How Good is the Goemans-Williamson MAX CUT Algorithm?

Howard Karloff *

Abstract

The celebrated semidefinite programming algorithm for MAX CUT introduced by Goemans and Williamson was known to have a performance ratio of at least $\alpha = \frac{2}{\pi}\min_{0<\theta\leq\pi}\frac{\theta}{1-\cos\theta}$ $(0.87856 < \alpha < 0.87857)$; the exact performance ratio was unknown. We prove that the performance ratio of their algorithm is exactly α. Furthermore, we show that it is impossible to add valid linear constraints to improve the performance ratio.

1 Introduction

The *performance ratio* of a randomized algorithm for a maximization problem is defined as follows. Let $W(I)$ be the (random) weight of the feasible solution it produces on instance I and let $OPT(I)$ be the weight of the optimal solution for instance I. The performance ratio is then

$$\inf \frac{E[W(I)]}{OPT(I)},$$

where the infimum is over instances with $OPT(I)$ positive.

In 1994, M. X. Goemans and D. P. Williamson published a new approximation algorithm for MAX CUT, the NP-Complete problem of finding a maximum cut in a nonnegatively weighted graph [GW]. (A summary of the algorithm is given in section 1.1. Readers unfamiliar with [GW] should read Section 1.1 before continuing.) Much of the mathematical programming community turned its attention to this new algorithm, not only because it improved the performance ratio for MAX CUT from 0.5 to at least 0.87856, but also because their paper was the first use of semidefinite programming in the area of approximation algorithms. Naturally researchers asked how far semidefinite programming could go.

* College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280. Research supported in part by NSF grant CCR-9319106. E-mail: howard@cc.gatech.edu.

STOC'96, Philadelphia PA,USA

In this paper, we analyze MAX CUT and the semidefinite programming algorithm proposed by Goemans and Williamson, hereafter called **Algorithm GW**. Goemans and Williamson proved a lower bound of $\alpha = \frac{2}{\pi}\min_{0<\theta\leq\pi}\frac{\theta}{1-\cos\theta}$ $(0.87856 < \alpha < 0.87857)$ on its performance ratio, a huge improvement over the 0.5 known before. But exactly how good is Algorithm GW? At no point did they prove an upper bound on the performance ratio of Algorithm GW. Conceivably the performance ratio of Algorithm GW substantially exceeded α. We will prove that its performance ratio is exactly α.

In addition, we study the effect of the addition of linear constraints to the semidefinite program of [GW]. For example, Feige and Goemans [FG] and Goemans and Williamson [GW] proposed adding to the semidefinite program a family of $4\binom{n}{3}$ linear constraints, in the hope of narrowing the gap between the optimal value of the semidefinite program and the weight of a max cut. These constraints, given in Section 2 and called the **FGW Constraints**, are valid for all cuts (that is, when the vector variables are actually the *integers* ± 1). It is known [BM] that with the FGW constraints, the optimal value of the semidefinite program is exactly the size of a maximum cut, if the input graph does not contain K_5 as a minor, in particular, if it is planar.

We will prove that the addition of any family of constraints, which, like the FGW constraints, are linear in the dot products and valid for all cuts, cannot improve the performance ratio; it will be exactly α, even with those constraints.

In addition to the performance ratio, another interesting quantity to study is the "integrality ratio," the minimum, over all weighted graphs, of the ratio between the weight of a maximum cut and the optimal value of the semidefinite relaxation for that graph. Unlike the performance ratio, which is affected both by the relaxation and the rounding procedure, the integrality ratio depends, of course, only on the relaxation. Goemans and Williamson did study the integrality ratio for their semidefinite relaxation, noting that it is particularly bad for the cycle C_5. We caution the reader that the results in the present paper say nothing about the integrality ratio. In fact, for the graphs we will construct, the weight of a max cut exactly equals the optimal value of the semidefinite program.

1.1 A Summary of Algorithm GW

Here we describe the algorithm of Goemans and Williamson. For more details, see [GW].

The Goemans and Williamson algorithm applies to MAX CUT, the problem of partitioning the vertices of a weighted graph $G = (V, E)$, $V = \{1, 2, ..., n\}$, into two sets, so as to maximize the total weight of the edges across the cut. Let us use $w_{ij} \geq 0$ to denote the nonnegative weight of the edge between vertices i and j, with $w_{ij} = 0$ if $\{i, j\} \notin E$.

This problem can be rephrased as follows. Assign an integer $x_i \in \{-1, +1\}$ to each vertex i, -1 representing the left side of the cut and $+1$ the right side, so as to maximize the sum, over all $i < j$, of $w_{ij}(1 - x_i x_j)/2$. The reason this works is that $(1 - x_i x_j)/2$ is 1 if i and j are on opposite sides of the cut and 0 if they're on the same side. Since MAX CUT is NP-Complete, clearly one cannot solve this problem exactly in polynomial time (unless P=NP). However, one can relax this program to a polynomial-time solvable one, as follows. Let v_i be an n-dimensional vector whose first component is x_i and whose others are 0. Clearly $||v_i|| = 1$ ($||v||$ being the Euclidean norm of v) and the objective function

$$\sum_{i<j} w_{ij} \frac{1 - x_i x_j}{2}$$

can be rewritten as

$$\sum_{i<j} w_{ij} \frac{1 - v_i \cdot v_j}{2}$$

for these vectors, where the usual multiplication of integers x_i and x_j has been replaced by the dot product of vectors v_i and $v_j \in \mathbf{R}^n$. Now consider this problem (SD):

Find vectors $v_1, v_2, ..., v_n \in \mathbf{R}^n$ to

$$\max \sum_{i<j} w_{ij} \frac{1 - v_i \cdot v_j}{2}$$

and such that $||v_i|| = 1$ for all i.

Since the vectors v_i given above satisfy $||v_i|| = 1$, they are feasible for this program. This makes (SD) a relaxation of MAX CUT and thus the optimal value of this program is at least as large as the weight of a maximum cut in G.

Interestingly, (SD) can be (almost exactly) solved in polynomial time. Before we sketch how, let us describe what we can do with the solution $v_1, v_2, ..., v_n$, which we assume, for this summary, is the sequence of exactly optimal vectors. Somehow, we want to convert the sequence of n vectors into a cut (or into a sequence of n -1's and $+1$'s), that is, we want to *round* the vectors to an integral solution, so that the total weight of the edges across the cut is close to the value of the objective function in (SD).

The rounding procedure proposed by Goemans and Williamson is remarkably simple: just choose a random n-dimensional hyperplane through the origin, putting on the "left" side L of the cut all the vertices on one side of the hyperplane and the remainder on the "right" side R. More formally, choose a random unit vector r (the normal to the random hyperplane) and define $L = \{i | r \cdot v_i \leq 0\}$ and $R = \{i | r \cdot v_i > 0\}$. What is the expected weight of this cut? By linearity of expectation, the expected weight is the sum, over all $i < j$, of w_{ij} times the probability that i and j are on opposite sides of the cut. It is not hard to prove that the probability that i and j are on opposite sides of the cut is proportional to the angle $\arccos(v_i \cdot v_j)$ between v_i and v_j; in fact, the probability is exactly $\arccos(v_i \cdot v_j)/\pi$. Thus the expected value $E[W]$ of the weight W of the random cut is exactly

$$\sum_{i<j} w_{ij} \frac{\arccos(v_i \cdot v_j)}{\pi}.$$

However, the optimal value z_P^* of program (SD) (it does have an optimum) is exactly

$$\sum_{i<j} w_{ij} \frac{1 - v_i \cdot v_j}{2}.$$

Thus, by a term-by-term analysis, we have

$$\begin{aligned} \frac{E[W]}{z_P^*} &= \frac{\sum_{i<j} w_{ij} \arccos(v_i \cdot v_j)/\pi}{\sum_{i<j} w_{ij}(1 - v_i \cdot v_j)/2} \\ &\geq \min_{i<j} \frac{\arccos(v_i \cdot v_j)/\pi}{(1 - v_i \cdot v_j)/2}. \end{aligned}$$

Letting

$$\alpha = \frac{2}{\pi} \min_{0<\theta\leq\pi} \frac{\theta}{1 - \cos\theta}$$

(θ representing $\arccos(v_i \cdot v_j)$), we have

$$E[W] \geq \alpha \cdot z_P^*,$$

and, since z_P^* is at least as large as the weight OPT of a maximum cut, we have

$$E[W] \geq \alpha \cdot OPT.$$

Fortunately, $\alpha \in (0.87856, 0.87857)$. It follows that Algorithm GW is an α-approximation algorithm for MAX CUT.

Now we return to semidefinite programming. To understand semidefinite programming, one must understand what it means for an $n \times n$ symmetric matrix Y to be positive semidefinite. There are three equivalent definitions:

1. All the eigenvalues of Y are nonnegative. (Recall that the eigenvalues of any symmetric matrix are real.)

2. For any $x \in \mathbf{R}^n$, $x^T Y x \geq 0$.

3. There is an $n \times n$ matrix B such that $Y = B^T B$.

A *semidefinite program* is an optimization problem of the following sort: Find reals $y_{11}, y_{12}, ..., y_{nn}$ so as to maximize $c^T y$ (where $y = (y_{11}, y_{12}, y_{13}, ..., y_{nn})^T$) subject to a finite number of linear constraints $a_i^T y \geq b_i$, and such that the $n \times n$ matrix $Y = (y_{ij})$ is symmetric and positive semidefinite. The only differences between semidefinite programming and linear programming are the immaterial difference that the number of variables is a perfect square, and the crucial difference that the matrix defined by the variables must be symmetric and positive semidefinite.

Semidefinite programs can be (almost exactly) solved in polynomial time; for more details, see [GW]. How does this help us? By the third definition of "positive semidefinite," we see that if Y is symmetric and positive semidefinite, then there is an $n \times n$ B such that $Y = B^T B$, and if B is an $n \times n$ matrix, then $Y = B^T B$ is positive semidefinite. To solve program (SD) on variables $v_1, v_2, ..., v_n \in \mathbf{R}^n$, we instead solve this problem:

Find a symmetric positive semidefinite matrix $Y = (y_{ij})$ to

$$\max \sum_{i<j} w_{ij} \frac{1-y_{ij}}{2}$$

such that $y_{ii} = 1$ for all i.

Given Y, we can find a B such that $Y = B^T B$ in polynomial time. Letting v_i be the ith column of B, we have $v_i \cdot v_j = y_{ij}$ and $||v_i|| = 1$, exactly what we wanted. Given vectors $v_1, v_2, ..., v_n \in \mathbb{R}^n$, we can build a matrix B whose ith column is v_i, and then let $Y = B^T B$; we will have Y symmetric and positive semidefinite, $v_i \cdot v_j = y_{ij}$ and $y_{ii} = 1$ again. Thus, program (SD) can be phrased as a semidefinite program and thus (almost exactly) solved in polynomial time.

2 Lower Bounds on a Performance Ratio

The troublesome graphs for the Goemans-Williamson algorithm are the graphs $J(m,t,b)$ defined as follows. The vertex set of $J(m,t,b)$ is the set of all $\binom{m}{t}$ t-element subsets of $\{1, 2, ..., m\}$, two t-subsets S, T being adjacent if and only if $|S \cap T| = b$ and $S \neq T$. (These graphs are similar to the Kneser graphs $K(m,t,b)$ used in [KMS], which have the same vertex set but with S and T adjacent if and only if $|S \cap T| \leq b$.) It will become clear that the smallest eigenvalue of the adjacency matrix of $J(m, m/2, b)$ plays a crucial role, so we now study the eigenvalues of $J(m,t,b)$. For $s = 0, 1, 2, ..., t$, let

$$\alpha_s = \sum_r (-1)^{s-r} \binom{s}{r} \binom{t-r}{b-r} \binom{m-t-s+r}{t-b-s+r},$$

with the usual convention that $\binom{u}{v} = 0$ if $v < 0$.

Theorem 1. Let $0 \leq t \leq m/2$ and $0 \leq b \leq t$. Then for $s = 0, 1, 2, ..., t$, α_s occurs as an eigenvalue of $J(m,t,b)$ with multiplicity $\binom{m}{s} - \binom{m}{s-1}$.

See [Knuth] for a beautiful proof of this theorem, or see [Delsarte]. Notice that since $\sum_{s=0}^{t} [\binom{m}{s} - \binom{m}{s-1}] = \binom{m}{t}$, these are all the eigenvalues of $J(m,t,b)$.

In fact, we will only need $J(m, m/2, b)$ for even positive m's.

Corollary 2. If m is an even positive integer, $0 \leq b \leq m/2$, then the eigenvalues of $J(m, m/2, b)$ are $\beta_0, \beta_1, ..., \beta_{m/2}$ (each with some positive multiplicity), where

$$\beta_s = \sum_r (-1)^{s-r} \binom{s}{r} \binom{m/2-r}{b-r} \binom{m/2-s+r}{b}.$$

The next theorem gives the exact value of the smallest eigenvalue of $J(m, m/2, b)$.

Theorem 3. Let m be an even positive integer and let $0 \leq b \leq m/12$. The smallest eigenvalue of $J(m, m/2, b)$ is

$$\binom{m/2}{b}^2 \left[\frac{4b}{m} - 1\right].$$

We defer the proof of Theorem 3. For now, let us say only that a simple calculation gives $\beta_1 = \binom{m/2}{b}^2 \left[\frac{4b}{m} - 1\right]$, so proving Theorem 3 is the same as proving that β_1 is the smallest of all the β's.

The Goemans-Williamson semidefinite program to approximately solve the MAX CUT instance with edge weight w_{ij} between edges i and j ($w_{ij} = w_{ji}$ for all i, j and $w_{ii} = 0$ for all i) is, as mentioned above, program (SD): Find reals y_{ij} to

$$\max z_P = \sum_{i<j} w_{ij} \frac{1-y_{ij}}{2},$$

subject to $y_{ii} = 1$ for all i and $Y = (y_{ij})$ is $n \times n$ and symmetric positive semidefinite.

In our case, we use $w_{ij} = 1$ if the ith set S is adjacent in $J(m, m/2, b)$ to the jth set T (and $i \neq j$), and 0 otherwise. Alternatively, $w_{S,T} = 1$ if $|S \cap T| = b$ (and $S \neq T$), and 0 otherwise. Thus the matrix of weights is exactly the adjacency matrix of $J(m, m/2, b)$.

Each vertex in $J(m, m/2, b)$, being a subset S of $\{1, 2, ..., m\}$ of weight $m/2$, can be represented by an m-vector of +1's and −1's, with the ith component +1 if $i \in S$ and the ith component −1 if $i \notin S$, then scaled by $1/\sqrt{m}$, so that the resulting m-vector v_S has unit length. Now, from the v_S's, build an $m \times n$ matrix B with the column indexed by S being the vector v_S; then set $Y = B^T B$. Y is symmetric positive semidefinite, is $n \times n$, and has $y_{ii} = 1$ for all i. Therefore Y is a feasible solution to (SD).

In fact, Y is an *optimal* solution to (SD), if $b \leq m/12$:

Theorem 4. If $b \leq m/12$ and m is even, then $Y = B^T B$ is an optimal solution to (SD).

Proof. To show that a feasible solution to a semidefinite program is optimal, we use the dual of the semidefinite program [GW]. Let $A = (w_{ij})$ be the matrix of weights on the edges of the n-vertex graph. Let $W_{tot} = \sum_{1 \leq i < j \leq n} w_{ij}$. The dual to the semidefinite program (SD) is the semidefinite program (D): Find reals $\gamma_1, \gamma_2, ..., \gamma_n$ to

$$\min z_D = \frac{1}{2} W_{tot} + \frac{1}{4} \sum_{i=1}^{n} \gamma_i,$$

such that

$$A + \mathrm{diag}(\gamma_1, \gamma_2, ..., \gamma_n)$$

is positive semidefinite, where $\mathrm{diag}(\gamma_1, \gamma_2, ..., \gamma_n)$ is the diagonal matrix whose ith diagonal entry is γ_i. It is known [GW] that $z_P \leq z_D$ for any pair of respectively (primal, dual) feasible solutions. In fact, (SD) and (D) attain optimal values z_P^* and z_D^*, respectively, and $z_P^* = z_D^*$.

Therefore, to prove that a particular feasible solution Y for (SD) is optimal, it suffices to exhibit a feasible solution $(\gamma_1, \gamma_2, ..., \gamma_n)$ to (D) such that $z_P = z_D$. Let γ be the negation of the smallest eigenvalue of $J(m, m/2, b)$. By Theorem 3, $\gamma = \binom{m/2}{b}^2 \left[1 - \frac{4b}{m}\right]$. A symmetric matrix being positive semidefinite if and only if its eigenvalues are nonnegative, the choice of γ guarantees that $\gamma_i = \gamma$ for all i is a feasible solution to (D).

Let z_P be the (primal) cost of the primal feasible Y given above and let z_D be the (dual) cost of dual feasible $(\gamma, \gamma, \gamma, ..., \gamma)$. We prove that $z_P = z_D$, so that both are optimal.

In this formulation,

$$\begin{aligned} z_P &= \frac{1}{4}\sum_{|S|=|T|=m/2} w_{S,T}(1 - v_S \cdot v_T) \\ &= \frac{1}{4}\sum_{|S|=|T|=m/2, |S\cap T|=b} (1 - v_S \cdot v_T). \end{aligned}$$

By symmetry, we can take $S = \{1, 2, ..., m/2\}$, and infer that

$$\begin{aligned} z_P &= \frac{1}{4}n\sum\left[1 - \frac{1}{\sqrt{m}}(\underbrace{1, 1, ..., 1}_{m/2}, \underbrace{-1, -1, ..., -1}_{m/2}) \cdot v_T\right] \\ &= \frac{1}{4}n\left[\binom{m/2}{b}^2 - \frac{1}{\sqrt{m}}\sum(\underbrace{1, ..., 1}_{m/2}, \underbrace{-1, ..., -1}_{m/2}) \cdot v_T\right] \end{aligned}$$

(where both summations are over all T such that $|T| = m/2$ and $|T \cap \{1, 2, ..., m/2\}| = b$)

$$\begin{aligned} &= \frac{1}{4}n\left[\binom{m/2}{b}^2 - \frac{1}{\sqrt{m}}\binom{m/2}{b}^2\frac{1}{\sqrt{m}}(4b - m)\right] \\ &= \frac{n}{2}\binom{m/2}{b}^2\left[1 - \frac{2b}{m}\right]. \end{aligned}$$

Also,

$$\begin{aligned} z_D &= \frac{1}{2}W_{tot} + \frac{1}{4}\sum_{i=1}^{n}\gamma_i \\ &= \frac{1}{4}n\binom{m/2}{b}^2 + \frac{1}{4}n\gamma \\ &= \frac{n}{4}\binom{m/2}{b}^2 + \frac{n}{4}\binom{m/2}{b}^2\left[1 - \frac{4b}{m}\right] \\ &= \frac{n}{2}\binom{m/2}{b}^2\left[1 - \frac{2b}{m}\right]. \end{aligned}$$

It follows that $z_P = z_D$, and by the discussion earlier in this proof, both are optimal. ■

Now let us study in more detail the optimal system of vectors v_S given above. Define $y_S = v_S\sqrt{m}$. Since the lth component of v_S is $\pm 1/\sqrt{m}$, each component of y_S is ± 1. Let us define a sequence of cuts $\mathcal{C}_l = (\mathcal{S}_l, \bar{\mathcal{S}}_l)$, $1 \leq l \leq m$, with $\mathcal{S}_l = \{S | S \ni l, |S| = m/2\}$. Letting $(y_S)_l$ denote the lth component of vector y_S, notice that for all S,

$$S \in \mathcal{S}_l \iff l \in S \iff (y_S)_l = +1.$$

When we solve the semidefinite program, we *hope* that the optimal vectors have the following property: there is an l such that each vector is ± 1 in component l, and 0 elsewhere; such vectors correspond naturally to cuts, and any solution of that form would be a maximum cut, since its weight would equal that of the optimum of the semidefinite program. Of course, the vectors y_S given above have ± 1 in *every* coordinate. This means that each vector v_S is the sum, over $l = 1, 2, 3, ..., m$, of a vector which is ± 1 in component l and 0 elsewhere (and then scaled down to unit length). The optimal family (v_S) can be viewed as an "average" of families of vectors representing the cuts $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m$. This motivates the following.

Lemma 5. If $b \leq m/12$, then the optimal value z_P^* of the semidefinite program is the average of the weights of cuts $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m$.

Proof. Where $(v_S)_l$ and $(y_S)_l$ denote the lth coordinates of v_S and y_S, respectively, we have

$$\begin{aligned} z_P^* &= \frac{1}{2}\sum_{S,T} w_{S,T}\frac{1 - v_S \cdot v_T}{2} \\ &= \frac{1}{2}\sum_{S,T} w_{S,T}\frac{1}{m}\sum_{l=1}^{m}\left[\frac{1 - m(v_S)_l(v_T)_l}{2}\right] \\ &= \frac{1}{m}\frac{1}{2}\sum_{S,T} w_{S,T}\sum_{l=1}^{m}\left[\frac{1 - (y_S)_l(y_T)_l}{2}\right] \\ &= \frac{1}{m}\sum_{l=1}^{m}\left[\frac{1}{2}\sum_{S,T} w_{S,T}\left[\frac{1 - (y_S)_l(y_T)_l}{2}\right]\right]. \end{aligned}$$

Since the weight of cut $\mathcal{C}_l$ is exactly

$$\frac{1}{2}\sum_{S,T} w_{S,T}\left[\frac{1 - (y_S)_l(y_T)_l}{2}\right],$$

we are done. ■

Lemma 6. Each cut $\mathcal{C}_l$ is a maximum cut (if $b \leq m/12$).

Proof. Because (SD) is a relaxation of MAX CUT, z_P^* is an upper bound on the weight of any cut. If any cut $\mathcal{C}_l$ were not a maximum cut, its weight would be strictly less than z_P^*, and so would the average weight of these cuts. ■

We have proven

Theorem 7. The weight of a maximum cut in $J(m, m/2, b)$ is exactly

$$z_P^* = \frac{n}{2}\binom{m/2}{b}^2\left[1 - \frac{2b}{m}\right]$$

if m is even and $b \leq m/12$.

Now we study what happens when additional constraints are added to the semidefinite program. The results on this subject were obtained in collaboration with Uri Feige, who showed how the author's original argument, which applied only to the FGW constraints, could be generalized.

Where a_{ij}, $1 \leq i < j \leq n$, and b are reals, let us call a constraint

$$\sum_{i<j} a_{ij}(z_i \cdot z_j) \geq b$$

valid if it is satisfied whenever each z_i is the *integer* ± 1. (Whether a linear constraint is valid doesn't depend, of course, on the w_{ij}'s.)

If

$$b' = \frac{-b + \sum_{i<j} a_{ij}}{2},$$

then by a simple linear transformation,

$$\sum_{i<j} a_{ij}(z_i \cdot z_j) \geq b \iff \sum_{i<j} a_{ij}\frac{1 - z_i \cdot z_j}{2} \leq b'.$$

Now build a complete graph H_a on $\{1, 2, ..., n\}$ with the (possibly negative) weight of edge $\{i, j\}$ being a_{ij}. Validity of the constraint is equivalent to saying that the weight of a maximum cut in H_a is at most b' (where we are including the trivial cut with one empty side as a cut).

Now suppose that $n = \binom{m}{m/2}$ for some positive even m. Label the v_S vectors above as $v_1, v_2, ..., v_n$ arbitrarily. Define $y_i = v_i\sqrt{m}$.

Theorem 8. Let a_{ij}, for all vertices $1 \le i < j \le n$, and b be reals such that the constraint

$$\sum_{i<j} a_{ij}(z_i \cdot z_j) \ge b$$

is valid. Then the constraint is necessarily satisfied when $z_i = v_i$ for all i.

Proof.

$$\begin{aligned} \sum_{i<j} a_{ij}(v_i \cdot v_j) &= \sum_{i<j} a_{ij} \sum_{l=1}^{m} (v_i)_l (v_j)_l \\ &= \frac{1}{m} \sum_{l=1}^{m} \sum_{i<j} a_{ij}(y_i)_l(y_j)_l. \end{aligned}$$

Clearly $(y_i)_l \in \{-1, +1\}$ for all vertices i. Therefore

$$\sum_{i<j} a_{ij}(y_i)_l(y_j)_l \ge b.$$

Hence

$$\frac{1}{m} \sum_{l=1}^{m} \sum_{i<j} a_{ij}(y_i)_l(y_j)_l \ge \frac{1}{m} \sum_{l=1}^{m} b = b. \blacksquare$$

Theorem 8 has the following consequence. Since the vectors $v_1, ..., v_n$ satisfy all valid constraints, and happen to be optimal for the semidefinite program (SD) associated with $J(m, m/2, b)$ (at least if $0 \le b \le m/12$), they are obviously optimal in the augmented program for $J(m, m/2, b)$ (if $b \le m/12$). Of course, the procedure that rounds the v_i's will proceed as before. However badly the algorithm rounded the vectors (v_i) beforehand, it will round them just as badly afterward.

For example, consider the FGW constraints, proposed by Feige, Goemans, and Williamson [FG, GW]: For each triple (i, j, k) of vertices, $1 \le i < j < k \le n$,

$$\begin{aligned} +z_i \cdot z_j + z_i \cdot z_k + z_j \cdot z_k &\ge -1 \\ -z_i \cdot z_j - z_i \cdot z_k + z_j \cdot z_k &\ge -1 \\ -z_i \cdot z_j + z_i \cdot z_k - z_j \cdot z_k &\ge -1 \\ +z_i \cdot z_j - z_i \cdot z_k - z_j \cdot z_k &\ge -1. \end{aligned}$$

The second, third, and fourth constraints in each block of four are obtained from the first one by negating z_i, z_j, or z_k, respectively. The reader can verify that these four constraints are valid. (The key is that given any three integers in $\{-1, +1\}$, not all three pairwise products can be -1.) It follows from Theorem 8, if $n = \binom{m}{m/2}$, that the vectors $z_i = v_i$ satisfy these vector constraints.

For another example, suppose we augment (SD) by adding all infinitely many possible valid constraints. That is, for each $a = (a_{ij})$, $1 \le i < j \le n$, we build a weighted complete graph H_a on $\{1, 2, ..., n\}$, with weight a_{ij} on edge $\{i, j\}$. We then calculate the maximum weight b' of a (possibly trivial) cut in H_a and add a constraint

$$\sum_{i<j} a_{ij} \frac{1 - v_i \cdot v_j}{2} \le b'.$$

This constraint is evidently valid, and furthermore, by looking at all a's, we are including all possible valid constraints (except that we are using the best possible right-hand side for each a). By Theorem 8, if $n = \binom{m}{m/2}$, then the vectors $v_1, v_2, ..., v_n$ will satisfy all the infinitely many additional constraints (of which the FGW constraints are but a subset), and, being optimal for (SD) for $J(m, m/2, b)$, will still be optimal afterward (if $b \le m/12$).

Now let us calculate just how badly Algorithm GW performs on $J(m, m/2, b)$, keeping in mind that the addition of more valid inequalities would not improve the outcome.

Lemma 9. The expected value of the weight W of the random cut produced from Y and the v_S's is exactly

$$\frac{n}{2}\binom{m/2}{b}^2 \frac{1}{\pi} \arccos\left[\frac{4b}{m} - 1\right].$$

Proof. The expected weight of the cut is

$$\begin{aligned} E[W] &= \frac{1}{2} \sum_{S,T:|S|=|T|=m/2} w_{S,T} \arccos(v_S \cdot v_T)/\pi \\ &= \frac{1}{2} \sum_{S,T:|S|=|T|=m/2, |S \cap T| = b} \arccos(v_S \cdot v_T)/\pi. \end{aligned}$$

By symmetry, we can take $S = \{1, 2, ..., m/2\}$ and get

$$E[W] = \frac{1}{2} n \sum \frac{1}{\pi} \arccos(v_S \cdot v_T),$$

where the summation is over all T such that $|T| = m/2$ and $|T \cap \{1, 2, ..., m/2\}| = b$. Thus

$$E[W] = \frac{n}{2}\binom{m/2}{b}^2 \frac{1}{\pi} \arccos\left[\frac{4b - m}{m}\right]. \blacksquare$$

We now know that the expected weight of the cut is exactly

$$E[W] = \frac{n}{2}\binom{m/2}{b}^2 \frac{1}{\pi} \arccos\left[\frac{4b}{m} - 1\right]$$

and that a maximum cut has weight exactly

$$\frac{n}{2}\binom{m/2}{b}^2 \left[1 - \frac{2b}{m}\right],$$

if $b \le m/12$. The ratio of $E[W]$ to the weight of a maximum cut is exactly

$$\frac{\frac{1}{\pi} \arccos\left[\frac{4b}{m} - 1\right]}{1 - \frac{2b}{m}}.$$

Where $\theta = \arccos[\frac{4b}{m} - 1]$, the ratio is $\frac{2}{\pi}\frac{\theta}{1 - \cos\theta}$. But $\alpha = \min_{0<\theta\le\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}$. By choosing b, m, and hence θ auspiciously, we will approach the α of [GW].

Theorem 10. For each $\epsilon > 0$, there are b and m, m even and positive and $0 \le b \le m/12$, such that the expected weight of the cut produced by Algorithm GW applied to $J(m, m/2, b)$ is at most $\alpha + \epsilon$ times the weight of a maximum cut.

Proof. A simple calculation shows that the unique θ^* that achieves the minimum value in the right-hand side of $\alpha = \min_{0<\theta\le\pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}$ corresponds to a value of b which is less than $m/12.5$, and hence is safely away from $m/12$. Because $\frac{2}{\pi}\frac{\theta}{1-\cos\theta}$ is continuous, we can find a $\delta > 0$ so that if $\theta \in [\theta^* - \delta, \theta^* + \delta]$, then $\frac{2}{\pi}\frac{\theta}{1-\cos\theta} \le \alpha + \epsilon$. We can certainly choose a rational $\gamma < \frac{1}{12.5}$ such that $\theta = \arccos(4\gamma - 1)$ lies in $[\theta^* - \delta, \theta^* + \delta]$. Then choose m even such that $b = \gamma m$ is an integer. But $\frac{2}{\pi}\frac{\theta}{1-\cos\theta}$ is precisely the ratio between $E[W]$ and the weight of a maximum cut. ■

Last, we prove Theorem 3.

Proof. We must prove that the smallest eigenvalue of $J(m, m/2, b)$ is

$$\binom{m/2}{b}^2 \left[\frac{4b}{m} - 1\right],$$

provided that m is even and that $0 \le b \le m/12$. Recall from Corollary 2 that the eigenvalues of $J(m, m/2, b)$ are $\beta_0, \beta_1, ..., \beta_{m/2}$, where

$$\beta_s = \sum_r (-1)^{s-r} \binom{s}{r} \binom{m/2 - r}{b - r} \binom{m/2 - s + r}{b}.$$

Since

$$\beta_1 = \binom{m/2}{b}^2 \left[\frac{4b}{m} - 1\right],$$

we must show that

$$\beta_1 \le \sum_r (-1)^{s-r} \binom{s}{r} \binom{m/2 - r}{b - r} \binom{m/2 - s + r}{b}$$

for $s = 0, 1, ..., m/2$.

Fix s, $0 \le s \le m/2$. Given r, let

$$t_r = \binom{m/2 - r}{b - r} \binom{m/2 - s + r}{b}$$

so that

$$\beta_s = (-1)^s \sum_r (-1)^r \binom{s}{r} t_r.$$

If s is even,

$$\beta_s \ge - \sum_{r \text{ odd}} \binom{s}{r} t_r;$$

and if s is odd,

$$\beta_s \ge - \sum_{r \text{ even}} \binom{s}{r} t_r.$$

Let us assume that $b \le m/2 - s + r$ and that $b \ge r$, for otherwise $t_r = 0$. Then

$$\frac{t_r}{\binom{m/2}{b}^2}$$

$$= \frac{(m/2 - r)!}{(m/2)!} \frac{(m/2 - s + r)!}{(m/2)!} \frac{b!}{(b - r)!} \frac{(m/2 - b)!}{(m/2 - s + r - b)!}$$

The product of the first and third fractions equals

$$\left[\frac{b(b-1)(b-2)\cdots(b-r+1)}{(\frac{m}{2})(\frac{m}{2}-1)(\frac{m}{2}-2)\cdots(\frac{m}{2}-r+1)}\right],$$

while the product of the second and fourth equals

$$\left[\frac{(\frac{m}{2}-b)(\frac{m}{2}-b-1)(\frac{m}{2}-b-2)\cdots(\frac{m}{2}-s+r-b+1)}{(\frac{m}{2})(\frac{m}{2}-1)(\frac{m}{2}-2)\cdots(\frac{m}{2}-s+r+1)}\right].$$

It is easy to verify that for $0 \le i < m/2$,

$$\frac{b - i}{\frac{m}{2} - i} \le \frac{b}{\frac{m}{2}}$$

and

$$\frac{\frac{m}{2} - b - i}{\frac{m}{2} - i} \le \frac{\frac{m}{2} - b}{\frac{m}{2}}.$$

Therefore

$$\frac{t_r}{\binom{m/2}{b}^2} \le \left(\frac{2b}{m}\right)^r \left(1 - \frac{2b}{m}\right)^{s-r}.$$

This is true whether or not $b \le m/2 - s + r$ and $b \ge r$. Let $\gamma = b/m \le 1/12$. Then

$$\frac{t_r}{\binom{m/2}{b}^2} \le (2\gamma)^r (1 - 2\gamma)^{s-r}.$$

For any p, let

$$x = \sum_{\text{odd } r} \binom{s}{r} p^r (1-p)^{s-r}$$

and

$$y = \sum_{\text{even } r} \binom{s}{r} p^r (1-p)^{s-r}.$$

Then $x + y = 1$ and $-x + y = (-p + (1-p))^s = (1-2p)^s$. It follows that

$$y = 1/2 + (1/2)(1-2p)^s$$

and

$$x = 1/2 - (1/2)(1-2p)^s.$$

Hence

$$\sum_{\text{odd } r} \binom{s}{r} (2\gamma)^r (1-2\gamma)^{s-r} = 1/2 - (1/2)(1-4\gamma)^s$$

and

$$\sum_{\text{even } r} \binom{s}{r} (2\gamma)^r (1-2\gamma)^{s-r} = 1/2 + (1/2)(1-4\gamma)^s.$$

Now

$$\frac{t_r}{\binom{m/2}{b}^2} \le (2\gamma)^r (1 - 2\gamma)^{s-r}.$$

So, if s is even,

$$\begin{aligned}\beta_s &\geq -\left[\sum_{\text{odd } r}\binom{s}{r}(2\gamma)^r(1-2\gamma)^{s-r}\right]\binom{m/2}{b}^2\\ &= -\left[\frac{1}{2}-\frac{1}{2}(1-4\gamma)^s\right]\binom{m/2}{b}^2\\ &\geq -\left[\frac{1}{2}+\frac{1}{2}(1-4\gamma)^s\right]\binom{m/2}{b}^2;\end{aligned}$$

if s is odd we have

$$\beta_s \geq -\left[\frac{1}{2}+\frac{1}{2}(1-4\gamma)^s\right]\binom{m/2}{b}^2.$$

So, in either case,

$$\beta_s \geq -\left[\frac{1}{2}+\frac{1}{2}(1-4\gamma)^s\right]\binom{m/2}{b}^2.$$

Now we need a technical claim.

Claim 11. If $p \in [0,1/3]$ and $s \geq 4$, then $(1-p)^s \leq 1-2p$.

Proof. Let $f(p) = \frac{(1-p)^s}{1-2p}$ for $p \in [0,1/3]$. $f'(p) = \frac{(1-p)^{s-1}}{(1-2p)^2}[2(1-p)-s(1-2p)]$. $f(0)=1$. We prove that $f'(p) \leq 0$ for all $p \in [0,1/3]$, thereby proving that $f(p) \leq 1$ for all $p \in [0,1/3]$.

$$\begin{aligned} f'(p) \leq 0 &\iff 2(1-p)-s(1-2p) \leq 0\\ &\iff 2-2p-s+2ps \leq 0\\ &\iff p(2s-2) \leq s-2\\ &\iff p \leq \frac{s-2}{2s-2}.\end{aligned}$$

Since $s \geq 4$, $\frac{s-2}{2s-2} \geq 1/3$. Therefore $p \leq 1/3$ implies that $p \leq \frac{s-2}{2s-2}$ and $f'(p) \leq 0$. ■

By the claim, because $\gamma \leq 1/12$, $(1-4\gamma)^s \leq 1-8\gamma$ if $s \geq 4$. So

$$\begin{aligned}\beta_s &\geq -\left[\frac{1}{2}+\frac{1}{2}(1-8\gamma)\right]\binom{m/2}{b}^2\\ &= (4\gamma-1)\binom{m/2}{b}^2\\ &= \left[\frac{4b}{m}-1\right]\binom{m/2}{b}^2,\end{aligned}$$

and we are finished, in the case $s \geq 4$.

Now we must dispense with $s = 0,1,2,3$ by showing that $\beta_s \geq \beta_1 = \binom{m/2}{b}^2\left[\frac{4b}{m}-1\right]$ for $s = 0,1,2,3$. Notice that if $b = 0$, then for any even m, $\beta_s = (-1)^s$. This means that the smallest eigenvalue is -1, as it should be. In what follows, we assume that $m \geq 12$, for otherwise $b \leq m/12$ makes $b = 0$.

The $s=0$ case is first. $\beta_0 = \binom{m/2}{b}^2$, the degree of a vertex in $J(m, m/2, b)$. Since this is positive and $\binom{m/2}{b}^2\left[\frac{4b}{m}-1\right]$ is negative, this part is easy.

The $s=1$ case is trivial.

The $s=2$ case is still not hard.

$$\frac{\beta_2}{\binom{m/2}{b}^2} = \frac{(\frac{m}{2}-b)}{\frac{m}{2}}\frac{(\frac{m}{2}-b-1)}{\frac{m}{2}-1} - 2\frac{b}{\frac{m}{2}}\frac{(\frac{m}{2}-b)}{\frac{m}{2}} + \frac{b}{\frac{m}{2}}\frac{(b-1)}{(\frac{m}{2}-1)}.$$

Since $\beta_1 = \binom{m/2}{b}^2\left[\frac{4b}{m}-1\right] < 0$, it suffices to prove that

$$\frac{(\frac{m}{2}-b)}{\frac{m}{2}}\frac{(\frac{m}{2}-b-1)}{\frac{m}{2}-1} \geq 2\frac{b}{\frac{m}{2}}\frac{(\frac{m}{2}-b)}{\frac{m}{2}}.$$

That

$$\frac{\frac{m}{2}-b-1}{\frac{m}{2}-1} \geq \frac{2b}{\frac{m}{2}}$$

holds follows from $b \leq m/12$ and $m \geq 12$. This means that

$$\frac{(\frac{m}{2}-b)}{\frac{m}{2}}\frac{(\frac{m}{2}-b-1)}{\frac{m}{2}-1} \geq 2\frac{b}{\frac{m}{2}}\frac{(\frac{m}{2}-b)}{\frac{m}{2}},$$

and the proof of the case $s=2$ is complete.

Now we attack $s=3$. A simple calculation gives

$$\begin{aligned}\frac{\beta_3}{\binom{m/2}{b}^2} = &-\left[\frac{(\frac{m}{2}-b)}{(\frac{m}{2})}\frac{(\frac{m}{2}-b-1)}{(\frac{m}{2}-1)}\frac{(\frac{m}{2}-b-2)}{(\frac{m}{2}-2)}\right]\\ &+3\left[\frac{b}{\frac{m}{2}}\right]\left[\frac{(\frac{m}{2}-b)}{\frac{m}{2}}\frac{(\frac{m}{2}-b-1)}{(\frac{m}{2}-1)}\right]\\ &-3\left[\frac{b}{\frac{m}{2}}\frac{(b-1)}{(\frac{m}{2}-1)}\right]\left[\frac{\frac{m}{2}-b}{\frac{m}{2}}\right]\\ &+\left[\frac{b}{\frac{m}{2}}\frac{(b-1)}{(\frac{m}{2}-1)}\frac{(b-2)}{(\frac{m}{2}-2)}\right].\end{aligned}$$

It clearly suffices to prove that

$$-\frac{(\frac{m}{2}-b)}{(\frac{m}{2})}\frac{(\frac{m}{2}-b-1)}{(\frac{m}{2}-1)}\frac{(\frac{m}{2}-b-2)}{(\frac{m}{2}-2)} - 3\frac{b}{\frac{m}{2}}\frac{(b-1)}{(\frac{m}{2}-1)}\frac{(\frac{m}{2}-b)}{\frac{m}{2}} \geq \frac{4b}{m}-1.$$

Simple calculations yield that

$$-\left(\frac{\frac{m}{2}-b}{\frac{m}{2}}\right)^3 \leq -\frac{(\frac{m}{2}-b)}{\frac{m}{2}}\frac{(\frac{m}{2}-b-1)}{\frac{m}{2}-1}\frac{(\frac{m}{2}-b-2)}{\frac{m}{2}-2}$$

and that

$$-3\frac{b}{\frac{m}{2}}\frac{b}{\frac{m}{2}}\frac{(\frac{m}{2}-b)}{\frac{m}{2}} \leq -3\frac{b}{\frac{m}{2}}\frac{(b-1)}{(\frac{m}{2}-1)}\frac{(\frac{m}{2}-b)}{\frac{m}{2}}.$$

This means that it is sufficient to prove that

$$-\left(\frac{\frac{m}{2}-b}{\frac{m}{2}}\right)^3 - 3\left(\frac{b}{\frac{m}{2}}\right)^2\frac{\frac{m}{2}-b}{\frac{m}{2}} \geq \frac{4b}{m}-1.$$

Let $\gamma = b/m$. The left-hand side equals

$$-(1-2\gamma)^3-3(2\gamma)^2(1-2\gamma) = -1+4\gamma-16\gamma^2+2\gamma-8\gamma^2+32\gamma^3.$$

Now

$$\begin{aligned} &-1+6\gamma-24\gamma^2+32\gamma^3 \geq 4\gamma-1\\ \iff\ & 32\gamma^3-24\gamma^2+2\gamma \geq 0\\ \iff\ & 16\gamma^2-12\gamma+1 \geq 0.\end{aligned}$$

The roots of $16\gamma^2 - 12\gamma + 1 = 0$ are $(3 \pm \sqrt{5})/8$. Since $(3 - \sqrt{5})/8 > 0.095$,

$$16\gamma^2 - 12\gamma + 1 \geq 0$$

if $\gamma \leq 0.095$. Because $\gamma \leq 1/12 \leq 0.095$, we are finished. ■

Although we don't need it, this paper wouldn't be complete without this conjecture:

Conjecture 12. For all even $m > 0$, for all $0 \leq b < m/4$, the smallest eigenvalue of $J(m, m/2, b)$ is exactly

$$\beta_1 = \binom{m/2}{b}^2 \left[\frac{4b}{m} - 1\right].$$

Possibly this is already known.

3 Acknowledgments

I am deeply grateful to David Williamson. It was David who originally suggested the problem of analyzing the effect of the FGW constraints on the performance of Algorithm GW. And without David's help in finding a maximum cut and solving (SD) for a particular graph, this paper wouldn't exist. I thank him also for his comments on an earlier draft of this paper.

For the far more general result on the power of additional inequalities than what appeared in an earlier draft, for allowing me to include the result, and for his comments on an earlier draft, I am very thankful to Uri Feige.

For his pointers to the literature on algebraic graph theory, I am grateful to Neil Calkin. I thank Leonard Schulman and Cristina Fernandes for their help in studying the eigenvalues and eigenvectors of the adjacency matrices of the $J(m, t, b)$'s. And I am thankful to Don Knuth for writing his notes "Combinatorial Matrices."

References

[BM] F. Barahona and A. R. Mahjoub, "On the Cut Polytope," *Mathematical Programming* 36 (1986), 157-173.

[Delsarte] P. Delsarte, "Hahn Polynomials, Discrete Harmonics, and t-Designs," SIAM Journal on Applied Math 34 (1978), 157-166.

[FG] U. Feige and M. X. Goemans, "Approximating the Value of Two Prover Proof Systems, With Applications to MAX 2SAT and MAX DICUT," *Proc. Third Israel Symposium on Theory of Computing and Systems*, Israel, 1995.

[GW] M. X. Goemans and D. P. Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming," to appear in *Journal of the Association for Computing Machinery*. A preliminary version of this paper appeared in *Proc. 26th ACM Symposium on the Theory of Computing* (1994), Montréal, 422-431.

[KMS] D. Karger, R. Motwani, and M. Sudan, "Approximate Graph Coloring by Semidefinite Programming," *Proc. 35th ACM Symposium on the Foundations of Computer Science* (1994), Santa Fe, 2-13.

[Knuth] D. Knuth, "Combinatorial Matrices," notes of Institut Mittag-Leffler, Nov., 1991, revised Jan., 1993.