Constructing worst case instances for semidefinite programming based approximation algorithms

Noga Alon [*] *Benny Sudakov* [†] *Uri Zwick* [‡]

Abstract

Semidefinite programming based approximation algorithms, such as the Goemans and Williamson approximation algorithm for the MAX CUT problem, are usually shown to have certain performance guarantees using local ratio techniques. Are the bounds obtained in this way tight? This problem was considered before by Karloff and by Alon and Sudakov. Here we further extend their results and show, for the first time, that the local analyses of the Goemans and Williamson MAX CUT algorithm, as well as its extension by Zwick, are tight for every possible relative size of the maximum cut in the sense that the expected value of the solutions obtained by the algorithms may be as small as the analyses ensure. We also obtain similar results for a related problem. Our approach is quite general and could possibly be applied to some additional problems and algorithms.

1 Introduction

MAX CUT is one of the most natural combinatorial optimization problems. An instance of MAX CUT is a graph. The goal is to partition the vertices of the graph into two sets such that the number, or the total weight, of the edges that cross the cut formed by this partition is maximized. Goemans and Williamson [GW95] describe an elegant approximation algorithm for the MAX CUT problem and show that its performance guarantee is at least $\alpha = \min_{0<\theta\le\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta} = 0.87856\ldots$. No polynomial time approximation algorithm for MAX CUT can have a performance ratio of more than $\frac{16}{17}$, unless P=NP (Håstad [Hås97], Trevisan et al. [TSSW96]).

The MAX CUT approximation algorithm of Goemans and Williamson [GW95] uses a *semidefinite programming* relaxation of the problem. In this relaxation, every vertex i of the graph has a unit vector $v_i \in R^n$ associated with it. The algorithm solves this relaxation and then uses a simple randomized rounding technique to convert the constellation of unit vectors obtained into a cut. To get a lower bound on the performance ratio of the algorithm, Goemans and Williamson consider the worst possible ratio between the probability that a given edge is in the cut, and the contribution of that edge to the optimal value of the semidefinite program. This worst case local ratio is attained when the angle θ between the two vectors v_i and v_j that correspond to the two endpoints of the edge is equal to $\theta_0 = \operatorname{argmin}_{0<\theta\le\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta} \simeq 2.331122\ldots$.

[*]Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. Email: noga@math.tau.ac.il. Research supported in part by a USA Israeli BSF grant and by a grant from the Israel Science Foundation.

[†]Department of Mathematics, Princeton University, Princeton, NJ 08540, USA and Institute for Advanced Study, Princeton, NJ 08540, USA. Email address: bsudakov@math.princeton.edu. Research supported in part by NSF grants DMS-0106589, CCR-9987845 and by the State of New Jersey.

[‡]Department of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. Email: zwick@cs.tau.ac.il.

Is the local analysis of the MAX CUT approximation algorithm of Goemans and Williamson [GW95] globally tight? In other words, are there graphs for which the optimal value of the relaxation is equal to the size of the maximum cut, *and* for which there is an optimal solution of the relaxation in which the angle between every two vectors that correspond to vertices in the graph that are connected by an edge is exactly, or very close to, θ_0? Karloff [Kar99] was the first to construct graphs that satisfy these conditions and show, therefore, that the local analysis of the MAX CUT approximation algorithm of Goemans and Williamson is indeed tight. Karloff's result was simplified by Alon and Sudakov [AS00].

Goemans and Williamson [GW95] give a better lower bound on the performance guarantee of their algorithm for graphs that have relatively large cuts. More specifically, for a graph $G = (V, E)$, let A be the *relative size* of the maximum cut of G, i.e., the ratio between the size (or weight) of the maximum cut and the number of edges (or total weight of the edges) of G. Note that $1/2 < A \leq 1$. It is shown in [GW95] that if $A > t_0 \simeq 0.84458$, where $t_0 = \text{argmin}_{0<t\leq 1} h(t)/t$ and $h(t) = \arccos(1-2t)/\pi$, then the performance ratio of the MAX CUT algorithm is at least $\alpha(A) = h(A)/A > \alpha$. Karloff [Kar99] and Alon and Sudakov [AS00] show that this lower bound is again tight, for every $t_0 \leq A \leq 1$.

What happens on graphs with $1/2 < A < t_0$? Goemans and Williamson [GW95] can only show that the performance ratio of their algorithm on such graphs is at least α. Zwick [Zwi99] presents a modification of the algorithm of Goemans and Williamson [GW95] that has a performance guarantee $\alpha'(A)$ strictly larger than $\alpha \simeq 0.87856$ for any $1/2 < A < t_0$. Furthermore, $\alpha'(A)$ approaches 1 as A decreases towards $1/2$.

In this paper we show, among other things, that the local analysis of the algorithms of Goemans and Williamson [GW95] and of Zwick [Zwi99] in the range $1/2 < A \leq t_0$ is again tight. Showing that the analysis of the MAX CUT algorithm is tight in the range $1/2 < A \leq t_0$ is a more challenging task than the corresponding task for the range $t_0 \leq A \leq 1$. To accomplish this task we construct, for any rational $-1 < \eta < 0$ and any rational $\frac{1}{2} < a \leq \frac{1-\eta}{2}$, a graph $G = (V, E)$ for which the size of the maximum cut is exactly $a|E|$, the optimal value of the relaxation is also equal to $a|E|$, *and* for which there is an optimal solution $v_1, v_2, \ldots v_n$ of the relaxation such that for every $\{i, j\} \in E$ we have either $v_i \cdot v_j = \eta$ or $v_i \cdot v_j = 1$. (Note that the requirement that the value of the relaxation be $a|E|$ determines the proportion of the edges for which the inner product should be $v_i \cdot v_j = \eta$.)

The graphs used by Alon and Sudakov [AS00] to show that the analysis of the MAX CUT algorithm is tight in the range $t_0 \leq A \leq 1$ are graphs arising from *Hamming association schemes* over the binary alphabet. (Karloff [Kar99] uses the related Kneser graphs.) The graphs we use here to show that the analysis is also tight in the range $\frac{1}{2} < A \leq t_0$ are obtained by composing Hamming graphs and *expander* graphs. More specifically, if H is an appropriate Hamming graph and B is an appropriate bipartite expander with b vertices on each of its sides, then the graph that we use is obtained by replacing each vertex of H by a clique on b vertices and replacing each edge of H by a copy of B.

We believe that the technique developed in this paper could be used to construct worst case instances for other semidefinite programming based approximation algorithms. To demonstrate it, we use our technique to show that local analysis of the MAX NAE-{3}-SAT algorithm of Zwick [Zwi99] is also tight. This is an even more demanding task, as will be explained later.

An instance of MAX NAE-{3}-SAT in the variables $x_1, x_2, \ldots, x_n$ is a weighted collection of triplets of the form $\langle z_1, z_2, z_3 \rangle$, where each z_i is a *literal*, i.e., one of the variables $x_1, x_2, \ldots, x_n$ or a negation of one of the variables, and the weights are non-negative. The three literals appearing in a triplet must be distinct. A triplet (z_1, z_2, z_3) is satisfied by an assignment of 0-1 values to the variables $x_1, x_2, \ldots, x_n$ if at least one of the literals in the triplet is assigned the value 0 and at least one is assigned the value 1. MAX NAE-{3}-SAT is an interesting problem as it can be seen as a generalization of both MAX CUT and of the problem of finding a maximum cut in 3-uniform hypergraphs.

The rest of this paper is organized as follows. In the next section we quickly review the MAX CUT approximation algorithm of Goemans and Williamson [GW95] and its extension by Zwick [Zwi99]. In Section 3 we present the construction of the graphs that show that the local analysis of the MAX CUT algorithms of [GW95] and [Zwi99] are tight. In Section 4 we then review the MAX NAE-{3}-SAT approximation algorithm of Zwick [Zwi99]. In Section 5 we modify the construction of Section 3 to show that the local analysis of the MAX NAE-{3}-SAT approximation algorithm is again tight. We end in Section 6 with some concluding remarks and open problems.

It is worth noting that our results here merely show that the analyses of the algorithms discussed are tight, and do not exclude the possibility that these algorithms (or some variants of them) may have a better performance either by showing that with non-negligible probability the rounding will output a solution that exceeds the expectation significantly or by proving that one can obtain other solutions to the corresponding semi-definite programs; solutions that may behave better in the rounding phase. Yet, the results here do show that some essentially novel ideas will be needed in order to improve the performance guarantees of the algorithms discussed.

2 Approximation algorithms for the MAX CUT problem

Let $G = (V, E)$ be a graph, where $V = \{1, \ldots, n\}$. We let $OPT(G)$ denote the size of the maximum cut of G. The Goemans and Williamson approximation algorithm for MAX CUT starts by solving the following semidefinite programming relaxation of the problem:

$$\max_{\|v_i\|^2=1} \sum_{\{i,j\}\in E} \frac{1 - v_i^t v_j}{2} ,$$

where each v_i ranges over all n-dimensional unit vectors. (All our vectors are considered to be column vectors and hence $v^t u$ is simply the inner product of v and u.) It is easy to see that the optimal value z^* of this program is at least as large as $OPT(G)$, the size of the maximum cut of G.

The algorithm of Goemans and Williamson [GW95] then rounds an optimal solution $v_1, \ldots, v_n$ of the semidefinite program by choosing a random unit vector r and defining $S = \{i \mid r^t v_i \leq 0\}$. This supplies a cut $(S, V{-}S)$ of the graph G. Let W denote the size of the random cut produced in this way and let $E[W]$ be its expectation. By linearity of expectation, the expected size is the sum, over all $\{i,j\} \in E$, of the probabilities that the vertices i and j lie in opposite sides of the cut. This last probability is precisely $\arccos(v_i^t v_j)/\pi$. Thus the expected value of the weight of the random cut is exactly $\sum_{\{i,j\}\in E} \frac{\arccos(v_i^t v_j)}{\pi}$. However, the optimal value z^* of the semidefinite program is equal to $z^* = \sum_{\{i,j\}\in E} \frac{1-v_i^t v_j}{2}$. Therefore the ratio between $E[W]$ and the optimal value z^* satisfies

$$\frac{E[W]}{z^*} = \frac{\sum_{\{i,j\}\in E} \arccos(v_i^t v_j)/\pi}{\sum_{\{i,j\}\in E}(1 - v_i^t v_j)/2} \geq \min_{\{i,j\}\in E} \frac{\arccos(v_i^t v_j)/\pi}{(1 - v_i^t v_j)/2} .$$

Denote $\alpha = \min_{0<\theta\leq\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}$. An easy computation shows that the minimum α is attained at $\theta = \theta_0 = 2.3311..$, the nonzero root of $\cos\theta + \theta\sin\theta = 1$, and that $\alpha \in (0.87856, 0.87857)$. Thus, $E[W] \geq \alpha \cdot z^*$, and since the value of z^* is at least as large as the weight OPT of the maximum cut, we conclude that $E[W] \geq \alpha \cdot OPT$. It follows that the Goemans-Williamson algorithm supplies an α-approximation for MAX CUT. Moreover, by the above discussion, the expected size of the cut produced by the algorithm is not better than $\alpha \cdot OPT$ if $OPT = z^*$ and if $v_1, \ldots, v_n$ is an optimal solution of the semidefinite program that satisfies $\frac{\arccos(v_i^t v_j)/\pi}{(1-v_i^t v_j)/2} = \alpha$ for every $\{i,j\} \in E$.

If the value of the semidefinite program is a large fraction of the total number of edges of G, the above reasoning, together with a simple convexity argument, is used in [GW95] to show that the performance of the algorithm is better. Let $h(t) = \arccos(1-2t)/\pi$ and let t_0 be the value of t for which $h(t)/t$ attains its minimum in the interval $(0,1]$. Then t_0 is approximately 0.84458. Define $a = z^*/|E|$. If $a \geq t_0$ then, as shown in [GW95], $E[W] \geq \frac{h(a)}{a} z^* \geq \frac{h(a)}{a} OPT$. Note that $A = OPT/|E| \leq a$. As $h(a)/a$ is an increasing function of a, for $A \geq t_0$, we have also that $E[W] \geq \frac{h(A)}{A} OPT$. Here, as before, the actual expected size of the cut produced by the algorithm is not better than $\frac{h(a)}{a} OPT$ if $OPT = z^*$ and if $v_1, \ldots, v_n$ is an optimal solution of the semidefinite programming problem that satisfies $v_i^t v_j = 1 - 2a$ for every $\{i,j\} \in E$.

Karloff [Kar99] and Alon and Sudakov [AS00] showed that the analysis of the algorithm of Goemans and Williamson [GW95] is tight for every $t_0 \leq a \leq 1$. More precisely, for any rational $t_0 \leq a \leq 1$, there are infinitely many graphs for which the size of the maximum cut OPT is equal to z^* and also $E[W] = (h(a)/a)z^* = (h(a)/a)OPT$. In the next section we extend this result even further and show that the analysis of the algorithm of Goemans and Williamson is tight for all $1/2 \leq a \leq 1$.

To show that the analysis of Goemans and Williamson [GW95] is also tight in the range $1/2 \leq a \leq t_0$, we construct, for every rational a in this range, an infinite sequence of graphs for which the size OPT of the maximum cut, and the optimal value z^* of the relaxation are both $a|E|$, and for which the relaxation has an optimal solution $v_1, v_2, \ldots, v_n$ such that for every $\{i,j\} \in E$ we have either $v_i^t v_j = \cos\theta_0$, or $v_i^t v_j = 1$. Indeed, the randomized rounding for such a solution satisfies $E[W] = \frac{h(t_0)}{t_0} z^* = \frac{h(t_0)}{t_0} OPT$, as for any edge ij for which $\frac{1-v_i^t v_j}{2} \neq 0$ we have $v_i^t v_j = \cos\theta_0$.

Zwick [Zwi99] describes a modification of the algorithm of Goemans and Williamson [GW95] that has a better performance guarantee in the range $1/2 \leq a \leq t_0$. His algorithm works as follows. After solving the relaxation and obtaining a, which is assumed to satisfy $a < t_0$, the algorithm finds the unique solutions $c = c(a)$ and $t = t(a)$ of the following two equations:

$$\frac{\arccos(c(1-2t)) - \arccos(c)}{t} = \frac{2c}{\sqrt{1-c^2(1-2t)^2}} \quad , \quad \frac{1-\frac{t}{a}}{\sqrt{1-c^2}} = \frac{1-2t}{\sqrt{1-c^2(1-2t)^2}} .$$

The algorithm then constructs a sequence of unit vectors $w_1, w_2, \ldots, w_n$ such that $w_i^t w_j = c(v_i^t v_j)$, for every $i \neq j$. The vectors $w_1, w_2, \ldots, w_n$, and not the vectors $v_1, v_2, \ldots, v_n$, are then rounded using a random hyperplane. It is shown in [Zwi99], using local analysis, that the performance ratio achieved by this algorithm is at least

$$\alpha'(a) = \left(\frac{1}{a} - \frac{1}{t(a)}\right) h_{c(a)}(0) + \frac{h_{c(a)}(t(a))}{t(a)} ,$$

where $h_c(t) = \arccos(c(1-2t))/\pi$. It is also shown there that this analysis is tight if the size of the maximum cut in the graph is $a|E|$ and if for every $\{i,j\} \in E$ we have either $v_i^t v_j = 1 - 2t(a)$ or $v_i^t v_j = 1$. It is not difficult to see that $a < t(a)$, for every $1/2 < a < t_0$.

3 Worst case instances for the MAX CUT algorithms

In this section we prove the following theorem

Theorem 3.1 *Let $-1 < \eta < 0$ and $\frac{1}{2} < a \leq \frac{1-\eta}{2}$ be rational numbers. Then, for infinitely many values of n there exists a graph $G = (V, E)$, $V = \{1, \ldots, n\}$ and a sequence $u_1, u_2, \ldots, u_n$ of unit vectors such that either $u_i^t u_j = \eta$ or $u_i^t u_j = 1$ for all $\{i, j\} \in E$, and the size of maximum cut in G is equal to*

$$\max_{\|v_i\|^2=1, v_i \in R^n} \sum_{\{i,j\} \in E} \frac{1 - v_i^t v_j}{2} = \sum_{\{i,j\} \in E} \frac{1 - u_i^t u_j}{2} = a|E| \ .$$

By the discussion in the previous section, it follows that the analyses of the algorithms of Goemans and Williamson [GW95] and of Zwick [Zwi99] are tight also in the range $1/2 \leq a \leq t_0$.

To prove Theorem 3.1 we first need to establish a connection between the smallest eigenvalue of a graph and the semidefinite relaxation of the MAX CUT problem. This is done in the following well known lemma, whose proof we include here for the sake of completeness.

Lemma 3.2 *Let G be a multigraph on the set $V = \{1, 2, \ldots, n\}$, with adjacency matrix $A = (a_{ij})$, where a_{ij} corresponds to the multiplicity of the edge between i and j. Let $\lambda_1 \geq \ldots \geq \lambda_n$ be the eigenvalues of $A = (a_{ij})$. Then*

$$\sum_{i<j} a_{ij} \frac{1 - v_i^t v_j}{2} \ \leq \ \frac{1}{2}|E(G)| - \frac{1}{4}\lambda_n \cdot |V(G)| \ = \ \frac{1}{2}|E(G)| - \frac{1}{4}\lambda_n \cdot n \ ,$$

for any set $v_1, \ldots, v_n$ of unit vectors in $R^k, k > 0$. In addition let $B = (b_{ij})$ be the $n \times k$ matrix whose rows are the vectors $v_1^t, \ldots, v_n^t$. Then equality holds if and only if each column of B is an eigenvector of A with eigenvalue λ_n.

Note that for every loopless graph G with edges, $\lambda_n < 0$, as the sum of the eigenvalues is the trace of A, which is 0.

Proof. Let $y_1, \ldots, y_k$ be the columns of B. By definition we have $\sum_{i=1}^k \|y_i\|^2 = \sum_{ij} b_{ij}^2 = \sum_{i=1}^n \|v_i\|^2 = n$. Therefore

$$\sum_{i<j} a_{ij} \frac{1 - v_i^t v_j}{2} \ = \ \frac{1}{2}|E| - \frac{1}{2}\sum_{i<j} a_{ij} v_i^t v_j \ = \ \frac{1}{2}|E| - \frac{1}{4}\sum_{i=1}^k y_i^t A y_i \ .$$

By the variational definition of the eigenvalues of A, for any vector $z \in R^n$, $z^t A z \geq \lambda_n \|z\|^2$ and equality holds if and only if $Az = \lambda_n z$. This implies that

$$\sum_{i<j} a_{ij} \frac{1 - v_i^t v_j}{2} \ \leq \ \frac{1}{2}|E| - \frac{1}{4}\lambda_n \sum_{i=1}^k \|y_i\|^2 \ = \ \frac{1}{2}|E| - \frac{1}{4}\lambda_n \cdot n \ .$$

Equality holds in the last expression if and only if each y_i is an eigenvector of A with eigenvalue λ_n. □

The main ingredient of our constructions are graphs arising from the Hamming Association Scheme over the binary alphabet. Let $V = \{v_1, v_2, \ldots\}$ be the set of all vectors of length m over the alphabet $\{-1, +1\}$. For any two vectors $x, y \in V$ denote by $d(x, y)$ their *Hamming distance*, that is, the number of coordinates in which they differ. The *Hamming graph* $H = H(m, b)$ is the graph whose vertex set is V and in which two vertices $x, y \in V$ are adjacent if and only if $d(x, y) = b$. Here we consider only even values of b which are greater than $m/2$. We may and will assume, whenever this is needed, that m is sufficiently large.

Consider any two adjacent vertices of $H(m, b)$, v_i and v_j. By the definition of H, the inner product $v_i^t v_j$ is $m - 2b$. Choose m and b such that $b > m/2$ is even and $\frac{m-2b}{m} = \eta$. This is always possible since η is a rational number, $-1 < \eta < 0$. Let $w_i = \frac{1}{\sqrt{m}} v_i$ for all i; thus $\|w_i\|^2 = 1$ and $w_i^t w_j = \eta$ for any pair of adjacent vertices.

Note that by definition, $H(m,b)$ is the Cayley graph of the multiplicative group $Z_2^m = \{-1,+1\}^m$ with respect to the set U of generators formed by all vectors with exactly b coordinates equal to -1, where vectors in the group multiply coordinate-wise. Therefore (see, e.g., [Lov93], Problem 11.8 and the hint to its solution) the eigenvectors of the adjacency matrix of $H(m,b)$ are the multiplicative characters χ_I of Z_2^m, where $\chi_I(x) = \prod_{i \in I} x_i$, I ranges over all subsets of $\{1,\ldots,m\}$. The eigenvalue corresponding to χ_I is $\sum_{x \in U} \chi_I(x)$. The eigenvalues of H are thus equal to the so called *binary Krawtchouk polynomials* (see [CHLL97]),

$$P_b^m(k) \; = \; \sum_{j=0}^{k} (-1)^j \binom{k}{j} \binom{m-k}{b-j} \; , \quad 0 \le k \le m \; . \tag{1}$$

The eigenvalue $P_b^m(k)$ corresponds to the characters χ_I with $|I| = k$ and thus has multiplicity $\binom{m}{k}$. Since $H(m,b)$ is a regular graph with degree $d = \binom{m}{b}$, it's largest eigenvalue is equal to d and its corresponding eigenvector is $(1,1,\ldots,1)$. In addition it was proved in [AS00] that if m big enough, then the smallest eigenvalue of $H(m,b)$ is $\lambda = P_b^m(1) = \frac{m-2b}{m}\binom{m}{b}$. By the above discussion this eigenvalue has multiplicity $\binom{m}{1} = m$ and eigenvectors $y_1,\ldots,y_m$ with ± 1 coordinates, where for each vertex $v_j = (v_{j1},\ldots,v_{jm})$, $y_i(v_j) = v_{ji}$. Therefore, the columns of the matrix, whose rows are the vectors w_i, are the eigenvectors $\frac{1}{\sqrt{m}} y_i$ of $A(H)$ corresponding to the eigenvalue λ.

Let $A = (a_{ij})$ be an $s \times s$ matrix and $B = (b_{pq})$ be a $t \times t$ matrix, then the *tensor product* of A and B is the $st \times st$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \ldots & a_{1s}B \\ a_{21}B & a_{22}B & \ldots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1}B & a_{s2}B & \ldots & a_{ss}B \end{pmatrix}$$

We need the following well known properties of eigenvalues and eigenvectors of tensor products of matrices.

Lemma 3.3 *Let A be a square matrix of order s with eigenvalues $\alpha_1,\ldots,\alpha_s$ and eigenvectors $e_1,\ldots,e_s$ and let B be a square matrix of order t with eigenvalues $\beta_1,\ldots,\beta_t$ and eigenvectors $f_1,\ldots,f_t$. Then the eigenvalues of the matrix $A \otimes B$ are equal to $\alpha_i\beta_j, i = 1,\ldots,s, j = 1,\ldots,t$ and their corresponding eigenvectors are $e_i \otimes f_j$.* $\square$

We also need the following result.

Lemma 3.4 *For every two integers $0 < Y < X$ and every integer $L > 0$, there is an integer g such that L divides the binomial coefficient $\binom{gX}{gY}$.*

Proof. If S,T are two positive integers and R is their sum, then for every prime p the maximum power of p that divides $\binom{R}{S} = \frac{R!}{S!T!}$ is p^h, where $h = \sum_{i \ge 1}(\lfloor R/p^i \rfloor - \lfloor S/p^i \rfloor - \lfloor T/p^i \rfloor)$. For each i, the i-th term in the sum above is either 0 or 1, and it is 1 if and only if there is a carry in the i-th rightmost digit when S and T are represented in base p and are being added to get R. Therefore, in order to prove the lemma it suffices to show that for every finite collection of primes P and for every positive integer u the following holds. There is an integer g such that for $Z = X - Y$, and for every $p \in P$, when gY and gZ are being added in base p there is a carry in at least u places. We proceed with a proof of this fact.

Fix a prime $p \in P$, and consider the representations of Y and Z in base p. If the rightmost non-zero digit in both of them appears at the same place, then there is some $g_1 > 0$ such that the rightmost non-zero digit of g_1Y is $p-1$, and as the digit of g_1Z in the same place is nonzero as well, there will be a carry in this position while the two numbers will be added. Otherwise assume, without loss of generality, that the rightmost nonzero digit of Y is to the right of the rightmost nonzero digit of Z. Choose $g_1 > 0$ such that the rightmost nonzero digit of g_1Z is $p-1$. If the digit of g_1Y in this position is nonzero, then when adding g_1Y and g_1Z there will be a carry here. Otherwise, by defining $g_1' = g_1(1+p^s)$, where s is chosen so that the right most nonzero digit of g_1Yp^s is at the same place as the rightmost nonzero digit of g_1Z we conclude that when adding $g_1'Y$ and $g_1'Z$ we have a carry in this place. We have thus shown that in all cases there is some positive g_1 such that when adding g_1Y and g_1Z there is a carry in at least one position. To get a carry in at least u positions we now take a sufficiently large integer m and define $g_p = g_1(1+p^m+p^{2m}+\ldots+p^{(u-1)m})$. If m is sufficiently large (as a function of Y, Z, g_1), then the representation of g_pY in base p consists of u pairwise disjoint blocks separated by zeros, where each block contains the representation of g_1Y. As the same description applies to gZ as well we conclude that indeed when adding g_pY and g_pZ in base p there will be a carry in at least u places.

It remains to combine all the different numbers g_p and obtain the required g. For each $p \in P$, let p^{t_p} be a power of p satisfying $p^{t_p} > max\{g_pY, g_pZ\}$. Note that if $g \equiv g_p \ (mod\ p^{t_p})$ then the right part of the representation of gY in base p is identical to the representation of g_pY in base p, and the same holds for gZ. By the Chinese Remainder Theorem there is an integer g satisfying $g \equiv g_p \ (mod\ p^{t_p})$ for all $p \in P$. It follows that p^u divides $\binom{gX}{gY}$ for all $p \in P$, completing the proof. $\square$

Having finished all necessary preparations, we are now ready to complete the proof of Theorem 3.1.

Proof of Theorem 3.1. Let $H = H(m,b)$ with $\frac{m-2b}{m} = \eta$ and adjacency matrix $A(H)$. By the above discussion this is a $d = \binom{m}{b}$ regular graph on 2^m vertices and the smallest eigenvalue of $A(H)$ is equal to $\lambda = \frac{m-2b}{m}\binom{m}{b} = \eta d$. Choose an appropriate m, such that $\frac{1-2a-\eta}{2a}\binom{m}{b}$ is an even, nonnegative integer. This is always possible, by Lemma 3.4, since a and η are rational numbers and $a \le (1-\eta)/2$. Pick H_1 to be any d_1 regular graph on $n_1 = \frac{1-2a-\eta}{2a}dd_1 + 1$ vertices such that if $\mu_1 \le \mu_2 \le \ldots \le \mu_{n_1-1} \le \mu_{n_1} = d_1$ are all eigenvalues of $A(H_1)$ and $\mu = max\{|\mu_1|, |\mu_{n_1-1}|\}$ then $\mu \le \frac{2a-1}{2a}d_1$. There are several known constructions of such expander graphs. In particular a random d_1-regular graph on n_1 vertices with high probability satisfies that $\mu = O(\sqrt{d_1})$ (see, e.g., [Fri91] and [FKS89]). By taking d_1 and n_1 sufficiently large we obtain a graph with the desired properties. Denote by I the identity matrix of order 2^m and by K_{n_1} a complete graph on n_1 vertices. It is easy to see that the eigenvalues of the adjacency matrix $A(K_{n_1})$ are equal to n_1-1 and -1 (the latter with multiplicity n_1-1) and the corresponding eigenvectors are the all one vector $\mathbf{1}_{n_1}$ and any maximal set of independent vectors in R^{n_1} whose sums of coordinates equal to zero. This implies that the adjacency matrices of H_1 and K_{n_1} have a common basis of eigenvectors.

Finally let G be a graph on $n = 2^m n_1$ vertices with the following adjacency matrix

$$A(G) = A(H) \otimes A(H_1) + I \otimes A(K_{n_1}).$$

In other words, G is obtained by replacing each vertex of H by a copy of K_{n_1}, and replacing each edge of H by (the double cover of) a copy of H_1. By the definition, $A(G)$ is a symmetric matrix with all entries equal either 1 or 0 and for every row of $A(G)$ the sum of its entries is equal to $d' = dd_1 + (n_1-1)$. Therefore it is an adjacency matrix of a d'-regular graph. The matrices $A(H)$ and I and also the matrices $A(H_1)$ and K_{n_1} have a common basis of eigenvectors. Thus by Lemma 3.3 we obtain that the same is true for $A(G)$, $A(H) \otimes A(H_1)$ and $I \otimes A(K_{n_1})$. Next we need to compute the smallest eigenvalue of $A(G)$. By Lemma 3.3 it is easy to see that the only two possibility for its value are $\lambda d_1 + (n_1-1)$ or $-\mu d - 1$. Since

$-1 < \eta < 0$ and $d = \binom{m}{b}$ is large enough, an easy computation shows that

$$\lambda d_1 + (n_1 - 1) \;=\; \eta d d_1 + \frac{1-2a-\eta}{2a} d d_1 \;=\; \frac{(1-2a)(1-\eta)}{2a} d d_1 \;<\; \frac{(1-2a)}{2a} d d_1 - 1 \;<\; -\mu d - 1\ ,$$

where in the penultimate inequality we used the fact that $\frac{1-2a}{2a}\eta d d_1 > 1$ for all sufficiently large d_1, and in the last inequality we used that $\mu \le \frac{2a-1}{2a} d_1$. Therefore we conclude that the smallest eigenvalue of $A(G)$ is $\lambda d_1 + (n_1 - 1)$. Furthermore, by Lemma 3.3 and the properties of $H(m,b)$ this eigenvalue has multiplicity m and its corresponding eigenvectors are equal to $z_1 = y_1 \otimes \mathbf{1}_{n_1}, \ldots, z_m = y_m \otimes \mathbf{1}_{n_1}$.

Clearly $z_i = (z_{i1}, \ldots, z_{in})$ is a vector with ± 1 coordinates. Thus the coordinates of z_i correspond to a cut in G of size equal to

$$\sum_{k<j} a_{kj}(G)\frac{1 - z_{ik} z_{ij}}{2} \;=\; \frac{1}{2}|E(G)| - \frac{1}{4} z_i^t A(G) z_i \;=\; \frac{1}{4} d' n - \frac{1}{4}(\lambda d_1 + (n_1 - 1))\|z_i\|^2$$

$$= \frac{1}{4}(d d_1 + \frac{1-2a-\eta}{2a} d d_1) n - \frac{1}{4}(\eta d d_1 + \frac{1-2a-\eta}{2a} d d_1) n \;=\; \frac{1}{2}\frac{1-\eta}{2} d d_1 n \;=\; a\frac{1}{2} d' n \;=\; a|E(G)|\ ,$$

where here we used the fact that $d' = d d_1 + n_1 - 1 = \frac{1-\eta}{2a} d d_1$. Thus the size of a maximum cut in G is equal to the optimal value of the semidefinite program (see Lemma 3.2). On the other hand let B be the $2^m \times m$ matrix whose rows are the vectors w_i and thus its columns are equal to $\frac{1}{\sqrt{m}} y_i$. Denote by $u_1, \ldots u_n$ the rows of the matrix $B \otimes \mathbf{1}_{n_1}$. By definition, $\|u_i\|^2 = 1$ and the columns of this matrix are the eigenvectors $\frac{1}{\sqrt{m}} y_i \otimes \mathbf{1}_{n_1} = \frac{1}{\sqrt{m}} z_i$ of $A(G)$ corresponding to its smallest eigenvalue $\lambda d_1 + (n_1 - 1)$. Therefore by Lemma 3.2 it follows that

$$\max_{\|v_l\|^2=1} \sum_{i<j} a_{ij}(G)\frac{1 - v_i^t v_j}{2} \;=\; \frac{1}{2}|E(G)| - \frac{1}{4}(\lambda d_1 + (n_1 - 1)) n \;=\; a|E(G)| \;=\; \sum_{i<j} a_{ij}\frac{1 - u_i^t u_j}{2}\ .$$

To finish the proof of the theorem note that by definition each u_i is equal to one of the vectors w_k. In addition if i and j are adjacent vertices in G then u_i, u_j are either equal to the same vector w_k, and then $u_i^t u_j = 1$, or are equal to w_k, w_l which correspond to adjacent vertices in H, and in that case $u_i^t u_j = \eta$. $\square$

4 Approximation algorithms for the MAX NAE-{3}-SAT problem

An instance of the MAX 2-XOR (or MAX NAE-{2}-SAT) problem in the variables $x_1, x_2, \ldots, x_n$ is composed of a (weighted) collection of pairs of the form $\langle z_1, z_2 \rangle$, where the z_i's are literals. A clause $\langle z_1, z_2 \rangle$ is satisfied by a 0-1 assignment to the variables $x_1, x_2, \ldots, x_n$ if and only if $z_1 \ne z_2$ under this assignment. It is easy to see that instances of MAX CUT are just instances of MAX 2-XOR with no negations. The approximation algorithms of [GW95] and [Zwi99] are, in fact, approximation algorithms for MAX 2-XOR, not just for MAX CUT. The performance guarantees obtained by these algorithms on MAX 2-XOR instances are the same as those obtained on MAX CUT instances.

An instance of MAX NAE-{3}-SAT is easily converted into an instance of MAX 2-XOR. Simply replace each triplet $\langle z_1, z_2, z_3 \rangle$ by the three pairs $\langle z_1, z_2 \rangle$, $\langle z_1, z_3 \rangle$ and $\langle z_2, z_3 \rangle$, giving each one of them a weight of 1/2. It is easy to check that the total weight of the triplets/pairs satisfied by this transformation is unchanged. Thus, the algorithm of [GW95] is also an approximation algorithm for MAX NAE-{3}-SAT with a performance ratio of at least $\alpha \simeq 0.87856$.

A better performance guarantee for the MAX NAE-{3}-SAT problem can be obtained as follows. It is convenient to adopt the notation $x_{n+i} = \bar{x}_i$, for $1 \leq i \leq n$. If we let $w_{ijk} \geq 0$ be the weight attached to the triplet $\langle x_i, x_j, x_k \rangle$ in a MAX NAE-{3}-SAT instance, then we can write the following semidefinite programming relaxation corresponding to the instance:

$$\begin{array}{ll} \max & \sum_{i<j<k} w_{ijk} \frac{3 - v_i^t v_j - v_i^t v_k - v_j^t v_k}{4} \\ \text{s.t.} & v_i^t v_i = 1 \ , \ v_i^t v_{n+i} = -1 \ , \text{ for } 1 \leq i \leq n \ , \\ & v_i^t v_j + v_i^t v_k + v_j^t v_k \geq -1 \ , \text{ for } 1 \leq i,j,k \leq 2n \ . \end{array}$$

If we round an optimal solution $v_1, v_2, \ldots, v_n$ of the above relaxation using a random hyperplane, then we still get a performance guarantee of only $\alpha \simeq 0.87856$. However, for *satisfiable* instances of MAX NAE-{3}-SAT a performance guarantee of at least $\beta_1 = \frac{3}{2\pi} \arccos(-\frac{1}{3}) \simeq 0.91226$ is obtained (see [AKMR96], [Zwi98]). The performance ratio obtained is no better than β_1 if there exist unit vectors $v_1, v_2, \ldots, v_n$ such that $v_i^t v_j = v_i^t v_k = v_j^t v_k = -\frac{1}{3}$, for every triplet $\langle x_i, x_j, x_k \rangle$ of the instance with nonzero w_{ijk}. (Due to the constraint $v_i^t v_j + v_i^t v_k + v_j^t v_k \geq -1$, such a collection of vectors is automatically an optimal solution of the relaxation.) We show in the next sections that such solutions do exist.

Zwick [Zwi99] obtains a performance guarantee of at least $\beta \simeq 0.908718$ for general, not necessarily satisfiable, instances of the problem by constructing a sequence of vectors $w_1, w_2, \ldots, w_n$ such that $w_i^t w_j = c(v_i^t v_j)$, for every $i \neq j$, where $c \simeq 0.9789916$, and then rounding $w_1, w_2, \ldots, w_n$, and not $v_1, v_2, \ldots, v_n$, using a random hyperplane. More specifically, the constants $\beta \simeq 0.908718$, $c \simeq 0.9789916$ and $\eta \simeq -0.74335866$ are the solutions of the three equations

$$c^2(\eta^2 + \frac{4}{\beta^2 \pi^2}) = 1 \ , \quad \frac{2\arccos(c\eta) + \arccos(c)}{\pi} = \beta(1-\eta) \ , \quad \frac{3}{2\pi} \arccos(-\frac{c}{3}) = \beta \ .$$

It is further shown in [Zwi99] that the performance ratio achieved by this algorithm on a given instance is no better than β if the optimal value of the relaxation is equal to the optimal value of the instance (no integrality gap), *and* if for every triplet $\langle x_i, x_j, x_k \rangle$ that appears in the instance either $v_i^t v_j = v_i^t v_k = v_j^t v_k = -\frac{1}{3}$ or two of the inner products $v_i^t v_j$, $v_i^t v_k$ and $v_j^t v_k$ are η and the third is 1. Furthermore, a fraction of about $r \simeq 0.278797$ of the triplets should be of the second type, otherwise an improved ratio may be obtained by varying c. (We omit the exact equation defining r.) We are again able to show that such instances do exist and hence the analysis is again tight.

5 Worst case instances for the MAX NAE-{3}-SAT algorithms

Let $\mathcal{H} = (V, E)$ be a 3-uniform hypergraph (every $e \in E$ satisfies $e \subseteq V$ and $|e| = 3$). A cut of $\mathcal{H}$ is again a partition of V into two sets S and $V - S$. A cut $(S, V - S)$ is said to cut a hyper-edge $e \in E$ if and only if $0 < |e \cap S| < 3$. A maximum cut of $\mathcal{H}$ is a cut that cuts the largest number of edges. It is easy to see that the problem of finding a maximum cut of $\mathcal{H}$ corresponds to a MAX NAE-{3}-SAT instance with no negations. We show that the analyses of the MAX NAE-{3}-SAT algorithms described in the previous section are tight even on such instances.

A hypergraph $\mathcal{H} = (V, E)$ has a cut of size $|E|$ if and only if it is 2-colorable. The MAX NAE-{3}-SAT instance corresponding to it is then satisfiable. The following theorem shows that the bound of $\beta_1 = \frac{3}{2\pi} \arccos(-\frac{1}{3}) \simeq 0.91226$ on the performance ratio achieved by the MAX NAE-{3}-SAT approximation algorithm described in the previous section on satisfiable instances is tight.

Theorem 5.1 *For infinitely many values of n, there exists a 2-colorable 3-uniform hypergraph $\mathcal{H} = (V, E)$, such that $V = \{1, 2, \dots, n\}$, and a sequence $w_1, w_2, \dots, w_n$ of unit vectors such that $w_i^t w_j = w_i^t w_k = w_j^t w_k = -\frac{1}{3}$ for every $\{i, j, k\} \in E$.*

Proof. It is easy to construct such an example for $n = 4$. Simply let E be composed of all subsets of $V = \{1, 2, 3, 4\}$ of size 3. It is easy to check that $S = \{1, 2\}$ is a cut that cuts all the edges. Let w_1, w_2, w_3 and w_4 be four unit vectors such that $w_i^t w_j = -\frac{1}{3}$, for every $1 \le i < j \le 4$. This can be done, for example, by taking $w_1 = \frac{1}{\sqrt{3}}(1, 1, 1)^t$, $w_2 = \frac{1}{\sqrt{3}}(1, -1, -1)^t$, $w_3 = \frac{1}{\sqrt{3}}(-1, 1, -1)^t$ and $w_4 = \frac{1}{\sqrt{3}}(-1, -1, 1)^t$.

This example is a special case of the following more general construction which supplies an infinite family of satisfiable instances of MAX NAE-{3}-SAT problem for which the analysis from [Zwi98] is tight. Let $H = H(m, b)$ with $b = 2m/3$. The vertex set $\{v_1, v_2, \dots\}$ of H consists of all ± 1 vectors of length m and two vectors are adjacent if the number of coordinates in which they differ is equal to $2m/3$. Let $w_i = \frac{1}{\sqrt{m}} v_i$ for all i; thus $\|w_i\|^2 = 1$ and $w_i^t w_j = \frac{m-2b}{m} = -1/3$ for any pair of adjacent vertices in H. Let $\mathcal{H}$ be the 3-uniform hypergraph whose edges are triples of the vertices in H that form a cycle of length 3. By definition, it is easy to see that three vectors which form a triangle in the graph H cannot have the same first coordinate. By partitioning vertices into two parts according to their first coordinate, we obtain therefore a 2-coloring of $\mathcal{H}$, as required. $\square$

We next show that the analysis of the performance of the MAX NAE-{3}-SAT algorithm of Zwick [Zwi99] on general, not necessarily satisfiable, instances in also tight. By the discussion in the previous section, this follows from the following theorem. Given a 3-uniform hypergraph $\mathcal{H}$, we let $OPT(\mathcal{H})$ be the size of the maximum cut of $\mathcal{H}$, and we let $z^*(\mathcal{H})$ be the optimal value of the semidefinite programming relaxation of the corresponding MAX NAE-{3}-SAT instance.

Theorem 5.2 *Let $-1 < \eta < -1/2$ be a rational number, and let $0 < r < 1$ and $\epsilon > 0$. Then, for infinitely many values of $|U|$ there exists a 3-uniform hypergraph $\mathcal{H} = (U, E)$, where $U = \{u_1, u_2, \dots\}$ and $E = E_1 \cup E_2$, and unit vectors $w_{u_1}, w_{u_2}, \dots$ such that for every $\{u_i, u_j, u_k\} \in E_1$, we have $w_{u_i}^t w_{u_j} = w_{u_i}^t w_{u_k} = w_{u_j}^t w_{u_k} = -\frac{1}{3}$, for every $\{u_i, u_j, u_k\} \in E_2$ exactly two of the inner products $w_{u_i}^t w_{u_j}$, $w_{u_i}^t w_{u_k}$, $w_{u_j}^t w_{u_k}$ are η and the third is 1, and such that*

$$OPT(\mathcal{H}) \;=\; z^*(\mathcal{H}) \;=\; \sum_{\{i,j,k\} \in E} \frac{3 - w_{u_i}^t w_{u_j} - w_{u_i}^t w_{u_k} - w_{u_j}^t w_{u_k}}{4}\,.$$

In addition $|E_2|$ is bounded by $r|E| \le |E_2| \le (r + \epsilon)|E|$.

Proof. The hypergraph $\mathcal{H} = (U, E)$ that we construct is the union of two hypergraphs $\mathcal{H}_1 = (U, E_1)$ and $\mathcal{H}_2 = (U, E_2)$, that is $E = E_1 \cup E_2$. We begin with the description of $\mathcal{H}_1$.

Let m and n be (large) integers (their values are specified at the end of the proof). Let H_1 be the graph $H_1 = H(m, b)$ with $b = 2m/3$ (we assume that m is divisible by 3) and let I be an identity matrix of order n. Consider a graph G_1 with adjacency matrix $A(G_1) = A(H_1) \otimes I$. Clearly G_1 is just a disjoint union of n copies of H_1. The vertex set $U = \{u_1, u_2, \dots\}$ of this graph consists of all pairs $\{(v, t) | v \in \{-1, 1\}^m, 1 \le t \le n\}$, and two vertices (v, t) and (v', t') are adjacent iff $t = t'$ and v and v' differ in exactly $2m/3$ coordinates. Let $w_u = \frac{1}{\sqrt{m}} v$ for all $u = (v, t)$; thus $\|w_u\|^2 = 1$ and $w_{u_i}^t w_{u_j} = \frac{m-2b}{m} = -1/3$ for any pair of adjacent vertices in G_1. Let $\mathcal{H}_1$ be a 3-uniform hypergraph, whose edges are the triples of the vertices in G_1 which form a cycle of length 3. It is easy to see that the number of edges in $\mathcal{H}_1$ is equal to $\frac{1}{6} n 2^m \binom{m}{2m/3} \binom{2m/3}{m/3}$. Let A be a subset of U containing all vertices (v, t) with first coordinate of v_i equal to one and let $B = U - A$. It follows easily from the definition that the three vertices of a 3-cycle in G_1 cannot all have the same first coordinate. Thus any 3-cycle in G_1 will intersect both A and B. Therefore we obtain a cut in the hypergraph $\mathcal{H}_1$ whose size is equal to the total number of edges of $\mathcal{H}_1$. Finally, since $w_{u_i}^t w_{u_j} = w_{u_i}^t w_{u_k} = w_{u_j}^t w_{u_k} = -1/3$ for any edge in $\mathcal{H}_1$ and the value of the semidefinite relaxation $z^*(\mathcal{H}_1)$ is always bounded by $|E(\mathcal{H}_1)|$ we conclude that

$$\sum_{\{u_i,u_j,u_k\}\in E(\mathcal{H}_1)} \frac{3 - w_{u_i}^t w_{u_j} - w_{u_i}^t w_{u_k} - w_{u_j}^t w_{u_k}}{4} = |E(\mathcal{H}_1)| = OPT(\mathcal{H}_1) = z^*(\mathcal{H}_1) = \frac{1}{6} n 2^m \binom{m}{2m/3}\binom{2m/3}{m/3}.$$

The construction of $\mathcal{H}_2$ is more involved than that of $\mathcal{H}_1$. We start by constructing an auxiliary multigraph G_2. Let H_2 be the graph $H_2 = H(m, b)$ with $b = \frac{1-\eta}{2}m$ (η is given at the statement of the theorem and can be made arbitrarily close to $-0.74335866...$). Let K_n be a complete graph on n vertices. The graph H_2 is d-regular with $d = \binom{m}{(1-\eta)m/2}$ and by the discussion in Section 3, the smallest eigenvalue of its adjacency matrix $A(H_2)$ is $\lambda = \frac{m-2b}{m}d = \eta d$. Let G_2 be a multigraph with adjacency matrix equal to $A(G_2) = (A(H_2) + dI/2) \otimes A(K_n)$, where I is an identity matrix of order 2^m. The vertex set of G_2 is again $U = \{(v,t) | v \in \{-1,1\}^m, 1 \le t \le n\}$ and two vertices (v,t) and (v',t') are connected by a unique edge if $t \neq t'$ and v and v' differ in exactly $\frac{1-\eta}{2}m$ coordinates or they are connected by $d/2$ parallel edges if $v = v'$ and $t \neq t'$. By definition G_2 is a $(3d(n-1)/2)$-regular multigraph and by Lemma 3.3 its smallest eigenvalue is equal to $(\lambda + d/2)(n-1) = (\eta + 1/2)d(n-1) < -3d/2 < 0$, where here we used the fact that n is sufficiently large. Let $w_u = \frac{1}{\sqrt{m}}v$ for all vertices (v,t) be as before; thus $\|w_u\|^2 = 1$ and $w_{u_i}^t w_{u_j} = \frac{m-2b}{m} = \eta$ or 1 for any pair of adjacent vertices of G_2. In addition $w_{u_i}^t w_{u_j} = 1$ if and only if $u_i = (v,t)$ and $u_j = (v,s)$ with $t \neq s$. Let $B = (b_{ij})$ be the $2^m n \times m$ matrix whose rows are equal to the vectors $w_u, u \in U$. Note that the elements in B are $\pm\frac{1}{\sqrt{m}}$. As in the proof of Theorem 3.1, we can see that the columns of B are eigenvectors of $A(G_2)$ that correspond to the smallest eigenvalue of $A(G_2)$. Let $OPT(G_2)$ be the size of the MAX CUT in G_2 and let $z^*(G_2)$ be the value of the semidefinite programming relaxation. Then by Lemma 3.2 we obtain that

$$OPT(G_2) = z^*(G_2) = \sum_{i<j} a_{ij}(G_2)\frac{1 - w_{u_i}^t w_{u_j}}{2} = \frac{1}{2}|E(G_2)| - \frac{1}{4}(\eta + 1/2)d(n-1)|V(G_2)| .$$

The coordinates of the first column of B produce the cut (A, B) (same as for $\mathcal{H}_1$) and its size is equal to $OPT(G_2)$, since the first column of B is an eigenvector of the smallest eigenvalue of $A(G_2)$.

Now we are ready to construct $\mathcal{H}_2$. Let $\mathcal{H}_2$ be the 3-uniform hypergraph on the vertex set $U = V(G_2)$, whose edges are the following triples of the vertices of G_2; $\{u_i, u_j, u_k\}$ belongs to $E(\mathcal{H}_2)$ if and only if $u_i = (v,t)$, $u_j = (v,t')$ and $u_k = (v'',t'')$ such that $t \neq t' \neq t''$ and v and v'' differ in exactly $(1-\eta)m/2$ coordinates. Note that by definition, the number of edges in $\mathcal{H}_2$ is equal to $\frac{1}{2}n(n-1)(n-2)2^m\binom{m}{(1-\eta)m/2}$ and they form cycles of length 3 in G_2. In addition every edge of G_2 connecting $u_i = (v,t)$ and $u_k = (v'',t'')$ (as above) is contained in exactly $2(n-2)$ edges of $\mathcal{H}_2$ and every pair of vertices $u_i = (v,t)$ and $u_j = (v,t')$ (as above) is contained in exactly $d(n-2)$ edges of $\mathcal{H}_2$. Since in the multigraph G_2 between the vertices $u_i = (v,t)$ and $u_j = (v,t')$ we have $d/2$ parallel edges, we can distribute them equally between all 3-cycles which correspond to the edges of $\mathcal{H}_2$, containing this pair of vertices. By doing this we obtain that every edge in the multigraph G_2 is contained in exactly $2(n-2)$ edges of $\mathcal{H}_2$. In this case the size of the MAX CUT in $\mathcal{H}_2$ is closely related to the size of the MAX CUT of G_2. Note that for any partition of the vertices $(X, U - X)$, the number of edges of $\mathcal{H}_2$ which crosses this cut is exactly $n-2$ times the number of edges of G_2 with the same property. Indeed, any edge from G_2 which connects X with $U - X$ is contained in $2(n-2)$ triples from $\mathcal{H}_2$. All of them also cross this cut, but every such triple we counted twice, since it contributes two edges of G_2 to the cut. Therefore we can conclude that the value of MAX CUT of $\mathcal{H}_2$ is equal to $OPT(\mathcal{H}_2) = (n-2)OPT(G_2)$ and this value is obtained on the cut (A, B), the same one as for the graph G_2. This, together with the above discussion implies that

$$\begin{aligned} OPT(\mathcal{H}_2) \;&\le\; z^*(\mathcal{H}_2) \;\le\; \max \textstyle\sum_{\{u_i,u_j,u_k\}\in E(\mathcal{H}_2)} \frac{3-y_{u_i}^t y_{u_j} - y_{u_i}^t y_{u_k} - y_{u_j}^t y_{u_k}}{4} \\ &= \tfrac{1}{2}\max \textstyle\sum_{\{u_i,u_j,u_k\}\in E(\mathcal{H}_2)} \left[\frac{1-y_{u_i}^t y_{u_j}}{2} + \frac{1-y_{u_i}^t y_{u_k}}{2} + \frac{1-y_{u_j}^t y_{u_k}}{2}\right] \;\le\; (n-2)\cdot \max \textstyle\sum_{\{u_i,u_j\}\in E(G_2)} \frac{1-y_{u_i}^t y_{u_j}}{2} \\ &= (n-2)\textstyle\sum_{i<j} a_{ij}(G_2)\frac{1-w_{u_i}^t w_{u_j}}{2} = (n-2)z^*(G_2) \;=\; (n-2)OPT(G_2) \;=\; OPT(\mathcal{H}_2)\;. \end{aligned}$$

Thus,

$$OPT(\mathcal{H}_2) \;=\; z^*(\mathcal{H}_2) \;=\; \sum_{\{u_i,u_j,u_k\}\in E(\mathcal{H}_2)} \frac{3 - w_{u_i}^t w_{u_j} - w_{u_i}^t w_{u_k} - w_{u_j}^t w_{u_k}}{4}\;.$$

Also, we know that for every edge $\{u_i, u_j, u_k\}$ of $\mathcal{H}_2$, two of the inner products $w_{u_i}^t w_{u_j}$, $w_{u_i}^t w_{u_k}$ and $w_{u_j}^t w_{u_k}$ are η and the third is 1.

Finally let $\mathcal{H}$ be the 3-uniform hypergraph with the same vertex set U and with edge set $E(\mathcal{H}_1)\cup E(\mathcal{H}_2)$. We clearly have $OPT(\mathcal{H}) \le z^*(\mathcal{H}) \le z^*(\mathcal{H}_1)+z^*(\mathcal{H}_2) = OPT(\mathcal{H}_1)+OPT(\mathcal{H}_2)$. On the other hand (A, B) is a MAX CUT of *both* $\mathcal{H}_1$ and $\mathcal{H}_2$, thus it is also a cut of $\mathcal{H}$ of size $OPT(\mathcal{H}_1) + OPT(\mathcal{H}_2)$. As the same vectors w_{u_i} were used for the two hypergraphs, we get that

$$\sum_{\{u_i,u_j,u_k\}\in E(\mathcal{H})} \frac{3 - w_{u_i}^t w_{u_j} - w_{u_i}^t w_{u_k} - w_{u_j}^t w_{u_k}}{4} \;=\; OPT(\mathcal{H}_1) + OPT(\mathcal{H}_2) \;=\; OPT(\mathcal{H}) \;=\; z^*(\mathcal{H})\;.$$

In addition, for every edge $\{u_i, u_j, u_k\} \in E(\mathcal{H})$, either two of the inner products $w_{u_i}^t w_{u_j}$, $w_{u_i}^t w_{u_k}$ and $w_{u_j}^t w_{u_k}$ are η and the third is 1 (if $\{u_i, u_j, u_k\} \in E(\mathcal{H}_2)$), or $w_{u_i}^t w_{u_j} = w_{u_i}^t w_{u_k} = w_{u_j}^t w_{u_k} = -\frac{1}{3}$ (if $\{u_i, u_j, u_k\} \in E(\mathcal{H}_1)$). Finally, recall that $|E(\mathcal{H}_1)| = \frac{1}{6}n2^m\binom{m}{2m/3}\binom{2m/3}{m/3}$ and that $|E(\mathcal{H}_2)| = \frac{1}{2}n(n-1)(n-2)2^m\binom{m}{(1-\eta)m/2}$ where m and n are (large) parameters that we are free to choose. By choosing appropriate values of m and n, and using the fact that

$$\binom{m}{2m/3}\binom{2m/3}{m/3} > 2^m > \binom{m}{(1-\eta)m/2}$$

for all sufficiently large m, it follows that we can control the proportion of the edges of the second type, and make it arbitrarily close to r, as required. $\square$

6 Concluding remarks

We have shown that lower bounds on the performance guarantees of semidefinite programming based approximation algorithms for the MAX CUT, MAX 2-XOR and MAX NAE-{3}-SAT problems obtained using local ratio arguments are indeed tight.

Furthermore, our constructions show that the analyses of these algorithms are tight even if arbitrary collections of valid constraints are added to the semidefinite programming relaxations of these problems. Let $a_{ij}, 1 \le i < j \le n$, and b be real numbers. A constraint

$$\sum_{i<j} a_{ij}(v_i^t v_j) \ge b$$

is said to be *valid* if it is satisfied whenever each v_i is an integer in $\{-1, 1\}$. Feige, Goemans and Williamson (see [FG95], [GW95]) proposed adding valid constraints to the semidefinite relaxations in the hope of narrowing the gap between the optimal value of the semidefinite program and the weight of the optimal solution. As all the coordinates of the vectors $u_1, u_2, \ldots$ of Section 3 and of the vectors $w_1, w_2, \ldots$ of Section 5 are equal to $\pm 1/\sqrt{m}$, it is not difficult to see that they satisfy any valid constraint. Thus the proofs of Theorems 3.1, 5.1 and 5.2 show that the addition of any collection of valid constraints does not improve the performance ratio of the above mentioned approximation algorithms for the MAX CUT, MAX 2-XOR and MAX NAE-{3}-SAT problems.

It is shown in [KZ97] that the 7/8 lower bound on the performance ratio of the MAX 3-SAT approximation algorithm, obtained again using a local ratio argument, is also tight. Does local analysis always produce tight results? We see no reason why this should always be the case. It would be interesting to find natural approximation algorithms for interesting constraint satisfaction problems for which local analysis is not tight. It would also be interesting to know whether the local analyses of the approximation algorithms of Feige and Goemans [FG95] (see also [Zwi00]) for the MAX 2-SAT and MAX DI-CUT problems are tight. This seems to require, however, some additional techniques.

Acknowledgment We would like to thank two anonymous referees for many helpful comments.

References

[AKMR96] N. Alon, P. Kelsen, S. Mahajan, and H. Ramesh. Approximate hypergraph coloring. *Nordic Journal of Computing*, 3:425–439, 1996.

[AS00] N. Alon and B. Sudakov. Bipartite subgraphs and the smallest eigenvalue. *Combinatorics, Probability & Computing*, 9:1–12, 2000.

[CHLL97] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes.* North-Holland, 1997.

[FG95] U. Feige and M.X. Goemans. Approximating the value of two prover proof systems, with applications to MAX-2SAT and MAX-DICUT. In *Proceedings of the 3nd Israel Symposium on Theory and Computing Systems, Tel Aviv, Israel*, pages 182–189, 1995.

[FKS89] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue in random regular graphs. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, Seattle, Washington*, pages 587–598, 1989.

[Fri91] J. Friedman. On the second eigenvalue and random walks in random d-regular graphs. *Combinatorica*, 11:331–362, 1991.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, Texas*, pages 1–10, 1997. Full version available as E-CCC Report number TR97-037.

[Kar99] H. Karloff. How good is the Goemans-Williamson MAX CUT algorithm? *SIAM Journal of Computing* 29 (1999), 336–350.

[KZ97] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, Florida*, pages 406–415, 1997.

[Lov93] L. Lovász. *Combinatorial Problems and Exercises.* North-Holland, 1993.

[TSSW96] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming (extended abstract). In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, Burlington, Vermont*, pages 617–626, 1996.

[Zwi98] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 201–210, 1998.

[Zwi99] U. Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems. In *Proceedings of the 31th Annual ACM Symposium on Theory of Computing, Atlanta, Georgia*, pages 679–687, 1999.

[Zwi00] U. Zwick. Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans. Submitted for publication, 2000.