Approximation Algorithms for MAX-3-CUT and Other Problems via Complex Semidefinite Programming*

Michel X. Goemans — MIT†

David P. Williamson — IBM Almaden‡

Dedicated to the memory of Daniel M. Lewin§

July 24, 2002

Abstract

A number of recent papers on approximation algorithms have used the square roots of unity, -1 and 1, to represent binary decision variables for problems in combinatorial optimization, and have relaxed these to unit vectors in real space using semidefinite programming in order to obtain near optimum solutions to these problems. In this paper, we consider using the cube roots of unity, 1, $e^{i2\pi/3}$, and $e^{i4\pi/3}$, to represent ternary decision variables for problems in combinatorial optimization. Here the natural relaxation is that of unit vectors in complex space. We use an extension of semidefinite programming to complex space to solve the natural relaxation, and use a natural extension of the random hyperplane technique introduced by the authors in [10] to obtain near-optimum solutions to the problems.

In particular, we consider the problem of maximizing the total weight of satisfied equations $x_u - x_v \equiv c \pmod 3$ and inequations $x_u - x_v \not\equiv c \pmod 3$, where $x_u \in \{0, 1, 2\}$ for all u. This problem can be used to model the MAX-3-CUT problem and a directed variant we call MAX-3-DICUT. For the general problem, we obtain a 0.793733-approximation algorithm. If the instance contains only inequations (as it does for MAX-3-CUT), we obtain a performance guarantee of $\frac{7}{12} + \frac{3}{4\pi^2}\arccos^2(-1/4) - \epsilon > 0.836008$. This compares with proven performance guarantees of .800217 for MAX-3-CUT (by Frieze and Jerrum [9]) and $\frac{1}{3} + 10^{-8}$ for the general problem (by Andersson, Engebretsen, and Håstad [3]). It matches the guarantee of .836008 for MAX-3-CUT found independently by de Klerk, Pasechnik, and Warners [6]. We show that all these algorithms are in fact equivalent in the case of MAX-3-CUT, and that our algorithm is the same as that of Andersson et al. in the more general case.

1 Introduction

The uses of linear programming for designing algorithms for combinatorial optimization problems has long been known. Since the seminal paper of Lovász [20], researchers have investigated the use of nonlinear programming, particularly semidefinite programming, since semidefinite programs can be solved in polynomial time (up to any prescribed accuracy). At first it was shown that

*A preliminary version of this paper appeared in the Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing, Crete, Greece, July 2001.

†Research supported by NSF under contracts 9623859-CCR and 0098018-CCR, and by the IBM University Partnership Program. Address: 77 Mass. Ave., Room 2-351, Cambridge, MA 02139. Email: `goemans@math.mit.edu`.

‡Address: 650 Harry Rd., K53/B1, San Jose, CA 95120. Email:`dpw@almaden.ibm.com`. WWW: `www.almaden.ibm.com/people/cs/dpw`.

§Please see the Dedication section at the end of the paper.

semidefinite programs could be solved via the ellipsoid method [12, 13]; later it was shown that practical interior-point methods for linear programs could be adapted to semidefinite programming [1, 23].

One particular use of semidefinite programming that has received a good deal of attention is its application to approximation algorithms. An α-approximation algorithm is a polynomial-time algorithm for a combinatorial optimization problem that produces a solution of value within a factor of α of the optimum value. A *randomized* α-approximation algorithm is an algorithm that runs in randomized polynomial time and produces a solution whose expected value (over the random choices of the algorithm) is within a factor of α of the optimum value. In [10], the authors introduced the following technique. An optimization problem with binary decisions (e.g. MAX CUT, MAX 2SAT) is modelled with n variables $x_i \in \{-1, +1\}$, and an objective function quadratic in these variables. These variables x_i are relaxed to n-dimensional unit length vectors v_i, and their products $x_i x_j$ to inner products $\langle v_i, v_j \rangle$. It is not difficult to show that this relaxation is a semidefinite program, and thus can be solved in polynomial time, up to any given accuracy. The solution to the relaxation is mapped to the solution in the original variables by choosing a random vector r from the n-dimensional normal distribution, and deriving the solution $x_i = \operatorname{sgn}(\langle r, v_i \rangle)$. The expected value of this solution can be compared to the value of the relaxation, and this leads to randomized α-approximation algorithms. This technique has been applied, with modifications, to several other problems in combinatorial optimization (e.g. [3, 4, 8, 9, 17, 22, 27, 30]). Almost any randomized approximation algorithm based on the hyperplane technique applied to a semidefinite relaxation can be derandomized, as was shown by Mahajan and Ramesh [21].

In the meantime, researchers in mathematical programming have shown that the interior-point methods that extend polynomial-time solvability from linear programming to semidefinite programming also extend polynomial-time solvability to the class of *symmetric cones*. This includes the second-order cone (also called the ice-cream cone), the cone of semidefinite matrices over real numbers/complex numbers/quaternions, and the cone of 3×3 semidefinite matrices over octonions. Toh and Trefethen showed that semidefinite programming over the complex domain is solvable in polynomial time [28]. Güler [14] showed that the self-scaled cones for which Nesterov and Todd [24, 25] had provided polynomial-time algorithms are exactly the symmetric cones.

We propose to take advantage of these developments by using semidefinite programming over the complex domain; we call this *complex semidefinite programming* (CSDP). Previously we modelled binary decisions with the square roots of unity ± 1; we now model ternary decisions with the cube roots of unity 1, $e^{i2\pi/3}$, and $e^{i4\pi/3}$. Given a model of this problem with an objective function quadratic in these variables, we relax these variables to unit length vectors $v_i \in \mathbb{C}^n$, where $\mathbb{C}^n$ is n-dimensional complex space, and relax their products $x_i x_j$ to inner products $\langle v_i, v_j \rangle$. This relaxation turns out to be a semidefinite program over the complex domain, and is solvable in polynomial time (up to any prescribed accuracy). We extend our randomized rounding technique by drawing a random vector r from the n-dimensional complex normal distribution, then mapping back to the original space by considering the angle θ of the inner product $se^{i\theta} = \langle r, v_i \rangle$.

We apply these ideas to the problem of maximizing the weight of satisfied two variable equations or inequations mod 3. In particular, in the problem Max 2-Lin-Mod-3 we have n variables x_i and m equations $x_i - x_j \equiv c \pmod 3$ or inequations $x_i - x_j \not\equiv c \pmod 3$. For each (in)equation j we are given a nonnegative weight w_j, and we must find values of the $x_i \in \{0, 1, 2\}$ that maximize the total weight of the satisfied (in)equations. Notice that our equations and inequations only involve $x_i - x_j$; we do not allow more general terms such as $x_i + x_j$. The problem Max 2-Lin-Mod-3 can be used to model other optimization problems. We give two examples here, Max-3-Cut and Max-3-Dicut. In Max-3-Cut, we are given a graph $G = (V, E)$ with a weight w_e on edge $e \in E$. We assume that the weights are nonnegative. For a partition of V into 3 sets V_0, V_1 and V_2, we let $\delta(V_0, V_1, V_2) = \{e = (u, v) \in E : u \in V_i, v \in V_j \text{ for } i \neq j\}$. The goal is to find a 3-partition (V_0, V_1, V_2) such that $w(\delta(V_0, V_1, V_2))$ is maximized where $w(F) = \sum_{e \in F} w_e$. MAX-3-DICUT is similarly defined except that G is now a digraph and we would like to maximize $w(\delta^+(V_0, V_1, V_2))$ where

$$\delta^+(V_0, V_1, V_2) = \{e = (u, v) \in E : u \in V_i, v \in V_j \text{ with } j - i \equiv 1 \pmod 3\}.$$

To model these two problems, we introduce a variable x_v for each $v \in V$. For the problem MAX-3-CUT, we introduce an inequation $x_u - x_v \not\equiv 0 \pmod 3$ of weight w_e for each edge $e = (u, v)$. For the problem MAX-3-DICUT, we introduce an equation $x_v - x_u \equiv 1 \pmod 3$ of weight w_e for each directed edge $e = (u, v)$.

Our main results are as follows. We obtain a randomized $(\rho - \epsilon)$-approximation algorithm for the general problem MAX 2-LIN-MOD-3 for any constant $\epsilon > 0$, where

$$\rho = \min_{-1/2 \leq x \leq 1} \frac{\frac{7}{24} + \frac{3}{4\pi^2} \arccos^2(-x) - \frac{3}{8\pi^2} \arccos^2(x - 0.5)}{\frac{1}{3} + \frac{2}{3}x} > 0.793733.$$

This implies a randomized 0.793733-approximation algorithm for MAX-3-DICUT. We also obtain a randomized $(\phi - \epsilon)$-approximation algorithm for instances of MAX 2-LIN-MOD-3 that have inequations only, where

$$\phi = \frac{7}{12} + \frac{3}{4\pi^2} \arccos^2(-1/4) > 0.836008.$$

This implies a randomized 0.836008-approximation algorithm for MAX-3-CUT. Our randomized algorithm seems to fit the framework of Mahajan and Ramesh [21] and we therefore expect that it can be derandomized using their approach.

Previously, Andersson, Engebretsen, and Håstad [3] considered a more general variant of the problem MAX 2-LIN-MOD-P (for two-variable equations mod p) and shown that they could obtain an $(\frac{1}{p} + \epsilon(p))$-approximation algorithm, where $\epsilon(p) > 0$ is a constant that depends on p; for our simpler version, they gave a proven performance guarantee of $\frac{1}{p}(1 + 10^{-8})$. Their algorithm is based on a (real) semidefinite programming relaxation involving p vectors for each variable. In the conclusion of their manuscript, the authors claim numerical evidence for a bound of $1/1.27 \approx .787$ in the case $p = 3$. This is actually not surprising since we are able to show that our complex semidefinite programming relaxation (for $p = 3$) and our rounding scheme are both equivalent to theirs; we can therefore prove that their performance guarantee is 0.793733. Frieze and Jerrum [9] give a $(\frac{k-1}{k} + \epsilon(k))$-approximation algorithm for the MAX k-CUT problem, and in particular a .800217-approximation algorithm for MAX-3-CUT. In [9, pg. 74], the authors state that by computing further terms in their series expansion, they are able to obtain a bound of .832718 for MAX-3-CUT. Frieze and Jerrum also use a (real) semidefinite programming relaxation and we can show that in the case of MAX-3-CUT, our complex relaxation and our rounding scheme is equivalent to theirs, and thus the performance guarantee of their algorithm is 0.836008. Independently, de Klerk, Pasechnik, and Warners [6] have shown numerically that the performance guarantee of Frieze and Jerrum is .836008. Since the appearance of an extended abstract of this paper [11], de Klerk and Pasechnik [5] have shown that the Frieze and Jerrum analysis also yields a performance guarantee of exactly $\phi - \epsilon$. In terms of negative results, Andersson et al. [3] have shown that no α-approximation algorithm for MAX 2-LIN-MOD-3 can have $\alpha > 17/18 \approx .944$ unless $P = NP$.

Our use of complex semidefinite programming is in some sense purely a modelling decision, since complex semidefinite programming is reducible to standard (real) semidefinite programming. However, we think it makes for cleaner models, algorithms, and analysis than the equivalent models using standard semidefinite programming. For instance, although de Klerk and Pasechnik [5] were eventually able to compute analytically the performance guarantee of Frieze and Jerrum for Max-3-Cut, the guarantee [6] was initially determined numerically, whereas our performance guarantee for Max-3-Cut is analytic. Also, we believe our model is somewhat cleaner. The modelling approach of Frieze and Jerrum [9] for Max-3-Cut of introducing a single 2-dimensional vector for each vertex and then relaxing the dimension restriction does not immediately extend to the case of Max-3-Dicut since in the real case the inner product of two vectors commute. To handle the more general case (of Max-3-Dicut or Max 2-Lin-Mod-3), Andersson et al. [3] define three 2-dimensional vectors for each vertex. This is also the general approach taken by de Klerk et al. [6]. The approach described in this paper allows the use of a single complex number (1-dimensional vector) for each vertex or each variable, which is then relaxed to a higher-dimensional vector. The fundamental reason why we can easily formulate a non-symmetric problem (such as Max 2-Lin-Mod-3, in which $x_u - x_v \not\equiv x_v - x_u$) comes from the fact that the inner product of two complex vectors does not commute.

Our paper is organized as follows. In Section 2, we review basic ideas of complex numbers and matrices, and introduce some notation we will be using. In Section 3, we introduce complex semidefinite programming. Section 4 shows how we model the Max 2-Lin-Mod-3 problem in terms of the cube roots of unity, and how its relaxation is a complex semidefinite program. We also show there that our relaxation is equivalent to Frieze and Jerrum's relaxation in the case of Max-3-Cut. In Section 5 we introduce our algorithm and present its analysis. Section 6 contains the proofs of equivalence between the various algorithms of Frieze and Jerrum [9], Andersson et al. [3], and ours. Section 7 contains a discussion of some of the complexity issues since almost all computations have to be carried out approximately.

2 Preliminaries

For a complex number $z = x + iy \in \mathbb{C}$, we denote its real part x by Re z and its imaginary part y by Im z. For the polar representation $z = re^{i\theta} = \cos(\theta) + i\sin(\theta)$, r is the modulus $|z|$ and θ is the argument $\arg z$. The complex conjugate of $z = x + iy$ is denoted by $\bar{z} = x - iy$.

Let $\mathbb{R}^{n\times n}$ (resp. $\mathbb{C}^{n\times n}$) denote the set of $n \times n$ matrices over the field $\mathbb{R}$ (resp. $\mathbb{C}$). We denote the transpose of a matrix $M = [m_{ij}]$ (of any size, including of a vector) by $M^T = [m_{ji}]$ and its adjoint by $M^* = \bar{M}^T = [\bar{m}_{ji}]$. We extend the operators Re and Im to complex matrices by letting Re M for $M \in \mathbb{C}^{n\times n}$ be the real matrix [Re m_{ij}] and Im M be the real matrix [Im m_{ij}], so that $M = \text{Re } M + i\text{Im } M$.

A real matrix $M \in \mathbb{R}^{n\times n}$ is *symmetric* if $M = M^T$ and *skew-symmetric* if $M = -M^T$. A complex matrix $M \in \mathbb{C}^{n\times n}$ is *Hermitian* if $M = M^*$. We let S_n denote the set of symmetric $n \times n$ real matrices and H_n the set of complex $n \times n$ Hermitian matrices. Note that for $M \in H_n$, Re M is symmetric and Im M is skew-symmetric. As an inner product over $\mathbb{C}^n$, we use $\langle u, v\rangle = v^*u$. Recall that $\langle v, u\rangle = \overline{\langle u, v\rangle}$. For any two matrices $A, B \in H_n$, where $A = (a_{ij})$ and $B = (b_{ij})$, we let $\langle A, B\rangle$ be the (Frobenius) inner product of the two matrices; that is, $\langle A, B\rangle = Tr(B^*A) = \sum_{i,j} \bar{b}_{ij}a_{ij}$ (which is also $Tr(BA)$ or $Tr(AB)$ since $A, B \in H_n$). When $A, B \in H_n$, it is easy to see that $\langle A, B\rangle \in \mathbb{R}$.

Recall that for a Hermitian matrix M, we have that $x^*Mx = \langle Mx, x\rangle$ is real for all $x \in \mathbb{C}^n$, and all its eigenvalues are real. A Hermitian matrix $M \in H_n$ is said to be *positive semidefinite* if $x^*Mx \geq 0$ for all $x \in \mathbb{C}^n$. A Hermitian matrix is positive semidefinite if and only if all of its eigenvalues are nonnegative. We denote a positive semidefinite matrix $M \in H_n$ by $M \succeq 0$. A Hermitian matrix $M = (m_{ij}) \in H_n$ is positive semidefinite if and only if it can be expressed as $M = VV^*$ for some $V \in H_n$; thus M is positive semidefinite if and only if there exist vectors

$v_1, \ldots, v_n \in \mathbb{C}^n$ such that $m_{ij} = \langle v_i, v_j \rangle$.

3 Complex Semidefinite Programming

Our algorithms use the fact that semidefinite programming over the reals can be generalized by semidefinite programming over the complex domain. For lack of a better term, we will call this *complex semidefinite programming* (CSDP). As in semidefinite programming, in CSDP we maximize or minimize a linear function over a square matrix $Z \in H_n$ subject to linear constraints on Z and the constraint that Z is positive semidefinite. More formally, we maximize or minimize $\langle C, Z\rangle$ for an objective matrix $C \in H_n$, subject to m linear constraints of the form $\langle A_i, Z\rangle \geq b_i$ or $\langle A_i, Z\rangle = b_i$ or $\langle A_i, Z\rangle \leq b_i$ where $A_i \in H_n$, and subject to $Z \succeq 0$, $Z \in H_n$.

In fact, CSDP is reducible to SDP, and thus is not a strict generalization of it. To see this, consider the linear transformation $\mathcal{T}$ below that maps matrices in H_n to matrices in S_{2n}:

$$\mathcal{T}(Z) = \left[\begin{array}{cc} \text{Re } Z & -\text{Im } Z \\ \text{Im } Z & \text{Re } Z \end{array}\right].$$

Because Re Z is symmetric and Im Z is skew-symmetric, $\mathcal{T}(Z)$ is symmetric. Observe that, for any $A, B \in H_n$,

$$\begin{aligned} \langle \mathcal{T}(A), \mathcal{T}(B)\rangle &= Tr\left(\left[\begin{array}{cc} \text{Re } B & \text{Im } B \\ -\text{Im } B & \text{Re } B \end{array}\right]\left[\begin{array}{cc} \text{Re } A & -\text{Im } A \\ \text{Im } A & \text{Re } A \end{array}\right]\right) \\ &= 2Tr(\text{Re } B \text{Re } A + \text{Im } B \text{Im } A) = 2\langle A, B\rangle. \end{aligned} \tag{1}$$

Suppose we are trying to solve the following canonical CSDP:

$$\begin{aligned} &\text{Max} \quad \langle C, Z\rangle \\ &\text{subject to:} \\ &\qquad \langle A_i, Z\rangle \leq b_i \qquad\qquad i = 1, \ldots, m \\ &\qquad Z \succeq 0 \\ &\qquad Z \in H_n. \end{aligned}$$

Then consider the following SDP:

$$\begin{aligned} &\text{Max} \quad \langle \mathcal{T}(C), Y\rangle \\ &\text{subject to:} \\ &\qquad \langle \mathcal{T}(A_i), Y\rangle \leq 2b_i \qquad\qquad i = 1, \ldots, m \\ &\qquad \left\langle \left[\begin{array}{cc} E_{ij} & 0 \\ 0 & -E_{ij} \end{array}\right], Y\right\rangle = 0 \qquad i, j = 1, \ldots, n, i < j \qquad (2) \\ &\qquad \left\langle \left[\begin{array}{cc} 0 & E_{ij} \\ E_{ij} & 0 \end{array}\right], Y\right\rangle = 0 \qquad i, j = 1, \ldots, n, i < j \qquad (3) \\ &\qquad Y \succeq 0 \\ &\qquad Y \in S_{2n}, \end{aligned}$$

where e_i is the ith unit vector and $E_{ij} = e_i e_j^T + e_j e_i^T$ (that is, it has a 1 in positions (i, j) and (j, i) and zeroes elsewhere). Observe that the two classes of constraints (2) and (3) ensure that Y has the form

$$Y = \left[\begin{array}{cc} A & -B \\ B & A \end{array}\right],$$

for some symmetric A and skew-symmetric B. Thus for any Y of this form, $\mathcal{T}^{-1}(Y)$ is well-defined and unique.

We claim the SDP is entirely equivalent to the CSDP, since if solution Z is feasible for the CSDP, then $\mathcal{T}(Z)$ is feasible for the SDP and has twice the objective function value, while if Y is feasible for the SDP, then $\mathcal{T}^{-1}(Y)$ is feasible for the CSDP with half the objective function value. This is easy to see given (1), except for showing that $Z \succeq 0$ iff $\mathcal{T}(Z) \succeq 0$. This follows since if x is an eigenvector with eigenvalue λ for Z, then the vector

$$y = \begin{pmatrix} \operatorname{Re} x \\ \operatorname{Im} x \end{pmatrix}$$

is an eigenvector of $\mathcal{T}(Z)$ of eigenvalue λ. Thus if Z has nonnegative eigenvalues, so does $\mathcal{T}(Z)$. Similarly, if the vector

$$\begin{pmatrix} x \\ y \end{pmatrix}$$

is an eigenvector of eigenvalue λ for $Y \in S_{2n}$, where $x, y \in \mathbb{R}^n$, then $z = x + iy$ is an eigenvector for $\mathcal{T}^{-1}(Y)$ of eigenvalue λ.

The polynomial-time solvability of semidefinite programs (up to any given degree of accuracy) and the reduction claimed above immediately implies that complex semidefinite programs are also solvable in polynomial time (up to any given degree of accuracy). Additionally, several interior-point methods used to solve SDPs can be generalized to CSDPs [7, 2]. For the purpose of our approximation algorithms, most of the computations have to be done approximately and this is discussed in Section 7.

4 Formulating Max 2-Lin-Mod-3

For combinatorial optimization problems, decisions which have two alternatives (such as, whether an edge/vertex is present, or whether a job precedes another) are generally formulated by introducing a binary variable $x \in \{0, 1\}$, or possibly $x \in \{-1, 1\}$. When there are three alternatives, we propose to introduce a complex variable $x \in \mathbb{C}$ which takes as values the third roots of unity, 1, $e^{i2\pi/3}$ and $e^{i4\pi/3}$. For simplicity of notation, we will denote $e^{i2\pi/3}$ by ω and $e^{i4\pi/3}$ by ω^2 (or ω^{-1}). We let R_3 denote the set of cube roots of unity, then: $R_3 = \{1, \omega, \omega^2\}$.

To illustrate this principle, we consider the optimization problem Max 2-Lin-Mod-3. In Max 2-Lin-Mod-3, we are given n variables x_u, and m equations $x_u - x_v \equiv c \,(\mathrm{mod}\ 3)$ or inequations $x_u - x_v \not\equiv c \,(\mathrm{mod}\ 3)$, where $c \in \{0, 1, 2\}$. The jth (in)equation is associated with a nonnegative, integer weight $w_j \geq 0$. We must choose a setting of the variables x_u to values in $\{0, 1, 2\}$ in such a way that we maximize the total weight of satisfied (in)equations. As we showed in the introduction, Max 2-Lin-Mod-3 can be used to model other optimization problems, including Max-3-Cut and Max-3-Dicut.

It is possible to extend the problem to include equations of the form $x_u \equiv c \,(\mathrm{mod}\ 3)$ or inequations $x_u \not\equiv c \,(\mathrm{mod}\ 3)$. This can be reduced to Max 2-Lin-Mod-3 in the following way. We introduce a new variable x_0, and change each equation $x_u \equiv c \,(\mathrm{mod}\ 3)$ to $x_u - x_0 \equiv c \,(\mathrm{mod}\ 3)$, and each inequation $x_u \not\equiv c \,(\mathrm{mod}\ 3)$ to $x_u - x_0 \not\equiv c \,(\mathrm{mod}\ 3)$. For any solution x to this instance of Max 2-Lin-Mod-3 we can obtain a solution x' to the original instance of the same value by setting $x'_u \equiv x_u - x_0 \,(\mathrm{mod}\ 3)$.

To model Max 2-Lin-Mod-3, we introduce a variable y_u for each variable x_u. Each variable y_u is constrained to belong to R_3. For any set of variables $x_u \in \{0, 1, 2\}$, we set $y_u = \omega^{x_u}$. For $y_u, y_v \in R_3$, observe that $\langle y_u, y_v \rangle$ can take only three values, namely the values in R_3. Furthermore, the

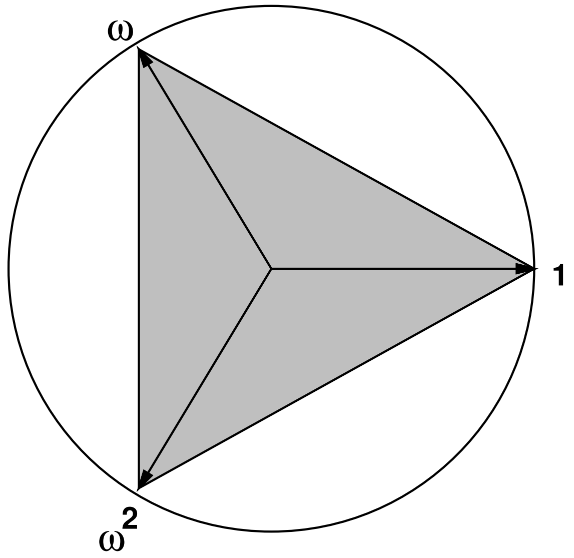

Figure 1: View of R_3 in the complex plane. The constraints force $\langle y_u, y_v\rangle$ to lie in the shaded region.

equation $x_u - x_v \equiv c \,(\mathrm{mod}\ 3)$ is satisfied if and only if $\langle y_u, y_v\rangle = \omega^c$, and the inequation is satisfied if and only if $\langle y_u, y_v\rangle \in R_3 \setminus \{\omega^c\}$. Thus the contribution of the jth equation $x_u - x_v \equiv c \,(\mathrm{mod}\ 3)$ to the objective function is

$$w_j \left(\frac{1}{3} + \frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle + \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle \right), \tag{4}$$

and the contribution of the jth inequation $x_u - x_v \not\equiv c \,(\mathrm{mod}\ 3)$ to the objective function is

$$w_j \left(\frac{2}{3} - \frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle - \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle \right). \tag{5}$$

Instead of imposing that $y_u \in R_3$ for every u, we relax it to:

- $\langle y_u, y_u\rangle = 1$, and
- $\alpha\langle y_u, y_v\rangle + \bar{\alpha}\langle y_v, y_u\rangle \geq -1$ for every u, v and every $\alpha \in R_3$.

The last constraint says that $\langle y_u, y_v\rangle$ belongs to the triangle with vertices R_3 in the complex plane. See Figure 1.

The relaxation we have obtained can now be formulated as a complex semidefinite program. Letting $Y = [\langle y_u, y_v\rangle] \in H_n$, we have the constraints

$$\langle E_{uu}, Y\rangle = 1 \tag{6}$$

for all u, and the constraints

$$\langle A^c_{uv}, Y\rangle \geq -1 \tag{7}$$

for every u, v and $c = 0, 1, 2$, where e_u is the vector with zeroes everywhere except for a 1 in the uth component, $E_{uu} = e_u e_u^T$, and $A^c_{uv} = \omega^c e_u e_v^T + \omega^{-c} e_v e_u^T$. Similarly, the objective function can be formulated as

$$\langle C, Y\rangle \tag{8}$$

where $C \in H_n$, by appropriately combining (4) and (5) for each equation/inequation. Our discussion implies that if we solve (8) subject to (6), (7) and $Y \succeq 0$, $Y \in H_n$, we obtain an upper bound on the optimum objective function value of Max 2-Lin-Mod-3.

We would like to mention that if Y is feasible for the above complex semidefinite programming relaxation then Y^T is also feasible. This is because ω and ω^2 are both cube roots of unity. However, in general, the objective function value of Y and Y^T are not equal. They will be equal in the special case in which every equation is of the form $x_i - x_j \equiv 0 \pmod 3$ and every inequation of the form $x_i - x_j \not\equiv 0 \pmod 3$. This includes the case of Max-3-Cut. If the objective function value is unchanged by replacing Y by Y^T, it is also unchanged by replacing it with $(Y + Y^T)/2$ which is a real matrix. This means that in this special case our complex semidefinite programming relaxation simply reduces to a (real) semidefinite programming relaxation. In the case of Max-3-Cut, the above argument shows that our relaxation is equivalent to the relaxation considered by Frieze and Jerrum [9]. In the algorithm and in our analysis, we will however not take advantage of the fact that the optimum solution can be assumed to be real in the case of Max-3-Cut.

5 The Algorithm and its Analysis

5.1 The algorithm

We can now describe the algorithm for Max 2-Lin-Mod-3. For a given problem instance, we solve the associated CSDP as described in the previous section in polynomial time up to any prescribed accuracy (see Section 7 for a discussion of complexity issues), and obtain vectors y_u for each u from the solution matrix Y in polynomial time by Cholesky decomposition (see also Section 7).

We then pick a random vector $p \in \mathbb{C}^n$ from the complex normal distribution. This is equivalent to drawing each entry p_k independently from the univariate complex normal distribution, which in turn is equivalent to drawing Re p_k and Im p_k independently from the univariate real normal distribution. The real normal distribution can be simulated using the uniform distribution on $[0, 1]$ (see Knuth [18, p. 117]).

We then consider the inner products $\langle p, y_u \rangle$ for each u. We create a solution x to the Max 2-Lin-Mod-3 instance by setting

$$x_u = \begin{cases} 0 & \text{if } \arg\langle p, y_u \rangle \in [0, 2\pi/3) \\ 1 & \text{if } \arg\langle p, y_u \rangle \in [2\pi/3, 4\pi/3) \\ 2 & \text{if } \arg\langle p, y_u \rangle \in [4\pi/3, 2\pi). \end{cases}$$

5.2 The analysis

We now turn to the analysis of the algorithm. As in [10], we first analyze the probability that a given equation $x_u - x_v \equiv c \pmod 3$ or inequation $x_u - x_v \not\equiv c \pmod 3$ is satisfied, solely given the inner product $\langle y_u, y_v \rangle$. We then compare this probability to the contribution made to the objective function for the given (in)equation, which also depends solely on the inner product $\langle y_u, y_v \rangle$. Since this inner product must belong to the triangle defined by R_3 in the complex plane (by the constraints of the CSDP), we find the minimum of the ratio of probability to objective function contribution over the triangle R_3. Finally, we use linearity of expectations and linearity of the objective function of our complex semidefinite program to derive the performance guarantee.

Throughout this paper, $\arccos(\cdot)$ has range in $[0, \pi]$.

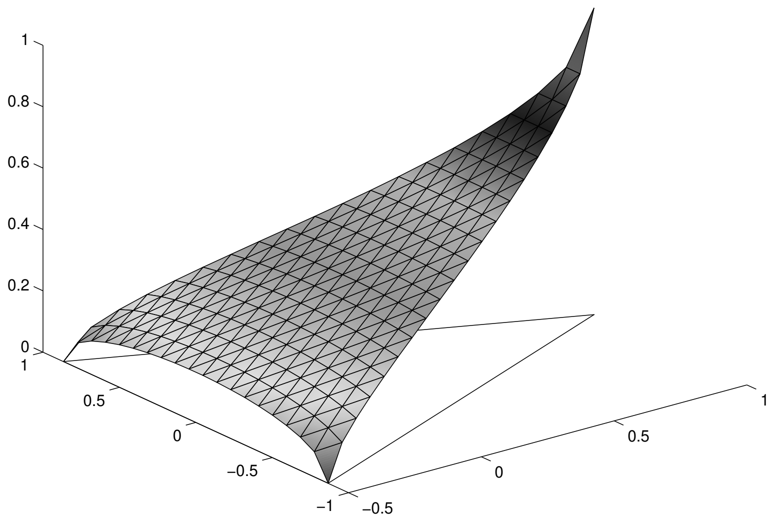

Figure 2: Probability of $x_u - x_v \equiv 0 \pmod 3$ being satisfied as a function of $\langle y_u, y_v \rangle$.

Theorem 1 *If $\langle y_u, y_v \rangle = re^{-i\theta}$ then the probability that the equation $x_u - x_v \equiv c$ (mod 3) is satisfied is*

$$
\begin{aligned}
P &= \frac{1}{3} + \frac{3}{8\pi^2}\left[2\arccos^2\left(-r\cos\left(\frac{2\pi c}{3}+\theta\right)\right)\right. \\
&\quad - \arccos^2\left(-r\cos\left(\frac{2\pi(c+1)}{3}+\theta\right)\right) \\
&\quad \left. - \arccos^2\left(-r\cos\left(\frac{2\pi(c-1)}{3}+\theta\right)\right)\right].
\end{aligned}
$$

For example, if $\langle y_u, y_v \rangle = 1$ ($r = 1$, $\theta = 0$) and $c = 0$, we have $P = \frac{1}{3} + \frac{3}{8\pi^2}\left(2\pi^2 - \frac{\pi^2}{9} - \frac{\pi^2}{9}\right) = 1$, while for $c = 1$, we have $P = \frac{1}{3} + \frac{3}{8\pi^2}\left(\frac{2\pi^2}{9} - \pi^2 - \frac{\pi^2}{9}\right) = 0$. If $\langle y_u, y_v \rangle = 0$, we have $P = \frac{1}{3} + \frac{3}{8\pi^2}\left(\frac{2\pi^2}{4} - \frac{\pi^2}{4} - \frac{\pi^2}{4}\right) = \frac{1}{3}$, as expected. Figure 2 shows the probability given in Theorem 1 over the entire range of values for $\langle y_u, y_v \rangle$ when $c = 0$.

By simply complementing the probability in Theorem 1, we get

Theorem 2 *If $\langle y_u, y_v \rangle = re^{-i\theta}$ then the probability that the equation $x_u - x_v \not\equiv c$ (mod 3) is satisfied is*

$$
\begin{aligned}
P &= \frac{2}{3} - \frac{3}{8\pi^2}\left[2\arccos^2\left(-r\cos\left(\frac{2\pi c}{3}+\theta\right)\right)\right. \\
&\quad - \arccos^2\left(-r\cos\left(\frac{2\pi(c+1)}{3}+\theta\right)\right) \\
&\quad \left. - \arccos^2\left(-r\cos\left(\frac{2\pi(c-1)}{3}+\theta\right)\right)\right].
\end{aligned}
$$

In order to be able to derive a performance guarantee for the algorithm, we need to compare the probabilities in Theorems 1 and 2 with the corresponding contribution to the objective function of the relaxation, namely to (4) and (5) respectively.

Theorem 3 *If $\langle y_u, y_v\rangle$ belongs to the triangle defined by R_3 then*

1. *the probability that $x_u - x_v \equiv c \pmod 3$ is satisfied is at least ρ times $\left(\frac{1}{3}+\frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle + \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right)$, where*
$$\rho = \min_{-1/2\le x\le 1} \frac{\frac{7}{24}+\frac{3}{4\pi^2}\arccos^2(-x) - \frac{3}{8\pi^2}\arccos^2(x-0.5)}{\frac{1}{3}+\frac{2}{3}x} > 0.793733.$$

2. *the probability that $x_u - x_v \not\equiv c \pmod 3$ is satisfied is at least $\phi = \frac{7}{12}+\frac{3}{4\pi^2}\arccos^2(-1/4) > 0.836008$ times $\left(\frac{2}{3}-\frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle - \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right)$.*

From this theorem, we can derive the claimed performance guarantee. Let OPT denote the optimum value of a Max 2-Lin-Mod-3 instance, and given a set of vectors $y_u \in \mathbb{C}^n$, let $Y = (\langle y_u, y_c\rangle)$. Let C be the objective function matrix for the instance as given in (8).

Theorem 4 *Given $y_u \in \mathbb{C}^n$ be a set of vectors such that $Y = (\langle y_u, y_v\rangle)$ is feasible for the constraints (6) and (7), and such that $\langle C, Y\rangle \ge (1-\epsilon)OPT$ for some $\epsilon > 0$, then the algorithm above is a randomized $(\rho-\epsilon)$-approximation algorithm for* Max 2-Lin-Mod-3*, and a randomized $(\phi-\epsilon)$-approximation algorithm for instances of* Max 2-Lin-Mod-3 *with inequations only.*

Proof: We let $E = \{(w,u,v,c) : x_u - x_v \equiv c \pmod 3 \text{ of weight } w\}$ be the set of equations of the Max 2-Lin-Mod-3 instance and $I = \{(w,u,v,c) : x_u - x_v \not\equiv c \pmod 3 \text{ of weight } w\}$ the set of the inequations, so that the objective function is

$$\langle C, Y\rangle = \sum_{(w,u,v,c)\in E} w\left(\frac{1}{3}+\frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle + \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right) + \sum_{(w,u,v,c)\in I} w\left(\frac{2}{3}-\frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle - \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right).$$

Thus if W is a random variable denoting the value of the solution produced by the randomized algorithm, we have that

$$\begin{aligned}
E[W] &= \sum_{(w,u,v,c)\in E} w\Pr[x_u - x_v \equiv c \pmod 3 \text{ satisfied}] + \sum_{(w,u,v,c)\in I} w\Pr[x_u - x_v \not\equiv c \pmod 3 \text{ satisfied}]\\
&\ge \rho \sum_{(w,u,v,c)\in E} w\left(\frac{1}{3}+\frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle + \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right)\\
&\quad + \phi \sum_{(w,u,v,c)\in I} w\left(\frac{2}{3}-\frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle - \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right)\\
&\ge \rho\langle C, Y\rangle \ge \rho(1-\epsilon)OPT \ge (\rho-\epsilon)OPT.
\end{aligned}$$

The inequalities above also imply that the performance guarantee is $\phi-\epsilon$ if $E = \emptyset$. △

In Section 7, we discuss the issue of how to obtain vectors y_u meeting the conditions of the Theorem above. In short, if Z^*_{CSDP} is the optimum value of the CSDP, we know that $Z^*_{CSDP} \ge OPT$, since the CSDP is a relaxation of the problem. Section 7 shows how to compute vectors whose objective function value is at least $(1-\epsilon)Z^*_{CSDP}$.

To prepare for the proof of Theorem 1, we first need a few simplications. Consider the equation $x_u - x_v \equiv c \pmod 3$, and the vectors y_u and y_v.

Claim 5 *The probability that $x_u - x_v \equiv c \pmod 3$ is satisfied depends only on $\langle y_u, y_v\rangle$. In particular, given $\langle y_u, y_v\rangle = re^{-i\theta}$, one can choose without loss of generality $y_u = e_1$ and $y_v = (re^{i\theta}, s, 0, \cdots, 0) \in \mathbb{C}^n$ with $r^2 + s^2 = 1$.*

Proof: Let $b_1, \cdots, b_n$ be an orthonormal basis of $\mathbb{C}^n$ such that $b_1 = y_u$ and $b_2 = c(y_v - \langle y_v, y_u\rangle y_u)$ with $c \in \mathbb{R}$ so that $b_2^* b_2 = 1$. Let $U^* \in \mathbb{C}^{n\times n}$ have b_i as ith column. Hence, $UU^* = I$ and thus $U^*U = I$ and U is unitary. Consider now the linear transformation that maps $x \in \mathbb{C}^n$ to Ux; this is a rotation. We first observe that $Uy_u = e_1$ and $Uy_v = (y_u^* y_v, cy_v^* y_v - c\overline{(y_u^* y_v)} y_u^* y_v, 0, \cdots, 0)$, with the second coefficient being real. Since U is unitary, for any $x, y \in \mathbb{C}^n$ we have $x^* y = (Ux)^*(Uy)$, and therefore any inner product is preserved after the linear transformation. Thus, $Uy_v = (re^{i\theta}, s, 0, \cdots, 0)$ where $r^2 + s^2 = 1$. Also, if we let $q = Up$ where p is distributed according to a complex normal distribution, we notice that (i) q is also distributed according to the same distribution (since $q^* q = p^* p$), (ii) $\langle p, y_u\rangle = \langle q, Uy_u\rangle$ and $\langle p, y_v\rangle = \langle q, Uy_v\rangle$. Therefore, if we consider Uy_u and Uy_v instead of y_u and y_v, we obtain exactly the same distribution for x_u and x_v, proving the claim. $\triangle$

By the claim, we can assume that $y_u = e_1$ and $y_v = (re^{i\theta}, s, 0, \cdots, 0)$. Now consider the inner products $\langle p, y_u\rangle$ and $\langle p, y_v\rangle$. We can thus restrict our attention to the first two coordinates of p; call them $p_1 e^{i\alpha_1}$ and $p_2 e^{i\alpha_2}$. Since these coordinates are drawn independently from the complex normal distribution, it is equivalent if we draw independently α_j uniformly from $[0, 2\pi)$ and p_j from the density function

$$f(r) = re^{-r^2/2}. \tag{9}$$

The probability that the equation $x_v - x_u \equiv c \,(\mathrm{mod}\ 3)$ is satisfied depends on $\arg\langle p, y_u\rangle$, and on $\arg\langle p, y_v\rangle$. The following lemma gives a precise dependence. In the lemma below, as in the rest of the paper, whenever we write $\Pr[\alpha \le \mu < \beta]$ we assume that $\alpha \le \beta$, that $\beta \le \alpha + 2\pi$, and that μ is determined up to multiples of 2π; that is, $\Pr[\alpha \le \mu < \beta] = \Pr[\alpha \le \mu + 2k\pi < \beta \text{ for some } k \in Z]$.

Lemma 6 *Let $\mu = \arg\langle p, y_v\rangle - \arg\langle p, y_u\rangle$. Then*

$$\Pr[x_v - x_u \equiv c \textit{ (mod 3) satisfied}] = \frac{3}{2\pi}\int_0^{2\pi/3} \Pr_\mu\left[\frac{2\pi c}{3} - \tau \le \mu < \frac{2\pi(c+1)}{3} - \tau\right] d\tau.$$

Proof: First note that $\langle p, y_u\rangle = p_1 e^{i\alpha_1}$, and thus $\arg\langle p, y_u\rangle = \alpha_1$, a uniformly distributed angle. Given μ, this implies that $\arg\langle p, y_v\rangle = \mu + \alpha_1$. If we add or subtract from $\arg\langle p, y_u\rangle$ and $\arg\langle p, y_v\rangle$ any multiple of $2\pi/3$, $x_v - x_u$ is unaffected. We can therefore replace α_1 by τ, where τ is uniformly distributed in $[0, 2\pi/3)$ and assume that $x_u = 0$.

Therefore,

$$\begin{aligned}\Pr[x_v - x_u \equiv c \,(\mathrm{mod}\ 3) \text{ satisfied}] &= \int_0^{2\pi/3} \Pr[x_v = c | \alpha_1 = \tau] \frac{3}{2\pi} d\tau \\ &= \frac{3}{2\pi}\int_0^{2\pi/3} \Pr_\mu\left[\frac{2\pi c}{3} \le \tau + \mu < \frac{2\pi(c+1)}{3}\right] d\tau.\end{aligned}$$

$\triangle$

We therefore need to study the distribution for $\mu = \arg\langle p, y_v\rangle - \arg\langle p, y_u\rangle$. Since $\langle p, y_u\rangle = p_1 e^{i\alpha_1}$ and $\langle p, y_v\rangle = p_1 r e^{i(\alpha_1 - \theta)} + p_2 s e^{i\alpha_2}$, we have that

$$\begin{aligned}\mu &= \arg(p_1 r e^{i(\alpha_1-\theta)} + p_2 s e^{i\alpha_2}) - \alpha_1 \\ &= \arg(p_1 r e^{-i\theta} + p_2 s e^{i(\alpha_2 - \alpha_1)}) \\ &= -\theta + \arg(p_1 r + p_2 s e^{i\alpha}),\end{aligned}$$

where $\alpha = \theta + \alpha_2 - \alpha_1$ is uniformly distributed and independent of p_1 and p_2. Note that we use crucially the fact that we are taking α modulo 2π. We define $\gamma = \arg(p_1 r + p_2 s e^{i\alpha})$ where p_1, p_2, α

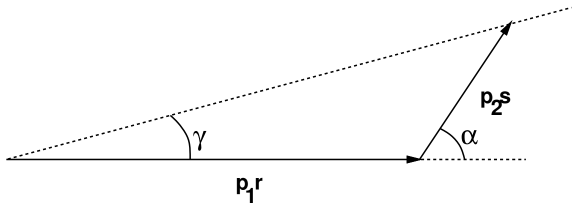

Figure 3: Angle γ.

are independent and p_1, p_2 are distributed according to the density (9) and α is uniform in $[0, 2\pi)$. Thus $\mu = \gamma - \theta$.

The next step in our analysis is to get a grasp of the distribution of γ. To do this, we will first need the following lemma.

Lemma 7 *For a given $\ell \geq 0$,*

$$\Pr[p_2 s \leq \ell p_1 r] = \frac{\ell^2 r^2}{(1-r^2) + \ell^2 r^2}.$$

Proof: Recall that p_1 and p_2 have density $f(r) = re^{-r^2/2}$. We have

$$\begin{aligned}
\Pr[p_2 s \leq \ell p_1 r] &= \int_0^\infty f(p_1) \int_0^{p_1 \ell r/s} f(p_2) dp_2 dp_1 \\
&= \int_0^\infty f(p_1) \cdot \left[-e^{-p_2^2/2}\right]_0^{p_1 \ell r/s} dp_1 \\
&= \int_0^\infty p_1 e^{-p_1^2/2} \left[1 - e^{-p_1^2 \ell^2 r^2/(2s^2)}\right] dp_1 \\
&= 1 - \int_0^\infty p_1 e^{-(p_1^2/2)(1+\ell^2 r^2/s^2)} dp_1 \\
&= 1 - \left[-\frac{1}{1+\ell^2 r^2/s^2} e^{-(p_1^2/2)(1+\ell^2 r^2/s^2)}\right]_0^\infty \\
&= 1 - \frac{1}{1+\ell^2 r^2/s^2} = \frac{\ell^2 r^2}{(1-r^2)+\ell^2 r^2},
\end{aligned}$$

where we have used the fact that $r^2 + s^2 = 1$. △

We can now prove the following lemma about the distribution of the angle γ.

Lemma 8 *For any $0 \leq \delta \leq 2\pi$, we have*

$$\Pr[0 \leq \gamma < \delta] = \frac{1}{2\pi}\left(\delta + \frac{r \sin\delta}{\sqrt{1 - r^2 \cos^2\delta}} \arccos(-r\cos\delta)\right).$$

Proof: We argue first for the case $\delta \leq \pi$, and obtain the general case later by symmetry. Consider Figure 3. Certainly if $\alpha \in (\pi, 2\pi)$ then $\gamma \notin [0, \delta]$, while if $\alpha \in [0, \delta]$, then $\gamma \leq \alpha$ and $\gamma \in [0, \delta]$. If $\alpha \in (\delta, \pi)$, we have $\gamma \leq \delta$ if and only if $p_2 s \leq p_1 r \frac{\sin\delta}{\sin(\alpha-\delta)}$, by the law of sines. Thus for $\delta \leq \pi$,

$$\Pr[0 \leq \gamma < \delta] = \frac{1}{2\pi}\left[\delta + \int_\delta^\pi \Pr\left[p_2 s \leq \frac{\sin\delta}{\sin(\alpha-\delta)} p_1 r\right] d\alpha\right],$$

where $\frac{\delta}{2\pi}$ is the probability that $\alpha \in [0,\delta]$ and the other term is the probability that $\gamma \in [0,\delta]$ given that $\alpha \in (\delta,\pi)$. For convenience, we change variables so that $\beta = \alpha - \delta$, and the probability becomes

$$\Pr[0 \leq \gamma < \delta] = \frac{1}{2\pi}\left[\delta + \int_0^{\pi-\delta} \Pr\left[p_2 s \leq \frac{\sin\delta}{\sin\beta} p_1 r\right] d\beta\right].$$

Then we get

$$\begin{aligned}
\Pr[0 \leq \gamma < \delta] &= \frac{1}{2\pi}\left[\delta + \int_0^{\pi-\delta} \frac{r^2\sin^2\delta}{(1-r^2)\sin^2\beta + r^2\sin^2\delta} d\beta\right] && (10)\\
&= \frac{1}{2\pi}\left(\delta + \left[\frac{r\sin\delta}{\sqrt{1-r^2\cos^2\delta}}\arctan\left(\frac{\tan\beta\sqrt{1-r^2\cos^2\delta}}{r\sin\delta}\right)\right]_{\beta=0}^{\beta=\pi-\delta}\right) && (11)\\
&= \frac{1}{2\pi}\left(\delta + \frac{r\sin\delta}{\sqrt{1-r^2\cos^2\delta}}\arctan\left(-\frac{\sqrt{1-r^2\cos^2\delta}}{r\cos\delta}\right)\right)\\
&= \frac{1}{2\pi}\left(\delta + \frac{r\sin\delta}{\sqrt{1-r^2\cos^2\delta}}\arccos(-r\cos\delta)\right)
\end{aligned}$$

where (10) follows by applying Lemma 7 and (11) follows from [26, pg. 441, number 360].

To obtain the proof statement for all values of δ, we use symmetry. First observe that for $0 \leq \delta \leq \pi$, we have that $\Pr[0 \leq \gamma < \delta] = \Pr[2\pi - \delta < \gamma \leq 2\pi]$. Thus (using the fact that γ takes any specific angle value with probability 0):

$$\begin{aligned}
\Pr[0 \leq \gamma < 2\pi - \delta] &= 1 - \Pr[0 \leq \gamma < \delta]\\
&= 1 - \frac{1}{2\pi}\left(\delta + \frac{r\sin\delta}{\sqrt{1-r^2\cos^2\delta}}\arccos(-r\cos\delta)\right)\\
&= \frac{1}{2\pi}\left((2\pi-\delta) + \frac{r\sin(2\pi-\delta)}{\sqrt{1-r^2\cos^2(2\pi-\delta)}}\arccos(-r\cos(2\pi-\delta))\right).
\end{aligned}$$

This is the same expression as in the statement of the lemma with δ replaced by $2\pi - \delta$. We have thus shown the lemma for all values of δ. △

For the proof of Theorem 1, it is useful to be able to compute $\Pr[\delta_1 \leq \gamma < \delta_2]$ for any δ_1, δ_2 even if one of them is outside the range $[0, 2\pi)$. For this purpose, we have the following corollary of Lemma 8.

Corollary 9 *For any δ, let*

$$g(r,\delta) = \frac{1}{2\pi}\left(\delta + \frac{r\sin\delta}{\sqrt{1-r^2\cos^2\delta}}\arccos(-r\cos\delta)\right).$$

Then, for any δ_1, δ_2 with $0 \leq \delta_2 - \delta_1 \leq 2\pi$, we have that

$$\Pr[\delta_1 \leq \gamma < \delta_2] = g(r,\delta_2) - g(r,\delta_1).$$

Proof: The corollary follows from the fact that (i) $g(r, 2\pi+\delta) = 1 + g(r,\delta)$ and (ii) for $0 \leq \delta_1 \leq \delta_2 \leq 2\pi$, we have that $\Pr[\delta_1 \leq \gamma < \delta_2] = \Pr[0 \leq \gamma < \delta_2] - \Pr[0 \leq \gamma < \delta_1]$. △

We can now prove Theorem 1.

Proof of Theorem 1: We first observe that

$$g(r,\delta)=\frac{1}{2\pi}\left(\delta-\frac{1}{2}\frac{d}{d\delta}\{\arccos^2(-r\cos\delta)\}\right).$$

By Lemma 6 and the fact that $\mu=\gamma-\theta$, we have that

$$\begin{aligned}
&\Pr[x_v-x_u\equiv c\,(\mathrm{mod}\ 3)\ \text{satisfied}]\\
&= \frac{3}{2\pi}\int_0^{2\pi/3}\Pr_\gamma\left[\frac{2\pi c}{3}+\theta-\tau\le\gamma<\frac{2\pi(c+1)}{3}+\theta-\tau\right]d\tau\\
&= \frac{3}{2\pi}\int_0^{2\pi/3}\left(g\left(r,\frac{2\pi(c+1)}{3}+\theta-\tau\right)-g\left(r,\frac{2\pi c}{3}+\theta-\tau\right)\right)d\tau \qquad (12)\\
&= \frac{3}{2\pi}\int_{2\pi c/3+\theta}^{2\pi(c+1)/3+\theta}g(r,\nu)d\nu-\frac{3}{2\pi}\int_{2\pi(c-1)/3+\theta}^{2\pi c/3+\theta}g(r,\nu)d\nu \qquad (13)\\
&= \frac{3}{2\pi}\frac{1}{2\pi}\left(\int_{2\pi c/3+\theta}^{2\pi(c+1)/3+\theta}\nu d\nu-\left[\frac{1}{2}\arccos^2(-r\cos\nu)\right]_{2\pi c/3+\theta}^{2\pi(c+1)/3+\theta}\right.\\
&\qquad\left.-\int_{2\pi(c-1)/3+\theta}^{2\pi c/3+\theta}\nu d\nu+\left[\frac{1}{2}\arccos^2(-r\cos\nu)\right]_{2\pi(c-1)/3+\theta}^{2\pi c/3+\theta}\right) \qquad (14)\\
&= \frac{3}{8\pi^2}\left[\left(\frac{2\pi(c+1)}{3}+\theta\right)^2+\left(\frac{2\pi(c-1)}{3}+\theta\right)^2-2\left(\frac{2\pi c}{3}+\theta\right)^2\right]\\
&\qquad+\frac{3}{8\pi^2}\left[2\arccos^2\left(-r\cos\left(\frac{2\pi c}{3}+\theta\right)\right)\right.\\
&\qquad\left.-\arccos^2\left(-r\cos\left(\frac{2\pi(c+1)}{3}+\theta\right)\right)-\arccos^2\left(-r\cos\left(\frac{2\pi(c-1)}{3}+\theta\right)\right)\right]\\
&= \frac{1}{3}+\frac{3}{8\pi^2}\left[2\arccos^2\left(-r\cos\left(\frac{2\pi c}{3}+\theta\right)\right)\right.\\
&\qquad\left.-\arccos^2\left(-r\cos\left(\frac{2\pi(c+1)}{3}+\theta\right)\right)-\arccos^2\left(-r\cos\left(\frac{2\pi(c-1)}{3}+\theta\right)\right)\right],
\end{aligned}$$

where (12) follows from Corollary 9, (13) from a change of variable, and (14) from our initial observation. △

Finally, we turn to the proof of Theorem 3. We first need the following lemma.

Lemma 10 $\arccos^2(x)$ *is convex.*

Proof: We have $\frac{d}{dx}\arccos(x)=\frac{-1}{\sqrt{1-x^2}}$. Thus,

$$\frac{d}{dx}\arccos^2(x)=2\arccos(x)\frac{-1}{\sqrt{1-x^2}}.$$

Now,

$$\frac{d^2}{dx^2}\arccos^2(x)=2\left(\frac{1}{1-x^2}+\arccos(x)\frac{-x}{(1-x^2)^{3/2}}\right).$$

For $x\le 0$, the above is the sum of nonnegative terms and hence nonnegative. For $x>0$, we have that $\arccos(x)\le\frac{\sqrt{1-x^2}}{x}$ since this is equivalent to $\tan\alpha\ge\alpha$ (after substituting $\cos\alpha$ for x).

Therefore, for $x > 0$, we have

$$\frac{d^2}{dx^2}\arccos^2(x) \geq 2\left(\frac{1}{1-x^2} - \frac{x\sqrt{1-x^2}}{(1-x^2)^{3/2}x}\right) = 0.$$

Thus, the second derivative is always nonnegative, proving the lemma. $\triangle$

Proof of Theorem 3: We want to show that

$$\Pr[x_v - x_u \equiv c \,(\text{mod } 3) \text{ satisfied}] \geq \rho\left(\frac{1}{3} + \frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle + \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right),$$

where

$$\rho = \min_{-1/2\leq x\leq 1} \frac{\frac{7}{24} + \frac{3}{4\pi^2}\arccos^2(-x) - \frac{3}{8\pi^2}\arccos^2(x-0.5)}{\frac{1}{3}+\frac{2}{3}x} > 0.793733.$$

We also want to show that

$$\Pr[x_u - x_v \not\equiv c \,(\text{mod } 3) \text{ satisfied}] \geq \phi\left(\frac{2}{3} - \frac{1}{3}\omega^{-c}\langle y_u, y_v\rangle - \frac{1}{3}\omega^{c}\langle y_v, y_u\rangle\right),$$

where $\phi = \frac{7}{12} + \frac{3}{4\pi^2}\arccos^2(-1/4) > 0.836008$. Given $\langle y_u, y_v\rangle = re^{-i\theta} \in R_3$, by Theorem 1 in the case of equations we are looking to show

$$\rho \geq \min_{c,r,\theta} \frac{\frac{3}{8\pi^2}\left[2\arccos^2\left(-r\cos\left(\frac{2\pi c}{3}+\theta\right)\right) - \arccos^2\left(-r\cos\left(\frac{2\pi(c+1)}{3}+\theta\right)\right) - \arccos^2\left(-r\cos\left(\frac{2\pi(c-1)}{3}+\theta\right)\right)\right]}{\frac{1}{3}(1+e^{-2\pi ic/3-i\theta}+e^{2\pi ic/3+i\theta})}.$$

Note that since we are minimizing over θ, and since all expressions containing θ contain the term $\theta + 2\pi c/3$, we can without loss of generality assume that $c = 0$. We can similarly assume that $c = 0$ in the case of inequations.

Given $c = 0$, observe that (4) and (5) become $\frac{1}{3}+\frac{2}{3}r\cos\theta$ and $\frac{2}{3}-\frac{2}{3}r\cos\theta$, respectively. Suppose we fix $r\cos\theta$ to be t; thus, (4) or (5) are fixed as well. Note that since $\langle y_u, y_v\rangle$ is constrained to be in the triangle defined by R_3, and $t = \text{Re}\,\langle y_u, y_v\rangle$, it must be the case $-1/2 \leq t \leq 1$. Since for any angle θ, $\cos(\theta) + \cos\left(\theta + \frac{2\pi}{3}\right) + \cos\left(\theta + \frac{4\pi}{3}\right) = 0$, we have that $r\cos\left(\theta + \frac{2\pi}{3}\right) + r\cos\left(\theta + \frac{4\pi}{3}\right) = -t$. Now consider the probability given in Theorem 1 or Theorem 2 for a fixed value of t and as a function of $x = r\cos\left(\theta + \frac{2\pi}{3}\right)$ (see Figure 4 for an visulization of x and t). For Theorem 1 we get

$$P = \frac{1}{3} + \frac{3}{4\pi^2}\arccos^2(-t) - \frac{3}{8\pi^2}\arccos^2(-x) - \frac{3}{8\pi^2}\arccos^2(x+t), \tag{15}$$

while for Theorem 2, we get

$$P = \frac{2}{3} - \frac{3}{4\pi^2}\arccos^2(-t) + \frac{3}{8\pi^2}\arccos^2(-x) + \frac{3}{8\pi^2}\arccos^2(x+t). \tag{16}$$

We now consider the case of equations and inequations separately. In the case of inequations, both terms $\frac{3}{8\pi^2}\arccos^2(-x)$ and $\frac{3}{8\pi^2}\arccos^2(x+t)$ are convex in x by Lemma 10, and P is symmetric in x around $-\frac{t}{2}$. Thus P is minimized when $x = -\frac{t}{2}$. We still need to compare

$$P = \frac{2}{3} - \frac{3}{4\pi^2}\arccos^2(-t) + \frac{3}{4\pi^2}\arccos^2\left(\frac{t}{2}\right) \tag{17}$$

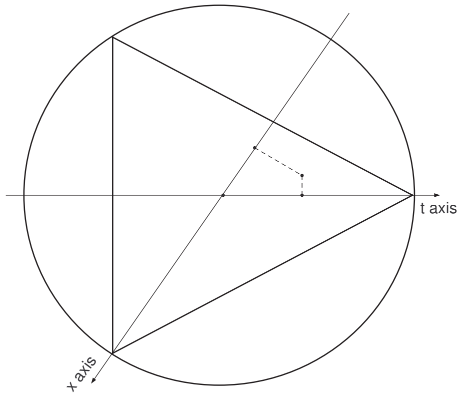

Figure 4: Visualization of x and t for a given point in R_3. Note that the value of t is its value on the real axis; x is its value on an axis tilted by $2\pi/3$.

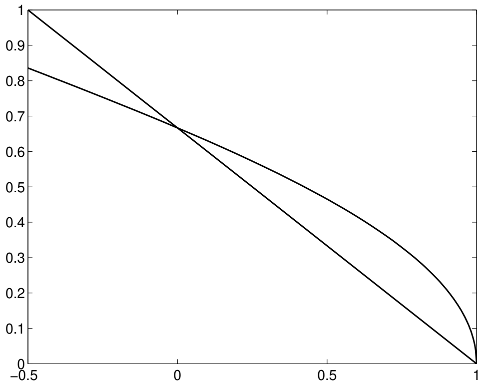

Figure 5: (17) vs. $\frac{2}{3} - \frac{2}{3}t$.

to $\frac{2}{3} - \frac{2}{3}t$ when $-\frac{1}{2} \leq t \leq 1$. We show below in Lemma 11 that the above function P is concave in t (see Figure 5). As a result, the minimum ratio is attained at one of the endpoints $t = -0.5$ or $t = 1$. Checking both, we derive that the worst-case bound is $\phi = \frac{7}{12} + \frac{3}{4\pi^2} \arccos^2(-1/4)$, attained for $t = -0.5$.

We now consider the case of equations. Here the argument is just the opposite. For fixed t, P (as given by (15)) is concave in x and symmetric with respect to $x = -\frac{t}{2}$. As a result, it is minimized for the maximum or minimum possible value for x, and this in turn is achieved when $\alpha e^{-i\theta} + \bar{\alpha} e^{i\theta} = -1$ for $\alpha = \omega$ or $\alpha = \omega^2$, i.e. $e^{-i\theta}$ is either on the top or bottom side of the triangle defined by R_3. Therefore, we either have that $x = r\cos\left(\theta + \frac{2\pi}{3}\right) = -0.5$ or $-x - t = r\cos\left(\theta + \frac{4\pi}{3}\right) = -0.5$. By symmetry we can assume that $x = -0.5$. Therefore, we only need to compare

$$\begin{aligned} P &= \frac{1}{3} + \frac{3}{4\pi^2} \arccos^2(-t) - \frac{3}{8\pi^2} \arccos^2(0.5) - \frac{3}{8\pi^2} \arccos^2(t - 0.5) \\ &= \frac{7}{24} + \frac{3}{4\pi^2} \arccos^2(-t) - \frac{3}{8\pi^2} \arccos^2(t - 0.5) \qquad (18) \end{aligned}$$

with $\frac{1}{3} + \frac{2}{3}t$ over the range of t, which is $-1/2 \leq t \leq 1$. The minimum ratio of these two functions is ρ by the definition of ρ. $\triangle$

Lemma 11 *The function* $f(t) = \frac{2}{3} - \frac{3}{4\pi^2} \arccos^2(-t) + \frac{3}{4\pi^2} \arccos^2(\frac{t}{2})$ *is concave for* $-1/2 \leq t \leq 1$.

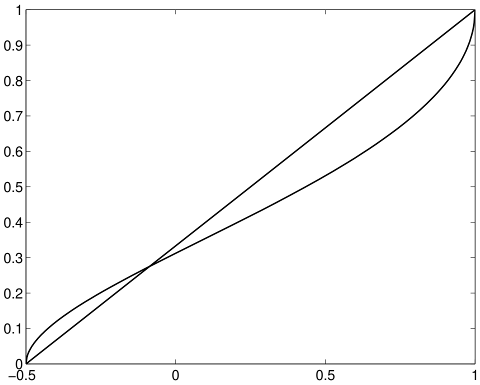

Figure 6: (18) vs. $\frac{1}{3} + \frac{2}{3}t$.

Proof: We have that

$$f'(t) = -\frac{3}{2\pi^2}\left(\arccos(-t)\frac{1}{\sqrt{1-t^2}}\right) + \frac{3}{4\pi^2}\left(\arccos(\frac{t}{2})\frac{-1}{\sqrt{1-(t/2)^2}}\right)$$

and that

$$\begin{aligned} f''(t) &= -\frac{3}{2\pi^2}\left(\frac{1}{1-t^2} + \arccos(-t)\frac{t}{(1-t^2)^{3/2}}\right) + \frac{3}{4\pi^2}\left(\frac{1/2}{1-(t/2)^2} - \arccos(t/2)\frac{t/4}{(1-(t/2)^2)^{3/2}}\right) \\ &= -\frac{3}{2\pi^2}\frac{1}{1-t^2}\left(1 + \arccos(-t)\frac{t}{\sqrt{1-t^2}}\right) + \frac{3}{8\pi^2}\frac{1}{1-(t/2)^2}\left(1 - \arccos(t/2)\frac{t/2}{\sqrt{1-(t/2)^2}}\right). \end{aligned}$$

This can be expressed as

$$f''(t) = \frac{3}{2\pi^2}\left(\frac{1}{4}g\left(\frac{t}{2}\right) - g(-t)\right), \tag{19}$$

where

$$g(x) = \frac{1}{1-x^2}\left(1 - \arccos(x)\frac{x}{\sqrt{1-x^2}}\right).$$

We claim that $g(x)$ is decreasing for $-1 \le x \le 1$, i.e. that for $-1 \le x \le 1$,

$$g'(x) = \frac{3x}{(1-x^2)^2} - \arccos(x)\frac{1+2x^2}{(1-x^2)^{2.5}} \le 0.$$

This is equivalent to saying that $h(x) := \frac{3x\sqrt{1-x^2}}{1+2x^2} \le \arccos(x)$. This follows from the fact that (i) $h(1) = 0 = \arccos(1)$ and that (ii) $h'(x) = \frac{3(1-4x^2)}{\sqrt{1-x^2}(1+2x^2)^2} \ge \frac{-1}{\sqrt{1-x^2}}$ for $-1 \le x \le 1$ since the latter is equivalent to $1 - 2x^2 + x^4 \ge 0$.

Equation (19) and the fact that $g(x)$ is decreasing imply that $f''(t)$ is also decreasing. Hence, for $-0.5 \le t \le 1$, we have that $f''(t) \le f''(-0.5) \approx -0.0205 < 0$, implying the concavity of $f(t)$ over $-0.5 \le t \le 1$. △

We end this section by observing that the proof of Theorem 3 allows us to characterize the graphs G for which our Max-3-Cut algorithm (and thus that of Frieze and Jerrum [9]) achieves its worst case. Note that if there is a feasible solution y_u such that for each edge $(u, v) \in E$, $\langle y_u, y_v \rangle = -1/2$, then the objective function (5) in this case will be $\sum_{(u,v)\in E} w_{uv}$, and this solution is optimal. The proof of Theorem 3 shows that for this solution, the expected value of the solution produced is precisely $\phi \sum_{(u,v)\in E} w_{uv}$. Graphs G for which this solution y is feasible are said to have a *strict vector 3-coloring* (Karger, Motwani, and Sudan [17]), which Karger et al. show is equivalent to $\vartheta(\bar{G}) = 3$ (see Lovász [20] for a definition of the ϑ function). This class of graphs includes all 3-colorable graphs that have a triangle. Observe that for the 3-colorable graphs, the optimum value of Max-3-Cut is the sum of the weight of all edges; in this case $OPT = Z^*_{CSDP} = \sum_{(u,v)\in E} w_{uv}$, but the expected value of the solution is ϕOPT. A similar observation about the worst-case performance of the algorithm of Frieze and Jerrum [9] was previously made by de Klerk et al. [6].

6 Equivalence of the relaxations and algorithms

In this section, we show that the relaxations and algorithms of Frieze and Jerrum [9] and Andersson et al. [3] are equivalent to ours in the case of Max-3-Cut. Independently, de Klerk et al. [6] have shown that their algorithm has the same worst-case performance guarantee as Frieze and Jerrum's algorithm [9] (although their relaxation is quite different). Furthermore we show that our relaxation is equivalent to that of Andersson et al. for the more general Max 2-Lin-Mod-3 problem; in addition, their algorithm and ours are also equivalent.

Some of the work is already done. We have shown the equivalence of our relaxation and the Frieze-Jerrum relaxation for Max-3-Cut in Section 4. Andersson et al. show that their relaxation is equivalent to the Frieze-Jerrum relaxation for Max-3-Cut.

We begin by showing that our relaxation is equivalent to that of Andersson et al. for Max 2-Lin-Mod-3. This equivalence implies that our relaxation and those of Frieze and Jerrum and Andersson et al. relaxations are equivalent for Max-3-Cut. To obtain a relaxation of Max 2-Lin-Mod-3, Andersson et al. [3] introduce three vectors $a^u_j \in \mathbb{R}^{3n}$ ($j = 0, 1, 2$) for each variable x_u. They impose the following constraints:

$$\langle a^u_j, a^u_j \rangle = 1 \text{ for all } u, j, \tag{20}$$

$$\langle a^u_j, a^u_{j'} \rangle = -\frac{1}{2} \text{ for all } u, j \neq j', \tag{21}$$

$$\langle a^u_j, a^v_{j'} \rangle \geq -\frac{1}{2} \text{ for all } u \neq v, j, j', \tag{22}$$

$$\langle a^u_j, a^v_{j+k} \rangle = \langle a^u_{j'}, a^v_{j'+k} \rangle \text{ for all } u, v, j, j', k, \tag{23}$$

where indices are taken modulo 3.

The objective function for a constraint $x_u - x_v \equiv c \pmod 3$ is then

$$\frac{1}{3} + \frac{2}{9}\sum_{j=0}^{2} \langle a^u_j, a^v_{j+c} \rangle, \tag{24}$$

and the complement to 1 for a constraint $x_u - x_v \not\equiv c \pmod 3$. We claim that the corresponding relaxation is exactly equivalent to our complex semidefinite program. This is shown below.

First, let us consider any set of complex vectors $y_u \in \mathbb{C}^n$. We show how to construct vectors $a^u_j \in \mathbb{R}^{2n}$ such that the vectors a have the same objective function contribution as the vectors y do. Given a vector $z \in \mathbb{C}^n$, let $f(z) \in \mathbb{R}^{2n}$ be $(\text{Re } z, \text{Im } z)$. Observe that

$$\langle f(z), f(z) \rangle = \langle z, z \rangle \tag{25}$$

and

$$\langle f(z), f(z')\rangle = \frac{1}{2}(\langle z, z'\rangle + \langle z', z\rangle). \tag{26}$$

Now define $a^u_j = f(\omega^j y_u)$ for every u and $j \in \{0,1,2\}$. We claim that (20)–(23) are all satisfied. (20) follows from (25) since

$$\langle f(\omega^j y_u), f(\omega^j y_u)\rangle = \langle \omega^j y_u, \omega^j y_u\rangle = \langle y_u, y_u\rangle = 1.$$

(21) follows from (26) since

$$\begin{aligned}\langle a^u_j, a^u_{j'}\rangle &= \frac{1}{2}(\langle \omega^j y_u, \omega^{j'} y_u\rangle + \langle \omega^{j'} y_u, \omega^j y_u\rangle) \\ &= \frac{1}{2}(\omega^{j-j'} + \omega^{j'-j})\langle y_u, y_u\rangle = -\frac{1}{2},\end{aligned}$$

for $j \neq j'$. (22) follows from

$$\begin{aligned}\langle a^u_j, a^v_{j'}\rangle &= \frac{1}{2}\left(\langle \omega^j y_u, \omega^{j'} y_v\rangle + \langle \omega^{j'} y_v, \omega^j y_u\rangle\right) \\ &= \frac{1}{2}\left(\omega^{j-j'}\langle y_u, y_v\rangle + \omega^{j'-j}\langle y_v, y_u\rangle\right) \geq -\frac{1}{2},\end{aligned}$$

because of the constraints $\alpha\langle y_u, y_v\rangle + \bar{\alpha}\langle y_v, y_u\rangle \geq -1$ for every u, v and every $\alpha \in R_3$. Finally, (23) follows from the fact that

$$\langle a^u_j, a^v_{j+k}\rangle = \frac{1}{2}\left(\omega^{-k}\langle y_u, y_v\rangle + \omega^k\langle y_v, y_u\rangle\right), \tag{27}$$

is independent of j. Furthermore, the objective function contribution (24) is

$$\frac{1}{3} + \frac{2}{9}\sum_{j=0}^{2}\langle a^u_j, a^v_{j+c}\rangle = \frac{1}{3} + \frac{1}{3}\left(\omega^{-c}\langle y_u, y_v\rangle + \omega^c\langle y_v, y_u\rangle\right),$$

by (27), and thus equal to (4).

On the other hand, consider any feasible solution to (20)–(23). Observe that (20) and (21) imply that, for every u, we have that $a^u_0 + a^u_1 + a^u_2 = 0$ and thus, for every u and v, that

$$\sum_{j=0}^{2}\sum_{j'=0}^{2}\langle a^u_j, a^v_{j'}\rangle = 0. \tag{28}$$

For every u, let us define

$$y_u = \frac{\sqrt{2}}{3}\sum_{j=0}^{2}\omega^{-j}a^u_j. \tag{29}$$

We claim that y_u is feasible for our complex semidefinite program and that its objective function value (4) is equal to (24). First, $\langle y_u, y_u\rangle = 1$ easily follows from (20) and (21). Secondly, for $\alpha = \omega^{-t}$, we have:

$$\alpha\langle y_u, y_v\rangle + \bar{\alpha}\langle y_v, y_u\rangle = \frac{4}{9}\sum_{j=0}^{2}\sum_{j'=0}^{2}\text{Re}\left(\alpha\langle \omega^{-j}a^u_j, \omega^{-j'}a^v_{j'}\rangle\right)$$

$$\begin{aligned}
&= \frac{4}{9}\sum_{j=0}^{2}\sum_{j'=0}^{2} \text{Re }\left(\omega^{-t+j'-j}\right)\langle a_j^u, a_{j'}^v\rangle \\
&= \frac{4}{9}\sum_{j=0}^{2}\sum_{j'=0}^{2}\left(\frac{1}{2} + \text{Re }\left(\omega^{-t+j'-j}\right)\right)\langle a_j^u, a_{j'}^v\rangle \\
&= \frac{2}{3}\sum_{j=0}^{2}\langle a_j^u, a_{j+t}^v\rangle \qquad (30) \\
&\geq -1,
\end{aligned}$$

where we have used (28) (for the third equality) and (22) (for the last inequality). Finally, using (30) with $t = c$, the term (4) of the objective function becomes:

$$\frac{1}{3} + \frac{2}{9}\sum_{j=0}^{2}\langle a_j^u, a_{j+c}^v\rangle,$$

which is precisely (24). This completes the equivalence between our complex semidefinite programming relaxation and the Andersson et al. [3] semidefinite programming relaxation.

Our algorithm and the one of Andersson et al. are also equivalent for Max 2-Lin-Mod-3. Their rounding scheme works as follows. Given vectors a_j^u, they generate one normally distributed (real) random vector r and x_u is set to be $\arg\max_j \langle r, a_j^u\rangle$. In our random scheme, since the distribution for p is invariant under multiplication by $e^{i\alpha}$, we can equivalently set

$$x_u = \begin{cases} 0 & \text{if } \arg\langle p, y_u\rangle \in [0, \pi/3) \cup [5\pi/3, 2\pi) \\ 1 & \text{if } \arg\langle p, y_u\rangle \in [\pi/3, \pi) \\ 2 & \text{if } \arg\langle p, y_u\rangle \in [\pi, 5\pi/3). \end{cases}$$

This is equivalent to setting x_u to be $\arg\max_j \text{Re } \langle p, \omega^j y_u\rangle$. Using (26), this can also be written as $\arg\max_j \langle f(p), f(\omega^j y_u)\rangle$. In our proof of the equivalence between our relaxation and the one of Andersson et al., we have shown that $a_j^u = f(\omega^j y_u)$ is an optimum solution for the Andersson et al. relaxation provided y_u constitutes an optimum solution to our relaxation. Since $f(p)$ is normally distributed, this shows the equivalence between our rounding scheme and the Andersson et al. scheme.

We now show that the rounding schemes of Frieze and Jerrum, Andersson et al., and this paper are also equivalent in the case of Max-3-Cut. The technique we use is inspired by the paper of de Klerk et al. [6]. We have already established the equivalence between our scheme and the one of Andersson et al, but we will nevertheless rederive it since the technique is quite different.

For Max-3-Cut, let $Y = [\langle y_u, y_v\rangle] \in S_n$ (with $y_u \in \mathbb{R}^n$) be a real optimum solution to both our complex relaxation and Frieze and Jerrum's relaxation. From the construction earlier in this section, we know that the $3n$ vectors $a_j^u = f(\omega^j y_u)$ constitute an optimum solution to the Andersson et al. relaxation. Let $Z = [\langle a_j^u, a_{j'}^v\rangle]$ where we order the rows and columns so that the 3 indices corresponding to u are consecutive. One can show that $Z = Y \otimes A$, where $\otimes$ denotes the Kronecker product ($A \otimes B$ is a matrix in which the (i, j) block is $a_{ij}B$) and

$$A = \begin{pmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1 & -1/2 \\ -1/2 & -1/2 & 1 \end{pmatrix}.$$

The various rounding schemes start by deriving from the positive semidefinite matrix Y or Z a set of vectors through a Cholesky decomposition $Y, Z = V^T V$ or $Y = V^* V$. The next step is to generate one or several normally distributed random vectors and deduce how to set x_u from the inner products of this (these) random vector(s) with the vectors from the Cholesky decomposition. Although a matrix Y or Z can be decomposed in many ways as $V^T V$ or $V^* V$, the probability that an edge (u, v) is in the 3-cut is independent of the choice of V in all these rounding schemes. As in the paper of de Klerk et al., we will carefully choose the Cholesky decomposition to show the equivalence of the rounding schemes.

The rounding schemes work as follows. In Frieze and Jerrum, three normally distributed random vectors, r_0, r_1 and r_2, are generated independently, and x_u is set to $\arg\max_j \langle r_j, y_u\rangle$. As mentioned previously, in Andersson et al., one normally distributed random vector r is generated and x_u is set to $\arg\max_j \langle r, a_j^u\rangle$. We will now focus on the Andersson et al. rounding scheme and show that, by carefully choosing the Cholesky decomposition of Z, we can simulate the other two rounding schemes, and as a result, the probability that an edge (u, v) is in the 3-cut is identical in all three rounding schemes (provided we start with Y for our scheme and Frieze and Jerrum's scheme, and Z for Andersson et al.'s scheme).

Consider two different Cholesky decompositions of the matrix Z given in the previous paragraph. To give these decompositions, we will use the following results about Kronecker products: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ and $(A \otimes B)^T = A^T \otimes B^T$. Thus $Z = Y \otimes A = (V^T V) \otimes A = (V^T \otimes B^T)(V \otimes B) = (V \otimes B)^T (V \otimes B)$, provided that $A = B^T B$. Here are 2 such choices for B:

$$B_1 = \sqrt{\frac{3}{2}} \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix},$$

and

$$B_2 = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}.$$

Thus if we use $V \otimes B_1$ or $V \otimes B_2$, the probability that edge (u, v) is in the 3-cut is the same.

First consider $V \otimes B_2$. For the Andersson et al. rounding scheme, we generate a normally distributed vector $r \in \mathbb{R}^{2m}$ and evaluate $r^T (V \otimes B_2)$. We set x_u to be j if the uth entry of $r^T (V \otimes b_k)$, where $B_2 = (b_0, b_1, b_2)$, is maximum for $k = j$. This can be seen to be equivalent to generating a complex normally distributed vector $p = (r_1 + ir_2, r_3 + ir_4, \cdots, r_{2m-1} + ir_{2m})$ and setting x_u to be j if $j = \arg\max_j \mathrm{Re}\,(\omega^j \langle p, V_u\rangle)$ where V_u is the uth column of V. This is the same as our complex rounding scheme.

Finally, we claim that using $V \otimes B_1$, we get a scheme equivalent to the randomized scheme of Frieze and Jerrum when applied to V. Notice that if we replace B_1 by $\tilde{B}_1 = \gamma(B_1 + \alpha J)$ where J is the all ones matrix, we do not change the setting of x_u in the Andersson et al. random process since we only increase the inner product between the random vector and the 3 vectors corresponding to u in the same monotone way. But choosing $\alpha = \frac{1}{3}\sqrt{\frac{3}{2}}$ and $\gamma = \sqrt{\frac{2}{3}}$, we get $\tilde{B}_1 = I$. But, in that case, the 3 entries in $r^T (V \otimes I)$ corresponding to u are equal to $q_j^T V_u$ $(j = 0, 1, 2)$ where $q_j = (r_{j+1}, r_{j+4}, r_{j+7}, \cdots)$. This corresponds exactly to the Frieze and Jerrum rounding scheme. This therefore shows that in the case of Max-3-Cut, the three rounding schemes will give the same probability of an edge being in the 3-cut if we start with the related optimum solutions as considered here.

7 Complexity issues

In order to make the algorithm polynomial, several complexity issues need to be taken care of. This is fairly straightforward but tedious, and is sketched below.

In particular, we need to show that we can, in polynomial time, meet the conditions of Theorem 4; namely, for a fixed $\epsilon > 0$, we can find a set of vectors $y_u \in \mathbb{C}^n$ such that $Y = (\langle y_u, y_v \rangle)$ is feasible and the objective function value is at least $(1-\epsilon)Z^*_{CSDP} \geq (1-\epsilon)OPT$, where Z^*_{CSDP} is the CSDP optimum and OPT is the optimum value of the Max 2-Lin-Mod-3 instance. When considering the objective function $\langle C, Y\rangle$ of CSDP in this section, we will not include the constant terms of (4) and (5) in C, and, as a result, we will have to increase $\langle C, Y\rangle$ by a quantity between $\frac{1}{3}\sum_j w_j$ and $\frac{2}{3}\sum_j w_j$ to get the true objective value.

We begin by observing that I is a feasible solution to the CSDP of value $\geq \frac{1}{3}\sum_j w_j$, so that $Z^*_{CSDP} \geq \frac{1}{3}\sum_j w_j$. Therefore, it is sufficient to find a feasible set of vectors y_u such that the objective function value is at least $Z^*_{CSDP} - \frac{\epsilon}{3}\sum_j w_j$.

In order to apply a polynomial-time algorithm to solve the CSDP, we first need to modify the input data so that it is rational. We do so in following way. In both the objective function and in the constraints, we replace ω by $\omega' = -\frac{1}{2} + i\lambda$ where λ is rational and $|\lambda - \frac{\sqrt{3}}{2}| \leq \delta$, for some small δ to be fixed later. Similarly, we approximate ω^2 with $\overline{\omega'} = -\frac{1}{2} - i\lambda$. This implies that whenever we have two vectors a, b such that $|\langle a, b\rangle| \leq 1$, we have that

$$\left|(\omega\langle a,b\rangle + \overline{\omega}\langle b,a\rangle) - \left(\omega'\langle a,b\rangle + \overline{\omega'}\langle b,a\rangle\right)\right| = \left|(\omega-\omega')\langle a,b\rangle + \overline{(\omega-\omega')}\langle b,a\rangle\right| \leq 2\delta. \tag{31}$$

Let CSDP$'$ denote the resulting CSDP after rounding ω and ω^2, C' its objective function matrix, and $Z^*_{CSDP'}$ its optimum value. We show now that CSDP and CSDP$'$ have approximately the same optimum values and the same solutions.

Given any feasible solution Y' to CSDP$'$, let $Y = \frac{2\delta}{1+2\delta}I + \frac{1}{1+2\delta}Y'$. We claim that Y is feasible for CSDP and of value $\langle C, Y\rangle \geq \langle C', Y'\rangle - 2\delta\sum_j w_j$. For this, first note that Y is also Hermitian positive semidefinite (since it is the convex combination of two Hermitian positive semidefinite matrices), and $y_{uu} = 1$ for all u. Furthermore,

$$\begin{aligned}
\omega\langle y_u, y_v\rangle + \overline{\omega}\langle y_v, y_u\rangle &= \frac{1}{1+2\delta}\left(\omega\langle y'_u, y'_v\rangle + \overline{\omega}\langle y'_v, y'_u\rangle\right) \\
&\geq \frac{1}{1+2\delta}\left(\omega'\langle y'_u, y'_v\rangle + \overline{\omega'}\langle y'_v, y'_u\rangle - 2\delta\right) \geq -1,
\end{aligned}$$

where we have used (31) and the analogue to (7) for CSDP$'$, showing that Y also satisfies (7). The objective function value

$$\begin{aligned}
\langle C, Y\rangle &= \frac{1}{1+2\delta}\langle C, Y'\rangle \geq \frac{1}{1+2\delta}\left(\langle C', Y'\rangle - \frac{2\delta}{3}\sum_j w_j\right) \\
&\geq \langle C', Y'\rangle - \frac{2\delta}{1+2\delta}\langle C', Y'\rangle - \frac{2\delta}{3(1+2\delta)}\sum_j w_j \\
&\geq \langle C', Y'\rangle - \frac{2\delta}{1+2\delta}\sum_j w_j \geq \langle C', Y'\rangle - 2\delta\sum_j w_j,
\end{aligned}$$

where the first inequality follows from (31) and the third from the fact that that $\langle C', Y'\rangle \leq \frac{2}{3}\sum_j w_j$ because of the analogue to (7).

Similarly, if we have a solution Y for CSDP, we can construct a feasible solution $Y' = \frac{2\delta}{1+2\delta}I + \frac{1}{1+2\delta}Y$ for CSDP of value $\langle C', Y'\rangle \geq \langle C, Y\rangle - \frac{2\delta}{1+2\delta}\sum_j w_j \geq \langle C, Y\rangle - 2\delta\sum_j w_j$ by the same argument as above.

We can now solve this modified relaxation CSDP$'$ in polynomial-time; using for example a potential-reduction interior-point algorithm for symmetric cones [2], we can obtain a strictly feasible solution Y' of value $\langle C', Y' \rangle \geq Z^*_{CSDP'} - \delta \sum_j w_j$ in $O(n^2 \log(\sum_j w_j + n^2) + n \log \frac{1}{\delta})$ iterations each involving the solution of linear systems of polynomial size. Initializing the interior-point algorithm is easy since I is a strictly feasible solution to the primal and a strictly feasible solution to the dual with duality gap $O(\sum_j w_j + n^2)$ can also be easily found. The n^2 term in the duality gap comes from the fact that we have $O(n^2)$ inequalities (7); this n^2 also makes the rank of the corresponding Euclidean Jordan algebra $O(n^2)$.

From this solution Y', we can derive a solution $Y = \frac{2\delta}{1+2\delta} I + \frac{1}{1+2\delta} Y'$ to the original CSDP of value $\langle C, Y \rangle \geq \langle C', Y' \rangle - 2\delta \sum_j w_j \geq Z^*_{CSDP'} - 3\delta \sum_j w_j \geq Z^*_{CSDP} - 5\delta \sum_j w_j$. In fact, for the next step, it is useful to consider instead $Y = \frac{4\delta}{1+4\delta} I + \frac{1}{1+4\delta} Y'$ which is also feasible for CSDP. Its value satisfies $\langle C, Y \rangle \geq \langle C', Y' \rangle - 5\delta \sum_j w_j \geq Z^*_{CSDP'} - 6\delta \sum_j w_j \geq Z^*_{CSDP} - 8\delta \sum_j w_j$. The main advantage is that now (7) is satisfied strictly: $\langle A^c_{uv}, Y \rangle \geq -\frac{1+2\delta}{1+4\delta}$.

Finally, we decompose the matrix Y into WDW^* in polynomial-time via a Cholesky decomposition, where D is a positive (real) diagonal matrix. To obtain the vectors y_u we would need to consider row u of W after having multiplied column v by $\sqrt{d_{vv}}$ for every v. Instead we approximate $\sqrt{d_{vv}}$ by a rational s_v where $(s_v - \sqrt{d_{vv}})^2 \leq \left(\frac{\delta}{4}\right)^2 d_{vv}$, multiply column v of W by s_v, and denote the resulting row u by $\hat{y}_u$. Although $\hat{y}_u$ is not necessarily of unit norm, observe that our rounding step is unaffected if we perform it on $\hat{y}_u$ instead of on $\tilde{y}_u = \hat{y}_u / ||\hat{y}_u||$. We claim that $\tilde{y}_u$ is a feasible solution for CSDP of value only slightly weaker than Z^*_{CSDP}. Note that the definition of s_v implies that $||y_u - \hat{y}_u||^2 \leq \delta^2/16$. This in turn implies that $1 + \delta/4 \geq ||\hat{y}_u|| \geq 1 - \delta/4$ and thus $||\hat{y}_u - \tilde{y}_u||^2 \leq \delta^2/16$. By the triangle inequality, we now have that

$$||y_u - \tilde{y}_u||^2 \leq ||y_u - \hat{y}_u||^2 + ||\hat{y}_u - \tilde{y}_u||^2 \leq \delta^2/8. \tag{32}$$

On the other hand, using $|\langle a, b \rangle + \langle b, a \rangle| \leq 2||a||||b||$ for any a, b, we have that

$$\begin{aligned}
&|\alpha\langle y_u, y_v\rangle + \bar{\alpha}\langle y_v, y_u\rangle - \alpha\langle \tilde{y}_u, \tilde{y}_v\rangle - \bar{\alpha}\langle \tilde{y}_v, \tilde{y}_u\rangle| \\
&\quad\leq\ |\langle \alpha y_u, \tilde{y}_v - y_v\rangle + \langle \tilde{y}_v - y_v, \alpha y_u\rangle| + |\langle \alpha(\tilde{y}_u - y_u), y_v\rangle + \langle y_v, \alpha(\tilde{y}_u - y_u)\rangle| \\
&\qquad\quad + |\langle \alpha(\tilde{y}_u - y_u), \tilde{y}_v - y_v\rangle + \langle \tilde{y}_v - y_v, \alpha(\tilde{y}_u - y_u)\rangle| \\
&\quad\leq\ 2||y_u||||\tilde{y}_v - y_v|| + 2||y_v||||\tilde{y}_u - y_u|| + 2||\tilde{y}_u - y_u||||\tilde{y}_v - y_v|| \\
&\quad\leq\ \sqrt{2}\delta + \frac{\delta^2}{4},
\end{aligned}$$

for any $|\alpha| = 1$, the last inequality following from (32). This shows that $\tilde{Y}$ now satisfies (7) provided that $\frac{1+2\delta}{1+4\delta} + \sqrt{2}\delta + \frac{\delta^2}{4} \leq 1$, which is true for δ sufficiently small. Furthermore, $\langle C, \tilde{Y} \rangle \geq \langle C, Y \rangle - \frac{1}{3}(\sqrt{2}\delta + \frac{\delta^2}{4}) \sum_j w_j$, implying that $\langle C, \tilde{Y} \rangle \geq Z^*_{CSDP} - 10\delta \sum_j w_j$. Setting $\delta = \epsilon/30$ achieves our goal.

Finally, to implement our randomized algorithm, we also need to generate a vector according to an approximately complex normal distribution, or an approximately real normal distribution in twice the dimension. The real normal distribution can be simulated using the uniform distribution on $[0, 1]$, as indicated in Knuth [18, p. 117]. One can show that this can be done in such a way that the probability of an (in)equation to be satisfied does not decrease by more than ϵ.

8 Extensions

The work described in this paper leads to many natural questions. One is whether the k complex roots of unity could be used in a similar way to approximate the Max-k-Cut problem. However, a straightforward extension does not work. Indeed, if we model the k sides of the partition with the k roots of unity $e^{il\pi/k}$ for $l = 0, 1, \cdots, k-1$ then the contribution of an edge (u, v) to the objective must be 1 if $\langle y_u, y_v \rangle = e^{il\pi/k}$ for $l \in \{1, 2, \cdots, k-1\}$. Since the objective function must be concave (as we are maximizing) and, for $k \geq 4$, we can write 0 as a convex combination of $e^{il\pi/k}$ for $l \in \{1, 2, \cdots, k-1\}$, we must have that the objective function contribution is at least 1 if $\langle y_u, y_v \rangle = 0$. This implies that for $Y = I$, we have an objective function of the relaxation equal to the sum of the weights of all the edges (and for $Y = I$ our algorithm would give a random partition). This gives a ratio between the optimum value and the relaxation no better than $\frac{k-1}{k}$ for a complete graph (as its size grows arbitrarily).

Acknowledgements

We thank Greg Sorkin, whose facility with the symbolic math program Derive showed that one of the nastier integrals in this paper wasn't as bad as we had thought. We also thank Mike Todd and Farid Alizadeh for numerous pointers to the literature on complex SDP and other extensions.

Dedication

This paper is dedicated to the memory of Danny Lewin, a fellow MIT grad student and colleague at Akamai Technologies, who perished on American Airlines Flight 11 during the attacks of September 11, 2001. Danny was the cofounder and CTO of Akamai Technologies. While Danny was best known in the academic world for the paper on which Akamai was based [16], he had also worked at MIT on the random checking of polynomial identities [19], the hardness of approximation algorithms [15], and on the topic of this paper, the use of semidefinite programming in approximation algorithms (independently deriving part of the work of Zwick [29]). We will always remember Danny for his extreme brilliance, his unlimited energy and enthusiasm, his constant encouragement, his relentless determination, and his loud laughter. In addition to his business success, he was destined to have a great research career. He will be dearly missed.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[2] F. Alizadeh and S. Schmieta. Symmetric cones, potential reduction methods and word-by-word extensions. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, pages 195–233. Kluwer Academic Publishers, 2000.

[3] G. Andersson, L. Engebretsen, and J. Håstad. A new way to use semidefinite programming with applications to linear equations mod p. *Journal of Algorithms*, 39:162–204, 2001.

[4] B. Chor and M. Sudan. A geometric approach to betweenness. *SIAM Journal on Discrete Mathematics*, 11:511–523, 1998.

[5] E. de Klerk and D. V. Pasechnik. On the performance guarantee of MAX 3-CUT approximation algorithms. Manuscript, 2001.

[6] E. de Klerk, D. V. Pasechnik, and J. P. Warners. On approximate graph colouring and MAX-k-CUT algorithms based on the ϑ-function. Manuscript, Oct. 2000.

[7] L. Faybusovich. Linear systems in Jordan algebras and primal-dual interior point algorithms. *Journal of Computational and Applied Mathematics*, 86:149–175, 1997.

[8] U. Feige and M. Goemans. Approximating the value of two proper proof systems, with applications to MAX-2SAT and MAX-DICUT. In *Proceeding of the Third Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[9] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18:67–81, 1997.

[10] 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.

[11] M. X. Goemans and D. P. Williamson. Approximation algorithms for MAX 3-CUT and other problems via complex semidefinite programming. In *Proceedings of the 33rd ACM Symposium on the Theory of Computing*, 2001.

[12] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[13] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer Verlag, Berlin, 1988.

[14] O. Güler. Barrier functions in interior point methods. *Mathematics of Operations Research*, 21:860–885, 1996.

[15] V. Guruswami, D. Lewin, M. Sudan, and L. Trevisan. A tight characterization of NP with 3 query PCPs. In *Proceedings of the 39th Symposium on the Foundations of Computer Science*, pages 8–17, 1998.

[16] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In *Proceedings of the 29th ACM Symposium on the Theory of Computing*, pages 654–663, 1997.

[17] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45:246–265, 1998.

[18] D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming.* Addison-Wesley, Reading, MA, second edition, 1981.

[19] D. Lewin and S. Vadhan. Checking polynomial identities over any field: Towards a derandomization? In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 438–447, 1998.

[20] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, IT-25:1–7, 1979.

[21] S. Mahajan and H. Ramesh. Derandomizing approximation algorithms based on semidefinite programming. *SIAM Journal on Computing*, 28:1641–1663, 1999.

[22] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. Technical report, CORE, Universite Catholique de Louvain, Belgium, 1997.

[23] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[24] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[25] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8:324–364, 1998.

[26] S. M. Selby, editor. *CRC Standard Mathematical Tables.* The Chemical Rubber Company, Cleveland, OH, twenty-first edition, 1973.

[27] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *Journal of the ACM*, 48:206–242, 2001.

[28] K.-C. Toh and L. N. Trefethen. The Chebyshev polynomials of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 20:400–419, 1998.

[29] 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*, pages 201–210, 1998.

[30] 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 31st ACM Symposium on the Theory of Computing*, pages 679–687, 1999.