k\nu n^{k-1} \qquad \text{for all } i, y, \psi. \tag{45}$$

Since by Theorem 4 we have $|d_i^{\psi}| \leq Cp\Gamma^k$ for all i, ψ, (45) yields

$$\begin{aligned}\left|\sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{y\in\{0,1\}^k} d_i^{\psi}\psi(y)\left[\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) - \nu^k\prod_{h=1}^{k}\tau_{ih}^{\psi}(y_h)\right]\right| &\leq Cp\Gamma^k \cdot s2^{2^k+2k}\nu n^{k-1}\\ &\leq Cp\Gamma^k \cdot \delta s 2^{2^k+2k+1}n^k \leq 2^{-k-2}\varepsilon m\end{aligned}\tag{46}$$

because $\delta = 1/(C\Gamma^k s 2^{2^k+4k+4})$; cf. step 2. Finally, combining (44) and (46), we conclude that OPT$'' \geq$ OPT$' - 2^{-k-2}\varepsilon m$, as desired. ∎

Proof of Proposition 25. Letting $\theta_{ih}^{\psi}(y) = |S_{ih}^{\psi} \cap \sigma^{-1}(y)|$ for all ψ, i, h and $y = 0, 1$, we have

$$\sum_{\psi\in\Psi}\sum_{v\in V^k}\mathbf{B}^{\psi}(z)\psi(\sigma(z)) = \sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{y\in\{0,1\}^k} d_i^{\psi}\psi(y)\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) \qquad \text{(cf. (44))}.$$

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

$$\begin{aligned}\tau_{ih}^{\psi}(1)\nu &\leq \sum_{P\in\mathcal{P}:P\subset S_{ih}^{\psi}} z_P \leq \sum_{P\in\mathcal{P}:P\subset S_{ih}^{\psi}} 1 + |\sigma^{-1}(1)\cap P| \leq |\sigma^{-1}(1)\cap S_{ih}^{\psi}| + s\\ &\leq |S_{ih}^{\psi}| + \nu \qquad\qquad [\text{as } n > n_0 = 2s/\delta].\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Similarly,

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

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

$$\left|\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) - \prod_{h=1}^{k}\tau_{ih}^{\psi}(y_h)\nu\right| \leq 2^{2k}\nu n^{k-1} \qquad \text{for all } i, y, \psi. \tag{47}$$

As $|d_i^{\psi}| \leq Cp\Gamma^k$ for all i, ψ by Theorem 4, (47) yields

$$\begin{aligned}\left|\sum_{\psi\in\Psi}\sum_{i=1}^{s}\sum_{y\in\{0,1\}^k} d_i^{\psi}\psi(y)\left[\prod_{h=1}^{k}\theta_{ih}^{\psi}(y_h) - \nu^k\prod_{h=1}^{k}\tau_{ih}^{\psi}(y_h)\right]\right| &\leq Cp\Gamma^k\cdot s2^{2^k+3k}\nu n^{k-1}\\ &\leq 2^{-k-2}\varepsilon m\end{aligned} \tag{48}$$

by the definition of δ. Thus, (47) and (48) imply 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* (**whp**) 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 **whp** $\mathcal{A}$ 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 27. *Suppose that* $np \geq c_0(\varepsilon)$ *for a number* $c_0(\varepsilon)$ *that depends only on* $\varepsilon > 0$. *The polynomial time algorithm* `ApxCSP` *from Theorem* 6 *approximates MAX CUT within* $1-\varepsilon$ *on* $G_{n,p}$*-bounded graphs.*

Proof. MAX CUT fits into the general CSP framework discussed in section 1.4 as follows. The set of variables is the vertex set of the input graph $G_* = (V, E_*)$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Moreover, each edge $e = \{v, w\} \in E_*$ yields the (binary) constraint

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

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 6 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

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

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

$$A(S,T) \leq 3|S \times T|p. \tag{49}$$

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, (49) 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 27 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 carry 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 27 encompasses this model. In fact, Theorem 27 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 ordered 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 $\mathcal{E}$ **whp** if the probability that $\mathcal{E}$ holds tends to 1 as $n \to \infty$. Moreover, for any k-SAT formula F we let $\mathcal{I}(F)$ denote the set of all subformulas F_* of F that contain at least $0.01m$ clauses. Furthermore,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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." In the first case we say that the algorithm *succeeds*; in the second case it *fails*.
2. If $F = F_k(n,p)$, then **whp** $\mathcal{A}$ succeeds on all formulas in $\mathcal{I}(F)$.

THEOREM 28. *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 depends only on ε. The polynomial time algorithm* `ApxCSP` *from Theorem* 6 *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 1.4 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 let $s_i = -1$ 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 nonzero, 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 $1 \leq i_1 < \cdots < i_k \leq n$ the entry of $\mathbf{A}_{\mathcal{F}}^{\psi}(i_1,\ldots,i_k)$ is 1 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 27, Chernoff bounds show that $\mathbf{A}_{\mathcal{F}}^{\psi}$ is $(1000,\gamma)$-bounded for any fixed $\gamma > 0$ **whp**. ∎

In particular, Theorem 28 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 $k \geq 3$, then for $m \geq c_0(\varepsilon)n^{\lceil k/2\rceil}$ 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, **whp** the polynomial time algorithm `ApxCSP` 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 [10]), and a number of authors have tailored algorithms specifically for this problem [8, 11, 16]. For even values of k, Theorem 27 matches the best known result [8].

6.3. Planted 3-SAT. 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 $i = |\{j \in \{1,2,3\} : \sigma(l_j) = 1\}| \geq 1$ include the clause $l_1 \vee l_2 \vee l_3$ with probability p_i independently.

In other words, $F(n,\vec{p})$ has a "planted" assignment σ, and each possible clause containing $i \geq 1$ satisfied literals under σ gets included with probability p_i independently.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Considering the clauses as *ordered* triples of literals, we see that the expected total number of clauses is $n^3p_3 + 3n^2p_2 + 3n^2p_1$. We say that a property holds **whp** if the probability that it holds tends to 1 as $n \to \infty$. The following result is due to Flaxman [15].

THEOREM 29. *There is a polynomial time algorithm* ***SpecSAT*** *such that for any* $\delta > 0$ *there is* $\zeta > 0$ *such that the following is true. Assume that either* $p_2 \geq \delta(p_1+p_3)$ *or* $p_1 \geq (1+\delta)p_3$ *or* $p_1 \leq (1-\delta)p_3$. *Moreover, assume that* $n^2(p_1 + 3p_2 + 3p_3) \geq \zeta$. *Then* ***SpecSAT*** *applied to* $F(n, \vec{p})$ *finds a satisfying assignment* **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 $G(\mathcal{F})$. Let T be the set of literals set to true under σ and $F = L \setminus T$. Consider $\delta > 0$ fixed, and let $\zeta > 0$ be the number promised by Theorem 29. Then there is a number $\delta' > 0$ such that for any triple $\vec{p}$ such that either $p_2 \geq \delta(p_1 + p_3)$ or $p_1 \geq (1+\delta)p_3$ or $p_1 \leq (1-\delta)p_3$ and $n^2(p_1+3p_2+3p_3) \geq \zeta$ one of the following is true:

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 $1F$ is at least $(\frac{1}{4} + \delta')|E(G(\mathcal{F}))|$.

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 $p_2 \leq \delta(p_1 + p_3)$ and $(1-\delta)p_3 \leq p_1 \leq (1+\delta)p_3$, 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 applies only if p_2 is identically zero. This is because 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 30. *There is a polynomial time algorithm* ***Find3SAT*** *(see Figure* 4*) that satisfies the following. Suppose that* $n^2(p_1 + 3p_2 + 3p_3) \geq \sqrt{n}\ln^{10} n$. *Then* ***Find3SAT*** *applied to* $F(n, \vec{p})$ *yields a satisfying assignment* **whp**.

Note that Theorem 30 requires the expected number of clauses to be at least $n^{3/2}\ln^{10} n$, whereas Theorem 29 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) [10]. But in order to refute the existence of a satisfying assignment of this type the best current polynomial time algorithm

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

ALGORITHM 31. `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 $\mathcal{R}$.

    Let $\delta = 10^{-20}$. Call `ApxCSP`($\mathcal{R}, 100, \delta$), and let τ' be the resulting assignment. Let τ'' be the Boolean complement 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."

FIG. 4. *Pseudocode for* `Find3SAT`.

requires $m \gg n^{3/2}$ [12]. In fact, techniques that allow one to improve the bound in Theorem 30 to $n^{3/2-\Omega(1)}$ may very well yield improved refutation algorithms (and vice versa).

In the rest of this section we let $\delta = 10^{-20}$. 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 satisfiable. Then, `Find3SAT` applies `ApxCSP` to $\mathcal{R}$. The following lemma shows that calling `ApxCSP` is feasible.

LEMMA 32. *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 $\gamma > 0$, $\mathcal{R}$ is a $(100, \gamma)$-bounded MAX CSP instance* **whp**.

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

LEMMA 33. *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 34. *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 30 is an immediate consequence of Theorem 29 and Lemmas 32–34.

Proof of Lemma 32. For a 3-SAT formula $\mathcal{F}$ over the variable set V we set up a 4-tensor $\mathbf{A}(\mathcal{F}) : L^4 \to \{0,1\}$ by letting

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

Then $\mathcal{F}$ is a $(100, \gamma)$-bounded MAX CSP instance if the tensor $\mathbf{A}(\mathcal{F})$ is $(100, \gamma)$-bounded. Hence, in order to prove Lemma 32 it is sufficient to establish the following,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

assuming that $p_2 \le \delta(p_1 + p_3)$ and $(1-\delta)p_3 \le p_1 \le (1+\delta)p_3$:

(50) $\quad \mathbf{A}(\mathcal{F}(n,\vec{p}))$ is $(100,\gamma)$-bounded **whp** $\quad$ for any fixed $\gamma > 0$.

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 (50):

(51) $\quad \mathbf{A}(F(n,\vec{q}))$ is $(33,\gamma)$-bounded **whp** $\quad$ for any fixed $\gamma > 0$.

To show (51), we employ the following result from [8, Lemma 3.3]. Let $\vec{J}$ signify the matrix with all entries equal to 1.

FACT 35. *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)*, and let $Q = nq^2$. Then $\|Q\vec{J} - \mathbf{B}(\mathcal{F})\| = o(n^2 Q)$.*

Proof of (51). Let $\mathbf{B} = \mathbf{B}(F(n,\vec{q}))$. The expected number $\mathrm{E}\,\|\mathbf{B}\|_\square$ of ones in $\mathbf{B}$ is $(2n)^4 Q$. This is because $\mathbf{B}((l_1,l_2),(l_3,l_4))$ equals 1 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 Fact 35

$$|Q|S \times T| - \mathbf{B}(S,T)| = \left|\left\langle (Q\vec{J} - \mathbf{B})\vec{1}_T, \vec{1}_S \right\rangle\right| \le \|Q\vec{J} - \mathbf{B}\| \cdot \|\vec{1}_T\| \cdot \|\vec{1}_S\|$$
$$= o(n^2 Q) \cdot \sqrt{|S \times T|} = o(Q|S \times T|) \qquad [\text{because } |S|, |T| \ge \gamma n^2].$$

Hence, $\mathbf{B}(S,T) \sim Q|S \times T|$ **whp**. As $Q \sim \hat{Q}$ **whp**, this implies that $\mathbf{B}(S,T) \sim \hat{Q}|S \times T|$, and thus $\mathbf{B}(S,T) \le 1.01\hat{Q}|S \times T|$ **whp**. Consequently, $\mathbf{B}$ is $(1.01,\gamma)$-bounded **whp**, whence (51) follows (with room to spare). ∎

Proof of Lemma 33. Without loss of generality we may assume that $\sigma(x) = 1$ for all $x \in V$. That is, we assume that the planted assignment is just the all-true assignment. Let $\alpha = 0.01$.

FACT 36. **Whp** *any assignment χ such that neither χ nor its Boolean complement χ' is within Hamming distance at most αn of σ fails to satisfy at least $\alpha^4 n^5 p_1 p_3 / 8$ clauses of $\mathcal{R}$.*

Proof. Suppose that neither χ nor χ' is within Hamming distance at most αn of σ. Then there are at least αn literals l such that $\sigma(l) = 1$ and $\chi(l) = 0$ and at least αn literals l' such that $\sigma(l') = 0$ and $\chi(l') = 1$. 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 $\chi(l_1) = \chi(l_2) = 0$ $(l_1, l_2 \in L)$. 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 $\sigma(l_4) = \chi(l_3) = \chi(l_4) = 0$ $(l_3, l_4 \in L)$. Then $(|\ell_z|, |\bar{\ell}_z|)_{z \in Z}$ is a family of mutually independent binomial random variables. For all $z \in V$ we have

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

Let us call z *bad* for χ if either $\ell_z < \mathrm{E}(|\ell_z|)/2$ or $\bar{\ell}_z < \mathrm{E}(|\bar{\ell}_z|)/2$. Then (52) 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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

at least $1-4^{-n}$ there be at most $n/2$ bad variables. Hence, by the union bound there is no assignment χ such that both χ and χ' are at Hamming distance more than αn from σ and χ has more than $n/2$ bad variables.

Thus, suppose that χ is an assignment that has less than $n/2$ bad variables. 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 χ does not satisfy. Consequently, χ fails to satisfy at least $\sum_{z \in V} \ell_z \bar{\ell}_z \geq \frac{n}{2} \cdot \frac{\alpha^4}{4} n^4 p_1 p_3 = \alpha^4 n^5 p_1 p_3 / 8$ clauses. ◻

To complete the proof of Lemma 33, we investigate assignments χ at Hamming distance at least $(1-\alpha)n$ and Hamming distance at most αn from σ.

FACT 37. **Whp** *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 fail 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 38. **Whp** *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, we show how Lemma 33 follows from Facts 36–38. Since the 4-SAT formula $\mathcal{R}$ is satisfiable, Theorem 6 entails that step 1 compute an assignment τ' that violates at most a δ fraction of all clauses of $\mathcal{R}$. By our assumption that $p_2 < \delta(p_1 + p_3)$ and $(1-\delta)p_3 \leq p_1 \leq (1+\delta)p_3$ the expected number of clauses in $\mathcal{R}$ is at most $2n^5 p_1 p_3$. Hence, by Chernoff bounds the total number of clauses in $\mathcal{R}$ is at most $3n^5 p_1 p_3$ **whp**. Since $\delta < \alpha^4/24$, we conclude that Fact 36 implies that either τ' or its Boolean complement τ'' is at Hamming distance at most αn from σ. Facts 37 and 38 imply that step 2 will identify the assignment among τ', τ'' that is at Hamming distance at most αn from σ. ◻

Proof of Lemma 34. We shall establish the following fact via a first moment argument. Let $\alpha = 0.01$.

FACT 39. **Whp** *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 33, Fact 39 shows that the Hamming distance of τ and σ decreases by a

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

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 39. 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. W.r.t. the first item, suppose that λ is a literal such 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 be 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**.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

REFERENCES

[1] N. Alon, A. Coja-Oghlan, H. Han, M. Kang, V. Rödl, and M. Schacht, *Quasi-randomness and algorithmic regularity for graphs with general degree distributions*, in Proceedings of the 34th International Colloquium on Automata, Languages and Programming, 2007, pp. 789–800.

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

[3] N. Alon, W. Fernandez de la Vega, R. Kannan, and M. Karpinski, *Random sampling and approximation of MAX-CSPs*, J. Comput. System Sci., 67 (2003), pp. 212–243.

[4] N. Alon and A. Naor, *Approximating the cut-norm via Grothendieck's inequality*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, pp. 72–80.

[5] S. Arora, D. Karger, and M. Karpinski, *Polynomial time approximation schemes for dense instances of NP-hard problems*, J. Comput. System Sci., 58 (1999), pp. 193–210.

[6] R. Boppana, *Eigenvalues and graph bisection: An average-case analysis*, in Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science, 1987, pp. 280–285.

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

[8] A. Coja-Oghlan, A. Goerdt, and A. Lanka, *Strong refutation heuristics for random k-SAT*, Combin. Probab. Comput., 16 (2007), pp. 5–28.

[9] R.A. Duke, H. Lefmann, and V. Rödl, *A fast approximation algorithm for computing the frequencies of subgraphs in a given graph*, SIAM J. Comput., 24 (1995), pp. 598–620.

[10] U. Feige, *Relations between average case complexity and approximation complexity*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 534–543.

[11] U. Feige, J.H. Kim, and E. Ofek, *Witnesses for non-satisfiability of dense random 3CNF formulas*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, pp. 497–508.

[12] U. Feige and E. Ofek, *Easily refutable subformulas of large random 3CNF formulas*, in Proceedings of the 31st International Colloquium on Automata, Languages and Programming, 2004, pp. 519–530.

[13] W. Fernandez de la Vega, *MAX-CUT has a randomized approximation scheme in dense graphs*, Random Structures Algorithms, 8 (1996), pp. 187–199.

[14] W. Fernandez de la Vega, R. Kannan, M. Karpinski, and S. Vempala, *Tensor decomposition and approximation algorithms for constraint satisfaction problems*, in Proceedings of the 37th Annual ACM Symposium on Theory of Computing, 2005, pp. 747–754.

[15] A. Flaxman, *A spectral technique for random satisfiable 3CNF formulas*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 357–363.

[16] J. Friedman, A. Goerdt, and M. Krivelevich, *Recognizing more unsatisfiable random k-SAT instances efficiently*, SIAM J. Comput., 35 (2005), pp. 408–430.

[17] A.M. Frieze and R. Kannan, *Quick approximation to matrices and applications*, Combinatorica, 19 (1999), pp. 175–220.

[18] S. Gerke and A. Steger, *The sparse regularity lemma and its applications*, in Surveys in Combinatorics, B. Webb, ed., Cambridge University Press, Cambridge, UK, 2005, pp. 227–258.

[19] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. ACM, 45 (1998), pp. 653–750.

[20] T. Gowers, *Lower bounds of tower type for Szemerédi's uniformity lemma*, Geom. Funct. Anal., 7 (1997), pp. 322–337.

[21] J. Hastad, *Some optimal inapproximability results*, J. ACM, 48 (2001), pp. 798–859.

[22] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, *Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?*, in Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004, pp. 146–154.

[23] Y. Kohayakawa, *Szemerédi's regularity lemma for sparse graphs*, in Foundations of Computational Mathematics, F. Cucker and M. Shub, eds., Springer, Berlin, 1997, pp. 216–230.

[24] V. Rödl, unpublished.

[25] E. Szemerédi, *Regular partitions of graphs*, in Problémes Combinatoires et Théorie des Graphes, Colloq. Internat. CNRS 260, CNRS, Paris, 1978, pp. 399–401.

[26] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson, *Gadgets, approximation, and linear programming*, SIAM J. Comput., 29 (2000), pp. 2074–2097.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.