A New Way of Using Semidefinite Programming with Applications to Linear Equations mod p

Gunnar Andersson, Lars Engebretsen, and Johan Håstad

Royal Institute of Technology

SE-100 44 Stockholm

SWEDEN

February 18, 2002

Suggested running head: New Use of Semidefinite Programming

Corresponding author: Johan Håstad, johanh@nada.kth.se

Abstract

We introduce a new method to construct approximation algorithms for combinatorial optimization problems using semidefinite programming. It consists of expressing each combinatorial object in the original problem as a constellation of vectors in the semidefinite program. When we apply this technique to systems of linear equations mod p with at most two variables in each equation, we can show that the problem is approximable within $(1-\kappa(p))p$, where $\kappa(p)>0$ for all p. Using standard techniques, we also show that it is **NP**-hard to approximate the problem within a constant ratio, independent of p.

List of symbols used:

α	lower case Greek letter alpha.
δ	lower case Greek letter delta.
ε	lower case Greek letter epsilon.
κ	lower case Greek letter kappa.
λ	lower case Greek letter lambda.
μ	lower case Greek letter mu.
π	lower case Greek letter pi.
σ	lower case Greek letter sigma.
θ	lower case Greek letter theta.
Θ	upper case Greek letter Theta.
φ	lower case Greek letter phi.
Φ	upper case Greek letter Phi.
ζ	lower case Greek letter zeta.
ξ	lower case Greek letter xi.
ℓ	lower case Latin letter ell.
$\int$	integral sign.
$\sum$	summation sign.
$\boldsymbol{R}$	bold face upper case Roman letter ar (denotes the set of reals).
$\boldsymbol{Z}$	bold face upper case Roman letter zed (denotes the set of integers).
$\perp$	perpendicular sign.
$\leftarrow$	left arrow.
$\mapsto$	right arrow with small bar (denotes “maps to”).
$\Longrightarrow$	double right arrow (denotes implication).
$\Longleftrightarrow$	double left-right arrow (denotes equivalence).
$\emptyset$	the empty set.
$\cup$	set union.
$\cap$	set intersection.
$\in$	belongs to sign.
$\vert \cdot \vert$	absolute value.
$\vert\vert \cdot \vert\vert$	Euclidean norm.
$\langle \cdot, \cdot \rangle$	angular brackets (denotes inner product)
$\forall$	universal quantifier.
$\exists$	existential quantifier.
$\sqrt{\cdot}$	square root sign.
$\times$	multiplication sign.

1 Introduction

Several combinatorial maximization problems have the following property: The naive algorithm which simply chooses a solution at random from the solution space is guaranteed to give a solution of expected weight at least some constant times the weight of the optimal solution. For instance, applying the above randomized algorithm to Max Cut yields a solution with expected weight at least half the optimal weight. For a long time, better polynomial time approximation algorithms than the randomized ones were not known to exist for many of the problems with the above property. This situation changed when Goemans and Williamson [5] showed that it is possible to use semidefinite programming to efficiently find a solution to Max Cut which is only a factor approximately 1.14 worse than the optimal value. Extending the techniques of Goemans and Williamson, Frieze and Jerrum [4] showed that it is possible to construct also for Max k-Cut a polynomial time approximation algorithm better than the simple randomized one.

The problem of systems of linear equations mod p is a basic and very general combinatorial problem, which exhibits the property described above: The naive randomized algorithm which chooses a solution at random approximates the problem within p. Recently, Håstad [7] studied systems of linear equations mod p with exactly k unknowns in each equation, and showed that it is **NP**-hard to approximate the problem within $p - \varepsilon$ for all $\varepsilon > 0$, all $p \geq 2$, and all $k \geq 3$.

In this paper we study the remaining problem of this type, systems of linear equations mod p with at most two unknowns in each equation, denoted by Max 2-Lin mod p. We also study systems of linear equations mod p with exactly two unknowns in each equation, denoted by Max E2-Lin mod p. When $p = 2$, this problem has been studied previously, but for $p > 2$ not much is known. We use semidefinite programming combined with randomized rounding to show, that for both Max 2-Lin mod p and Max E2-Lin mod p it is possible to do better than the naive randomized heuristic. Specifically, we show that there exists, for all p, a randomized polynomial time algorithm which approximates both problems within $(1-\kappa(p))p$, where $\kappa(p) > 0$ for all p. On the negative side, we show that it is **NP**-hard to approximate Max E2-Lin mod p within some constant performance ratio, independent of p.

The usual way to use semidefinite programming in approximation algorithms is to formulate the problem as an integer program, and then relax this program to a semidefinite one. In order to approximate Max k-Cut, Frieze and Jerrum [4] instead associated a vector with each vertex, and added constraints enforcing the vectors to have certain properties. To refine their technique, we let each variable in the system of linear equations be represented by a constellation of several vectors. By adding suitably chosen constraints to the semidefinite program, we make sure that the solution to the semidefinite program has the same type of symmetries as the solution to the original problem.

Our approach is in some sense dual to the one of Frieze and Jerrum. We use many vectors to represent one variable and one random vector in the rounding; they use one vector for each variable and many random vectors in the rounding. Our algorithm can be used also for Max k-Cut, since Max k-Cut is a special case of Max E2-Lin mod k. It is not clear *a priori* how our method and the method of Frieze and Jerrum relate to each other. We elucidate on this and show, using local analysis, that the performance ratio of our algorithm cannot be better than the one of the algorithm of Frieze and Jerrum, and we have obtained numerical evidence that the algorithms actually achieve the same performance ratio.

2 Preliminaries

We start by defining our problems and our performance measure.

Definition 1. We denote by Max Ek-Lin mod p the problem of, given a system of m linear equations mod p with exactly k variables in each equation together with positive weights w_i, $i = 1, 2, \ldots m$, maximizing the total weight of satisfied equations.

Definition 2. We denote by MAX k-LIN MOD p the problem of, given a system of m linear equations mod p with at most k variables in each equation together with positive weights w_i, $i = 1, 2, \ldots m$, maximizing the total weight of satisfied equations.

Definition 3. Let P be a maximization problem. For an instance x of P let $\operatorname{opt}(x)$ be the optimal value. A C-approximation algorithm is an algorithm that on any input x outputs a value V such that $\operatorname{opt}(x) \geq V \geq \operatorname{opt}(x)/C$.

In our positive results, i.e., examples of algorithms achieving a certain approximation ratio, the algorithm in fact does more than required by the above definition. Apart from finding the number V it also finds an assignment which, on the average, satisfies equations of total weight V. We think of this assignment as the main output of the algorithm and hence concentrate on the description on how to find it using a randomized procedure. The numerical value V, which is found without randomness, is only a side effect.

In our negative results, i.e., proofs that no efficient approximation algorithm exists, we rule out even algorithms only giving the numerical approximation without producing a feasible solution.

From now on, p always denotes a prime, although all our results generalize to composite p. Regarding the lower bound, it is easy to see, that if p is a prime factor in m we can convert a MAX E2-LIN MOD p instance to an equivalent MAX E2-LIN MOD m instance by multiplying each equation with m/p. Since we show a constant lower bound, independent of p, the lower bound generalizes. We will show later how to generalize our upper bounds to composite p.

To get acquainted with the above definitions, we now show that a simple randomized heuristic, which can be derandomized by the method of conditional probabilities, for MAX 2-LIN MOD m has performance ratio m. Since an equation $ax_i - bx_{i'} = c \bmod m$ can only be satisfied if $\gcd(a, b, m)$ divides c, we can assume that all equations have this property.

Algorithm 1. Takes as its input an instance of MAX 2-LIN MOD m, $m = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$, with variables $x_1, \dots, x_n$. Outputs an assignment with expected weight at least a fraction $1/m$ of the weight of the satisfiable equations in the instance. The algorithm picks the values of x_i uniformly at random.

Since the behavior modulo different $p_s^{\alpha_s}$ is independent it is sufficient to establish the lemma below.

Lemma 1. *If we guess an assignment to the x_i mod $p_s^{\alpha_s}$ uniformly at random, an equation of the form $ax_i - bx_{i'} = c \bmod p_s^{\alpha_s}$ is satisfied with probability at least $1/p_s^{\alpha_s}$.*

Proof. If either a or b is a unit mod $p_s^{\alpha_s}$, the proof is trivial. Otherwise, $\gcd(a, b) = p_s^t$ for some $t \geq 1$, and in this case we can divide a, b and c by p_s^t to produce an equivalent equation

$$\frac{a}{p_s^t} x_i - \frac{b}{p_s^t} x_{i'} = \frac{c}{p_s^t} \bmod p_s^{\alpha_s - t}. \tag{1}$$

This equation will be satisfied with probability greater than $1/p_s^{\alpha_s}$. □

Corollary 2. *There exists, for all $m \geq 2$, a deterministic algorithm for* MAX 2-LIN MOD m *with performance ratio m.*

Proof. Algorithm 1 satisfies any satisfiable equation with probability at least $1/m$. With this in mind, the corollary follows from the fact that the optimum of an instance is at most the weight of the satisfiable equations, and that the algorithm can be derandomized by using the standard technique of conditional probabilities. In this method one determines the values of the variables one by one making sure that the expected number of satisfied equation, conditioned upon the choices made so far, never decreases [2, Chapter 15]. □

2.1 Earlier work

Goemans and Williamson [5] construct an approximation algorithm for Max Cut by studying a relaxation of an integer quadratic program. Frieze and Jerrum [4] extend the methods of Goemans and Williamson, and thereby construct an approximation algorithm for Max k-Cut. To obtain an intuitive understanding of our algorithms, it is instructive to study these particular algorithms.

For a graph $G = (V, E)$ with vertices $V = \{1, \dots, n\}$, we introduce for each vertex i in the graph a variable $y_i \in \{-1, 1\}$. If we denote by w_{ij} the weight of the edge (i, j), the weight of the maximum cut in the graph is given by the optimum of the integer quadratic program

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i<i'} w_{ii'} \frac{1 - y_i y_{i'}}{2} \\ \text{subject to} & y_i \in \{-1, 1\} \text{ for all } i. \end{array} \tag{2}$$

The partition (V_1, V_2) yielding the optimum cut can be formed as

$$V_1 = \{i : y_i = 1\}, \tag{3}$$

$$V_2 = \{i : y_i = -1\}. \tag{4}$$

It is **NP**-hard to solve (2) optimally. To obtain a polynomial time approximation algorithm, Goemans and Williamson [5] construct a relaxation of the above integer quadratic program by using n-dimensional real vectors v_i instead of the integer variables y_i. The products $y_i y_{i'}$ are then replaced by the inner products $\langle v_i, v_{i'} \rangle$. This gives the semidefinite program

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i<i'} w_{ii'} \frac{1 - \langle v_i, v_{i'} \rangle}{2} \\ \text{subject to} & \langle v_i, v_i \rangle = 1 \text{ for all } i. \end{array} \tag{5}$$

Given any $\varepsilon > 0$, this program can be solved within an additive error of ε in time polynomial in n, $\log \frac{1}{\varepsilon}$ and the number of bits needed to represent the weights

$w_{ii'}$ [1]. To obtain from the solution to the semidefinite relaxation a partition of the graph, the algorithm selects a random vector r, uniformly distributed on the unit sphere in $\boldsymbol{R}^n$, and sets

$$V_1 = \{i : \langle v_i, r\rangle > 0\}, \tag{6}$$

$$V_2 = \{i : \langle v_i, r\rangle < 0\}. \tag{7}$$

Vectors satisfying $\langle v_i, r\rangle = 0$ can be assigned a part in the partition arbitrarily since they occur with probability zero.

We note that both the integer quadratic program and the semidefinite relaxation exhibit a symmetry inherent to the Max Cut problem: If we negate each y_i and each v_i, respectively, the solution is unaffected. This is a natural property of an algorithm for Max Cut, since it does not matter which of the parts in the partition of V we choose to call V_1, as long as we call the other part V_2.

In their approximation algorithm for Max k-Cut, Frieze and Jerrum [4] face a complication similar to ours: representing variables which can take one of k values. To do this, they use a regular $(k-1)$-simplex centered at the origin. If the vertices of the simplex are $\{a_1, a_2, \dots, a_k\}$ the Max k-Cut problem can be formulated as

$$\begin{array}{ll} \text{maximize} & \dfrac{k-1}{k} \displaystyle\sum_{i<i'} w_{ii'} \bigl(1 - \langle y_i, y_{i'}\rangle\bigr) \\ \text{subject to} & y_i \in \{a_1, a_2, \dots, a_k\} \text{ for all } i. \end{array} \tag{8}$$

The partition is formed according to

$$V_j = \{i : y_i = a_j\}. \tag{9}$$

Since $\langle a_i, a_j\rangle = -\frac{1}{k-1}$ when $i \neq j$ and $\langle a_i, a_i\rangle = 1$, the natural way to relax this program to a semidefinite one is to use vectors v_i which are not constrained to the vertices of a simplex:

$$\begin{aligned} &\text{maximize} && \frac{k-1}{k}\sum_{i<i'} w_{ii'}\bigl(1-\langle v_i, v_{i'}\rangle\bigr) \\ &\text{subject to} && \langle v_i, v_i\rangle = 1 \text{ for all } i, \\ &&& \langle v_i, v_{i'}\rangle \geq -\tfrac{1}{k-1} \text{ for all } i \neq i'. \end{aligned} \tag{10}$$

To obtain from the solution to the semidefinite relaxation a partition of the graph, the algorithm selects k independent random vectors $r_1, r_2, \dots, r_k$ from a spherically symmetric distribution in $\boldsymbol{R}^n$, and sets

$$V_j = \bigl\{i : \langle v_i, r_j\rangle \geq \langle v_i, r_{j'}\rangle \text{ for all } j' \neq j\bigr\}. \tag{11}$$

When $k = 2$ this algorithm is equivalent to the Max Cut algorithm of Goemans and Williamson.

2.2 Our construction

Our goal is to generalize the algorithm of Goemans and Williamson to Max 2-Lin mod p. We first construct an approximation algorithm for systems of linear equations where the equations are of the form

$$x_i - x_{i'} = c. \tag{12}$$

A problem in applying the approach of Frieze and Jerrum is that it has no "metric" information, it can only express equality and non-equality and thus it would only know how to distinguish equations of the form $c = 0$ and $c \neq 0$ but not to distinguish equations with $c = 1$ and $c = 2$. The reason for this is that the algorithm chooses p random vectors without any linear structure. Our way of getting a linear structure is by representing each variable x_i by a constellation of p vectors, $\{u_0^i, u_1^i, \dots, u_{p-1}^i\}$. The idea now is that if a certain value of a corresponds to a particular constellation of vectors then the value $a + c$ corresponds the same set of vectors but the order is given a cyclic shift of

c. This implies that an equation $x - y = c$ is modeled by taking pairwise inner products of vectors in the constellation of x with vectors in the constellation of y where the pairing is such that a vector in the constellation of y is paired with a vector at distance c in the cyclic ordering. If the constellations correspond to legitimate encodings that satisfy the equation this would imply that each inner product is formed between two identical vectors resulting in unit value.

The rounding can now be performed by using *one* random vector r. The partitions would then be constructed as

$$V_j = \{x_i : \langle u_j^i, r\rangle \geq \langle u_{j'}^i, r\rangle \text{ for all } j' \neq j\}, \tag{13}$$

and all variables in V_j are assigned the value $-j$.

We create a consistent linear structure of these constellations by requiring that for all i, i' and all j, j', k,

$$\langle u_j^i, u_{j+k}^{i'}\rangle = \langle u_{j'}^i, u_{j'+k}^{i'}\rangle. \tag{14}$$

If we denote by $w_{ii'c}$ the weight of the equation $x_i - x_{i'} = c$, we can thus write our restricted version of the Max E2-Lin mod p problem as the following program:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i,i',c} w_{ii'c}\left(\frac{p-1}{p^2}\sum_{j=0}^{p-1}\langle u_j^i, u_{j+c}^{i'}\rangle + \frac{1}{p}\right) \\ \text{subject to} & \langle u_j^i, u_j^i\rangle = 1 \ \forall i, j, \\ & \langle u_j^i, u_{j'}^i\rangle = -\frac{1}{p-1} \ \forall i \forall j \neq j', \\ & \langle u_j^i, u_{j'}^{i'}\rangle \in \{1, -\frac{1}{p-1}\} \ \forall i \neq i' \forall j, j', \\ & \langle u_j^i, u_{j+k}^{i'}\rangle = \langle u_{j'}^i, u_{j'+k}^{i'}\rangle \ \forall i, i', j, j', k. \end{array} \tag{15}$$

To simplify the terminology, we will now define formally the constellation of vectors associated with each variable in the above program.

Definition 4. For each variable $x_i \in \boldsymbol{Z}_p$ we construct an object henceforth called a *simplicial porcupine* in the following way:

We take p vectors $\{u_j^i\}_{j=0}^{p-1}$ and add the following constraints to the semidefinite program:

$$\langle u_j^i, u_k^i \rangle = \begin{cases} 1 & \text{when } j = k, \\ -\frac{1}{p-1} & \text{otherwise,} \end{cases} \tag{16a}$$

for all i and all $j, k \in \boldsymbol{Z}_p$,

$$\langle u_j^i, u_{j+k}^{i'} \rangle = \langle u_{j'}^i, u_{j'+k}^{i'} \rangle \tag{16b}$$

for all i, i' and all $j, j', k \in \boldsymbol{Z}_p$, and

$$\langle u_j^i, u_k^{i'} \rangle \geq -\frac{1}{p-1} \tag{16c}$$

for all i, i' and all $j, k \in \boldsymbol{Z}_p$.

We can now relax the program (15) to a semidefinite one, and then apply the rounding procedure described above. For completeness, we write out the semidefinite relaxation:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i,i',c} w_{ii'c} \left(\frac{p-1}{p^2} \sum_{j=0}^{p-1} \langle u_j^i, u_{j+c}^{i'} \rangle + \frac{1}{p} \right) \\ \text{subject to} & \langle u_j^i, u_j^i \rangle = 1 \ \forall i, j, \\ & \langle u_j^i, u_{j'}^i \rangle = -\frac{1}{p-1} \ \forall i \forall j \neq j', \\ & \langle u_j^i, u_{j'}^{i'} \rangle \geq -\frac{1}{p-1} \ \forall i \neq i' \forall j, j', \\ & \langle u_j^i, u_{j+k}^{i'} \rangle = \langle u_{j'}^i, u_{j'+k}^{i'} \rangle \ \forall i, i', j, j', k. \end{array} \tag{17}$$

When we are to analyze the rounding procedure, we want to study the inner products $X_j = \langle u_j^i, r \rangle$. Unfortunately, the random variables X_j are dependent, which complicates the analysis. We would obtain a simpler analysis if the vectors corresponding to a variable were orthogonal, since then the corresponding inner products would be independent. It is easy to construct such a semidefinite program. All constraints change accordingly and for each equation the terms

$$\frac{1}{p}\sum_{j=0}^{p-1}\langle v_j^i, v_{j+c}^{i'}\rangle \tag{18}$$

are included in the objective function. Such a construction gives the semidefinite program

$$\begin{aligned}
\text{maximize}\quad & \sum_{i,i',c} w_{ii'c}\left(\frac{1}{p}\sum_{j=0}^{p-1}\langle v_j^i, v_{j+c}^{i'}\rangle\right)\\
\text{subject to}\quad & \langle v_j^i, v_j^i\rangle = 1\ \forall i, j,\\
& \langle v_j^i, v_{j'}^i\rangle = 0\ \forall i \forall j \neq j',\\
& \langle v_j^i, v_{j'}^{i'}\rangle \geq 0\ \forall i \neq i' \forall j, j',\\
& \langle v_j^i, v_{j+k}^{i'}\rangle = \langle v_{j'}^i, v_{j'+k}^{i'}\rangle\ \forall i, i', j, j', k.
\end{aligned} \tag{19}$$

We use the same rounding procedures in both cases: x_i is assigned the value $-j$ if $\langle v_j^i, r\rangle > \langle v_{j'}^i, r\rangle$ for all $j' \neq j$. It is this program we will analyze in Sec. 3.

Definition 5. For each variable $x_i \in \boldsymbol{Z}_p$ we construct an object henceforth called an *orthogonal porcupine* in the following way:

We take p vectors $\{v_j^i\}_{j=0}^{p-1}$ and add the following constraints to the semidefinite program:

$$\langle v_j^i, v_k^i\rangle = \begin{cases} 1 & \text{when } j = k,\\ 0 & \text{otherwise,}\end{cases} \tag{20a}$$

for all i and all $j, k \in \boldsymbol{Z}_p$,

$$\langle v_j^i, v_{j+k}^{i'}\rangle = \langle v_{j'}^i, v_{j'+k}^{i'}\rangle \tag{20b}$$

for all i, i' and all $j, j', k \in \boldsymbol{Z}_p$, and

$$\langle v_j^i, v_k^{i'}\rangle \geq 0 \quad \text{for all } i, i' \text{ and all } j, k \in \boldsymbol{Z}_p. \tag{20c}$$

When no confusion can arise, we will simply call the above object a *porcupine*.

In fact, the simplicial and orthogonal formulations are equally good, in terms of the quality of the relaxation.

Theorem 3. *The simplicial and orthogonal porcupine models achieve the same performance ratio for the restriction of* MAX E2-LIN MOD p *to equations of the form* $x_i - x_{i'} = c$.

Proof. An orthogonal porcupine $\{v_j^i\}_{j=0}^{p-1}$ can be transformed into a simplicial porcupine $\{u_j^i\}_{j=0}^{p-1}$ by letting

$$b^i = \frac{1}{p}\sum_{j=0}^{p-1} v_j^i, \tag{21}$$

$$u_j^i = \sqrt{\frac{p}{p-1}}(v_j^i - b^i). \tag{22}$$

The vector b^i is usually called the *barycenter* of the vectors $\{v_j^i\}_{j=0}^{p-1}$. The orthogonality of v_k^i, $k = 0, 1, \ldots p-1$ implies that

$$\langle u_j^i, u_j^i \rangle = \frac{p}{p-1}\left(\frac{(p-1)^2}{p^2} + (p-1)\cdot\frac{1}{p^2}\right) = 1, \tag{23}$$

while for $j \neq k$

$$\langle u_j^i, u_k^i \rangle = \frac{p}{p-1}\left(-2\cdot\frac{p-1}{p^2} + (p-2)\cdot\frac{1}{p^2}\right) = -\frac{1}{p-1} \tag{24}$$

and thus (16a) is satisfied. The constraints (20b) imply the constraints (16b). Also, the constraints (20b) and (20c) together imply the constraints (16c). To see this, it is enough to show that

$$\begin{aligned} -1/p &\leq \langle v_j^i - b^i, v_{j'}^{i'} - b^{i'} \rangle \\ &= \langle v_j^i, v_{j'}^{i'} \rangle - \langle v_j^i, b^{i'} \rangle - \langle b^i, v_{j'}^{i'} \rangle + \langle b^i, b^{i'} \rangle. \end{aligned} \tag{25}$$

Now note that the constraints (20b) imply that

$$\langle v_j^i, b^{i'} \rangle = \langle b^i, v_{j'}^{i'} \rangle = \langle b^i, b^{i'} \rangle, \tag{26}$$

and thus

$$\langle v_j^i - b^i, v_{j'}^{i'} - b^{i'} \rangle = \langle v_j^i, v_{j'}^{i'} \rangle - \langle b^i, b^{i'} \rangle \geq -\|b^i\| \|b^{i'}\| = -1/p. \tag{27}$$

Consider the contribution to the objective function from the equation $x_i - x_{i'} = c$ in the two models. The simplicial porcupine gives

$$\begin{aligned} &\frac{p-1}{p^2} \sum_{j=0}^{p-1} \langle u_j^i, u_{j+c}^{i'} \rangle + \frac{1}{p} \\ &= \frac{1}{p} \sum_{j=0}^{p-1} \langle v_j^i, v_{j+c}^{i'} \rangle - \frac{1}{p^2} \langle \sum_{k=0}^{p-1} v_k^i, \sum_{k=0}^{p-1} v_k^{i'} \rangle + \frac{1}{p} \\ &\geq \frac{1}{p} \sum_{j=0}^{p-1} \langle v_j^i, v_{j+c}^{i'} \rangle \end{aligned} \tag{28}$$

with equality if and only if the orthogonal porcupines $\{v_j^i\}_{j=0}^{p-1}$ and $\{v_j^{i'}\}_{j=0}^{p-1}$ have the same barycenters.

A simplicial porcupine $\{u_j^i\}_{j=0}^{p-1}$ can be transformed into an orthogonal porcupine $\{v_j^i\}_{j=0}^{p-1}$ by letting

$$v_j^i = \sqrt{\frac{1}{p}} u_\perp + \sqrt{\frac{p-1}{p}} u_j^i, \tag{29}$$

where $u_\perp$ is a unit vector such that $\langle u_\perp, u_j^i \rangle = 0$ for all i, j. This construction results in the barycenters of all orthogonal porcupines coinciding if the same $u_\perp$ is used for all simplicial porcupines. Also, the constraints (20b) will be satisfied in the orthogonal porcupine if the constraints (16b) are satisfied in the simplicial porcupine. This implies that we can assume that the barycenters of all orthogonal porcupines coincide. For, using the transformations (22)–(29), we can transform an arbitrary family of orthogonal porcupines into a family of orthogonal porcupines with coinciding barycenters without decreasing the objective function.

The probability of the equation $x_i - x_{i'} = c$ being satisfied after the randomized rounding is

$$\begin{aligned} & p \times \Pr[x_i \leftarrow c \text{ and } x_{i'} \leftarrow 0] \\ & = p \times \Pr\Biggl[\bigcap_{j=0}^{p-1} \Bigl(\langle v^i_{-c}, r\rangle \geq \langle v^i_j, r\rangle\Bigr) \quad \cap \\ & \qquad\qquad \bigcap_{j=0}^{p-1} \Bigl(\langle v^{i'}_0, r\rangle \geq \langle v^{i'}_j, r\rangle\Bigr)\Biggr]. \end{aligned} \tag{30}$$

The transformations between the different types of porcupines only involve scaling both sides of the inequalities by the same positive factor or adding the same constant to both sides. Hence $\Pr[x_i \leftarrow c \text{ and } x_{i'} \leftarrow 0]$ is unaffected. □

When studying the MAX E2-LIN MOD p problem, we will use orthogonal porcupines. Let us show that our construction is a relaxation of MAX E2-LIN MOD p.

Lemma 4. *Given an instance of* MAX E2-LIN MOD *p with all equations of the form $x_i - x_{i'} = c$ and the corresponding semidefinite program (19), the optimum of the former can never be larger than the optimum of the latter.*

Proof. Suppose that we have an assignment π to the variables x_i, such that x_i is assigned the value $\pi(x_i)$. Let $\{\hat{e}_j\}_{j=0}^{p-1}$ be orthonormal unit vectors in $\boldsymbol{R}^p$ and set

$$v^i_j = \hat{e}_{j+\pi(x_i)} \quad \text{for all } i \text{ and all } j \in \boldsymbol{Z}_p, \tag{31}$$

where indices are calculated using modulo p arithmetic. The sum (18) corresponding to an equation $x_i - x_{i'} = c$ then takes the value

$$\frac{1}{p}\sum_{j=0}^{p-1} \langle v^i_j, v^{i'}_{j+c}\rangle = \frac{1}{p}\sum_{j=0}^{p-1} \langle \hat{e}_{j+\pi(x_i)}, \hat{e}_{j+c+\pi(x_{i'})}\rangle. \tag{32}$$

If the equation $x_i - x_{i'} = c$ is satisfied, then $\pi(x_i) = \pi(x_{i'}) + c$, and

$$\frac{1}{p}\sum_{j=0}^{p-1}\langle \hat{e}_{j+\pi(x_i)}, \hat{e}_{j+c+\pi(x_{i'})}\rangle = 1. \tag{33}$$

On the other hand, if the equation is not satisfied, then $\pi(x_i) \neq \pi(x_{i'}) + c$, and

$$\frac{1}{p}\sum_{j=0}^{p-1}\langle \hat{e}_{j+\pi(x_i)}, \hat{e}_{j+c+\pi(x_{i'})}\rangle = 0. \tag{34}$$

Thus, the maximum of the semidefinite program can never be less than the optimum of the Max E2-Lin mod p instance. □

3 Our algorithms

In this section we use the relaxations constructed in Sec. 2.2 to formulate an algorithm approximating Max 2-Lin mod p within $(1-\kappa(p))p$, where $\kappa(p) > 0$, for all p. The algorithm is constructed in three steps. First, we describe an algorithm which works for instances of Max E2-Lin mod p where all equations are of the form $x_i - x_{i'} = c$. This algorithm is then generalized to handle instances where also equations of the form $x_i = c$ are allowed. Finally, the resulting algorithm is generalized once more to handle general Max 2-Lin mod p instances.

3.1 Equations of the form $x_i - x_{i'} = c$

We use the semidefinite program (19) constructed in Sec. 2.2. We can now formulate the algorithm to approximate Max E2-Lin mod p restricted to instances where every equation is of the form $x_i - x_{i'} = c$. Below, κ is a constant, which is to be determined during the analysis of the algorithm. Given a set of linear equations, we run both Algorithm 1 and the following algorithm:

Algorithm 2. Construct and solve the semidefinite program (19). Use the vectors obtained from the optimal solution to the semidefinite program to obtain

an assignment to the variables x_i in the following way: A vector r is selected by independently choosing each component as an $\mathrm{N}(0,1)$ random variable. Then, for each porcupine $\{v_j^i\}_{j=0}^{p-1}$ we find the j maximizing $\langle v_j^i, r\rangle$, and set $x_i = -j$.

We take as our result the maximum of the results obtained from Algorithms 1 and 2. By Corollary 2, we will always approximate the optimum within $(1-\kappa)p$ if the optimum weight is less than $1-\kappa$ times the total weight of all equations. Thus, when analyzing the performance of Algorithm 2, we can assume that the optimum is at least $1-\kappa$ times the weight of all equations. Intuitively, this means that for most equations, the two porcupines involved will be almost perfectly aligned.

Lemma 5. *If the objective function is at least $1-\kappa$ times the total weight of all equations, then equations of total weight at least $1-2\kappa/\varepsilon$ times the total weight of the instance have the property that the corresponding terms (18) in the objective function evaluate to at least $\sqrt{1-\varepsilon}$.*

Proof. Let μ be the fraction of the equations with the property that the corresponding terms (18) in the objective functions are less than $\sqrt{1-\varepsilon}$. Then, the inequality

$$\mu\sqrt{1-\varepsilon} + (1-\mu) \geq 1-\kappa \tag{35}$$

must always hold. When we solve for μ we obtain $\mu \leq \kappa/(1-\sqrt{1-\varepsilon}) \leq 2\kappa/\varepsilon$. □

The conclusion in Lemma 5 is useful because of the following lemma.

Lemma 6. *There is a universal constant ε such that for all primes p, for any equation $x_i - x_{i'} = c$ whose corresponding terms (18) in the objective function sum up to at least $\sqrt{1-\varepsilon}$,*

$$\Pr[\text{equation satisfied}] > \frac{3}{2p}. \tag{36}$$

One acceptable value for ε *is* $3 \cdot 10^{-7}$.

Proof. For notational simplicity we assume that the value of the corresponding term in objective function is exactly $\sqrt{1-\varepsilon}$ since a larger value only increases the probability of satisfying the equation. Thus we have

$$\frac{1}{p}\sum_{j=0}^{p-1}\langle v_j^i, v_{j+c}^{i'}\rangle = \sqrt{1-\varepsilon}. \tag{37}$$

By the constraints (20b), the identity (37) implies that

$$v_{j+c}^{i'} = \sqrt{1-\varepsilon}v_j^i + \sqrt{\varepsilon}e_j^c, \tag{38}$$

where e_j^c is orthogonal to v_j^i and $\|e_j^c\| = 1$.

Definition 6. For a fixed equation $x_i - x_{i'} = c$, let $X_j = \langle v_j^i, r\rangle$, $Y_j = \langle v_{j+c}^{i'}, r\rangle$, and $Z_j = \langle e_j^c, r\rangle$.

By the construction of the porcupines and the choice of r, the X_j are i.i.d. $\mathrm{N}(0,1)$ and the Z_j are, possibly dependent, $\mathrm{N}(0,1)$. However, for each fixed j, X_j and Z_j are independent.

The proof needs four lemmas following from undergraduate probability theory stated and proved in Appendix A.

The randomized rounding succeeds if the "chosen" vectors are v_j^i and $v_{j+c}^{i'}$, respectively, for some j. Another way to state this is that we want to estimate the probability that some j maximizes Y_j, given that the very same j maximizes X_j.

We will first show that the theorem holds for large p: Let $A(\delta)$ be the event that the largest X_j is at least $(1+\delta)\sqrt{2\ln p}$ and all other X_j are at most $(1+\delta/2)\sqrt{2\ln p}$. By Lemma 33,

$$\Pr[A(\delta)] > \frac{1}{2p^{2\delta+\delta^2}(1+\delta)\sqrt{\pi\ln p}}\left(1 - \frac{1}{2\ln p} - \frac{1}{2p^{\delta}\sqrt{\pi\ln p}}\right). \tag{39}$$

Next, let us study the Z_j. Let

$$B(\delta,\varepsilon)=\bigcap_{j=0}^{p-1}\left\{|X_j-Y_j|<\frac{\delta}{4}\sqrt{(1-\varepsilon)2\ln p}\right\}. \tag{40}$$

Since $Y_j=\sqrt{1-\varepsilon}X_j+\sqrt{\varepsilon}Z_j$, we can use Lemma 34 to obtain

$$\Pr[\overline{B(\delta,\varepsilon)}]<\frac{4p^{1-\delta^2(1-\varepsilon)/32\varepsilon}}{\delta}\sqrt{\frac{2\varepsilon}{(1-\varepsilon)\pi\ln p}}. \tag{41}$$

The equation is satisfied if both $A(\delta)$ and $B(\delta,\varepsilon)$ occur. The probability that this happens can be bounded by

$$\Pr[A(\delta)\cap B(\delta,\varepsilon)]\geq\Pr[A(\delta)]-\Pr[\overline{B(\delta,\varepsilon)}]. \tag{42}$$

Now set $\delta=10^{-2}$ and $\varepsilon=10^{-6}$. Then

$$\Pr[\overline{B(\delta,\varepsilon)}]\leq 400p^{-2}\sqrt{\frac{2\cdot 10^{-6}}{3\ln p}}\leq\frac{1}{2p}, \tag{43}$$

and thus it is sufficient to establish that $\Pr[A(\delta)]\geq 2/p$. If we multiply the estimate of $\Pr[A(\delta)]$ given in (39) by $p/2$ we get an increasing function in p and a direct verification shows that it is larger than 1 for $p\geq 20$.

Now it remains to be shown that the theorem is valid also for $p\leq 19$. Let $C(\delta)$ be the event that the difference between the largest and the second-largest X_j is at least δ, and let $D(\delta)$ be the event that, for all j, $|X_j-Y_j|\leq\delta/2$. By Lemmas 35 and 36,

$$\Pr[C(\delta)]\geq 1-\frac{p^2\delta}{(p-1)\sqrt{2\pi}}, \tag{44}$$

$$\Pr[\overline{D(\delta)}]\leq\frac{4p}{\delta}\sqrt{\frac{\varepsilon}{\pi}}. \tag{45}$$

The equation is satisfied if both $C(\delta)$ and $D(\delta)$ occur. Assuming

$$\delta\leq\frac{(2p-3)(p-1)\sqrt{2\pi}}{4p^3} \tag{46}$$

and

$$\varepsilon \leq \frac{(2p-3)^2\delta^2\pi}{256p^4} \tag{47}$$

we see that the probability that this happens can be bounded by

$$\Pr[C(\delta) \cap D(\delta)] \geq \Pr[C(\delta)] - \Pr[\overline{D(\delta)}] \geq 1 - \frac{2p-3}{4p} - \frac{2p-3}{4p} = \frac{3}{2p}. \tag{48}$$

For $p \leq 19$ we see that it is sufficient to take $\varepsilon = 3 \cdot 10^{-7}$. □

Putting the pieces together we obtain:

Theorem 7. *There exists a universal constant κ such that there exists, for all primes p, a randomized polynomial time algorithm approximating systems of linear equations mod p of the form $x_i - x_{i'} = c$ within $(1-\kappa)p$. One acceptable value of κ is 10^{-8}.*

Proof. The algorithm is as described above. Denote by w the total weight of the instance. If the optimum is at most $(1-\kappa)w$, Algorithm 1 approximates the solution within $(1-\kappa)p$.

Otherwise, by Lemma 5, equations with total weight at least $(1-2\kappa/\varepsilon)w$ have the property that the corresponding terms in the objective function in the semidefinite program evaluate to at least $\sqrt{1-\varepsilon}$ in the optimal solution. By Lemma 6, if we choose $\varepsilon = 10^{-7}$ and $\kappa = 10^{-8}$, these equations are satisfied with probability at least $3/2p$, over the choice of the random vector r. Thus, the expected weight of the solution obtained by the rounding is at least $12w/10p > w(1-\kappa)/p$. □

It is straightforward to adapt the algorithm to handle equations with one unknown. Simply introduce a new variable x_0 which should take the value zero. Each equation of the form $x_i = c$ is replaced by $x_i - x_0 = c$. If $x_0 \neq 0$ in the optimal solution, we transform the solution according to $x_i \leftarrow x_i - x_0$. This new assignment to the variables satisfies exactly the same equations as the original one.

Finally, since nothing in Sec. 3.1 actually uses that p is prime, the results hold also for composite p—indeed, they hold for equations over any finite Abelian group. In this case, the vectors in the porcupines correspond to the elements of the group; the number of vectors in each porcupine is equal to the order of the group.

3.2 General equations

In this section we extend the algorithm from Sec. 3.1 to handle general Max 2-Lin mod p instances. We do this by associating $p-1$ porcupines, $\{v_j^{i,1}\}_{j=0}^{p-1}$ up to $\{v_j^{i,p-1}\}_{j=0}^{p-1}$, with each variable x_i. These porcupines are supposed to represent x_i, $2x_i$, up to $(p-1)x_i$, respectively. The porcupines are constructed as described in Definition 5, with the constraints (20) generalized to

$$\langle v_j^{i,\ell}, v_k^{i,\ell} \rangle = \begin{cases} 1 & \text{when } j = k, \\ 0 & \text{otherwise,} \end{cases} \tag{49a}$$

for all i, all $j,k \in \boldsymbol{Z}_p$, and all $\ell \in \boldsymbol{Z}_p^*$;

$$\langle v_j^{i,\ell}, v_{j+k}^{i',\ell'} \rangle = \langle v_{j'}^{i,\ell}, v_{j'+k}^{i',\ell'} \rangle \tag{49b}$$

for all i, i', all $j, j', k \in \boldsymbol{Z}_p$, and all $\ell, \ell' \in \boldsymbol{Z}_p^*$;

$$\sum_{j=0}^{p-1} \langle v_j^{i,\ell}, v_{j''}^{i'',\ell''} \rangle = \sum_{j=0}^{p-1} \langle v_j^{i',\ell'}, v_{j''}^{i'',\ell''} \rangle \tag{49c}$$

for all i, i', i'', all $j'' \in \boldsymbol{Z}_p$ and all $\ell, \ell', \ell'' \in \boldsymbol{Z}_p^*$; and

$$\langle v_j^{i,\ell}, v_k^{i',\ell'} \rangle \geq 0 \tag{49d}$$

for all i, i', all $j, k \in \boldsymbol{Z}_p$, and all $\ell, \ell' \in \boldsymbol{Z}_p^*$.

The condition (49c) ensures that all porcupines have the same barycenter.

We would want the porcupines to be dependent in such a way that $x_i = c$ is equivalent to $kx_i = kc$, but since the resulting constraint is not linear, this seems hard to achieve. Instead, we allow the porcupines corresponding to the same variable to vary freely. Somewhat surprisingly, it turns out that this enables us to construct an algorithm which approximates MAX 2-LIN MOD p within $p - \kappa(p)$, where $\kappa(p) > 0$ for all p, but tends to zero as p grows to infinity.

To handle equations of the form $ax_i = c$ we introduce a new variable x_0. Our algorithm will be designed in such a way that x_0 always gets the value 0. Each equation $ax_i = c$ can thus be changed to $ax_i - x_0 = c$. For each equation $ax_i - bx_{i'} = c$ we include the terms

$$\frac{1}{p(p-1)} \sum_{k=1}^{p-1} \sum_{j=0}^{p-1} \langle v_j^{i,ka}, v_{j+kc}^{i',kb} \rangle \tag{50}$$

in the objective function. We remind the reader that all indices are counted modulo p. Since we use the same type of objective function as in the algorithm for equations of the form $x_i - x_{i'} = c$, the simplicial and orthogonal porcupine models achieve the same performance ratio also for general instances of MAX 2-LIN MOD p. We analyze orthogonal porcupines.

Lemma 8. *Given an instance of* MAX 2-LIN MOD p *and the corresponding semidefinite program constructed as described above, the optimum of the former can never be larger than the optimum of the latter.*

Proof. Suppose that we have an assignment π to the variables x_i. Let $\{\hat{e}_j\}_{j=0}^{p-1}$ be orthonormal unit vectors in $\boldsymbol{R}^p$ and set

$$v_j^{i,\ell} = \hat{e}_{j+\ell\pi(x_i)} \tag{51}$$

for all i, all $j \in \boldsymbol{Z}_p$ and all $\ell \in \boldsymbol{Z}_p^*$. The terms (50) corresponding to an equation

$ax_i - bx_{i'} = c$ are then

$$\begin{aligned}&\frac{1}{p(p-1)}\sum_{k=1}^{p-1}\sum_{j=0}^{p-1}\langle v_j^{i,ka}, v_{j+kc}^{i',kb}\rangle \\ &\quad = \frac{1}{p(p-1)}\sum_{k=1}^{p-1}\sum_{j=0}^{p-1}\langle \hat{e}_{j+ka\pi(x_i)}, \hat{e}_{j+kc+kb\pi(x_{i'})}\rangle.\end{aligned} \tag{52}$$

If the equation is satisfied by the assignment π, then $ka\pi(x_i) = kb\pi(x_{i'}) + kc$, and

$$\frac{1}{p(p-1)}\sum_{k=1}^{p-1}\sum_{j=0}^{p-1}\langle \hat{e}_{j+ka\pi(x_i)}, \hat{e}_{j+kc+kb\pi(x_{i'})}\rangle = 1. \tag{53}$$

On the other hand, if the equation is not satisfied, then $ka\pi(x_i) \neq kb\pi(x_{i'}) + kc$, and

$$\frac{1}{p(p-1)}\sum_{k=1}^{p-1}\sum_{j=0}^{p-1}\langle \hat{e}_{j+ka\pi(x_i)}, \hat{e}_{j+kc+kb\pi(x_{i'})}\rangle = 0. \tag{54}$$

Thus, the maximum of the semidefinite program can never be less than the optimum of the Max 2-Lin mod p instance. □

The intuition behind the algorithm for Max 2-Lin mod p is as follows: There are $p-1$ porcupines corresponding to each variable. Each of these porcupines give one vote on a "good" assignment to the corresponding variable. Since there are at most $p-1$ different votes, a random guess among these votes satisfies any equation with probability almost $1/(p-1)$, which is more than $1/p$. We now give the details of this algorithm by a suitable generalization of Algorithm 2.

Algorithm 3. Construct and solve the above semidefinite program. Use the vectors obtained from the optimal solution to the semidefinite program to obtain an assignment to the variables x_i in the following way: A vector r is selected by independently choosing each component as an $\mathrm{N}(0,1)$ random variable. Then, we do the following:

```
Find the $j \in \boldsymbol{Z}_p$ maximizing $\langle v_j^{0,1}, r \rangle$.
Set $t \leftarrow j$.
For each $i \in [0..n]$,
  For all $j \in \boldsymbol{Z}_p$, set $q_{i,j} \leftarrow 0$.
  For all $k \in \boldsymbol{Z}_p^*$,
    Find the $j \in \boldsymbol{Z}_p$ maximizing $\langle v_j^{i,k}, r \rangle$.
    Set $q_{i,k^{-1}(j-t)} \leftarrow q_{i,k^{-1}(j-t)} + \frac{1}{p-1}$.
```

Finally, given the resulting $q_{i,j}$, each variable x_i, but x_0, is given the value $-j$ with probability $q_{i,j}$. The variable x_0 is given the value 0.

Remark 1. By the choice of t in Algorithm 3 above, $q_{0,0}$ will always be non-zero. This turns out to be essential for analyzing equations containing x_0.

To obtain our estimate of the optimum of the Max 2-Lin mod p instance, we take the maximum of the results obtained from Algorithms 1 and 3.

By Corollary 2 and Lemma 8, Algorithm 1 is a $(1-\kappa)p$-approximation algorithm if the optimum weight of the semidefinite program is less than $1-\kappa$ times the weight of all equations. Thus, when analyzing the performance of Algorithm 3, we can assume that the optimum of the semidefinite program is at least $1-\kappa$ times the weight of all equations. By this assumption and Lemma 5, equations of total weight at least $1-2\kappa/\varepsilon$ times the weight of the instance have the property that the sum of the corresponding terms (50) in the objective functions is at least $\sqrt{1-\varepsilon}$. Let us now study an arbitrary equation $ax_i - bx_{i'} = c$ with that property. I.e.,

$$\frac{1}{p(p-1)} \sum_{k=1}^{p-1} \sum_{j=0}^{p-1} \langle v_j^{i,ka}, v_{j+kc}^{i',kb} \rangle \geq \sqrt{1-\varepsilon}. \tag{55}$$

We want to show that this equation is satisfied with probability a little bit larger than $1/p$. Let us study the details of the selection procedure in Algorithm 3. Informally, we expect the following:

- By the condition (55) we expect the vectors $v_j^{i,ka}$ and $v_{j+kc}^{i',kb}$ to be almost perfectly aligned, for all j and k.

- For each k, this should imply, that if some j maximizes $\langle v_j^{i,ka}, r\rangle$ then, with high probability over the choice of r, we will have that $j' = j + kc$ maximizes $\langle v_{j'}^{i',kb}, r\rangle$.
- In terms of $q_{i,j}$ this means that

$$q_{i,a^{-1}j} = q_{i',b^{-1}(j+c)} \tag{56}$$

should hold for each j with high probability.

- If the above equivalence holds for all j, the randomized assignment at the end of Algorithm 3 will find a satisfying assignment with probability at least $1/(p-1)$.

We now formalize this intuition.

Definition 7. $X_j^k = \langle v_j^{i,ak}, r\rangle$ and $Y_j^k = \langle v_{j+kc}^{i',bk}, r\rangle$.

Remark 2. By the construction of the porcupines and the choice of r, the X_j^k are i.i.d. $\mathrm{N}(0,1)$.

Definition 8. Let ε_k be defined by the relation

$$\frac{1}{p}\sum_{j=0}^{p-1}\langle v_j^{i,ka}, v_{j+kc}^{i',kb}\rangle = \sqrt{1-\varepsilon_k}. \tag{57}$$

Remark 3. By the constraints (49), the above definitions imply that

$$Y_j^k = \sqrt{1-\varepsilon_k}X_j^k + \sqrt{\varepsilon_k}Z_j^k, \tag{58}$$

where $Z_j^k \in \mathrm{N}(0,1)$.

Lemma 9. *Let $ax_i - bx_{i'} = c$ be an arbitrary equation with the property that the corresponding terms in the objective function satisfy the bound (55), and let*

A_k be the event that the same j maximizes X_j^k and Y_j^k. Then, for all $\delta > 0$,

$$\Pr\left[\overline{A_k}\right] \le \frac{p^2\delta}{(p-1)\sqrt{2\pi}} + \frac{4p}{\delta}\sqrt{\frac{\varepsilon_k}{\pi}}. \tag{59}$$

Proof. Let $X_{(p)}^k$ and $X_{(p-1)}^k$ be the maximum and the second maximum, respectively, of the X_j^k. Define the events $B_k(\delta)$ and $C_k(\delta)$ as follows:

$$B_k(\delta) = \left\{X_{(p)}^k > X_{(p-1)}^k + \delta\right\}, \tag{60}$$

$$C_k(\delta) = \bigcap_{i=0}^{p-1} \left\{\left|X_i^k - Y_i^k\right| < \frac{\delta}{2}\right\}. \tag{61}$$

If both $B_k(\delta)$ and $C_k(\delta)$ occur, then A_k must occur. Furthermore, if there exists some δ such that $B_k(\delta)$ and $C_k(\delta)$ both occur with high probability, A_k will also occur with high probability. For,

$$B_k(\delta) \cap C_k(\delta) \subseteq A_k \implies \Pr\left[\overline{A_k}\right] \le \Pr\left[\overline{B_k(\delta)}\right] + \Pr\left[\overline{C_k(\delta)}\right]. \tag{62}$$

By Lemma 35 we obtain the bound

$$\Pr\left[\overline{B_k(\delta)}\right] \le \frac{p^2\delta}{(p-1)\sqrt{2\pi}}, \tag{63}$$

and by the relation (58) and Lemma 36 we obtain

$$\Pr\left[\overline{C_k(\delta)}\right] \le \frac{4p}{\delta}\sqrt{\frac{\varepsilon_k}{\pi}}. \tag{64}$$

When the bounds (62), (63), and (64) are combined, the lemma follows. □

Lemma 10. *For fixed i and i', let A_k be the event that the same j maximizes X_j^k and Y_j^k. Then, if A_k occur for all k, we are ensured that $q_{i,j} = q_{i',b^{-1}(aj+c)}$ for all $j \in \boldsymbol{Z}_p$.*

Proof. Fix i and i'. Initially in the algorithm, all $q_{i,j}$ are zero. Suppose q_{i,j_1} is incremented in Algorithm 3 for a certain $k = ak_1$. Then the maximum was obtained at $j = t + kj_1 = t + ak_1j_1$.

We now want to show that this implies that $q_{i',b^{-1}(aj_1+c)}$ is also incremented. Since A_{k_1} occurs, for $k = bk_1$ the maximum in the loop for i' appears at $j+k_1c = t+ak_1j_1+k_1c$ and hence q_{i',j_2} is incremented for $j_2 = (bk_1)^{-1}(t+ak_1j_1+k_1c-t) = b^{-1}(aj_1+c)$. The lemma now follows. □

Lemma 11. *Let $ax_i - bx_{i'} = c$ be an arbitrary equation with the property that the corresponding terms in the objective function satisfy the bound (55). Then,*

$$\Pr\left[\bigcap_{j\in \boldsymbol{Z}_p} q_{i,j} = q_{i',b^{-1}(aj+c)}\right] \geq 1 - \frac{p^2\delta}{\sqrt{2\pi}} - \frac{4p(p-1)}{\delta}\sqrt{\frac{\varepsilon}{\pi}}, \tag{65}$$

where $\delta > 0$ is arbitrary.

Proof. By Lemmas 9 and 10,

$$\begin{aligned}\Pr\left[\bigcap_{j=0}^{p-1} q_{i,a^{-1}j} = q_{i',b^{-1}(j+c)}\right] &\geq \Pr\left[\bigcap_{k=1}^{p-1} A_k\right] \\ &\geq 1 - \sum_{k=1}^{p-1}\Pr\left[\overline{A_k}\right] \\ &\geq 1 - \frac{p^2\delta}{\sqrt{2\pi}} - \frac{4p}{\delta\sqrt{\pi}}\sum_{k=1}^{p-1}\sqrt{\varepsilon_k}\,.\end{aligned} \tag{66}$$

Since the function $x \mapsto \sqrt{1-x}$ is concave when $x \in [0,1]$, we can apply Jensen's inequality, which states that

$$\sum_i a_i f(x_i) \leq f\left(\sum_i a_i x_i\right) \tag{67}$$

for any positive a_i such that $\sum_i a_i = 1$ and any concave function f [8]. Applied to our case, this inequality implies that

$$\sqrt{1-\varepsilon} \leq \sum_{k=1}^{p-1}\frac{\sqrt{1-\varepsilon_k}}{p-1} \leq \sqrt{1 - \sum_{k=1}^{p-1}\frac{\varepsilon_k}{p-1}}\,, \tag{68}$$

where the first inequality follows from the bound (55) combined with the relation

(57), and the second from Jensen's inequality. Thus,

$$\sum_{k=1}^{p-1} \frac{\varepsilon_k}{p-1} \leq \varepsilon. \tag{69}$$

Using the Cauchy-Schwartz inequality, we obtain from (69) the bound

$$\sum_{k=1}^{p-1} \sqrt{\varepsilon_k} \leq \sqrt{(p-1)\sum_{k=1}^{p-1} \varepsilon_k} \leq (p-1)\sqrt{\varepsilon}. \tag{70}$$

When this is inserted into (66), the proof is complete. □

Lemma 12. *If $q_{0,0} > 0$ and $q_{i,j} = q_{i',b^{-1}(aj+c)}$ for all i, i' and all $j \in \boldsymbol{Z}_p$, then the equation $ax_i - bx_{i'} = c$ will be satisfied with probability at least $1/(p-1)$.*

Proof. By the construction of the system of linear equations there are no equations $ax_i - bx_{i'} = c$ where $i = 0$. If $i' \neq 0$ the $q_{i,j}$ and $q_{i',j}$, computed using the probabilistic construction described above, are used to independently assign values to x_i and $x_{i'}$. Thus,

$$\Pr[\text{equation satisfied}] = \sum_j q_{i,j} q_{i',b^{-1}(aj+c)} = \sum_j q_{i,j}^2, \tag{71}$$

where the second equality follows from the initial requirement in the formulation of the lemma. Now since any nonzero value of $q_{i,j}$ is at least $1/(p-1)$ we have

$$\sum_j q_{i,j}^2 \geq \frac{1}{p-1} \sum q_{i,j} = \frac{1}{p-1}. \tag{72}$$

If $i' = 0$ we know that $b = 1$ and $x_{i'} = 0$. Then

$$\Pr[\text{equation satisfied}] = q_{i,-a^{-1}c} = q_{0,0}. \tag{73}$$

Since $q_{0,0} \neq 0$ we know, by the construction of Algorithm 3, that $q_{0,0} \geq 1/(p-1)$, and the lemma follows. □

Theorem 13. *It is possible to choose $\varepsilon(p) > 0$ such that, for all primes p,*

$$\Pr[\text{equation satisfied}] > \frac{2}{2p-1} \tag{74}$$

for all equations with the property that the corresponding terms in the objective function are at least $\sqrt{1-\varepsilon(p)}$.

Proof. It follows immediately from the construction of Algorithm 3, together with Lemmas 9–12, that

$$\Pr[\text{equation satisfied}] > \frac{1}{p-1}\left(1 - \frac{p^2\delta}{\sqrt{2\pi}} - \frac{4p(p-1)}{\delta}\sqrt{\frac{\varepsilon}{\pi}}\right). \tag{75}$$

Setting

$$\delta(p) = \frac{\sqrt{2\pi}}{8p^3}, \tag{76}$$

$$\varepsilon(p) = \frac{\delta(p)^2\pi}{1024p^4(p-1)^2} = \frac{\pi^2}{32768p^{10}(p-1)^2}, \tag{77}$$

and substituting into (75) we obtain

$$\Pr[\text{equation satisfied}] > \frac{1}{p-1}\left(1 - \frac{1}{8p} - \frac{1}{8p}\right) \geq \frac{2}{2p-1}. \tag{78}$$

□

As an immediate corollary, the main theorem follows. It is proved in exactly the same way as Theorem 7.

Theorem 14. *For all primes p, there exists a randomized polynomial time algorithm approximating* Max 2-Lin mod p *within $(1-\kappa(p))p$, where $\kappa(p) > 0$ for all p.*

Proof. The algorithm is as described above. Denote by w the total weight of the instance and set $\kappa(p) = \varepsilon(p)/6p$ where $\varepsilon(p)$ is given in Theorem 13. If the optimum is at most $(1-\kappa)w$, Algorithm 1 approximates the solution within $(1-\kappa)p$.

Otherwise, Lemma 5 tells us that equations with total weight at least $(1 - 2\kappa(p)/\varepsilon(p))w = (1-\frac{1}{3p})w$ have the property that the corresponding terms in the objective function in the semidefinite program evaluate to at least $\sqrt{1-\varepsilon(p)}$ in the optimal solution. By Theorem 13, the resulting solution will, on the average, satisfy equations of weight at least

$$\frac{2(1-\frac{1}{3p})w}{2p-1} \geq \frac{w}{p(1-\frac{1}{6p})} \geq \frac{w}{p(1-\kappa)}. \tag{79}$$

□

If we use the explicit value of $\kappa(p)$ from the proof of Theorem 13, we see that Max 2-Lin mod p is approximable within $p - \Theta(p^{-12})$.

It is possible to generalize the algorithm to Max 2-Lin mod m for composite m: First notice that since equations where $\gcd(a, b, m)$ does not divide c can never be satisfied, we can remove them from the instance. Assume that the total weight of all remaining equations is w. If the optimum is less than $(1-\kappa)w$, there is nothing to prove since we can simply apply Algorithm 1, while if it at least $(1-\kappa)w$ we consider a prime factor p of m and proceed as follows: We determine values $\{a_i\}_{i=1}^n$ mod p such that when setting $x_i = a_i + px'_i$ we get a system mod m/p in x'_i such that the weight of the satisfiable equations is at least $w/p(1-\kappa(p))$. The result then follows by applying Algorithm 1 to this resulting system yielding a solution that satisfies equations of weight at least $w/m(1-\kappa(p))$. The condition that an equation remains satisfiable is simply a linear equation mod p and by the assumption that it is possible to find a solution mod m that satisfies almost all equations, desired values a_i can be found by the approximation algorithm for a prime modulus.

In fact, since the only property of $\boldsymbol{Z}_p$ used in the algorithm and the proof of correctness is the existence of a multiplicative inverse, our algorithm generalizes to linear equations over any finite field.

4 Max k-Cut and comparison to the algorithm of Frieze and Jerrum

In this section, we go back to simplicial porcupines to ease the comparison with the algorithm of Frieze and Jerrum [4], which is described in Sec. 2.1. We observe that MAX k-CUT is a special case of MAX E2-LIN MOD k: That the edge (i, i') is to be cut is equivalent to exactly one of the equations $x_i - x_{i'} = c$, for $c = 1, 2, \ldots, k-1$, being satisfied. This corresponds to the term

$$\sum_{c=1}^{k-1} \left(\frac{k-1}{k^2} \sum_{j=0}^{k-1} \langle u_j^i, u_{j+c}^{i'} \rangle + \frac{1}{k} \right) \tag{80}$$

in the objective function. Note that if we use the fact that $\sum_j u_j^i = 0$ for all i, we obtain exactly the same objective function (8) as Frieze and Jerrum used. Thus, it is possible to solve MAX k-CUT by formulating the instance as a MAX E2-LIN MOD k instance and solve it using the methods developed in Sec. 3.1. It is interesting to study how this method of solution compares to that of Frieze and Jerrum.

Another, seemingly good, strategy to improve the algorithm of Frieze and Jerrum is to change the rounding procedure by adding constraints forcing the random vectors to be far apart.

We show that the two approaches outlined above to some extent are equivalent to the relaxation (10) with the original randomized rounding strategy. Notice, however, that Frieze and Jerrum's semidefinite program cannot be used for MAX E2-LIN MOD k as their objective function is not able to represent equations of the form $x_i - x_{i'} = c$.

4.1 Using porcupines for the rounding

Frieze and Jerrum round the solution to their semidefinite program using k random vectors $r_0, \ldots, r_{k-1}$ where the components of each r_i can be chosen as independent $\mathrm{N}(0, 1)$ variables. At first, it seems that it would be better to instead choose a random porcupine. To make the situation more like the rest of the paper we fist scale all vectors by a factor $\sqrt{n}$ to make each component $N(0, 1/\sqrt{n})$. This scaling does not change the algorithm.

Definition 9. A *random orthogonal porcupine* is a porcupine chosen as follows: The first vector s_0 in the porcupine is chosen uniformly at random. Then, for each $i \geq 1$, the vector s_i is chosen uniformly at random from the subspace orthogonal to the space spanned by the vectors $s_0, \ldots, s_{i-1}$. Finally all vectors are normalized. When no confusion can arise, we will simply call the above object a *random porcupine.*

One could also imagine using a *random simplicial porcupine*, defined in the obvious way. We note in passing that a theorem analogous to Theorem 3 holds for random porcupines.

Theorem 15. *Rounding using a random orthogonal porcupine is equivalent to rounding using a random simplicial porcupine.*

Proof. Let $\{s_i\}_{i=0}^{k-1}$ be an orthogonal porcupine and

$$s'_i = \sqrt{\frac{k}{k-1}} \left(s_i - \frac{1}{k} \sum_j s_j \right). \tag{81}$$

It is easy to verify that $\{s'_i\}_{i=0}^{k-1}$ is a simplicial porcupine. The probability that the edge (i, i') is not cut after the rounding is

$$k \times \Pr\left[\bigcap_{j=1}^{k-1} \left(\langle v^i, s'_0 \rangle \geq \langle v^i, s'_j \rangle \right) \cap \bigcap_{j=1}^{k-1} \left(\langle v^{i'}, s'_0 \rangle \geq \langle v^{i'}, s'_j \rangle \right) \right] \tag{82}$$

where v^i and $v^{i'}$ are vectors from the semidefinite program. Using the same argument as in the proof of Theorem 3, we conclude that this probability is the same for the orthogonal and simplicial porcupine models. □

We now relate the rounding procedure proposed above to the rounding procedure of Frieze and Jerrum. The first thing to notice is that the k random vectors $r_0 \dots, r_{k-1}$ are in fact close to a random orthogonal porcupine with high probability.

Lemma 16. *Let $\varepsilon \le 1$. Construct the random vectors $r_0, \dots, r_{k-1}$ by choosing the components of each vector as independent $\mathrm{N}(0, 1/\sqrt{n})$ random variables. Then*

$$\mathrm{E}[\langle r_i, r_j \rangle] = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{83}$$

$$\Pr[|\langle r_i, r_j \rangle - \mathrm{E}[\langle r_i, r_j \rangle]| > \varepsilon] \in O\bigl(1/n\varepsilon^2\bigr). \tag{84}$$

Proof. If X and Y are independent $\mathrm{N}(0, 1/\sqrt{n})$ random variables,

$$\mathrm{E}[X^2] = 1/n, \tag{85}$$

$$\mathrm{E}[X^4] = 3/n^2, \tag{86}$$

$$\mathrm{E}[XY] = 0, \tag{87}$$

$$\mathrm{E}[X^2Y^2] = \mathrm{E}[X^2]E[Y^2] = 1/n^2, \tag{88}$$

which implies that

$$\mathrm{Var}[X^2] = 2/n^2, \tag{89}$$

$$\mathrm{Var}[XY] = 1/n^2. \tag{90}$$

Since the components of the vectors $r_0, \dots, r_{k-1}$ are independent $\mathrm{N}(0, 1/\sqrt{n})$ random variables,

$$\mathrm{E}[\langle r_i, r_j \rangle] = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{91}$$

$$\mathrm{Var}[\langle r_i, r_j \rangle] = \begin{cases} 2/n & \text{when } i = j, \\ 1/n & \text{otherwise.} \end{cases} \tag{92}$$

The above equations combined with Chebyshev's inequality complete the proof.

□

Note that we regard k as a constant and hide it in the $O(\cdot)$ notation. We now study the generation of the random porcupine in greater detail.

Definition 10. Let R be the matrix whose columns are $r_0, \ldots, r_{k-1}$ and let G be the Cholesky factorization of $R^T R$, i.e., G is an upper triangular matrix such that $G^T G = R^T R$. (By construction, $R^T R$ is positive definite with probability one, and thus a unique G exists with probability one.) Define the matrix S by $S = RG^{-1}$.

Since the matrix S constructed in Definition 10 is an orthonormal $(n \times k)$-matrix and the matrix G used to construct S is upper triangular, multiplying R by G^{-1} from the right is equivalent to performing a Gram-Schmidt orthogonalization of the random vectors $r_0, \ldots, r_{k-1}$. Thus, the vectors $s_0, \ldots, s_{k-1}$, forming the columns of S, constitute a random porcupine.

Lemma 17. *Suppose that*

$$\bigl|\langle r_j, r_\ell\rangle - \mathrm{E}[\langle r_j, r_\ell\rangle]\bigr| \le \varepsilon \tag{93}$$

for all j, ℓ. Then all elements of $G - I$ are $O(\varepsilon)$.

Proof. Since the Cholesky factorization is unique for symmetric positive definite matrices, it follows from the factorization algorithm [6, Algorithm 5.2-1] that $|G_{jj} - 1| \in O(\varepsilon)$, and $|G_{j\ell}| \in O(\varepsilon)$ when $j \neq \ell$. □

Corollary 18. *Construct the random vectors $r_0, \ldots, r_{k-1}$ by choosing the components of each vector as independent $\mathrm{N}(0, 1/\sqrt{n})$ random variables. Construct the vectors $s_0, \ldots, s_{k-1}$ by performing a Gram-Schmidt orthogonalization of the vectors $r_0, \ldots, r_{k-1}$. Let v be any vector in $\boldsymbol{R}^n$, and v_r be the projection of v into the subspace spanned by the vectors $r_0, \ldots, r_{k-1}$. With probability at least*

$1 - O(1/n\varepsilon^2)$ *over the choice of* $r_0, \dots, r_{k-1}$,

$$|\langle v, s_j - r_j \rangle| < \|v_r\| O(\varepsilon). \tag{94}$$

Proof. Let e_j be the k-dimensional vector with zeros in all components but the jth. Then

$$\|s_j - r_j\| = \|S(I - G)e_j\| = \|(I - G)e_j\| \in O(\varepsilon), \tag{95}$$

since, by Lemma 17, all elements of $I - G$ are $O(\varepsilon)$. □

The second important property of the rounding procedure is that the probability of a "photo finish" in the rounding procedure is small.

Lemma 19. *Let* v *be any vector in* $\boldsymbol{R}^n$ *and* v_r *be the projection of* v *into the subspace spanned by the vectors* $r_0, \dots, r_{k-1}$. *Then,*

$$\Pr[|\langle v, s_j - s_\ell \rangle| < \|v_r\| \delta] \in O(\delta). \tag{96}$$

Proof. By construction, the vectors $s_0, \dots, s_{k-1}$ are orthogonal unit length vectors with random orientation. Thus, we can instead view the situation as follows: We select a random unit length k-dimensional vector w, and compute the probability that

$$|\langle w, s \rangle| \in O(\varepsilon), \tag{97}$$

where $s = s_j - s_\ell$. But this probability is $O(\varepsilon)$ for any k-dimensional vector s of constant length. □

Corollary 20. *The probability that the edge* (i, i') *is not cut can be written as*

$$\sum_{j=0}^{k-1} \Pr\left[\bigcap_{\substack{\ell=0 \\ \ell \neq j}}^{k-1} \left\{ \langle v^i, r_j \rangle \geq \langle v^i, r_\ell \rangle \right\} \cap \bigcap_{\substack{\ell=0 \\ \ell \neq j}}^{k-1} \left\{ \langle v^{i'}, r_j \rangle \geq \langle v^{i'}, r_\ell \rangle \right\} \right]. \tag{98}$$

Suppose that

$$\big|\langle r_j, r_\ell\rangle - \mathrm{E}[\langle r_j, r_\ell\rangle]\big| \le \varepsilon \tag{99}$$

for all j, ℓ. Given that (i, i') is not cut, the probability that the above inequalities hold with a margin of at least $\|v_r^i\|O(\varepsilon)$ and $\|v_r^{i'}\|O(\varepsilon)$, respectively, is $1 - O(\varepsilon)$.

Proof. By Corollary 18, $\langle v, r_j\rangle$ and $\langle v, s_j\rangle$ differ by at most $\|v_r\|O(\varepsilon)$, and by Corollary 19

$$\Pr[|\langle v, s_j - s_\ell\rangle| < \|v_r\|\delta] \in O(\delta). \tag{100}$$

If we select $\delta \in O(\varepsilon)$ this completes the proof, since there is only a constant number of inequalities in (98). □

We can now fit the pieces together.

Theorem 21. *The probability of the edge (i, i') being cut when the solution to the semidefinite program of Frieze and Jerrum is rounded using k random vectors differs by a factor $1 + O(n^{-1/3})$ from the probability of it being cut when a random orthogonal porcupine is used in the rounding.*

Proof. It follows from Corollaries 18 and 20 that the probability that the edge (i, i') is cut when the rounding procedure uses $s_0, \dots, s_{k-1}$ differs from the probability that it is cut when the rounding procedure uses $r_0, \dots, r_{k-1}$ by a factor

$$1 - O\big(1/n\varepsilon^2\big) - O\big(\varepsilon\big). \tag{101}$$

If we choose $\varepsilon = n^{-1/3}$ this factor is $1 - O(n^{-1/3})$. □

4.2 Using porcupines to represent variables

Traditionally, the analysis of the performance ratio of semidefinite programming based approximation algorithms is done using local analysis. In our case this corresponds to finding the worst possible configuration of two porcupines (or vectors).

Theorem 22. *Consider the edge (i,i'). For each configuration of vectors v^i and $v^{i'}$ from Frieze and Jerrum's semidefinite program there exists a configuration of simplicial porcupines $\{u^i_j\}_{j=0}^{k-1}$ and $\{u^{i'}_j\}_{j=0}^{k-1}$ such that the ratio between the probability of the edge being cut after rounding and the corresponding term in the objective function is the same for the two configurations.*

Corollary 23. *Using local analysis, the performance ratio of the porcupine algorithm for* Max k-Cut *is no less than that obtained by Frieze and Jerrum.*

Proof of Theorem 22. We can without restriction choose coordinate system in such a way that

$$v^i = (1, 0, \ldots), \tag{102}$$

$$v^{i'} = (\lambda, \sqrt{1-\lambda^2}, 0, \ldots), \tag{103}$$

where $\lambda \geq -1/(k-1)$. Let $w_j \in \boldsymbol{R}^{k-1}$, $j = 0, \ldots, k-1$, be the vertices of a regular k-simplex with $\|w_j\| = 1$. Suppose that w_j has the coordinates $w_j = (w_{j,1}, \ldots, w_{j,k-1})$, and consider a simplicial porcupine $\{u^i_j\}_{j=0}^{k-1}$ which we wish to put in correspondence with v^i. Let L_i be the $(k-1)$-dimensional subspace spanned by $\{u^i_j\}_{j=0}^{k-1}$. By symmetry, we can assume that the coordinates of u^i_j in L_i are $(w_{j,1}, \ldots, w_{j,k-1})$. We construct another simplicial porcupine $\{u^{i'}_j\}_{j=0}^{k-1}$ (corresponding to $v^{i'}$) with the following properties. Let $L_i^{\perp} = L_{i'} - L_i$. Denote with $\pi_L(v)$ the projection of v onto the subspace L. Then $u^{i'}_j$ can be assumed to have the coordinates $\sqrt{1-\lambda^2}(w_{j,1}, \ldots, w_{j,k-1})$ in $L_i^{\perp}$ (again by symmetry) and satisfy $\pi_{L_i}(u^{i'}_j) = \lambda u^i_j$. We note that $\{u^i_j\}_{j=0}^{k-1}$ and $\{u^{i'}_j\}_{j=0}^{k-1}$ satisfy the constraints (16a) and (16c).

If we scale each random vector by a factor $1/\sqrt{n}$, we can view the rounding scheme of Frieze and Jerrum as if it chooses k random vectors $r_0, \ldots, r_{k-1}$ where the components of each r_j are independent $\mathrm{N}(0, 1/\sqrt{n})$ variables. This process is equivalent to choosing a random variable from the kn-dimensional normal distribution with mean zero and covariance matrix $\frac{1}{n}I$, where I denotes the unit matrix.

Consider the following way to generate the random vectors $s_0, \ldots, s_{k-1}$:

$$s_j = \sqrt{\frac{k-1}{k}} \sum_{\ell=1}^{k-1} w_{j,\ell} t_\ell + \sqrt{\frac{1}{k}} t_0 \tag{104}$$

where the components of each t_j, $j = 0, \ldots, k-1$, are independent $N(0, 1/\sqrt{n})$. Denote with $s_{j,m}$ the mth component of s_j, $0 \le j \le k-1$ and $1 \le m \le n$. Then $s_{j,m} \in N(0, 1/\sqrt{n})$ for all j and m. Furthermore,

$$\mathrm{E}[s_{j,m} s_{j',m'}] = \begin{cases} 1/n & \text{when } j = j' \text{ and } m = m', \\ 0 & \text{otherwise.} \end{cases} \tag{105}$$

Therefore the $s_{j,m}$ variables can be viewed as the components of a single random variable with the kn-dimensional normal distribution with mean zero and covariance matrix $\frac{1}{n}I$. This implies that rounding using the random vectors $s_0, \ldots, s_{k-1}$ is equivalent to rounding using the vectors $r_0, \ldots, r_{k-1}$.

Using the same techniques as in the proof of Theorem 3, it can be shown that we instead of the random vectors defined in (104) can perform the randomized rounding using the vectors

$$s'_j = \sum_{\ell=1}^{k-1} w_{j,\ell} t_\ell \tag{106}$$

for $j = 0, \ldots, k-1$. We let $t_\ell = (\xi_\ell, \zeta_\ell, \ldots)$ where $\xi_\ell, \zeta_\ell \in \mathrm{N}(0, 1/\sqrt{n})$ for all l. The rest of the coordinates are $\mathrm{N}(0, 1/\sqrt{n})$ as well but are not used in the calculations below.

Let us now compute the probability of the edge being cut using the approach of Frieze and Jerrum. Let A_j^i be the event that $\langle v^i, s_0' \rangle \geq \langle v^i, s_j' \rangle$. Then,

$$\begin{aligned} &\Pr[(i,i') \text{ is not cut}] = \\ &\quad = k \times \Pr[x_i \leftarrow 0 \text{ and } x_{i'} \leftarrow 0] = \\ &\quad = k \times \Pr\left[\bigcap_{j=1}^{k-1} \left(A_j^i \cap A_j^{i'}\right)\right]. \end{aligned} \tag{107}$$

Equations (102), (103) and (106) immediately imply that

$$A_j^i \iff \sum_{\ell=1}^{k-1} (w_{0,\ell} - w_{j,\ell})\xi_\ell \geq 0, \tag{108}$$

$$\begin{aligned} A_j^{i'} \iff &\lambda \sum_{\ell=1}^{k-1} (w_{0,\ell} - w_{j,\ell})\xi_\ell + \\ &\sqrt{1-\lambda^2} \sum_{\ell=1}^{k-1} (w_{0,\ell} - w_{j,\ell})\zeta_\ell \geq 0. \end{aligned} \tag{109}$$

Finally, we focus on the randomized rounding used to obtain a cut from a configuration of porcupines. The random vector r used in the rounding can be assumed to satisfy

$$\pi_{L_i}(r) = (\xi_1, \xi_2, \ldots, \xi_{k-1}) \tag{110}$$

$$\pi_{L_i^\perp}(r) = (\zeta_1, \zeta_2, \ldots, \zeta_{k-1}) \tag{111}$$

where $\xi_i, \zeta_i \in \mathrm{N}(0,1)$ for all i. Let B_j^i be the event that $\langle u_0^i, r \rangle \geq \langle u_j^i, r \rangle$. Then,

$$\begin{aligned} &\Pr[(i,i') \text{ is not cut}] = \\ &\quad = k \times \Pr[x_i \leftarrow 0 \text{ and } x_{i'} \leftarrow 0] = \\ &\quad = k \times \Pr\left[\bigcap_{j=1}^{k-1} \left(B_j^i \cap B_j^{i'}\right)\right]. \end{aligned} \tag{112}$$

Equations (110) and (111) imply that

$$B_j^i \iff \sum_{\ell=1}^{k-1}(w_{0,\ell}-w_{j,\ell})\xi_\ell \geq 0, \tag{113}$$

$$\begin{aligned} B_j^{i'} \iff &\lambda\sum_{\ell=1}^{k-1}(w_{0,\ell}-w_{j,\ell})\xi_\ell + \\ &\sqrt{1-\lambda^2}\sum_{\ell=1}^{k-1}(w_{0,\ell}-w_{j,\ell})\zeta_\ell \geq 0, \end{aligned} \tag{114}$$

which shows that the probability of the edge being cut is indeed identical in both cases.

To finish the proof, we just note that the corresponding terms in the objective functions in both cases evaluate to $\frac{k-1}{k}(1-\lambda)$. □

We cannot conclude that the performance ratios are the same as there might exist porcupine configurations which cannot be put in correspondence with feasible solutions to (10). Also, the configurations used in the above proof might not be optimal for the semidefinite program. Using local analysis, we have obtained numerical evidence that the performance ratios are indeed the same, but we have not been able to prove it formally.

5 Negative results

In this section we show that there exists a universal constant, such that it is **NP**-hard to approximate Max E2-Lin mod p within that constant. We do this by reducing Max E3-Lin mod p to Max E2-Lin mod p. This reduction is valid for all primes $p \geq 3$. However, since the quality of this reduction deteriorates when p increases we cannot use it to show that it is **NP**-hard to approximate Max E2-Lin mod p within a constant factor independent of p. To deal with this, we also make a reduction from Max E3-Lin mod 2 to Max E2-Lin mod p. While this reduction is only valid for large enough p, it guarantees that it is **NP**-hard to approximate Max E2-Lin mod p within a constant factor independent of p. The two reductions combined then give the desired result.

The reductions we construct are through *gadgets* which are local reduction taking one constraint of the original problem and constructing one or more constraints, possibly using weights, of the target problem. These constrains use the variables of the original problem and also possibly some new auxiliary variables. A gadget is an α-*gadget* if whenever the original constraint is satisfied, then the auxiliary variables can be adjusted to satisfy constraints of total weight exactly α while if the constraint is not satisfied than the obtainable maximum is exactly $\alpha - 1$ [9]. The usefulness of gadgets follows from the below theorem.

Theorem 24. *Suppose there is an α-gadget reducing constraint satisfaction problem A to constraint satisfaction problem B, and that is **NP**-hard to, given an instance of A of total weight w, distinguish instances where the maximal weight of simultaneous satisfiable constraints is at least $a_1 w$ and and when it is at most $a_2 w$. Then it is **NP**-hard to approximate B within a factor*

$$\frac{\alpha + a_1 - 1}{\alpha + a_2 - 1}. \tag{115}$$

Proof. This theorem is standard so let us only sketch the proof. By the properties of a gadget an instance of A where the maximal simultaneous satisfiable weight is aw is transformed into an instance of B where the maximal simultaneous satisfiable weight is $(\alpha + a - 1)w$. The theorem follows directly from this. □

5.1 Small p

For the case $p = 2$, it is possible to use the methods of Trevisan et al. [9] to construct a gadget reducing Max E3-Lin mod 2 to Max E2-Lin mod 2. When this gadget is combined with the hardness results by Håstad [7], it follows, as described in [7], that it is **NP**-hard to approximate Max E2-Lin mod 2 within $12/11 - \varepsilon$. We now show how to construct a gadget which can be used to show hardness results for Max E2-Lin mod p when $p \geq 3$. Note, that although

Trevisan et al. [9] have constructed an algorithm which computes optimal gadgets, this algorithm seems to be on no use to us to construct the gadgets for $p \geq 3$; the running time of the algorithm is simply too large to run it in on existing computers.

We start with an instance of Max E3-Lin mod p of total weight w. For each equation in the instance we construct a number of equations with two variables per equation. By the result of Håstad [7], for any $\varepsilon > 0$ it is **NP**-hard to distinguish the case when the maximal simultaneous satisfiable weight is $(1-\varepsilon)w$ and when it is $(1+\varepsilon)w/p$. This theorem applies to the case when all coefficients in the equations are equal to one. Thus, we can assume that, for all i, the ith equation in the Max E3-Lin mod p instance is of the form

$$x_{i_1} + x_{i_2} + x_{i_3} = c. \tag{116}$$

For an arbitrary equation of this form we now construct the corresponding equations in the Max E2-Lin mod p instance. Consider assignments to the variables x_{i_1}, x_{i_2}, and x_{i_3} with the property that $x_{i_1} = 0$. There are p^2 such assignments, and p of those are satisfying. For each of the $p^2 - p$ unsatisfying assignments

$$(x_{i_1}, x_{i_2}, x_{i_3}) \leftarrow (0, a, b) \quad a + b \neq c \tag{117}$$

we introduce a new auxiliary variable $y_{i,a,b}$ and construct the following triple of equations:

$$x_{i_1} - y_{i,a,b} = 0, \tag{118a}$$

$$x_{i_2} - y_{i,a,b} = a, \tag{118b}$$

$$x_{i_3} - (p-2)y_{i,a,b} = b. \tag{118c}$$

The variable $y_{i,a,b}$ is an auxiliary variable that appears only in this triple of equations. Our Max E2-Lin mod p instance contains $3m(p^2 - p)$ equations if the Max E3-Lin mod p instance contains m equations.

Lemma 25. *When $p \geq 3$ is prime, the above construction is a $(p-1)(p+3)$-gadget.*

Proof. Let π be an assignment to the x_i and the $y_{i,a,b}$, such that the number of satisfied equations in the Max E2-Lin mod p instance is maximized. Since each fixed $y_{i,a,b}$ occurs only in three equations, we can assume that $\pi(y_{i,a,b})$ is such that as many as possible of these three equations are satisfied. We now study some arbitrary equation

$$x_{i_1} + x_{i_2} + x_{i_3} = c \tag{119}$$

from the Max E3-Lin mod 2 instance, and the corresponding $3(p^2-p)$ equations of type (118) from the Max E2-Lin mod p instance. Assume that

$$\pi(x_{i_1}, x_{i_2}, x_{i_3}) = (s_1, s_2, s_3), \tag{120}$$

and that (s_1, s_2, s_3) satisfies (119). Then, for arbitrary a and b such that $a+b \neq c$ there is no assignment to $y_{i,a,b}$ such that all corresponding equations (118) containing $y_{i,a,b}$ are satisfied. For, if we sum the three equations in a triple, the left hand side becomes $s_1 + s_2 + s_3$ and the right hand side $a+b$. If all equations in the triple (118) were satisfied, then this new equation would also be satisfied. But $a + b \neq c$ by construction, which contradicts this assumption. We can, however, always satisfy one of the three equations containing $y_{i,a,b}$ by choosing $\pi(y_{i,a,b}) = \pi(x_{i_1})$. In some cases it is possible to satisfy two of the three equations. In fact, exactly $3(p-1)$ of the p^2-p triples of type (118) have this property.

To see this, remember that each triple (118) corresponds to an assignment which do not satisfy (119). There are exactly $3(p-1)$ ways to construct unsatisfying assignments (u_1, u_2, u_3) with the property that (s_1, s_2, s_3) and (u_1, u_2, u_3)

differ in exactly one position. Such an assignment corresponds to the triple

$$x_{i_1} - y_{i,a,b} = 0, \tag{121a}$$

$$x_{i_2} - y_{i,a,b} = u_2 - u_1, \tag{121b}$$

$$x_{i_3} - (p-2)y_{i,a,b} = u_3 - (p-2)u_1. \tag{121c}$$

With the assignment $\pi(y_{i,a,b}) = u_1$, all of the equations are satisfied in the assignment (u_1, u_2, u_3) and since (s_1, s_2, s_3) and (u_1, u_2, u_3) differ in exactly one position, the assignment (s_1, s_2, s_3) must satisfy two equations. Furthermore, two different unsatisfying assignments (u_1, u_2, u_3) and (u_1', u_2', u_3'), both with the property that they differ from the satisfying assignment in exactly one position, can never correspond to the same triple. For, if that were the case, the equations

$$u_2 - u_1 = u_2' - u_1' \tag{122}$$

$$u_3 - (p-2)u_1 = u_3' - (p-2)u_1' \tag{123}$$

$$u_k = u_k' \quad \text{for some } k \in \{1,2,3\} \tag{124}$$

would have to be simultaneously satisfied. This, however, implies that $u_k = u_k'$ for all k. Summing up, the contribution to the objective function in the Max E2-Lin mod p instance is

$$2 \times 3(p-1) + \big((p^2 - p) - 3(p-1)\big) = (p-1)(p+3). \tag{125}$$

Let us now assume that (s_1, s_2, s_3) does not satisfy (119). Then for exactly one triple, namely $a = s_2 - s_1$ and $b = s_3 - (p-2)s_1$, all three equations containing $y_{i,a,b}$ can be satisfied. By a similar argument as above, triples where exactly two equations can be satisfied are in one-to-one correspondence with not satisfying triplets (u_1, u_2, u_3) differing from (s_1, s_2, s_3) in exactly one coordinate. There are exactly $3(p-2)$ such triples, and in the remaining triples one equation can be satisfied. The contribution to the objective function in the Max E2-Lin

MOD p is

$$\begin{aligned}&3+2\times 3(p-2)+\bigl((p^2-p)-(3(p-2)+1)\bigr)\\&\quad=(p-1)(p+3)-1.\end{aligned}\tag{126}$$

□

Theorem 26. *For all $\varepsilon > 0$ and all $p \geq 3$, it is **NP**-hard to approximate* MAX E2-LIN MOD p *within* $(p^2+3p)/(p^2+3p-1)-\varepsilon$.

Proof. This follows from the above quoted result of Håstad [7], Lemma 25, and Theorem 24. □

5.2 Large p

Recently, Håstad showed that it is **NP**-hard to approximate MAX E3-LIN MOD 2 within $2-\varepsilon$ for any constant $\varepsilon > 0$ [7]. In his proof, Håstad constructs a PCP which in each round reads from the proof three bits, b_f, b_{g_1} and b_{g_2}, where f, g_1 and g_2 are functions. The equations constructed in the instance are of the form $x_f + x_{g_1} + x_{g_2} = \{0,1\}$. For each equation, f and g_1 are chosen independently at random, and then g_2 is defined pointwise, in such a way that $f(x) + g_1(x) = g_2(x)$ with probability $1-\varepsilon$.

In our construction, we encode such an equation as a number of equations with two variables in each equation. Let θ be a number mod p. (We will need some properties of θ later.) In our MAX E2-LIN MOD p instance, we have the new variables y_g, y_{g_1,g_2} and y_f. We arrange things so that they take values $x_g(1+\theta)$, $x_{g_1}+\theta x_{g_2}$ and x_f, respectively. Since the equations $x_f+x_{g_1}+x_{g_2}=0$ and $x_f + x_{g_2} + x_{g_1} = 0$ are satisfied simultaneously we can assume that y_{g_1,g_2} and y_{g_2,g_1} appear in the same type of equations with the same probabilities.

Definition 11. Denote by w_{g_1,g_2} the total weight of the equations containing x_{g_1} and x_{g_2} (in this order) and by w_f the total weight of the equations containing x_f. Also, let the total weight of all equations containing x_g be w_g and the total weight of all equations be w.

Remark 4. Since each equation contains two g variables and one f variable, $2w = \sum_g w_g$ and $w = \sum_f w_f$. Also, since each g can be either the first or the last entry, $w_g = 2\sum_{g_2} w_{g,g_2}$.

We now construct the equations in our instance of Max E2-Lin mod p. The variable z used below should be thought of as zero, it is included merely to produce a Max E2-Lin mod p instance. First, for any g we have the equations

$$y_g - z = h \quad \text{for } h \in \{0, 1 + \theta\}, \tag{127}$$

each with weight w_g. To make sure that coding of pairs is done correctly we add equations

$$y_{g_1,g_2} - y_{g_1} = h \quad \text{for } h \in \{0, \pm\theta\}, \tag{128}$$

and

$$y_{g_1,g_2} - y_{g_2} = h \quad \text{for } h \in \{0, \pm 1\}, \tag{129}$$

each with weight w_{g_1,g_2}. To make sure that x_f is coded in a legal way we have equations

$$y_f - z = h \quad \text{for } h \in \{0, 1\}, \tag{130}$$

each with weight w_f. Finally, we include equations corresponding to the original equation from the Max E3-Lin mod 2 instance. Every such equation has the same weight as the original equation. If the original equation is of the form $x_f + x_{g_1} + x_{g_2} = 0$, we include the equations

$$y_f - y_{g_1,g_2} = h \quad \text{for } h \in \{0, \pm 1 - \theta\}. \tag{131}$$

If the right-hand side of the original equation is 1, we use the right-hand sides

$h \in \{\pm 1, -\theta\}$ instead in (131).

The only property from θ that we need is that variables that satisfy an equation of type (127) or (130) do not satisfy other equations for "incorrect" reasons. It turns out to be sufficient that the numbers $h_1 + h_2\theta$ for $h_1, h_2 \in \{0, \pm 1\}$ are all distinct and thus we can set $\theta = 3$ which works for any $p \geq 9$. Let us now turn to analyzing the set of equations.

As mentioned above, the variable z should be thought of as zero. By the following lemma, it can actually never be optimal to have $z \neq 0$.

Lemma 27. *There always exists an optimal solution to the systems of linear equations with* $z = 0$.

Proof. Suppose that $z = c \neq 0$ in the optimal solution. Since all equations are of the form $x - y = k$, the following transformation does not make any satisfied equation unsatisfied: $y_f \leftarrow y_f - c$, $y_g \leftarrow y_g - c$, $y_{g_1,g_2} \leftarrow y_{g_1,g_2} - c$ and $z \leftarrow 0$. □

By Lemma 27, we can assume that $z = 0$ in the optimal solution. We will implicitly use this assumption in the following two lemmas, which show that it is always optimal to encode y_f and y_g correctly.

Lemma 28. *For each* f, y_f *is always either* 0 *or* 1 *in an optimal solution.*

Proof. Each variable y_f appears in equations of total weight $5w_f$. If y_f is either 0 or 1, the weight of all satisfied equations containing y_f is at least w_f, otherwise this weight is at most w_f (only one of type (131) for each left-hand side). Thus we can assume that an optimal solution has y_f equal to 0 or 1. □

Lemma 29. *For each* g, y_g *is always either* 0 *or* $1 + \theta$ *in an optimal solution.*

Proof. If y_g is either 0 or $1+\theta$, the weight of all satisfied equations containing y_g is at least w_g, otherwise this weight is at most $\sum_{g_2} w_{g,g_2} + w_{g_2,g} = w_g$ (only one of each of the types (128) and (129) for each left-hand side). Thus we can assume that an optimal solution has y_g equal to 0 or $1 + \theta$. □

In view of Lemma 29 we can write the value of y_g in the optimal assignment as $x_g(1 + \theta)$ for $x_g \in \{0, 1\}$. Next we have

Lemma 30. *For all g_1 and g_2, $y_{g_1,g_2} = x_{g_1} + x_{g_2}\theta$ in an optimal solution.*

Proof. Consider the equations containing y_{g_1,g_2}. If we satisfy one equation of each of the types (128) and (129) then at least equations of weight $2w_{g_1,g_2}$ are satisfied while otherwise the weight is at most $2w_{g_1,g_2}$ (namely at most one of those two equations and all equations of type (131)). Now assume that the two equations are satisfied with right hand sides $h_1\theta$ and h_2, respectively where $h_1, h_2 \in \{0, \pm 1\}$. Subtracting the two equations yield $(x_{g_1} - x_{g_2})(1 + \theta) = h_2 - h_1\theta$. For each value of $x_{g_1} - x_{g_2}$ we have one "natural" solution $h_1 = -h_2 = x_{g_2} - x_{g_1}$ and by the assumption on θ there are no other possibilities. Clearly this natural solution corresponds to the value of y_{g_1,g_2} stated in the lemma. □

We are now ready to prove the main theorem.

Theorem 31. *When $p \geq 11$, it is **NP**-hard to approximate* Max E2-Lin mod p *within* $12/11 - \varepsilon$ *for all* $\varepsilon > 0$.

Proof. By Lemmas 28, 29 and 30, we know that the optimal assignment is given by a correct encoding. It then satisfies equations of type (127) with a total weight of

$$\sum_g w_g = 2w, \tag{132}$$

and equations of type (130) with a total weight of

$$\sum_f w_f = w, \tag{133}$$

and equations of types (128) and (129) each of weight w. Thus, if the corresponding assignment to the binary variables satisfies equations of weight t, we satisfy equations of total weight $5w + t$ in our transformed case. By the result of Håstad [7] it is **NP**-hard to distinguish the cases when t is $w - \varepsilon_1$ and when t is $w/2 + \varepsilon_2$ for arbitrarily small $\varepsilon_1, \varepsilon_2 > 0$, whence it follows that it is **NP**-hard to approximate Max E2-Lin mod p within $12/11 - \varepsilon$ for any $\varepsilon > 0$. □

When we combine this result with the results for small p, we obtain the following general result:

Theorem 32. *For all primes p, it is **NP**-hard to approximate* Max E2-Lin mod p *within* $70/69 - \varepsilon$.

Proof. For $p = 2$ we use the hardness result by Håstad [7]. For $p \in \{3, 5, 7\}$ we use Theorem 26, and for $p \geq 11$ we use Theorem 31. □

6 Conclusions

We have shown that there exists a randomized polynomial time algorithm approximating Max 2-Lin mod p within $p - \Theta(p^{-12})$. For the special case of Max 2-Lin mod p, where the equations are either of the form $x_i - x_{i'} = c$ or $x_i = c$, we have shown that there exists a randomized polynomial time algorithm approximating the problem within $(1 - 10^{-8})p$. We have numerical evidence that the performance ratio in the latter, simpler case is actually 1.27 when $p = 3$. In fact, we have not tried to find the tightest possible inequalities in our proofs; our primary goal was to show a performance ratio less than p. Most likely, our bounds can be improved significantly.

We have also shown that it is **NP**-hard to approximate Max E2-Lin mod p within $70/69 - \varepsilon$. Of major interest at this point is, in our opinion, to determine if the lower bounds are in fact increasing with p, or if there exists a polynomial time algorithm approximating Max 2-Lin mod p within some constant ratio.

7 Acknowledgments

We are most grateful to Madhu Sudan for helpful discussions on these subjects. Ron Rivest raised the question of an algorithm for arbitrary finite fields. We also thank two anonymous referees for many detailed comments helping the presentation and pin-pointing some inaccuracies.

References

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

[2] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* John Wiley & Sons, New York, 1991.

[3] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley & Sons, New York, second edition, 1962.

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

[5] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[6] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* North Oxford Academic Publishing, Oxford, 1983.

[7] Johan Håstad. Some optimal inapproximability results. In *Proceedings of Twenty-ninth Annual ACM Symposium on Theory of Computing*, pages 1–10, El Paso, Texas, May 1997. ACM Press. Accepted for publication in *Journal of the ACM.*

[8] David G. Luenberger. *Linear and Nonlinear Programming.* Addison Wesley, Reading, second edition, 1973.

[9] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.

A Four lemmas from elementary probability theory.

Let

$$\varphi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}} \tag{134}$$

be the density function of a normal distribution with standard deviation 1 and let

$$\Phi(x) = \int_{-\infty}^{x} \varphi(t)\, dt. \tag{135}$$

As is well known [3], we have

$$\varphi(x)\left(\frac{1}{x} - \frac{1}{x^3}\right) < 1 - \Phi(x) < \frac{\varphi(x)}{x}, \tag{136}$$

when $x > 0$. This bound will be used to prove the following lemmas.

Lemma 33. *Let $X_0, \dots, X_{p-1}$ be independent identically distributed* $\mathrm{N}(0,1)$ *random variables. Denote the maximum of the X_i by $X_{(p)}$, and the second maximum by $X_{(p-1)}$. Then, for any $\delta > 0$,*

$$\begin{aligned}\Pr\Big[X_{(p)} \geq (1+\delta)\sqrt{2\ln p} \bigcap X_{(p-1)} \leq (1+\delta/2)\sqrt{2\ln p}\Big] \\ > \frac{1}{2p^{2\delta+\delta^2}(1+\delta)\sqrt{\pi\ln p}}\left(1 - \frac{1}{2\ln p} - \frac{1}{2p^{\delta}\sqrt{\pi\ln p}}\right).\end{aligned} \tag{137}$$

Proof. Since the X_i are i.i.d. $\mathrm{N}(0,1)$, we know that

$$\Pr[X_{(p)} \geq x \cap X_{(p-1)} \leq y] = p(1-\Phi(x))\Phi(y)^{p-1} \tag{138}$$

when $x \geq y$. We now apply the bound (136) on $\Phi(x)$. This bound, together with the fact that $\delta > 0$, implies that

$$\begin{aligned} 1 - \Phi&\left((1+\delta)\sqrt{2\ln p}\right) \\ &> \frac{1}{\sqrt{2\pi} p^{(1+\delta)^2}}\left(\frac{1}{(1+\delta)\sqrt{2\ln p}} - \frac{1}{(1+\delta)^3(2\ln p)^{3/2}}\right) \\ &> \frac{1}{2p^{1+2\delta+\delta^2}(1+\delta)\sqrt{\pi\ln p}}\left(1 - \frac{1}{2\ln p}\right), \end{aligned} \tag{139}$$

and that

$$\begin{aligned} \Phi\left((1+\delta/2)\sqrt{2\ln p}\right) &> 1 - \frac{1}{\sqrt{2\pi} p^{(1+\delta/2)^2}(1+\delta/2)\sqrt{2\ln p}} \\ &> 1 - \frac{1}{2p^{1+\delta}\sqrt{\pi\ln p}}. \end{aligned} \tag{140}$$

Using $(1-a)^b \geq 1 - ab$, valid for positive a and $b > 1$ we see that

$$\left(1 - \frac{1}{2p^{1+\delta}\sqrt{\pi\ln p}}\right)^{p-1} \geq 1 - \frac{1}{2p^{\delta}\sqrt{\pi\ln p}}. \tag{141}$$

Inserting the derived bounds into (138), using $(1-x)(1-y) \geq 1 - x - y$, the lemma follows. □

Lemma 34. *Let X and Z be i.i.d.* $\mathrm{N}(0,1)$ *and $\varepsilon \in [0,1]$. Then, for any $\delta > 0$,*

$$\begin{aligned} \Pr&\left[\left|\left(1 - \sqrt{1-\varepsilon}\right)X - \sqrt{\varepsilon}Z\right| > \frac{\delta}{4}\sqrt{2(1-\varepsilon)\ln p}\right] \\ &\leq \frac{4p^{-\delta^2(1-\varepsilon)/32\varepsilon}}{\delta}\sqrt{\frac{2\varepsilon}{(1-\varepsilon)\pi\ln p}}. \end{aligned} \tag{142}$$

Proof. Let $W = (1 - \sqrt{1-\varepsilon})X - \sqrt{\varepsilon}Z$. Since X and Z are independent, $W \in \mathrm{N}(0,\sigma)$, where

$$\sigma = \sqrt{\left(1 - \sqrt{1-\varepsilon}\right)^2 + \varepsilon} \leq \sqrt{2\varepsilon}. \tag{143}$$

Since $\Pr[|W| > w] = 2(1 - \Phi(w/\sigma))$, we can use the bound (136).

$$\begin{aligned}\Pr\left[|W| > \frac{\delta}{4}\sqrt{2(1-\varepsilon)\ln p}\right] &= 2\left(1 - \Phi\left(\frac{\delta}{4\sigma}\sqrt{2(1-\varepsilon)\ln p}\right)\right) \\ &\leq 2\frac{4\sigma}{\delta\sqrt{2(1-\varepsilon)\ln p}} \times \frac{p^{-\delta^2(1-\varepsilon)/16\sigma^2}}{\sqrt{2\pi}} \qquad (144) \\ &\leq \frac{4p^{-\delta^2(1-\varepsilon)/32\varepsilon}}{\delta}\sqrt{\frac{2\varepsilon}{(1-\varepsilon)\pi\ln p}}.\end{aligned}$$

□

Lemma 35. *Let* $X_0, \ldots, X_{p-1}$ *be i.i.d.* $\mathrm{N}(0,1)$ *random variables. Denote the maximum of the* X_i *by* $X_{(p)}$*, and the second maximum by* $X_{(p-1)}$*. Then*

$$\Pr\Big[X_{(p)} > X_{(p-1)} + \delta\Big] > 1 - \frac{p^2\delta}{(p-1)\sqrt{2\pi}}. \qquad (145)$$

Proof. Since the X_i are independent,

$$\Pr\Big[X_{(p)} > X_{(p-1)} + \delta\Big] = p \times \Pr\left[\bigcap_{i=1}^{p-1} X_0 > X_i + \delta\right]. \qquad (146)$$

To compute the latter probability we condition on X_0.

$$\Pr\left[\bigcap_{i=1}^{p-1} X_0 > X_i + \delta\right] = \int_{-\infty}^{\infty} \Phi^{p-1}(x-\delta)\varphi(x)\,dx. \qquad (147)$$

To bound $\Phi^{p-1}(x-\delta)$, we use the mean value theorem. (In the following equations, $\xi \in [x-\delta, x]$.)

$$\begin{aligned}\Phi^{p-1}(x-\delta) &= \big(\Phi(x) - \delta\varphi(\xi)\big)^{p-1} \\ &\geq \Phi^{p-1}(x) - p\delta\varphi(\xi)\Phi^{p-2}(x) \qquad (148) \\ &\geq \Phi^{p-1}(x) - \frac{p\delta}{\sqrt{2\pi}}\Phi^{p-2}(x).\end{aligned}$$

From this bound on $\varphi(x)$, we obtain

$$\begin{aligned}
&\int_{-\infty}^{\infty} \Phi^{p-1}(x-\delta)\varphi(x)\,dx \\
&\geq \int_{-\infty}^{\infty} \Phi^{p-1}(x)\varphi(x)\,dx - \frac{p\delta}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \Phi^{p-2}(x)\varphi(x)\,dx \\
&= \frac{1}{p} - \frac{p\delta}{(p-1)\sqrt{2\pi}}.
\end{aligned} \tag{149}$$

□

Lemma 36. *Let X and Z be i.i.d.* $\mathrm{N}(0,1)$ *and* $\varepsilon \in [0,1]$*. Then, for any* $\delta > 0$*,*

$$\Pr\Big[\Big|\Big(1-\sqrt{1-\varepsilon}\Big)X - \sqrt{\varepsilon}Z\Big| > \delta/2\Big] \leq \frac{4}{\delta}\sqrt{\frac{\varepsilon}{\pi}}. \tag{150}$$

Proof. Since X and Z are independent,

$$\Big(1-\sqrt{1-\varepsilon}\Big)X - \sqrt{\varepsilon}Z \in \mathrm{N}(0,\sigma), \tag{151}$$

where

$$\sigma = \sqrt{\Big(1-\sqrt{1-\varepsilon}\Big)^2 + \varepsilon} \leq \sqrt{2\varepsilon}. \tag{152}$$

Thus,

$$\begin{aligned}
\Pr\Big[\Big|\Big(1-\sqrt{1-\varepsilon}\Big)X - \sqrt{\varepsilon}Z\Big| > \delta/2\Big] &\leq 2\Big(1-\Phi\big(\delta/2\sigma\big)\Big) \\
&\leq \frac{4\sigma}{\delta\sqrt{2\pi}} \\
&\leq \frac{4}{\delta}\sqrt{\frac{\varepsilon}{\pi}}.
\end{aligned} \tag{153}$$

□