Electronic Colloquium on Computational Complexity, Report No. 104 (2007)

On the Advantage over Random for Maximum Acyclic Subgraph

Moses Charikar* Konstantin Makarychev† Yury Makarychev‡

Abstract

In this paper we present a new approximation algorithm for the MAX ACYCLIC SUBGRAPH *problem. Given an instance where the maximum acyclic subgraph contains $1/2+\delta$ fraction of all edges, our algorithm finds an acyclic subgraph with $1/2+\Omega(\delta/\log n)$ fraction of all edges.*

1 Introduction

The focus of this paper is the MAX ACYCLIC SUBGRAPH problem which is the following:

Definition 1.1. *Given a directed graph $G = (V, E)$, find the largest subset of edges which are acyclic. Equivalently, find an ordering of the vertices so as to maximize the number of edges going forward.*

A simple randomized algorithm achieves a factor 1/2 for this problem: Simply pick a random ordering of the vertices. In fact, one can achieve factor 1/2 by an even simpler algorithm: Pick an arbitrary ordering of the vertices π and its reverse π^R. One of them has at least $1/2$ fraction of the edges in the forward direction. Improving the $1/2$ approximation for MAX ACYCLIC SUBGRAPH is a long standing open problem. The motivating question for our work was whether it is possible to beat this 1/2 approximation.

In fact, algorithms very slightly better than 1/2 are known: Berger and Shor [4] showed how to get $1/2 + \Omega(1/\sqrt{d_{max}})$, where d_{max} is the maximum vertex degree in the graph. Later Hassin and Rubinstein [9] proposed another algorithm with the same approximation guarantee, but better running time in certain cases. Note that these algorithms achieve such a guarantee for any graph: the guarantee does not depend on the value of the optimal solution.

Let us measure the objective as a fraction of the number of edges. If the optimal value OPT is $1-\varepsilon$, for a very small ε, we could use the best known approximation algorithm for the complementary problem (MIN FEEDBACK ARC SET) to beat the random algorithm. Using the $O(\log n \log\log n)$ algorithm of Seymour [20], for instances where $OPT = 1-\varepsilon$ and $\varepsilon = O(1/(\log n \log\log n))$, we can indeed beat $1/2$ for MAX ACYCLIC SUBGRAPH. This yields an approximation ratio of $1/2 + \Omega(1/(\log n \log\log n))$ for the problem.

To summarize, we can beat random for instances where OPT is very close to 1. For instances where OPT is smaller, we do not know of any techniques which perform better than random. (As mentioned, there are algorithms which have the guarantee $1/2 + \Omega(1/\sqrt{d_{max}})$.)

Recently, a related question of approximating the advantage over random has been studied for several basic optimization problems (see [3, 5, 6, 7, 10, 11, 12, 13]). These studies give a fresh perspective on these optimization problems and motivated the development of new techniques to extract information from mathematical programming relaxations for them.

Definition 1.2. *Let $G = (V, E)$ be a directed graph on n vertices; and let $\pi : \{1, \dots, n\} \to V$ be a linear arrangement of its vertices. Then the* advantage or gain[1] over random *of the arrangement π is equal to the fraction of edges going forward minus the fraction of edges going backward. We denote the gain over random by* $\mathrm{gain}(G, \pi)$.

If a linear arrangement has value $1/2+\delta$ for MAX ACYCLIC SUBGRAPH, then the gain of this arrangement is 2δ. The question of beating random for MAX ACYCLIC SUBGRAPH can be phrased thus: Given an instance with optimal gain δ, can we guarantee that we produce a solution with gain $f(\delta)$?

Note that the usual notion of approximation only focuses on instances where the optimal gain δ is close to 1 (undoubt-

*Department of Computer Science, Princeton University, Princeton, NJ 08540. Email: moses@cs.princeton.edu. Supported by NSF ITR grant CCR-0205594, NSF CAREER award CCR-0237113, MSPA-MCS award 0528414, and an Alfred P. Sloan Fellowship.

†IBM T.J. Watson Research Center, Yorktown Heights, NY 10598. Email: konstantin@us.ibm.com. This work was done while the author was at the Department of Computer Science, Princeton University. Supported by an IBM Graduate fellowship and Gordon Wu fellowship.

‡Microsoft Research, One Microsoft Way, Redmond, WA 98052. Email: yurym@microsoft.com. This work was done while the author was at the Department of Computer Science, Princeton University. Supported by a Gordon Wu fellowship.

[1] This concept has been referred to in the literature as both *advantage over random* and *gain*. We will use both terms interchangeably.

ISSN 1433-8092

edly a very interesting question). We ask what guarantee is possible as a function of δ for all values of $\delta \in (0, 1)$.

Such guarantees were developed for MAX CUT by Charikar and Wirth [6]: Given a MAX CUT instance, for which the optimal solution has gain δ, we can find a cut with gain $\Omega(\delta/\log(1/\delta))$ and $\Omega(\delta/\log n)$; the former approximation guarantee is optimal (if the Unique Games Conjecture is true) as was shown by Khot and O'Donnell [15].

There are some parallels between MAX CUT and MAX ACYCLIC SUBGRAPH, since a random assignment achieves factor 1/2 for both problems. For MAX CUT, this was indeed the best known until the seminal work of Goemans and Williamson [8] using semidefinite programming (SDP). In a sequence of later papers, our understanding of the MAX CUT SDP has vastly improved. Arguably, MAX ACYCLIC SUBGRAPH is a more complex problem than MAX CUT. Linear programming (LP) relaxations for the problem have been studied intensively in the mathematical programming community (it is sometimes referred to as the linear ordering problem). For more information we refer the reader to the papers of Newman [16] and Newman and Vempala [18]. However the best known approximation for the problem still remains $1/2$. Newman [17] recently studied an SDP relaxation for the problem and gave some evidence to suggest that the SDP might be useful in beating the $1/2$ approximation (in particular, the SDP does well on the known gap instances for the LP).

In this work, we give an $O(\log n)$ approximation for the advantage over random for MAX ACYCLIC SUBGRAPH. In other words, given an instance where $OPT = 1/2 + \delta$, we find a solution of value $1/2+\Omega(\delta/\log n)$. Prior to our work, no non-trivial guarantees were known even for instances where OPT was close to 1, say $1 - 1/\log n$. In contrast, our algorithm gives a non-trivial guarantee even for OPT close to $1/2$. As a byproduct, we obtain a $1/2 + \Omega(1/\log n)$ approximation for MAX ACYCLIC SUBGRAPH — very slightly better than the $1/2+\Omega(1/(\log n \log\log n))$ alluded to earlier that comes from Seymour's algorithm [20]. Note that the known hardness results for MAX ACYCLIC SUBGRAPH [14, 19] imply that the advantage over random version has a constant factor hardness.

Vertex ordering problems like MAX ACYCLIC SUBGRAPH seem more complex than the constraint satisfaction problems that have been recently explored with the lens of approximating the advantage over random. It is somewhat surprising therefore that we obtain results for MAX ACYCLIC SUBGRAPH that match the corresponding guarantee for MAX CUT. Despite the similarity in the statement of the result, the techniques are quite different. The $\log n$ in MAX CUT comes from the tail of the Gaussian distribution, while the $\log n$ in our result comes from the number of different distance scales in a linear arrangement. Roughly speaking, our results show how ordering information from one distance scale in the optimum solution can be exploited algorithmically. Extending these ideas further to exploiting information from multiple distance scales simultaneously is a promising avenue for obtaining a constant better than 1/2 approximation for MAX ACYCLIC SUBGRAPH. This would be an exciting result indeed.

Our Results. Our main result is as follows.

Theorem 1.3. *There exists a randomized polynomial time algorithm that given a directed graph G finds a linear arrangement π of its vertices with gain over random at least $\Omega(\delta/\log n)$, where δ is the maximum possible gain.*

We show a connection between the advantage over random and the cut norm of the adjacency matrix of the graph G. In Section 2, we present a simple algorithm that finds a linear arrangement with advantage over random proportional to the cut norm of the adjacency matrix of the graph. Then, in Section 3, we prove using Fourier analysis techniques that the cut norm of the adjacency matrix is within a $\log n$ factor of the optimal gain. We also give an example that shows that our analysis is tight.

2 Approximation Algorithm

It will be convenient for us to express different quantities in terms of the adjacency matrix W_G of the directed graph G. For unweighted graphs, we define W_G as follows:

$$W_G(u,v) = \begin{cases} 1, & \text{if } (u,v) \in E; \\ -1, & \text{if } (v,u) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

If both $(u, v) \in E$ and $(v, u) \in E$ then $W_G(u, v) = 0$. For weighted graphs,

$$W_G(u,v) = \mathrm{weight}((u,v)) - \mathrm{weight}((v,u)),$$

where $\mathrm{weight}((u, v)) = 0$ if $(u, v) \notin E$. Below $|E|$ denotes the total weight of all edges.

The gain over random is equal to

$$\mathrm{gain}(G,\pi) = \frac{1}{|E|} \sum_{i<j} W_G(\pi_i, \pi_j).$$

In other words the gain is equal to the sum of the elements in the upper triangle of the matrix W_G, in which rows and columns are arranged according to π, divided by the number of edges.

Let us now describe our approach to solving the problem. First partition the vertices of the graph into three sets A, B and C in a special way. Then randomly permute vertices in each of these sets. Finally, with probability a half output all vertices in the order A, B, C and with probability a half in the order C, A, B.

It is easy to see that all edges from A to B go forward; and all edges from B to A go backward. On the other hand, all other edges go backward or forward with probability exactly a half. Hence, the expected gain is equal to

$$\frac{1}{|E|}\sum_{u\in A;\, v\in B} W_G(u,v). \tag{1}$$

We just showed that the maximum gain is greater than or equal to (1). It turns out that the converse is also true up to an $O(\log n)$ factor. That is, there always exist disjoint sets A and B for which

$$\frac{1}{|E|}\sum_{u\in A;\, v\in B} W_G(u,v) \geq \frac{\mathrm{gain}(G,\pi)}{O(\log n)},$$

where π is the optimal permutation of the vertices. This statement is the main technical component of the proof and we prove it in the next section.

2.1 Combinatorial Interpretation of Proof

As a prelude to the technical analysis in Section 3, we give a simplified overview. The goal of the analysis is to show that if there is an ordering π with gain δ, then there are subsets A and B of such that placing all vertices of A before vertices of B gives gain at least $\delta/\log n$. (In fact, the sets A and B we construct in our analysis are not disjoint, but this is easy to fix.)

Define the length of an edge to be the distance between its end points in the ordering π. We can group edges geometrically by length into $O(\log n)$ groups. If the gain of the ordering π is δ, at least one of these groups must have gain $\delta/\log n$.

We construct the sets A and B by random sampling the positions in the ordering π. The selection probabilities vary periodically with position. Roughly speaking, if we have period P, this targets the group of edges with edge lengths $\Theta(P)$. Since one of the groups has gain $\delta/\log n$, the sampling process targeted towards that group will generate sets A and B such that the corresponding ordering has gain $\delta/\log n$.

In our proof later, this sampling is incorporated into a certain bilinear form (3) we analyze. This expression involves terms $x_k(r)$ and $y_k(r)$ (defined later) that corresponds to selecting sets A and B randomly where the selection probabilities vary periodically with position in the ordering π. For appropriate choice of the period our analysis shows that the bilinear form constructed must have value at least $\delta/\log n$.

This is a somewhat simplistic explanation that ignores several issues. In fact, edges could have both positive and negative contributions to the bilinear form constructed (positive contributions come from edges going from A to B, negative contributions come from edges going from B to A). Our intuitive explanation focussed on the contribution from one group of edges, but we need to ensure that the potential negative contributions of the other groups do not overwhelm this. The Fourier machinery we use allows us to properly account for positive and negative contributions.

Now, we do not actually know the optimal ordering π, so we do not really perform this sampling in our algorithm to obtain sets A and B. Instead, we focus on a quantity called the cut norm which we define in the next section. Our existential proof is merely an analysis tool that allows us to prove that the cut norm of the adjacency matrix is large if the gain of *some* ordering is large. The actual algorithm uses Alon and Naor's SDP based approximation for the cut norm. This yields sets A and B from which we obtain an ordering of the vertices.

2.2 Efficient Implementation

We now show how to efficiently find sets A and B that maximize (1) within a constant factor and thus obtain an $O(\log n)$ approximation for the maximum gain problem. As alluded to before, this problem is closely related to the problem of finding the cut norm of the matrix W_G, which can be approximately solved using the algorithm proposed by Alon and Naor [2].

Definition 2.1. *The cut norm of a matrix $W(u,v)$ is equal to*

$$\|W\|_C = \max_{A,B\subset V}\left|\sum_{u\in A;\, v\in B} W(u,v)\right|.$$

Note that for skew-symmetric matrices

$$\sum_{u\in A;\, v\in B} W(u,v) = -\sum_{u\in A;\, v\in B} W(v,u);$$

and therefore

$$\|W\|_C = \max_{A,B\subset V}\sum_{u\in A;\, v\in B} W(u,v).$$

In this definition it is not required that the sets A and B are disjoint. However, given arbitrary sets A and B, we can always find disjoint sets A' and B' such that

$$\sum_{u\in A';v\in B'} W(u,v) \geq \frac{1}{4}\cdot\sum_{u\in A;\, v\in B} W(u,v). \tag{2}$$

In order to do so, we simply partition the vertices of the graph G into two random sets X and Y. Then set $A' = A\cap X$; $B' = B\cap Y$.

Lemma 2.2. *For every skew-symmetric matrix G, the sets A' and B' (as described above) are disjoint and satisfy the following equality:*

$$\mathbb{E}\left[\sum_{u\in A',v\in B'} W(u,v)\right] = \frac{1}{4}\sum_{u\in A,v\in B} W(u,v).$$

Proof. The sets A' and B' are disjoint, since the sets X and Y are disjoint. Now, for every distinct vertices $u \in A$ and $v \in B$, the probability that $u \in A'$ and $v \in B'$ equals $1/4$. (Note, that the diagonal entries of W are equal to zero.) ☐

We now state the result of Alon and Naor [2].

Theorem 2.3 (Alon and Naor [2]). *There exists a randomized polynomial time algorithm that given a matrix $W(u,v)$ finds two subsets of indices A and B such that*

$$\left|\sum_{u\in A;\, v\in B} W(u,v)\right| \geq \alpha_{AN}\cdot \|W\|_C,$$

where $\alpha_{AN} \approx 0.56$.

Applying Lemma 2.2, we get the following corollary.

Corollary 2.4. *There exists a randomized polynomial time algorithm that given a skew-symmetric matrix $W(u,v)$ finds two disjoint subsets A and B such that*

$$\sum_{u\in A;\, v\in B} W(u,v) \geq \frac{\alpha_{AN}}{4}\cdot \|W\|_C.$$

This Corollary implies that the algorithm described in the beginning of the section finds a linear arrangement with gain $\alpha_{AN}/4 \cdot \|W_G\|_C/|E|$. Thus, we proved:

Lemma 2.5. *There exists a randomized polynomial time algorithm that given a directed graph G with adjacency matrix $W_G(u,v)$ finds a linear arrangement of the vertices with gain*

$$\frac{\alpha_{AN}}{4}\cdot\frac{\|W_G\|_C}{|E|}.$$

3 Cut Norm of Skew-Symmetric Matrices

In this section we will prove that for every linear arrangement π (particularly, for the optimal linear arrangement)

$$\|W_G\|_C \geq \Omega\left(\frac{\mathrm{gain}(G,\pi)|E|}{\log n}\right),$$

which will conclude the proof of Theorem 1.3. Fix an arbitrary linear arrangement π and denote

$$w_{kl} = W(\pi_k, \pi_l).$$

Remark 3.1. *In this paper, we use the discrete Fourier* sine *transform, which is an analog of the discrete Fourier transform, but is less well known. For this reason, we briefly describe it and prove the inversion formula in the Appendix.*

Theorem 3.1. *Let W be an $n\times n$ skew-symmetric matrix. Define*

$$S^+ = S^+(W) = \sum_{1\leq k<l\leq n} w_{kl}.$$

Then

$$\|W\|_C \geq \Omega\left(\frac{|S^+|}{\log n}\right).$$

Remark 3.2. *Recall, that $\mathrm{gain}(G,\pi) = S^+(W_G)/|E|$, where the columns and rows in W_G are ordered according to the permutation π.*

We need several lemmas.

Lemma 3.2. *Let $\hat{S}_t$ be the discrete Fourier sine transform of a sequence $S_1,\dots,S_{n-1}$, defined as follows*

$$\hat{S}_t = \sum_{k=1}^{n-1}\sin\Big(\frac{\pi k t}{n}\Big) S_k.$$

Then

$$\max_t |\hat{S}_t| \geq \frac{\pi}{2\log n + O(1)}\sum_{k=1}^{n-1} S_k.$$

Proof. The inverse Fourier sine transform is given by

$$S_k = \frac{2}{n}\sum_{t=1}^{n-1}\sin\Big(\frac{\pi k t}{n}\Big)\hat{S}_t.$$

Hence,

$$\begin{aligned}\sum_{k=1}^{n-1} S_k &= \frac{2}{n}\sum_{k=1}^{n-1}\sum_{t=1}^{n-1}\sin\Big(\frac{\pi k t}{n}\Big)\hat{S}_t\\ &= \frac{2}{n}\sum_{t=1}^{n-1}\left(\sum_{k=1}^{n-1}\sin\Big(\frac{\pi k t}{n}\Big)\right)\hat{S}_t.\end{aligned}$$

Note that[2]

$$\sum_{k=1}^{n-1}\sin\frac{\pi k t}{n} = \frac{1}{2}\frac{(1-(-1)^t)\sin(\frac{\pi t}{n})}{1-\cos(\frac{\pi t}{n})} = \frac{1-(-1)^t}{2\tan(\frac{\pi t}{2n})}.$$

Then

$$\begin{aligned}\sum_{k=1}^{n-1} S_k &\leq \frac{2}{n}\sum_{t=1}^{n-1}\left|\frac{1-(-1)^t}{2\tan(\frac{\pi t}{2n})}\right| |\hat{S}_t|\\ &\leq \frac{2}{n}\left(\sum_{\substack{t=1\\ t \text{ is odd}}}^{n-1}\frac{2n}{\pi t}\right)\max_t|\hat{S}_t|\\ &= \frac{2}{\pi}(\log n + O(1))\max_t|\hat{S}_t|,\end{aligned}$$

[2] See Corollary 5.2 in the Appendix for more details.

here we used that $\tan x > x$ for $x \in (0, \pi/2)$. Therefore,

$$\max_t |\hat{S}_t| \geq \frac{\pi}{2\log n + O(1)} \sum_{k=1}^{n-1} S_k.$$

$\square$

Lemma 3.3. *Let W be an $n \times n$ skew-symmetric matrix. Define*

$$S_k = \sum_{j=1}^{n-k} w_{j,j+k}$$

for $1 \leq k \leq n-1$. And let $\hat{S}_t$ be the discrete Fourier sine transform of S_k. Then

$$\max_{-1 \leq x_k, y_l \leq 1} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k y_l \geq \max_t |\hat{S}_t|.$$

Proof. Let $t_0 = \operatorname{argmax}_t |\hat{S}_t|$. For every k, l from 1 to n and r from 0 to $n-1$ define

$$\begin{aligned} x_k(r) &= \sin\Big(\frac{\pi(k+r)t_0}{n}\Big); \\ y_l(r) &= -\cos\Big(\pi\frac{(k+r)t_0}{n}\Big). \end{aligned}$$

Find the average value of the bilinear form

$$\sum_{k=1}^{n} \sum_{k=1}^{n} w_{kl} x_k(r) y_l(r) \qquad (3)$$

over r from 0 to $n-1$. Write

$$\begin{aligned} &\frac{1}{n} \sum_{r=0}^{n-1} \left(\sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k(r) y_l(r) \right) = \\ &\qquad = \frac{1}{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \left(\sum_{r=0}^{n-1} x_k(r) y_l(r) \right) w_{kl}. \end{aligned}$$

Observe, that

$$\begin{aligned} &\sum_{r=0}^{n-1} x_k(r) y_l(r) = \\ &= -\sum_{r=0}^{n-1} \sin\Big(\frac{\pi(k+r)t_0}{n}\Big) \cos\Big(\frac{\pi(l+r)t_0}{n}\Big) \\ &= -\frac{1}{2} \sum_{r=0}^{n-1} \sin\Big(\frac{\pi(k-l)t_0}{n}\Big) + \sin\Big(\frac{\pi(k+l+2r)t_0}{n}\Big) \\ &= \frac{n}{2} \cdot \sin\frac{\pi(l-k)t_0}{n}. \end{aligned}$$

Hence,

$$\begin{aligned} \frac{1}{n} \sum_{r=0}^{n-1} \Big(\sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k(r) y_l(r) \Big) &= \\ &= \frac{1}{2} \sum_{k=1}^{n} \sum_{l=1}^{n} \sin\Big(\frac{\pi(l-k)t_0}{n}\Big) w_{kl} \\ &= \sum_{1 \leq k < l \leq n} \sin\Big(\frac{\pi(l-k)t_0}{n}\Big) w_{kl} \\ &= \sum_{j=1}^{n-1} \sin\Big(\frac{\pi j t_0}{n}\Big) S_j = \hat{S}_{t_0}. \end{aligned}$$

here we used that the matrix W is skew-symmetric. We got that the average value of bilinear form (3) is $\hat{S}_{t_0}$, therefore, there exists r for which the absolute value of (3) is at least $|\hat{S}_{t_0}|$. Since the bilinear form is an odd function as a function of x (when y is fixed), the maximum of (3) is at least $|\hat{S}_{t_0}|$. $\square$

Corollary 3.4. *Let W, S_k and $\hat{S}_t$ be as in Lemma 3.3. Then*

$$\max_{x_k, y_l \in \{-1,1\}} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k y_l \geq \max_t |\hat{S}_t|.$$

Proof. The maximum of the bilinear form is attained at a vertex of the cube. $\square$

The following observation is due to Alon and Naor [2]. We prove it here for completeness.

Lemma 3.5 (Alon and Naor [2])**.**

$$\|W\|_C \geq \frac{1}{4} \max_{x_k, y_l \in \{-1,1\}} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k y_l.$$

Proof. Fix x_k and y_l for which the maximum of the bilinear form is attained. Define sets $I^+ = \{k : x_k = 1\}$, $I^- = \{k : x_k = -1\}$, $J^+ = \{l : y_l = 1\}$ and $J^- = \{l : y_l = 1\}$. Now notice that

$$\begin{aligned} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k y_l &= \sum_{\substack{k \in I^+ \\ l \in J^+}} w_{kl} + \sum_{\substack{k \in I^- \\ l \in J^-}} w_{kl} \\ &\quad - \sum_{\substack{k \in I^+ \\ l \in J^-}} w_{kl} - \sum_{\substack{k \in I^- \\ l \in J^+}} w_{kl}. \end{aligned}$$

Each of the terms on the right hand side does not exceed the cut norm $\|W\|_C$ in absolute value. Hence

$$\sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k y_l \leq 4\|W\|_C.$$

$\square$

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let

$$S_k = \sum_{j=1}^{n-k} w_{j,j+k}$$

for $1 \le k \le n-1$. Then

$$S^+ = \sum_{k=1}^{n-1} S_k.$$

By Lemma 3.2 and Corollary 3.4,

$$\max_{x_k, y_l \in \{-1,1\}} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{kl} x_k y_l \ge \frac{\pi}{2\log n + O(1)} \sum_{k=1}^{n-1} S_k.$$

Now by Lemma 3.5,

$$\|W\|_C \ge \frac{\pi}{8\log n + O(1)} \sum_{k=1}^{n-1} S_k = \frac{\pi\, S^+}{8\log n + O(1)}.$$

□

4 Lower bound

We will now prove that the bound in Theorem 3.1 is essentially tight.

Theorem 4.1. *For every $n > 1$, there exists a nonzero $n \times n$ skew-symmetric matrix W such that*

$$\|W\|_C \le O(|S^+|/\log n), \tag{4}$$

where $S^+ = S^+(W)$ is defined in Theorem 3.1.

Proof. Consider the matrix W defined by

$$w_{kl} = \sum_{t=1}^{n} \sin \frac{\pi(l-k)t}{n+1}.$$

Clearly, the matrix W is skew-symmetric. Let us compute S^+ (see Lemma 5.3 in the Appendix for details).

$$\begin{aligned} S^+ &= \sum_{1\le k<l\le n} w_{kl} = \sum_{1\le k<l\le n} \sum_{t=1}^{n} \sin \frac{\pi(l-k)t}{n+1} \\ &= \sum_{t=1}^{n} \sum_{k=0}^{n-1} (n-k) \sin \frac{\pi k t}{n+1} \\ &= \frac{1}{2} \sum_{t=1}^{n} \frac{n + (-1)^t}{\tan(\frac{\pi t}{2(n+1)})}. \end{aligned}$$

Replace $1/\tan x$ with $1/x + O(1)$,

$$S^+ \ge \frac{1}{2} \sum_{t=1}^{n} \left(\frac{2n^2}{\pi t} + O(n) \right) = \frac{n^2}{\pi} (\log n + O(1)).$$

We are now going to estimate $\|W\|_C$. Pick sets I and J that maximize $\left|\sum_{k\in I;\, l\in J} w_{kl}\right|$. We have

$$\begin{aligned} \|W\|_C &= \Big| \sum_{k\in I;\, l\in J} w_{kl} \Big| \\ &= \Big| \sum_{k\in I;\, l\in J} \sum_{t=1}^{n} \sin\Big(\frac{\pi(k-l)t}{n+1}\Big) \Big| \\ &= \Big| \sum_{k\in I;\, l\in J} \sum_{t=1}^{n} \sin\Big(\frac{\pi k t}{n+1}\Big) \cos\Big(\frac{\pi l t}{n+1}\Big) \\ &\qquad - \cos\Big(\frac{\pi k t}{n+1}\Big) \sin\Big(\frac{\pi l t}{n+1}\Big) \Big| \\ &\le \Big| \sum_{k\in I;\, l\in J} \sum_{t=1}^{n} \sin\Big(\frac{\pi k t}{n+1}\Big) \cos\Big(\frac{\pi l t}{n+1}\Big) \Big| \\ &\qquad + \Big| \sum_{k\in I;\, l\in J} \sum_{t=1}^{n} \cos\Big(\frac{\pi k t}{n+1}\Big) \sin\Big(\frac{\pi l t}{n+1}\Big) \Big|. \end{aligned}$$

Estimate the first term. Let x_k be the indicator of the set I: $x_k = 1$ if $k \in I$, $x_k = 0$ otherwise. Let y_k be the indicator of the set J. Then the first term equals

$$\begin{aligned} T_I &\equiv \sum_{t=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \sin\Big(\frac{\pi k t}{n+1}\Big) \cos\Big(\frac{\pi l t}{n+1}\Big) x_k y_l \\ &= \sum_{t=1}^{n} \left(\sum_{k=1}^{n} \sin\Big(\frac{\pi k t}{n+1}\Big) x_k \right) \left(\sum_{l=1}^{n} \cos\left(\frac{\pi l t}{n+1}\right) y_l \right) \\ &= \sum_{t=1}^{n} \hat{x}_t \hat{y}_t, \end{aligned}$$

where $\hat{x}_t$ is the Fourier *sine* transform of x_t

$$\hat{x}_t = \sum_{k=1}^{n} \sin\Big(\frac{\pi k t}{n+1}\Big) x_k, \quad 1 \le t \le n;$$

and $\hat{y}_t$ is the discrete Fourier *cosine* transform of y_k (extended by $y_0 = y_{n+1} = 0$)

$$\hat{y}_t = \sum_{k=1}^{n} \cos\Big(\frac{\pi k t}{n+1}\Big) y_k, \quad 0 \le t \le n+1.$$

By the Cauchy–Schwartz inequality and Bessel's inequality, we have

$$\begin{aligned} |T_I| &= \Big| \sum_{t=1}^{n} \hat{x}_t \hat{y}_t \Big| \le \sqrt{\sum_{t=1}^{n} \hat{x}_t^2 \sum_{t=1}^{n} \hat{y}_t^2} \\ &\le \frac{n+1}{2} \sqrt{\sum_{k=1}^{n} x_k^2 \sum_{l=1}^{n} y_l^2} \le \frac{n^2}{2} + O(n). \end{aligned}$$

Similarly, the second term is at most $n^2/2 + O(n)$. Hence

$$\|W\|_C = \Big|\sum_{k\in I;\, l\in J} w_{kl}\Big| \le n^2 + O(n) \le \frac{\pi S^+}{\log n}(1+o(1)).$$

This finishes the proof. □

We presented a matrix W with real entries for which bound (4) holds. This matrix corresponds to a directed graph with weighted edges. However, it can be transformed to a matrix with entries -1, 0 and 1, which corresponds to an unweighted directed graph.

Corollary 4.2. *There exists a matrix $\tilde{W}$ with entries -1, 0 and 1 that satisfies bound (4).*

Proof. Let W be the matrix from Theorem 4.1. By scaling, we may assume that the largest entry in W equals 1 in absolute value. Let $N = 4n^4$. We construct the matrix $\tilde{W}$ by replacing each entry w_{ij} of W with an $N \times N$ block matrix R^{ij} that has the following properties. First, each entry of R^{ij} is either -1, 0 or 1. Second, for every two sets $A, B \subset \{1, \dots, N\}$,

$$\frac{1}{N^2}\left|\sum_{k\in A,\, l\in B} R^{ij}_{kl} - w_{ij}|A||B|\right| < \frac{2}{\sqrt{N}} = \frac{1}{n^2}. \quad (5)$$

We prove that such matrix R^{ij} exists using the probabilistic method (see Alon and Berger [1] for a similar argument). Let every entry of R^{ij} be equal to $\mathrm{sgn}(w_{ij})$ with probability $|w_{ij}|$ and equal to 0 with probability $1-|w_{ij}|$. Then, by the Chernoff bound, for fixed sets $A, B \subset \{1, \dots, N\}$, we have

$$\Pr\left(\left|\sum_{k\in A,\, l\in B} R^{ij}_{kl} - w_{ij}|A||B|\right| > \frac{2N^2}{\sqrt{N}}\right) < 2\exp\left(-2\cdot(2N^{3/2})^2/N^2\right) \le 2e^{-8N}.$$

Since there are 2^{2N} distinct pairs of sets A and B, and $2^{2N}\cdot 2e^{-8N} < 1$, there exists a matrix R^{ij} that satisfies inequality (5) for all sets A and B simultaneously. (To ensure that the matrix $\tilde{W}$ is skew-symmetric, we use this argument to find matrices R^{ij} for $i<j$; we let $R^{ji} = -(R^{ij})^T$.)

We verify that the matrix $\tilde{W}$ satisfies bound (4). Let us estimate the cut norm $\|\tilde{W}\|_C$. Let A and B be the sets of indices such that

$$\|\tilde{W}\|_C = \left|\sum_{k\in A,\, l\in B} \tilde{w}_{kl}\right|.$$

Denote the restriction of A to the set of row indices of submatrices R^{i*} by A_i; denote the restriction of B to the set of column indices of submatrices R^{*j} by B_j. Then

$$\begin{aligned}
\|\tilde{W}\|_C &= \left|\sum_{\substack{1\le i\le n\\ 1\le j\le n}} \sum_{\substack{k\in A_i\\ l\in B_j}} R^{ij}_{kl}\right| \\
&\le N^2\left|\sum_{\substack{1\le i\le n\\ 1\le j\le n}} w_{ij}\frac{|A_i|}{N}\frac{|B_j|}{N}\right| + N^2\sum_{\substack{1\le i\le n\\ 1\le j\le n}}\frac{1}{n^2} \\
&\le N^2\left(\|W\|_C + 1\right) \le 2N^2\|W\|_C.
\end{aligned}$$

Here, first, we used that, $0 \le |A_i|/N \le 1$ and $0 \le |B_j|/N \le 1$, so $\sum_{ij} w_{ij}\cdot|A_i|/N\cdot|B_j|/N$ does not exceed the cut norm; second, we used that at least one entry in W equals to 1 in absolute value, so $\|W\|_C \ge 1$. On the other hand, we can estimate $S^+(\tilde{W})$ as follows (assume without loss of generality that $S^+(W) \ge 0$)

$$\begin{aligned}
|S^+(\tilde{W})| &= \sum_{1\le i<j\le n}\sum_{k,l} R^{ij}_{kl} \\
&\ge \sum_{1\le i<j\le n} N^2(w_{ij} - 1/n^2) \\
&\ge N^2(S^+(W) - 1/2) \\
&\ge N^2\, S^+(W)(1-o(1)).
\end{aligned}$$

Here we used that $S^+(W) = \Omega(\log n)\|W\|_C = \Omega(\log n) = \omega(1)$. Combining the bounds for $\|\tilde{W}\|_C$ and $|S^+(\tilde{W})|$ with bound (4), we get

$$\|\tilde{W}\|_C \le O(S^+(\tilde{W})/\log n).$$

This concludes the proof. □

References

[1] N. Alon and E. Berger. The Grothendieck constant of random and pseudo-random graphs. Discrete Optimization, to appear.

[2] N. Alon and A. Naor. Approximating the Cut-Norm via Grothendieck's Inequality. Proceedings of the thirty sixth Annual ACM Symposium on Theory of Computing, pp. 72–80, 2004.

[3] G. Andersson, L. Engebretsen, and J. Håstad. A New Way of Using Semidefinite Programming with Applications to Linear Equations mod p. Journal of Algorithms, vol. 39, pp. 162–204, 2001.

[4] B. Berger and P. Shor. Approximation algorithms for the maximum acyclic subgraph problem. Proceedings of the first Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 236–243, 1990.

[5] M. Charikar, K. Makarychev, and Y. Makarychev. Near-Optimal Algorithms for Maximum Constraint Satisfaction Problems. Proceedings of the eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 62–68, 2007.

[6] M. Charikar and A. Wirth. Maximizing quadratic programs: extending Grothendieck's inequality. Proceedings of the forty fifth Annual IEEE Symposium on Foundations of Computer Science, pp. 54–60, 2004.

[7] L. Engebretsen and V. Guruswami. Is constraint satisfaction over two variables always easy? Random Struct. Algorithms, vol. 25, no. 2, pp. 150–178, 2004.

[8] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[9] R. Hassin and S. Rubinstein. Approximations for the maximum acyclic subgraph problem. Source Information Processing Letters, vol. 51, issue 3, pp. 133–140, 1994.

[10] G. Hast. Approximating MAX kCSP - Outperforming a random assignment with almost a linear factor. Proceedings of the thirty second International Colloquium on Automata, Languages and Programming, pp. 956–968, 2005.

[11] G. Hast. Beating a Random Assignment. APPROX 2005, pp. 134–145.

[12] J. Håstad. Every 2-CSP allows nontrivial approximation. Proceedings of the thirty-seventh Annual ACM Symposium on Theory of Computing, pp. 740–746, 2005.

[13] J. Håstad and V. Srinivasan. On the advantage over a random assignment. Random structures and Algorithms, vol. 25:2, pp. 117–149, 2004.

[14] V. Kann. On the Approximability of NP-complete Optimization Problems. Ph. D. thesis, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.

[15] S. Khot and R. O'Donnell. SDP gaps and UGC-hardness for MaxCutGain. Proceedings of the 47^{th} Annual IEEE Symposium on Foundations of Computer Science, pp. 217–226, 2006.

[16] A. Newman. The Maximum Acyclic Subgraph Problem and Degree-3 Graphs. Proceedings of APPROX 2001, pp. 147–158.

[17] A. Newman. Cuts and Orderings: On Semidefinite Relaxations for the Linear Ordering Problem. Proceedings of APPROX 2004, pp. 195–206.

[18] A. Newman and S. Vempala. Fences Are Futile: On Relaxations for the Linear Ordering Problem. Proceedings of the 8^{th} Conference on Integer Programming and Combinatorial Optimization, pp. 333–347, 2001.

[19] C. H. Papadimitriou and M. Yannakakis. Optimization, Approximation, and Complexity Classes. J. Comput. System Sci. 43, pp. 425–440, 1991.

[20] P. D. Seymour. Packing directed circuits fractionally. Combinatorica (15), pp. 281–288, 1995.

5 Appendix

In the Appendix, we remind the reader the definition of the discrete Fourier sine transform and prove the inversion formula. Then we verify two equalities (Corollary 5.2 and Lemma 5.3) that we used in the paper.

Let f be a function from $\{1,\dots,n-1\}$ to $\mathbb{R}$. Then its discrete Fourier sine transform is defined as

$$\hat{f}(t) = \sum_{k=1}^{n-1} \sin\Big(\frac{\pi k t}{n}\Big) f(k).$$

The inversion formula is given by

$$f(k) = \frac{2}{n}\sum_{t=1}^{n-1} \sin\Big(\frac{\pi k t}{n}\Big) \hat{f}(t).$$

To prove this formula we need to verify that the functions

$$k \mapsto \sin(\pi k t/n)$$

form an orthonormal basis, i.e.

$$\frac{2}{n}\sum_{k=1}^{n-1} \sin\Big(\frac{\pi k s}{n}\Big)\sin\Big(\frac{\pi k t}{n}\Big) = \begin{cases} 1, & \text{if } s = t; \\ 0, & \text{if } s \neq t. \end{cases} \tag{6}$$

We use the following simple lemma.

Lemma 5.1. *For every $n > 1$ and $0 < t < n$,*

$$\sum_{k=0}^{n-1} e^{\frac{\pi k t i}{n}} = \begin{cases} 1 + i\cot\frac{\pi t}{2n}, & t \text{ is odd;} \\ 0, & t \text{ is even.} \end{cases}$$

Proof. We have

$$\sum_{k=0}^{n-1} e^{\frac{\pi kti}{n}} = \sum_{k=0}^{n-1} \left(e^{\frac{\pi ti}{n}}\right)^k = \frac{1-e^{\pi ti}}{1-e^{\frac{\pi ti}{n}}}.$$

If t is even, then $e^{\pi ti} = 1$ and we are done. Otherwise,

$$\frac{1-e^{\pi ti}}{1-e^{\frac{\pi ti}{n}}} = \frac{2}{1-e^{\frac{\pi ti}{n}}} = \frac{2(1-e^{-\frac{\pi ti}{n}})}{2(1-\mathrm{Re}(e^{\frac{\pi ti}{n}}))} \tag{7}$$

$$= 1 + i\frac{\sin\frac{\pi t}{n}}{1-\cos\frac{\pi t}{n}} = 1 + i\cot\frac{\pi t}{2n}.$$

□

Corollary 5.2. *For every $n > 1$ and $0 < t < n$,*

$$\sum_{k=0}^{n-1} \sin\frac{\pi kt}{n} = \begin{cases} \cot\frac{\pi t}{2n}, & t \text{ is odd;} \\ 0, & t \text{ is even.} \end{cases}$$

$$\sum_{k=0}^{n-1} \cos\frac{\pi kt}{n} = \begin{cases} 1, & t \text{ is odd;} \\ 0, & t \text{ is even.} \end{cases}$$

Proof of expression (6):

$$\frac{2}{n}\sum_{k=1}^{n-1} \sin\Big(\frac{\pi ks}{n}\Big)\sin\Big(\frac{\pi kt}{n}\Big) = \frac{2}{n}\sum_{k=0}^{n-1} \sin\Big(\frac{\pi ks}{n}\Big)\sin\Big(\frac{\pi kt}{n}\Big)$$

$$= \frac{1}{n}\sum_{k=0}^{n-1}\cos\frac{\pi(s-t)k}{n} - \frac{1}{n}\sum_{k=0}^{n-1}\cos\frac{\pi(s+t)k}{n}.$$

If $s \neq t$, then the sums above are equal by Corollary 5.2. Hence the whole expression is equal to zero. If $s = t$, then

$$\frac{2}{n}\sum_{k=1}^{n-1} \sin\Big(\frac{\pi ks}{n}\Big)\sin\Big(\frac{\pi kt}{n}\Big) = 1 - \frac{1}{n}\sum_{k=0}^{n-1}\cos\frac{2\pi\cdot kt}{n} = 1.$$

This finishes the proof of the inversion formula. □

Lemma 5.3. *For every $n > 1$ and $1 \le t \le n$,*

$$\sum_{k=0}^{n-1}(n-k)\sin\frac{\pi kt}{n+1} = \frac{n+(-1)^t}{2\tan\frac{\pi t}{2(n+1)}}.$$

Proof. We have

$$\sum_{k=0}^{n-1}(n-k)\sin\frac{\pi kt}{n+1} =$$

$$= \mathrm{Im}\left(\sum_{k=0}^{n-1}(n-k)\, e^{\frac{\pi kti}{n+1}}\right) = \mathrm{Im}\left(\sum_{l=0}^{n-1}\sum_{k=0}^{l} e^{\frac{\pi kti}{n+1}}\right)$$

$$= \mathrm{Im}\left(\sum_{l=0}^{n-1}\frac{1-e^{\frac{\pi t(l+1)i}{n+1}}}{1-e^{\frac{\pi ti}{n+1}}}\right) = \mathrm{Im}\left(\frac{n-\sum_{l=1}^{n} e^{\frac{\pi tli}{n+1}}}{1-e^{\frac{\pi ti}{n+1}}}\right)$$

$$= \mathrm{Im}\left(\frac{n+1}{1-e^{\frac{\pi ti}{n+1}}} - \frac{1-e^{\pi ti}}{\left(1-e^{\frac{\pi ti}{n+1}}\right)^2}\right).$$

Similarly to (7), we get

$$\mathrm{Im}\left(\frac{n+1}{1-e^{\frac{\pi ti}{n+1}}}\right) = \frac{n+1}{2\tan\frac{\pi t}{2(n+1)}};$$

and

$$\mathrm{Im}\frac{1-e^{\pi ti}}{\left(1-e^{\frac{\pi ti}{n+1}}\right)^2} = \frac{\mathrm{Im}((1+(-1)^{t+1})(1-e^{-\frac{\pi ti}{n+1}})^2)}{\left(1-e^{\frac{\pi ti}{n+1}}\right)^2\left(1-e^{-\frac{\pi ti}{n+1}}\right)^2}$$

$$= \frac{2(1+(-1)^{t+1})(1-\cos\frac{\pi t}{n+1})\sin\frac{\pi t}{n+1}}{(2-2\cos\frac{\pi t}{n+1})^2}$$

$$= \frac{(1+(-1)^{t+1})\sin\frac{\pi t}{n+1}}{2(1-\cos\frac{\pi t}{n+1})} = \frac{1+(-1)^{t+1}}{2\tan\frac{\pi t}{2(n+1)}}.$$

□

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/