TWO NUMBER-THEORETIC PROBLEMS THAT ILLUSTRATE THE POWER AND LIMITATIONS OF RANDOMNESS

By

Andrew Shallue

A dissertation submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

(Mathematics)

at the

UNIVERSITY OF WISCONSIN – MADISON

2007

Abstract

This thesis contains work on two problems in algorithmic number theory. The first problem is to give an algorithm that constructs a rational point on an elliptic curve over a finite field. A fast and easy randomized algorithm has existed for some time. We prove that in the case where the finite field has characteristic 2, there is a deterministic algorithm with the same asymptotic running time as the existing randomized algorithm.

The second problem is the random modular subset sum problem. Let t, n, m be given, and let $a_1, \ldots, a_n \in \mathbb{Z}/m\mathbb{Z}$ be chosen uniformly at random. The goal is to find a subset of the a_i that sum to t in $\mathbb{Z}/m\mathbb{Z}$. Define the density of a subset sum problem to be $n/(\log_2 m)$.

For the case of constant density greater than 1, we apply the multi-set birthday problem to give the first algorithm that uses less time and space than dynamic programming. In particular, for parameter $k < n$ and problems of density greater than k, the algorithm uses $\widetilde{O}(m^{1/\log_2 k})$ time and space. It is a randomized algorithm, and will output a solution with probability $1 - e^{-\Omega(n)}$, where the constant depends on k and on the density.

Contents

Chapter 1

Introduction

Algorithmic number theory is the design, analysis, and implementation of number-theoretic algorithms. As a result, researching in this field requires creativity, mathematical skill, and experience with experimentation. Algorithmic number theorists seek faster algorithms by applying the latest mathematical theorems, finding constructive proofs, and designing new protocols. Researchers strive to prove that the algorithm computes what it claims to compute, and provide rigorous bounds on the resource usage required. Finally, sharp analysis involves making simplifying assumptions about computers, and only by programming and running an algorithm can the practitioner be fully confident that it is effective.

The problems studied by algorithmic number theorists vary across all aspects of number theory. The most classical problems are those of factoring an integer into primes and determining whether a given integer is prime or composite. While these problems have not lost importance, recent research has branched into all aspects of number theory, including class field theory, elliptic curves, modular forms, arithmetic varieties, and much more.

One general research program is to classify problems by their running time. At its coarsest this separates algorithms into polynomial-time and exponential-time (of course, these are not the only possibilities). Polynomial-time means that on a given input, the

running time of the algorithm is at most a constant times a polynomial of the input length. For example, if an integer n is the input, the input length is $O(\log n)$ since it takes $\log_2 n$ bits to store n in a computer's memory. Thus a polynomial-time algorithm is one that runs using at most $O(\log n)^c$ bit operations for some constant c, while an exponential algorithm would take $\Omega(n^c)$ bit operations.

A vital tool in this field is the use of randomness. Although mysterious, it is an empirical fact that for many computational problems in discrete mathematics the fastest known algorithms are those that use randomness. Despite the evident power, mathematicians are uncomfortable with even the tiniest possibility of error, and so a continuous research program in algorithmic number theory is to find deterministic algorithms, i.e. those that make no use of random bits. A great breakthrough in this program was the discovery of the long sought after deterministic polynomial time prime-testing algorithm by Agrawal, Kayal, and Saxena [1].

This thesis presents new algorithms for two separate problems. The first is a derandomization of a classic problem involving constructing points on elliptic curves over finite fields. The second is a series of new algorithms for the dense case of the random modular subset sum problem. While still exponential, significant gains are made through heavy use of randomness in the algorithm, requiring significant understanding of the tools of the theory of probability for a rigorous analysis. In this way this thesis showcases the role that randomness plays in algorithmic number theory.

1.1 Results

The following is a summary of the main results in this thesis.

Theorem. *Let E be an elliptic curve over a finite field $\mathbb{F}$ of characteristic 2 given by a Weierstrass equation. Then there exists a* deterministic *algorithm that outputs a nontrivial point of $E(\mathbb{F})$ using $O(\log^2 q)$ bit operations.*

This is the first deterministic algorithm for constructing points on elliptic curves over finite fields. Notably, it matches the asymptotic running time of the long-standing randomized algorithm used in practice. Christiaan E. van de Woestijne independently discovered a similar algorithm that accomplishes the same result in the odd characteristic case, and extended his result to cover the case of even characteristic. The work of both authors appears in [28].

The next several theorems cover new work on the dense case of the random modular subset sum problem, abbreviated RMSS. The parameters of this problem include a modulus m and the number of input elements n. The density of the problem is defined to be $n/\log_2 m$. An important fact is that we expect such problems to have a solution, so the probability that the algorithm fails takes into account the probability that no solution exists.

Theorem (2-set)**.** *Assume $m = 2^{cn/2}$ with $c < 1/2$, so the RMSS problem has density greater than 4. Then there is a randomized algorithm that runs using time and space $\widetilde{O}(m^{1/2})$ and finds a solution with probability greater than $1 - e^{-\Omega(n)}$.*

Theorem (4-set)**.** *Assume $m = 2^{cn/4}$ with $c < 3/5$, so the RMSS problem has density greater than $20/3$. Then there is a randomized algorithm that runs using time and space $\widetilde{O}(m^{1/3})$ and finds a solution with probability greater than $1 - e^{-\Omega(n)}$.*

Theorem (k-set)**.** *Fix a parameter $k < n$ with $n > O(k(\log k)^2)$, and assume $m = 2^{cn/k}$ for $c < \frac{\log k}{\log k + 4}$, giving problems of density greater than $k(1 + \frac{4}{\log k})$. Then there is a*

randomized algorithm that runs using time and space $\widetilde{O}(m^{1/\log k})$ *and finds a solution with probability greater than* $1 - e^{-\Omega(n)}$.

For each of these theorems, the probability of success is over the random bits of the algorithm and also over the random choice of inputs. The constant in the exponent of the success probability depends on c for all algorithms (and on k for the k-set algorithm), and in such a way that the probability of success increases with increasing density.

Each of these theorems gives an algorithm for RMSS that is the fastest known for the given range of densities. Rather than prove the expected number of solutions is at least 1, for each algorithm we prove the stronger result that a solution is found with probability exponentially close to 1 on all inputs. Normally such results are achieved through the use of the Chernoff bound, but Chernoff is not applicable in our case since the random variables that arise are not independent. Improvements are made through increased understanding of the relationships between variables which are neither uniform nor independent.

1.2 Notation

Each of the two algorithmic problems will have their own notation and objects under consideration that will need definition, but here we give some notation common to both and outline our model of computation.

All logarithms will have base 2.

We will use "Big-Oh" and "Soft-Oh" notation to give bounds on the resource usage of our algorithms. The amounts given will be in terms of bit complexity, so for example the addition of two n-bit integers takes $O(n)$ bit operations. Throughout this thesis we

assume naive arithmetic, i.e. we assume that the four standard arithmetic operations are performed using the classical "grade school" algorithms. An area of active research is the development of asymptotically faster algorithms. However, we choose to use naive arithmetic since the main results are in correctness proofs rather than running time analysis, and even here arithmetic plays a small role in the operations performed, both conceptually and asymptotically. For further discussion of fast arithmetic see [31].

We now give formal definitions of "Big-Oh" notation and some of its variations, following [31] and [4].

Definition 1.1 (Big-Oh)**.** *Let $f, g : \mathbb{N} \to \mathbb{R}$ be positive functions. We say f is $O(g)$ if there exists $a, N \in \mathbb{N}$ such that $f(n) \leq ag(n)$ for all $n \geq N$.*

Definition 1.2 (Big-Omega)**.** *Let $f, g : \mathbb{N} \to \mathbb{R}$ be positive functions. We say f is $\Omega(g)$ if there exists $a > 0$ and $N \in \mathbb{N}$ such that $f(n) \geq ag(n)$ for all $n \geq N$.*

Definition 1.3 (Soft-Oh)**.** *Let $f, g : \mathbb{N} \to \mathbb{R}$ be positive functions. We say f is $\widetilde{O}(g)$ if there exists $c, N \in \mathbb{N}$ such that $f(n) \leq g(n)(\log_2(3 + g(n)))^c$ for all $n \geq N$.*

The use of $\widetilde{O}$ will be convenient for highlighting the main term of a complicated running time expression.

Acknowledgments I thank my advisor Eric Bach for being an essential part of my graduate education. I also thank Matt Darnall for introducing me to the basics of martingale theory, and to Tom Kurtz and Dieter van Melkebeek for helping me dig myself out of painful misconceptions about probability. Thanks to NSF award CCF-8635355 and the William F. Vilas Trust Estate for monetary support. Finally, I thank my wife Heather Evert for her unbounded love and moral support.

Chapter 2

Constructing Points on Elliptic Curves over Finite Fields

2.1 Preliminaries and Previous Work

The birational classification of algebraic varieties defines an elliptic curve as an algebraic curve of genus 1 over an algebraically closed field. However, for our purposes it is most convenient to define an elliptic curve as the set of points (x, y) over some arbitrary field that satisfy the Weierstrass equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \ ,$$

assuming that the equation is non-singular. We also assume that the curve includes one point at infinity called $\mathcal{O}$, with homogeneous coordinates $(0 : 1 : 0)$. Algebraic geometers are most interested in elliptic curves over algebraically closed fields, number theorists in curves over $\mathbb{Q}$ or its extensions, while cryptographers are most interested in curves over finite fields.

It is a remarkable fact that the points of E over a field form a group with $\mathcal{O}$ as the identity element. For further information and a wealth of other remarkable facts about elliptic curves, the reader is referred to [29].

Due to the rich interplay between their geometric and algebraic properties, elliptic curves have fascinated mathematicians for over a century. In recent years the study of elliptic curves has been enriched by numerous computational results and applications of elliptic curves to classical computational questions. Birch and Swinnerton-Dyer [6] published their famous conjecture, one of the Clay Institute's Millennium Problems, after gathering data on numerous curves. Elliptic curves form an integral part of the integer factorization algorithm first developed by Lenstra [19]. This algorithm has broken several factoring records, and is still of practical use today. Several prime proving algorithms, one of which is due to Atkin and Morain [3], involve elliptic curves. Finally, new public key cryptosystems use the group of points on an elliptic curve since there are no known general subexponential algorithms for solving the discrete logarithm problem on such groups. The research in this area has exploded in the last couple of years; for a brief introduction see [17].

Let E be an elliptic curve over a finite field $\mathbb{F}$ of characteristic p a prime. Due to the Hasse bound, it is well-known that if $|\mathbb{F}| > 5$ then E must have points other than $\mathcal{O}$. However, it is another matter entirely to construct such points. Other than brute force search, the only algorithm known was the following naive probabilistic one [17, Section 6.2]. Substitute a random value for x in the Weierstrass equation and see if the resulting quadratic equation can be solved. If so, solve for the y coordinate, using a probabilistic square root algorithm if char $\mathbb{F}_q \neq 2$. Trivial though it seemed, finding a deterministic polynomial time algorithm that constructs points posed a challenge.

This challenge was articulated by N. Koblitz [17] who said "there is no polynomial time (in $\log q$) *deterministic* algorithm known for writing down a large number of points on an arbitrary elliptic curve E over $\mathbb{F}_q$." Earlier, R. Schoof in [26], in addition to

demonstrating the first polynomial time algorithm for computing the number of points of an elliptic curve over a finite field, reported that an algorithm of Shanks could also be used to find $|E(\mathbb{F}_q)|$. The algorithm required first computing a point in $E(\mathbb{F}_q)$, to which Schoof remarked, "in practice there is no problem in finding a point P, but I do not know how to prove that computing a point in $E(\mathbb{F}_q)$ is easy."

The problem of finding a deterministic polynomial-time algorithm for constructing points on elliptic curves over finite fields has now been solved. In the summer of 2005 the author completed the case of finite fields of characteristic 2. Simultaneously, Christiaan E. van de Woestijne completed the odd characteristic case and then successfully adapted his methods to characteristic 2. An essential component of this work came from his thesis [30], where he partially solved the long-standing problem of taking square roots modulo p in deterministic polynomial time. For point construction, the results of both cases were combined and a joint paper [28] presented at ANTS VII. The following section contains an excerpt from this paper, presenting the work on the characteristic 2 case done by the author.

2.2 Elliptic curves in characteristic 2

The goal of this section is to prove the following theorem.

Theorem 2.1. *Let E be an elliptic curve over a finite field $\mathbb{F}$ of characteristic 2. Then there exists a* deterministic *algorithm that outputs a nontrivial point of $E(\mathbb{F})$ using $O(\log^2 q)$ bit operations, assuming standard arithmetic.*

We assume that E is given by a nonsingular Weierstrass equation. Then by [29,

Appendix A] we know that E has a Weierstrass equation of one of two following forms:

$$Y^2 + a_3 Y = X^3 + a_4 X + a_6 \quad \text{if } j(E) = 0$$
$$Y^2 + XY = X^3 + a_2 X^2 + a_6 \quad \text{if } j(E) \neq 0 \ .$$

Here $j(E)$, the *j-invariant* of E, partitions elliptic curves into classes of curves isomorphic over $\overline{\mathbb{F}}$. Using the coefficients from the general Weierstrass equation in the previous section, in characteristic 2 we define $j(E) := a_1^{12}/\Delta$ where Δ is the discriminant of E.

In the case when $\mathbb{F}$ is finite of order 2^r, let Tr stand for the trace map from $\mathbb{F}$ to $\mathbb{F}_2$, which is defined by

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{F}_2}(x) := x + x^2 + x^{2^2} + \cdots + x^{2^{r-1}} \ .$$

For motivation, consider the problem of finding rational points on

$$Y^2 + Y = f(X).$$

Lemma 2.2. *If $f \in \mathbb{F}[X]$ is linear, then there exists a deterministic polynomial-time algorithm that returns an $\mathbb{F}$-rational point of $Y^2 + Y = f(X)$.*

Proof. It is known that the valid X-coordinates are exactly $x \in \mathbb{F}$ satisfying $\mathrm{Tr}(f(x)) = 0$ [5, Sect. 6.6]. This suggests the following algorithm. First precompute $a \in \mathbb{F}$ such that $\mathrm{Tr}(f(a)) = 1$. Since $x \mapsto \mathrm{Tr}(f(x))$ is a linear map over $\mathbb{F}_2$, we can deterministically compute the required a using linear algebra. Now, one of x or $x + a$ must be a valid X-coordinate.

Given such an x, it remains to solve for Y. Here we have an advantage over the case of odd characteristic in that there exist deterministic polynomial-time algorithms for solving quadratics ([5, Chap. 6], [4, Sect. 7.4]). □

For more general f, the new idea is to look for points on the threefold

$$f(x_1) + f(x_2) + f(x_3) = y^2 + y \ .$$

Elements of the form y^2+y are exactly those in Ker(Tr), and form an index two subgroup of $\mathbb{F}^+$. Thus one of the three terms must itself be of the form $y^2 + y$.

With this in mind, we define

$$g(x) = x^{-2} \cdot (x^3 + a_2x^2 + a_6), \text{ and}$$

$$h(x) = x^3 + a_4x + a_6 \ .$$

Now let V_1 and V_2 be threefolds given by the equations

$$V_1 : g(x) + g(y) + g(z) = w^2 + w$$

$$V_2 : h(x) + h(y) + h(z) = w^2 + a_3w \ .$$

Geometrically, V_1 and V_2 are each the quotient of $E \times E \times E$ by the action of a Klein 4-group of automorphisms.

To construct a point on E we will construct a computable rational map from a parameterizable surface to the appropriate threefold. Once we have a point on the threefold it will be easy to get rational points on E. The surfaces we need are given by the equations

$$S_1 : x + y + xy(x + y)^{-1} + a_2 = w^2 + w$$

$$S_2 : x^2y + y^2x + a_6 = w^2 + a_3w \ .$$

Lemma 2.3. *Let $\mathbb{F}$ be a field of characteristic 2. There exist rational maps $\phi_1 : S_1 \to V_1$ and $\phi_2 : S_2 \to V_2$ over $\mathbb{F}$ which are invertible on their images, given by*

$$\phi_1 : (x, y, w) \mapsto (x,\ y,\ xy(x + y)^{-1},\ w)$$

$$\phi_2 : (x, y, w) \mapsto (x,\ y,\ x + y,\ w) \ .$$

Proof. First consider ϕ_1, the map that will be used in the case when $j(E) \neq 0$. Recall that $g(x) = x + a_2 + a_6 x^{-2}$. We have

$$\begin{aligned} g(x) + g(y) + g\left(\frac{xy}{x+y}\right) &= x + y + \frac{xy}{x+y} + 3a_2 + a_6\left(\frac{1}{x} + \frac{1}{y} + \frac{x+y}{xy}\right)^2 \\ &= x + y + \frac{xy}{x+y} + a_2 \\ &= w^2 + w \end{aligned}$$

since (x, y, w) is a point on S_1. Hence $(x,\ y,\ xy(x+y)^{-1},\ w)$ is a point on V_1.

Next consider ϕ_2, the map that will be used when $j(E) = 0$. We have

$$\begin{aligned} h(x) + h(y) + h(x+y) &= x^3 + a_4 x + y^3 + a_4 y + (x+y)^3 + a_4(x+y) + 3a_6 \\ &= x^2 y + y^2 x + a_6 \\ &= w^2 + a_3 w \end{aligned}$$

since (x, y, w) is a point on S_2.

Note that given a point in the image of one of these maps we can trivially find its preimage on the surface, so that both maps are invertible on their images. □

A useful geometric interpretation of these maps is that the image of ϕ_1 is contained in the intersection of V_1 with $x^{-1} + y^{-1} + z^{-1} = 0$, while the image of ϕ_2 is contained in the intersection of V_2 with $x + y + z = 0$.

These maps now play a critical role in the proof of Theorem 2.1. Beyond the statement of the theorem, we assume that the field $\mathbb{F}$ has more than 4 elements. This is no loss since for such small fields we can compute all the points of $E(\mathbb{F})$ quite quickly.

Proof. There are two cases to consider, since E can either have j-invariant zero or nonzero. In both cases our strategy is to deterministically find points on the appropriate surface, map them to the threefold, and then construct a point on E.

First assume that $j(E) \neq 0$. For arbitrary c the equation

$$x + y + \frac{xy}{x+y} = c$$

is equivalent to the genus 0 curve $C: \ x^2 + y^2 + xy + c(x+y) = 0$ except when $x = y$. However, if (x, y) is a point on C with $x = y$ then it must be the point $(0, 0)$, so not much is lost. We have the generic solution $(0, c)$ and from this get all points of C through the rational parameterization

$$\begin{aligned} y &= tx + c \\ x &= \frac{tc}{1 + t + t^2} \ . \end{aligned}$$

Thus we have a family of rational points on S_1 parameterized by t and w which can be mapped to points on V_1 via ϕ_1. It now remains to compute rational points of E.

For $a \in \mathbb{F}^\times$ consider the set

$$\{v^2 + av \mid v \in F\}.$$

This set is an additive subgroup of $\mathbb{F}^+$ of index 2, so if $g(x) + g(y) + g(z) = w^2 + w$ then at least one of $g(x)$, $g(y)$, $g(z)$ is itself of the form $v^2 + v$. Discover which it is by applying Tr, call the input x, and deterministically solve the quadratic to find v. From $v^2 + v = x^{-2}(x^3 + a_2x^2 + a_6)$ we now have

$$(vx)^2 + x(vx) = x^3 + a_2x^2 + a_6$$

and hence a point on E.

Suppose instead that $j(E) = 0$. We wish to compute points on S_2. Taking $y = u^2$, we transform the equation for S_2 as follows:

$$\begin{aligned} xy(x+y) + a_6 &= w^2 + a_3 w \\ x^2u^2 + xu^4 + a_6 &= w^2 + a_3 w \\ a_3xu + xu^4 + a_6 &= (w+xu)^2 + a_3(w+xu) \ . \end{aligned}$$

Now, choose y and compute its square root u (possible deterministically since squaring is an automorphism). There are at most four bad choices of y to avoid, corresponding to the roots of $u^4 + a_3u$. Since our field size is greater than 4, we know that good choices exist. If $u^4 + a_3u \neq 0$, the equation $x(a_3u + u^4) + a_6 = z^2 + a_3z$ is linear in x and hence for any given z, we easily compute the unique value for x. Now the point $(x, y, z + xu)$ is a point on S_2, which we map to V_2 via ϕ_2.

It remains to find a point on E. Mirroring the argument in the previous case, one of $h(x)$, $h(y)$, and $h(z)$ has the form $v^2 + a_3v$. Discover which it is, call the input x, and solve the quadratic $v^2 + a_3v = h(x)$ for v. Output (x, v) as a rational point on E.

Finally, we briefly consider the running time. Besides standard arithmetic, the algorithm applies the Tr map and solves quadratic equations. As far as standard arithmetic, addition and subtraction take $O(\log q)$ bit operations while multiplication and inversion take $O(\log q)^2$ [4]. If we assume that squaring is implemented for characteristic 2 fields using shift registers and thus constant, the Tr map involves r additions and thus takes $O(\log q)^2$ bit operations. Finally, if we use Berlekamp's algorithm at a cost of $O((\deg f + p)(\log f)^2)$ bit operations [4], then in the case of quadratic equations over characteristic 2 fields this also takes time $O(\log q)^2$. □

An important question to analyze is how many of the $\mathbb{F}$-rational points of E are

obtained by this algorithm. The next theorem will demonstrate that the number is quite large, in particular at least a nonzero proportion. First we need a definition.

Definition 2.4. *We define two points* $P = (x_1, x_2, x_3, y)$ *and* $P' = (x'_1, x'_2, x'_3, y')$ *on* V *to be* disjoint *if the sets* $\{x_1, x_2, x_3\}$ *and* $\{x'_1, x'_2, x'_3\}$ *are disjoint.*

Theorem 2.5. *Let* $\mathbb{F}$ *be a finite field of order* $q = 2^r$ *with* $q > 4$*. The number of disjoint points of* V_1 *that arise from Theorem 2.1 is at least* $(q-4)/6$*.*

Proof. Throughout, assume that the parameter w in S_1 is fixed. Allowing different values could improve the bound, but that analysis has not yet been done.

It was noted before that S_1 can be transformed into a genus 0 curve $C : x^2 + y^2 + xy + c(x + y) = 0$, with C having only gained the point $(0, 0)$. Let $C'(\mathbb{F})$ be the points of C except for $(0, 0)$, $(c, 0)$, and $(0, c)$.

It can easily be confirmed that if (x, y) is a point on C', then

$$\sigma_1(x, y) := (x,\ xy(x + y)^{-1}),$$

$$\sigma_2(x, y) := (y, x)$$

are points on C'. We conclude that the group $G = \langle \sigma_1, \sigma_2 \rangle$ acts on $C'(\mathbb{F})$, is isomorphic to Sym(3), and splits the points of C' into orbits of size 6. For the last statement, note that $x = y$ implies $(x, y) = (0, 0)$ and $y = xy(x + y)^{-1}$ implies $y = 0$. Thus the stabiliser in Sym(3) of any point has index 6, giving an orbit of size 6.

Any coordinate only appears in its orbit, and each orbit yields the same set $\{x,\ y,\ xy(x + y)^{-1}\}$. Thus each orbit when mapped via ϕ_1 yields a disjoint point on V_1.

It remains to count the number of orbits. If r is odd, $t^2 + t + 1$ is irreducible over $\mathbb{F}$ and hence all $t \in \mathbb{F}$ are valid. Thus C has $q + 1$ points, but after discarding $(0, 0)$, $(c, 0)$,

and $(0, c)$ we are left with $(q-2)/6$ orbits. If r is even, t^2+t+1 splits and hence there are $q-2$ valid t, leaving us with $(q-4)/6$ orbits. □

We note that the case with $j(E) = 0$ yields a similar bound, since fixing w in S_2 yields a curve of genus 0 that also breaks up into orbits of size 6, each element of the orbit resulting in the same triple $(x,\ y,\ x+y)$.

Corollary 2.6. *The algorithm in Theorem 2.1 can construct at least* $(q-4)/3$ *points of* E.

Proof. By Theorem 2.5, the algorithm in Theorem 2.1 yields at least $(q-4)/6$ disjoint points of V. This yields at least $(q-4)/6$ different x-coordinates of E, and hence at least $(q-4)/3$ distinct points of E. □

Chapter 3

Random Dense Subset Sums

3.1 Problem Statement

Let $a_1, a_2, \ldots, a_n, t \in \mathbb{Z}/m\mathbb{Z}$ be given. The modular subset sum problem is to find a subset of the a_i that sum to t in $\mathbb{Z}/m\mathbb{Z}$, i.e. to find $x_i \in \{0, 1\}$ such that

$$\sum_{i=1}^{n} a_i x_i = t \mod m \ . \tag{3.1}$$

The corresponding decision problem is to determine whether or not there exists $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ that satisfies equation 3.1.

A subset sum problem is called random if n, m, and t are all fixed parameters but the a_i are drawn uniformly at random from $\mathbb{Z}/m\mathbb{Z}$. In addition to being interesting in their own right, random subset sum problems accurately model problems that arise naturally in number theory and combinatorics. We will use the shorthand MSS for modular subset sum and RMSS for random modular subset sum.

A useful way of classifying subset sum problems is by density.

Definition 3.1. *The* density *of a subset sum problem is* $\frac{n}{\log_2 m}$. *Problems with density less than one are called* sparse, *while those with density greater than one are called* dense.

In Chapters 4 and 5 we will give a series of new randomized algorithms for the random dense modular subset sum problem, all of which are rigorously analyzed, and

each of which is the fastest known algorithm for a given range of densities. These algorithms vastly outperform the best known deterministic algorithms. Consequently, they showcase the computational power of randomized algorithms and the important role the theory of probability plays in their analysis.

3.2 Previous Work

The subset sum problem is of great practical and theoretical interest. Its decision version was proven NP-complete by R. Karp in his seminal 1972 paper on reductions among combinatorial problems [16]. It has seen application in the creation of public key cryptosystems [10], and is a vital tool in discovering Carmichael numbers [13].

Trivial algorithms for MSS include brute force enumeration at $O(2^n)$ time and constant space, basic time-space tradeoff at $O(2^{n/2})$ time and space, and dynamic programming at $O(n \cdot m)$ time and space. Schroeppel and Shamir [27] were first to discover a nontrivial method for solving the subset sum problem. Their algorithm takes time $O(2^{n/2})$ and space $O(2^{n/4})$, using a technique of decomposition that is reflected in this thesis.

Despite its status as an NP-complete problem, many cases are quite tractable. If m is polynomial in n (giving a very dense instance), the problem is solvable in polynomial time using dynamic programming. More sophisticated methods can improve the running time, for example [7] achieved a running time of $O(n^{7/4}/\log^{3/4} n)$ for instances of density $(m/\log m)^{1/2}$. For sparse instances, the current favored technique is that of lattice basis reduction. If we have density $d < 0.64$, then Lagarias and Odlyzko [18] proved that almost all (as n goes to infinity) subset sum problems are solvable using this technique.

The density bound was improved to 0.98 in [8].

The new results in this thesis are a continuation of a recent line of research that focuses on dense subset sum problems with m superpolynomial in n. In 2005 Flaxman and Przydatek [11] extended the range of subset sum problems that could be solved in polynomial time, so that for $m = 2^{O(\log n)^2}$ their algorithm runs in time $O(n^{3/2})$. At about the same time Lyubashevsky [20] applied Wagner's algorithm for the k-set birthday problem [32] and performed a more detailed probabilistic analysis. Whereas Wagner only claimed the expected number of solutions the algorithm produced would be at least one, Lyubashevsky proved his new algorithm outputs a solution with high probability for every instance of RMSS. However, the cost of this improved analysis was a slower running time. Taking as a parameter the number of sets k that the a_i are divided into and assuming $k = n^\epsilon$ for $\epsilon < 1$, Wagner's general algorithm takes time and space $\widetilde{O}(m^{1/\log k})$ while Lyubashevsky's takes time and space $\widetilde{O}(m^{2/\log k})$.

The new subset sum algorithms presented in this thesis all share a common philosophy with Wagner's algorithm for the k-set birthday problem. This philosophy is that dense RMSS problems have many solutions, and hence searching randomly for a solution will be more profitable than searching through brute force. Of course, pathological instances do exist and can be easily constructed, but by choosing our a_i randomly we ensure these pathological cases are rare. For even denser RMSS problems, the general algorithm searches for an easier to find solution with specific properties, and the task of analysis is to ensure one exists with high probability.

3.3 Solving Multi-Set Birthday Problems

Since the work of Wagner [32] and Lyubashevsky [20] plays such a prominent role in the algorithms that follow, it is worthwhile to provide more details than the brief overview in the previous section.

A familiar tool in the design and analysis of randomized algorithms is the 2-set birthday problem, commonly called the birthday problem or birthday paradox. Often this is applied to the problem of finding a collision among the elements of one list, but this is not the context of interest to us. Given two lists of elements drawn uniformly and independently at random from $\mathbb{Z}/m\mathbb{Z}$, the birthday problem is to find two elements, one from each list, whose sum is zero (in $\mathbb{Z}/m\mathbb{Z}$). The surprising fact (for those first introduced to probability) is that if the lists each have size $\sqrt{m}$ then we expect that such a sum will occur. In fact, there is even an algorithm to find the solution (if it exists) which takes time and space $\widetilde{O}(\sqrt{m})$. This birthday problem can be generalized to the case where we have many lists instead of just two.

Definition 3.2. *Let $L_1, \ldots, L_k$ be lists of elements drawn uniformly and independently at random from $\mathbb{Z}/m\mathbb{Z}$. Then the* k-set birthday problem *is to find $\ell_i \in L_i$ such that $\ell_1 + \cdots + \ell_k = 0 \mod m$.*

Note that if $|L_i| = m^{1/k}$ for all i, then the number of possible sums is $(m^{1/k})^k = m$. Since the elements of each list are uniformly generated, the sums themselves are uniformly distributed on $\mathbb{Z}/m\mathbb{Z}$ and hence the expected number of solutions to the k-set birthday problem in this case is 1. However, finding a solution is another matter and for now no algorithm is known that only takes time and space $O(m^{1/k})$.

While the literature has scattered instances of applications of the 4-set birthday

problem, it appears that Wagner [32] is the first to state the k-set problem in full generality and to give an algorithm that beats $O(m^{1/2})$ in resource usage. This algorithm can be easily generalized to many other groups besides $\mathbb{Z}/m\mathbb{Z}$, but we focus on this case since it is applicable to MSS.

The key subroutine used in Wagner's algorithm is the join operation $\bowtie_{[a,b)}$, $a, b \in \mathbb{R}$, where $[a, b)$ denotes the interval from a to b. We define $L_1 \bowtie_{[a,b)} L_2$ to be the list of elements $\ell_1 + \ell_2 \in [a, b)$ where $\ell_i \in L_1$. In later chapters we instantiate the join operator with Algorithm ListMerge.

Let $p = m^{-1/\log k}$. The algorithm proceeds as follows. Consider the elements of the lists L_i as integers in the interval $[-\frac{m}{2}, \frac{m}{2})$. Apply $\bowtie_{[-mp/2,mp/2)}$ to pairs of lists, yielding $k/2$ lists at level 1. At level λ, apply $\bowtie_{[-mp^{\lambda+1}/2,mp^{\lambda+1}/2)}$ to pairs of lists. At level $\log k$, we have one list whose elements are integers in the interval $[-\frac{1}{2}, \frac{1}{2})$, and thus any element is a solution to the k-set birthday problem. The algorithm uses $O(k \cdot m^{1/\log k})$ time and space. In addition, Wagner gave a slightly different algorithm that solves the 4-set birthday problem using time and space $O(m^{1/3})$.

The key piece missing from turning Wagner's algorithm into a MSS algorithm was generating a large number of uniformly distributed elements. This was accomplished by Lyubashevsky [20] who noted that if the density of the problem is greater than k, random subset sums of n/k of the a_i will be very close to uniformly generated, as long as the a_i are themselves randomly generated. Break up the a_i into k sets of size n/k, and let L_i be $m^{1/\log k}$ random subsets of the ith set. Then applying Wagner's k-set algorithm to the L_i will solve the subset sum problem.

Lyubashevsky also changed the join operation $\bowtie_{[-R/2,R/2)}$ as follows. Suppose that L_1 and L_2 each contain n/p^2 elements rather than $1/p$. Break up the interval $[-\frac{R}{2}, \frac{R}{2})$

into $1/p$ intervals of length Rp, and pick a b from L_1 from each interval. Then search for c such that $b + c \in [-\frac{Rp}{2}, \frac{Rp}{2})$. In this way the elements that pass from level to level remain uniform and independent, and by applying the Chernoff bound we can prove that $L_1 \bowtie L_2$ will again contain at least n/p^2 elements with probability at least $1 - e^{-\Omega(n)}$. In this way Lyubashevsky strengthened the analysis by proving that a solution must exist with high probability, rather than only proving that the expected number of solutions is positive. However, the cost of this is an increase in the algorithm's resource usage to $\widetilde{O}(m^{2/\log k})$ time and space.

In this thesis we will extend these results in two directions. First, we will apply the 2-set and 4-set birthday problems to yield 2-set and 4-set algorithms for the RMSS problem. This will yield the first algorithms to perform faster than $O(2^{n/2})$ time for dense problems of constant density. Secondly, we apply stronger tools from probability theory to improve the list-merging lemma of Lyubashevsky. We prove that if L_1 and L_2 contain at least n/p elements, then $L_1 \bowtie L_2$ also contains at least n/p elements with high probability. Rather than maintaining the independence of the elements at all levels, it is enough to be close to pairwise independent.

Finally, all of these new algorithms find a solution with probability at least $1 - e^{-\Omega(n)}$. Note that this probability is over all possible choices of a_i, as well as the coin flips the algorithm performs, and that the constant in the exponent depends upon the density.

3.4 Probability Preliminaries

We will assume the reader is well aware of the basics of probability distributions and spaces, including conditional probability and conditional expectation. All random variables we consider will be discrete and defined on finite probability spaces since all random variables in our algorithms will have these properties. In particular, whenever we say a random variable is defined on an interval I we mean that it is defined on the integers in that interval. However, most of the definitions have continuous analogues and many of the results are true in a much more general setting. For a standard reference see [12].

Often we will abuse notation and have a symbol such as ℓ stand for both a random variable and a value that random variable can take. In this case $\Pr[\ell]$ means the probability that the random variable ℓ takes value ℓ.

Consider the following random variable $X_{\boldsymbol{a}}$ taking values on $\mathbb{Z}/m\mathbb{Z}$, where $\boldsymbol{a} = (a_1, \ldots, a_n)$ and each a_i is drawn uniformly and independently from $\mathbb{Z}/m\mathbb{Z}$. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be an n-bit vector, where each element is drawn uniformly and independently from $\{0, 1\}$. Then treating the x_i and a_i as random variables, we define

$$X_{\boldsymbol{a}} := \sum_{i=1}^{n} x_i a_i \mod m \ . \tag{3.2}$$

In the case where $\boldsymbol{a}$ is fixed and understood from context (say where it is the input of an RMSS instance) we will suppress the $\boldsymbol{a}$ in the notation.

Note that for fixed $\boldsymbol{a}$ and varying $\boldsymbol{x}$, the collection $\{X_{\boldsymbol{a}}(\boldsymbol{x})\}$ is a collection of independent random variables .

Intuitively, if our subset sum problem is dense, X should be close to the uniform distribution U on $[-\frac{m}{2}, \frac{m}{2})$. The assumption that X is effectively uniform is the starting

point for all the new algorithms being presented. Thus it is vital that we formalize what we mean by "close."

Definition 3.3. *Let X and Y be random variables taking values in a probability space A. The* statistical distance *between X and Y, denoted $\Delta(X,Y)$, is*

$$\Delta(X,Y) = \frac{1}{2}\sum_{a\in A} |Pr[X=a] - Pr[Y=a]| \quad .$$

While intuitive, it is often hard to prove theorems with this definition. For that reason we give the following essentially equivalent definition of what it means for distributions to be close.

Definition 3.4. *Let D be a discrete probability distribution on a finite set A. For $S \subset A$ let $D(S) = \sum_{s\in S} D(s)$. Distributions D and D' are* statistically indistinguishable within *ϵ if*

$$|D(S) - D'(S)| < \epsilon \quad \textit{for every } S \subset A.$$

The next two propositions are standard results that expand our usage of this new tool.

Proposition 3.5. *Let X, Y be two random variables taking values in a set A. For any predicate $f : A \to \{0,1\}$,*

$$|Pr[f(X)=1] - Pr[f(Y)=1]| \leq \Delta(X,Y) \quad .$$

Proof. Let S be the subset of A that is the preimage of 1 under f. Then

$$|\Pr[f(X)=1] - \Pr[f(Y)=1]| = |X(S) - Y(S)| \quad .$$

Proposition A.2 in the Appendix proves that X and Y are statistically indistinguishable within ϵ if and only if $\Delta(X,Y) < \epsilon$. In fact, the stronger result is proven that $|X(S) - Y(S)|$ is at most $\Delta(X,Y)$ for all subsets S of A. Thus

$$|\Pr[f(X) = 1] - \Pr[f(Y) = 1]| \leq \Delta(X,Y) \ .$$

□

Proposition 3.6. *Let $X_1, \ldots X_N$ and $Y_1, \ldots, Y_N$ be two lists of independent random variables. Then*

$$\Delta((X_1, \ldots, X_N), (Y_1, \ldots, Y_N)) \leq \sum_{i=1}^{N} \Delta(X_i, Y_i) \ .$$

Proof. We will prove this statement by induction on N. For ease of notation let x_i stand for $\Pr[X_i = a_i]$. For the base case, suppose $N = 2$. Then

$$\begin{aligned}
\Delta((X_1, X_2), (Y_1, Y_2)) &= \frac{1}{2} \sum_{\boldsymbol{a}} |\Pr[(X_1, X_2) = \boldsymbol{a}] - \Pr[(Y_1, Y_2) = \boldsymbol{a}]| \\
&= \frac{1}{2} \sum_{a_1} \sum_{a_2} |x_1 x_2 - y_1 y_2| \quad \text{by independence} \\
&= \frac{1}{2} \sum_{a_1} \sum_{a_2} |x_1(x_2 - y_2) + y_2(x_1 - y_1)| \\
&\leq \frac{1}{2} \sum_{a_1} \sum_{a_2} |x_2 - y_2| + |x_1 - y_1| = \sum_{i=1}^{2} \Delta(X_i, Y_i)
\end{aligned}$$

since $x_1, y_2 \leq 1$. For the inductive step suppose that

$$\Delta((X_1, \ldots, X_{N-1}), (Y_1, \ldots, Y_{N-1})) \leq \sum_{i=1}^{N-1} \Delta(X_i, Y_i)$$

and for ease of notation write $\boldsymbol{a}'$ for $(a_1, \ldots, a_{N-1})$ and x' for $\Pr[(X_1, \ldots, X_{N-1}) = \boldsymbol{a}']$. Then

$$
\begin{aligned}
&\Delta((X_1,\ldots,X_N),(Y_1,\ldots,Y_N)) \\
&= \frac{1}{2}\sum_{\boldsymbol{a}} |\Pr[(X_1,\ldots,X_N)=\boldsymbol{a}] - \Pr[(Y_1,\ldots,Y_N)=\boldsymbol{a}]| \\
&= \frac{1}{2}\sum_{\boldsymbol{a}'}\sum_{a_N} |\Pr[(X_1,\ldots,X_{N-1})=\boldsymbol{a}']x_N - \Pr[(Y_1,\ldots,Y_{N-1})=\boldsymbol{a}']y_N| \\
&= \frac{1}{2}\sum_{\boldsymbol{a}'}\sum_{a_N} |x'(x_N-y_N)+y_N(x'-y')| \\
&\leq \frac{1}{2}\sum_{\boldsymbol{a}'}\sum_{a_N} |x_N-y_N|+|x'-y'| \leq \Delta(X_N,Y_N)+\sum_{i=1}^{N-1}(X_i,Y_i)
\end{aligned}
$$

by induction. □

The main goal of our work with probability distributions will be to attain good tail bounds. We next state several standard bounds that form the starting point towards this goal.

Lemma 3.7 (Markov's Inequality, [12])**.** *If X is any random variable with finite mean then*

$$\Pr[|X| \geq t] \leq \frac{\mathrm{E}[|X|]}{t} \quad \textit{for any } t > 0.$$

Lemma 3.8 (Chebyshev's Inequality, [12])**.** *If X is any random variable with finite mean μ and nonzero variance σ^2, then*

$$\Pr[|X-\mu| \geq t\sigma] \leq \frac{1}{t^2} \quad \textit{for any } t > 0.$$

Lemma 3.9 (Chernoff Bound, [23])**.** *Suppose that $x_1,\ldots,x_n$ are independent random variables taking the values 1 and 0 with probabilities p and $1-p$, respectively, and consider their sum $X=\sum_{i=1}^n x_i$. Then for all $0 \leq \theta \leq 1$,*

$$\Pr[X \geq (1+\theta)pn] \leq e^{-\frac{\theta^2}{3}pn} \ .$$

Lemma 3.10 (Bernstein's Inequality, [21]). *Let random variables $X_1, \ldots, X_n$ be independent, with $|X_k - \mathrm{E}[X_k]| \leq b$ for each k. Let $S_n = \sum X_k$ and let S_n have expected value μ and variance V (the sum of the variances of the X_k). Then for any $t \geq 0$,*

$$\Pr[S_n - \mu \geq t] \leq e^{-\frac{t^2}{2V(1+(bt/3V))}} \ .$$

The Chernoff bound will not be used in the work that follows. We include it as the prototypical exponential tail bound. Bernstein's inequality is presented to show that stronger results are possible with more detailed information on the random variable S_n. In our analysis of the k-set result a martingale version of Bernstein will prove essential.

The following proposition is stated more generally than the result stated in the reference cited. The reason is that we will apply Markov's inequality with different parameters in order to optimize the density of subset sum problems on which our algorithms succeed.

While the details are left to the appendix, essentially the proof involves showing that $\{X_{\boldsymbol{a}} : \{0,1\}^n \to \mathbb{Z}/m\mathbb{Z}\}_{\boldsymbol{a} \in (\mathbb{Z}/m\mathbb{Z})^n}$ is a universal (and hence almost universal) family of hash functions, and then applying the leftover hash lemma. Here we encode elements of $\mathbb{Z}/m\mathbb{Z}$ as bit strings of length cn, $c < 1$.

Proposition 3.11 (Impagliazzo, Naor [14]). *Let $m = 2^{cn}$ with $c < 1$. Then for all $\gamma > 0$, the probability over all choices of input vector $\boldsymbol{a} = (a_1, \ldots, a_n)$ that $\Delta(X_{\boldsymbol{a}}, U) < 2^{-\frac{(1-c)n}{2}+\gamma}$ is greater than $1 - 2^{-\gamma}$.*

Proof. See the Appendix. □

Definition 3.12. *We call the multiset $\boldsymbol{a} = (a_1, \ldots, a_n)$* well-distributed *if it is one of the good choices from Proposition 3.11, i.e.*

$$\Delta(X_{\boldsymbol{a}}, U) < 2^{-\frac{(1-c)n}{2}+\gamma} \ .$$

So for example, choosing $\gamma = \frac{(1-c)n}{4}$ tells us that $X_{\boldsymbol{a}}$ is exponentially close to uniform for all but an exponentially small fraction of $\boldsymbol{a}$. Thus if the a_i are chosen uniformly at random, we lose very little by assuming that $X_{\boldsymbol{a}}$ is uniform. This is the only place we use the fact that our subset sum problem is random, and it is possible to apply the same algorithms to MSS instances with the additional assumption that the a_i are well-distributed. This might be preferable if a constructive criterion could be found for $\boldsymbol{a}$ being well-distributed, but for now that remains an open problem. Note that a necessary condition for being well-distributed is that the a_i contain no common factor. If $\gcd(a_1, \ldots, a_n, m) \neq 1$ then X is only nonzero on a subgroup of $\mathbb{Z}/m\mathbb{Z}$, and thus is far from uniform.

Chapter 4

Basic Algorithms

In this chapter we present two new algorithms for RMSS. These are the 2-set Algorithm and the 4-set Algorithm, and solve RMSS by solving the 2-set and 4-set birthday problems over $\mathbb{Z}/m\mathbb{Z}$. Though asymptotically slower than the general algorithm presented in Chapter 5, they may be applied to problems of density greater than 4 and 20/3, respectively, making them the fastest known algorithms for these density ranges, and the first nontrivial algorithms known for problems of constant density greater than 1. They have the added advantage of being conceptually simpler, and serve as an ideal introduction to the ideas underlying the general algorithm.

4.1 The 2-set Algorithm

The previous chapter gave us tools with which we could justify the assumption that the distribution $X_{\boldsymbol{a}}$ given by 3.2 above is uniform. This then suggests the following algorithm. First split the a_i into two sets (hence the title of this section). Generate $\sqrt{m}$ random subsets of each half, and expect that there will be a collision by the birthday paradox. Since we ultimately want a solution with high probability, we need a high probability version of the 2-set birthday paradox.

The situation is modeled as follows: two types of balls (say n_1 white and n_2 red)

are thrown uniformly at random into m urns. Let S be the number of urns containing balls of both colors. We need an upper bound on $\Pr[S = 0]$, especially in the situation where $n_1 = n_2 = \lceil \sqrt{m} \rceil$. Probability distributions related to this situation are studied extensively in [22]; for our purposes the following lemma will suffice.

Lemma 4.1. *Let* $n_1 = n_2 = \lceil \sqrt{m} \rceil$. *Then* $Pr[S = 0] \leq 1 - e^{-1} + e^{-3/2}$.

Proof. Will will prove the result assuming $n_1 = n_2 = \sqrt{m}$, from which the result stated in the lemma follows.

We have from [22, sec. 4] an upper bound of $(1 - n_1/m)^{n_2} e^{-\lambda} + 1 - e^{-\lambda}$ where $\lambda = \frac{(n_1)^2}{2(m-n_1+1)}$.

Under our assumption we have $\lambda = \frac{m}{2(m-\sqrt{m}+1)}$ and the bounds $e^{-1} \leq e^{-\lambda} \leq e^{-1/2}$ (the lower bound assumes $m/2 < m - \sqrt{m} + 1$ and hence requires $m \geq 4$). Now note that

$$\begin{aligned}(1 - n_1/m)^{n_2} e^{-\lambda} + 1 - e^{-\lambda} &= (1 - 1/\sqrt{m})^{\sqrt{m}} e^{-\lambda} + 1 - e^{-\lambda} \\ &\leq e^{-1} e^{-1/2} + 1 - e^{-1} \ .\end{aligned}$$

□

Let the bound on $\Pr[S = 0]$ in Lemma 4.1 be called β. Calculation reveals that $-1/4 < \log \beta < -1/5$, and that β is approximately 0.855.

Algorithm 1 displays pseudocode for the first of our new RMSS algorithms. In order to apply Proposition 3.11, each of the two sets must be dense, and thus we need $\log m < n/2$ (the size of each of the two sets) rather than $\log m < n$. We have required the even stricter condition $\log m < n/4$ for technical reasons that will be made clear in the

Algorithm 1 2-set algorithm for RMSS

Input: modulus m, target t, $a_1, \ldots, a_n \in \mathbb{Z}/m\mathbb{Z}$.
Require: $\log m < n/4$
Output: bit vector $\boldsymbol{x}$ such that $\sum_{i=1}^{n} a_i x_i = t \mod m$ or *Probably No Solution*
Step 1
for $j = 0$ to $\lceil \sqrt{m} \rceil$ **do**
  generate random bits $x_1, x_2, \ldots, x_{\lfloor n/2 \rfloor}$
  store value $t - \sum_{i=1}^{\lfloor n/2 \rfloor} a_i x_i \mod m$ in a table along with the bits
end for
sort table by value
Step 2
for $j = 0$ to $\lceil \sqrt{m} \rceil$ **do**
  generate random bits $x_{\lfloor n/2 \rfloor + 1}, \ldots, x_n$
  calculate $z = \sum_{i=\lfloor n/2 \rfloor + 1}^{n} a_i x_i \mod m$
  search for z in the table using binary search
  if collision found **then**
    output $\boldsymbol{x} = (x_1, \ldots, x_n)$
    break; {that is, end the algorithm}
  end if
end for
repeat Steps 1 and 2 n times
output *Probably No Solution*

analysis. This means that Algorithm 1 only comes with a rigorous analysis for RMSS problems of density greater than 4, although we suspect it will work just as well on problems of density between 2 and 4. Also note that the algorithm is repeated n times before it gives up and outputs *Probably No Solution*. This is because the chance of a collision is some fixed probability and we iterate (assuming independent trials) to make the probability of failure exponentially small.

Theorem 4.2. *Assume $m = 2^{cn/2}$ with $c < 1/2$ (thus problem has density greater than 4). Then Algorithm 1 runs in time $O(n^3 \cdot \sqrt{m})$ and space $O(n^2 \cdot \sqrt{m})$, both of which are $\widetilde{O}(\sqrt{m})$.*

With the a_i chosen uniformly at random from $\mathbb{Z}/m\mathbb{Z}$, the algorithm outputs a solution with probability at least

$$(1 - 2 \cdot 2^{-\frac{(1-2c)n}{8}})(1 - (\beta + 2 \cdot 2^{-\frac{(1-2c)n}{8}})^n) = 1 - e^{-\Omega(n)} \ .$$

Proof. The algorithm output is error free, so the work in proving correctness is to calculate the probability of failure. Since we only have control when the a_i are well-distributed, we will assume that if they are not the algorithm fails. Thus the probability of success is the probability that each of the two halves of $\boldsymbol{a}$ are well-distributed, times the probability that at least one of the n trials of the algorithm finds a solution under the assumption that the sets are well-distributed.

Let $\boldsymbol{a}_1 = (a_1, \ldots, a_{\lfloor n/2 \rfloor})$, and let $\gamma = \frac{(1-2c)n}{8}$ be the parameter choice in Proposition 3.11. Then for all but a $2^{-\frac{(1-2c)n}{8}}$ fraction of possible $\boldsymbol{a_1}$, the distribution X is within $2^{-\frac{n}{8}}$ of being uniform. Note that n has been replaced by $n/2$ in the statement of the proposition since $|\boldsymbol{a}_1| = n/2$. Thus the probability that both halves are well-distributed

is greater than

$$(1 - 2^{-\frac{(1-2c)n}{8}})(1 - 2^{-\frac{(1-2c)n}{8}}) > 1 - 2 \cdot 2^{-\frac{(1-2c)n}{8}} \quad .$$

Assume that the sets in Steps 1 and 2 contain uniformly distributed elements in the range $[0, m)$. Then by Lemma 4.1 the probability that the sets do not collide is at most $\beta < 1$.

One trial of the algorithm requires at most $2\sqrt{m}$ random elements of $\mathbb{Z}/m\mathbb{Z}$. Let $\boldsymbol{a}_2 = (a_{\lfloor n/2 \rfloor + 1}, \ldots, a_n)$, and define random variables $X_{\boldsymbol{a}_1}$ and $X_{\boldsymbol{a}_2}$ by 3.2 as was done in Section 3.4. Define random variables X_i and U_i, $1 \leq i \leq 2\sqrt{m}$, such that half the X_i are copies of $X_{\boldsymbol{a}_1}$, half copies of $X_{\boldsymbol{a}_2}$, and U_i has the uniform distribution on $\mathbb{Z}/m\mathbb{Z}$. Recall that the X_i are independent.

Let the predicate f take the value 1 if no collision occurs between t minus an element from the first $\sqrt{m}$ variables and an element from the second $\sqrt{m}$ variables, where all operations are in $\mathbb{Z}/m\mathbb{Z}$. For example, we have $\Pr[f(U_1, \ldots, U_{2\sqrt{m}}) = 1] \leq \beta$. Then by Proposition 3.5 and Proposition 3.6 we have

$$|\Pr[f(X_1, \ldots, X_{2\sqrt{m}}) = 1] - \Pr[f(U_1, \ldots, U_{2\sqrt{m}}) = 1]| \leq \sum_{i=1}^{2\sqrt{m}} \Delta(X_i, U_i) \quad .$$

Applying Proposition 3.11, this quantity is at most $2\sqrt{m} \cdot 2^{-\frac{n}{8}} = 2 \cdot 2^{-\frac{(1-2c)n}{8}}$.

So one trial fails with probability at most $\beta + 2 \cdot 2^{-\frac{(1-2c)n}{8}}$. For $c < \frac{1}{2}$ and n large enough, this is bounded away from 1. The failure after n trials is thus smaller than $(\beta + 2 \cdot 2^{-\frac{(1-2c)n}{8}})^n$.

Putting this all together, we have that the probability that the algorithm succeeds is at least

$$(1 - 2 \cdot 2^{-\frac{(1-2c)n}{8}})(1 - (\beta + 2 \cdot 2^{-\frac{(1-2c)n}{8}})^n)$$

which given $c < \frac{1}{2}$ can be written as $1 - e^{-\Omega(n)}$.

Finally we analyze the complexity. The storage is $\sqrt{m}$ entries, each of which is $n/2$ bits and an element of $\mathbb{Z}/m\mathbb{Z}$. For time complexity, Steps 1 and 2 each repeat a calculation $\sqrt{m}$ times which takes $n \log m$ bit operations. These steps are repeated at worst n times. □

4.2 The 4-set Algorithm

In this section we present the second of our new algorithms for RMSS, which gives an improvement by solving the 4-set birthday problem. We start by giving an overview of Wagner's [32] algorithm for the 4-set birthday problem, assuming for ease of exposition that the target is 0. First divide the a_i into 4 sets, and generate lists of random subsets for each of size N. Assume that the elements of each list are uniformly generated. Choose $p < 1$, and for each pair of lists L_1, L_2 find the elements $b + c \in L_1 + L_2$ such that $b + c \in [-\frac{mp}{2}, \frac{mp}{2})$. Now solve the 2-set birthday problem on the pair of merged lists.

Observe that if we choose $p = m^{1/3}$ and $N = \frac{1}{p}$, then the expected number of elements in the merged lists is $N^2 p = \frac{1}{p}$. If we now take sums of the elements of the merged lists, there are $\frac{1}{p^2}$ sums in an interval of length $mp = \frac{1}{p^2}$, and so the expected number of zero sums is 1.

The work of this section is to flesh out these ideas and to prove the stronger correctness result that a solution is found with high probability.

Let $L_1 + L_2 = \{b + c \mod m \mid b \in L_1, c \in L_2\}$ and note that $\ell \in L_1 + L_2$ is distributed uniformly on $\mathbb{Z}/m\mathbb{Z}$. Let I be some interval of length mp for $p < 1$, say the interval

$[a, mp + a)$ (wrapping modulo m if necessary). Since $\ell \in L_1 + L_2$ is a discrete random variable, we do not necessarily have $\Pr[\ell \in I] = p$. However, the next proposition provides bounds for this probability.

Proposition 4.3. *Let $p < 1$ be given. For any interval $I = [a, mp + a)$ of length mp and for $\ell \in L_1 + L_2$, we have*

$$\frac{p}{2} \leq \Pr[\ell \in I] \leq p$$

assuming that $mp \geq 2$ or $mp = 1$.

Proof. We start with the fact that ℓ follows a uniform distribution on $\mathbb{Z}/m\mathbb{Z}$ since the elements of L_1 and L_2 are uniform and the addition is done modulo m. There are at most mp integers in I and so $\Pr[\ell \in I] \leq p$.

For the lower bound note that $\Pr[\ell \in I] \geq \Pr[\ell \in [a, \lfloor mp \rfloor + a)]$. The number of integers in this interval is $\lfloor mp \rfloor$, and so we have

$$\Pr\left[\ell \in [a, \lfloor mp \rfloor + a)\right] = \frac{\lfloor mp \rfloor}{m} = \frac{mp - \epsilon}{m} \quad \text{for some } 0 \leq \epsilon < 1,$$
$$\text{and} \quad \frac{mp - \epsilon}{m} > p - \frac{1}{m} \geq \frac{p}{2}$$

since $1/m \leq p/2$ by assumption.

If $mp = 1$ then I contains exactly one integer. In this case

$$\Pr[\ell \in I] = \frac{1}{m} = \frac{mp}{m} = p \ .$$

□

Note that $\Pr[\ell \in I] = p$ if mp is a positive integer. Also note that all our intervals will have a left closed bracket and right open bracket so that these bounds stay true if the interval bounds are integers.

Next we fix notation needed for the list merging lemma, some of which has been hinted at in the section introduction. Let L_1 and L_2 be lists of independent, uniformly generated elements of $\mathbb{Z}/m\mathbb{Z}$ with $|L_1| = |L_2| = N$. Choose $p = m^{-1/3}$, and let $N = \frac{\alpha}{p}$ where $\alpha > 1$. Later we will set $\alpha = O(n)$. Let $b_1, \ldots, b_N$ be the elements of L_1 and $c_1, \ldots, c_N$ the elements of L_2.

In considering elements of $L_1 + L_2$, the sum is performed in $\mathbb{Z}/m\mathbb{Z}$ but we then map the sums to the interval $[-\frac{m}{2}, \frac{m}{2})$ in order to determine if the sums are in the restricted interval $[-\frac{mp}{2}, \frac{mp}{2})$.

Lemma 4.4 (4-set ListMerge)**.** *Using the notation outlined above, let A be the following event: for every $b \in L_1$, there exists $c \in L_2$ such that $b + c \in [-\frac{mp}{2}, \frac{mp}{2})$. Then*

$$\Pr[A] \geq 1 - (\alpha/p)e^{-\alpha/2} \ .$$

Proof. First fix $b \in L_1$. We wish to show that with high probability there is at least one $c \in L_2$ such that $c \in I_b = [-\frac{mp}{2} - b, \frac{mp}{2} - b)$, wrapping modulo m so the interval always has length mp.

By Proposition 4.3 we have $\Pr[c \in I_b] \geq \frac{p}{2}$ and hence $\Pr[c \notin I_b] \leq 1 - \frac{p}{2}$. Since the c_j are independent, the probability that none of the c_j fall in I_b is at most

$$\left(1 - \frac{p}{2}\right)^N = \left(1 - \frac{p}{2}\right)^{\frac{2}{p} \cdot \frac{\alpha}{2}} \leq (e^{-1})^{\alpha/2} \ .$$

Now consider the event that there exists $b \in L_1$ such that no $c \in L_2$ falls in I_b. By using the first two terms of the inclusion-exclusion principle we see that the probability of this event is at most $(\alpha/p)e^{-\alpha/2}$ and thus conclude that

$$\Pr[A] \geq 1 - (\alpha/p)e^{-\alpha/2} \ .$$

□

Definition 4.5. *Call a sublist S of $L_1 + L_2$* row distinct *if $|S| = N$ and $b_1 \neq b_2$ for any two elements $b_1 + c_1$, $b_2 + c_2$ of S.*

Lemma 4.6. *Let $I = [-\frac{mp}{2}, \frac{mp}{2})$ and let S be a sublist of $L_1 + L_2$. Then conditioned on all elements being in I, the elements of S are uniformly distributed on I. If in addition S is row distinct, then, conditioned on all elements being in I, the elements of S are independent.*

Proof. Let $\lfloor mp \rfloor$ be the number of integers in I, and let ℓ be an element of S. Recall that ℓ is uniformly distributed on $\mathbb{Z}/m\mathbb{Z}$. Then for any $x \in I$,

$$\Pr[\ell = x \mid \ell \in I] = \frac{\Pr[\ell = x]}{\Pr[\ell \in I]} = \frac{1/m}{\lfloor mp \rfloor / m} = \frac{1}{\lfloor mp \rfloor}$$

and thus ℓ is uniformly distributed on I.

Next suppose that S is row distinct. Reindexing the c_j and b_j, suppose that the columns $c_1, \ldots, c_r$ support elements $\ell_1, \ldots, \ell_u$ where $\ell_j = c_i + b_j$. Note that the b_j are distinct and the ℓ_j have uniform distributions. Then

$$\begin{aligned}
&\Pr[\ell_1, \ldots, \ell_u] \\
&= \sum_{z_1} \cdots \sum_{z_r} \Pr[c_1 = z_1 \wedge \cdots \wedge c_r = z_r] \Pr[b_1 = \ell_1 - c_1 \wedge \cdots \wedge b_u = \ell_u - c_t] \\
&= m^r \cdot \left(\frac{1}{m}\right)^r \cdot \left(\frac{1}{m}\right)^u \quad \text{since the } b_j \text{ and } c_j \text{ are independent} \\
&= \Pr[\ell_1] \cdots \Pr[\ell_u] \ .
\end{aligned}$$

However, this is only a step towards our goal, because we want independence where the variables are conditioned on being in I. From the work above we easily conclude that

$$\Pr[\ell_1 \in I \wedge \cdots \wedge \ell_u \in I] = \Pr[\ell_1 \in I] \cdots \Pr[\ell_u \in I]$$

by writing the left hand side as a sum of $\Pr[\ell_1, \ldots, \ell_u]$ terms. We now conclude that

$$\begin{aligned}\Pr[\ell_1, \ldots, \ell_u \mid \ell_1, \ldots, \ell_u \in I] &= \frac{\Pr[\ell_1, \ldots, \ell_u]}{\Pr[\ell_1 \in I \wedge \cdots \wedge \ell_u \in I]} \\ &= \prod_{j=1}^{u} \frac{\Pr[\ell_j]}{\Pr[\ell_j \in I]} = \prod_{j=1}^{u} \Pr[\ell_j \mid \ell_j \in I] \quad .\end{aligned}$$

□

At this point it is worth providing some intuition for the notion of row distinct. If we accept all elements $\ell \in L_1 + L_2$ that are in the interval $[-\frac{mp}{2}, \frac{mp}{2})$, they could be highly dependent. For example, $(b_1 + c_1) + (b_2 + c_2) = (b_1 + c_2) + (b_2 + c_1)$. To visualize these dependencies, make a table with the b_j on the left and the c_j on the top. Note that ℓ_4 is determined by ℓ_1, ℓ_2, ℓ_3 since $\ell_4 = \ell_2 + \ell_3 - \ell_1$.

	c_1	c_2	$\cdots$	c_N
b_1	ℓ_1	ℓ_3		
b_2	ℓ_2	ℓ_4		
$\vdots$	$\vdots$			
b_N				

If the list S is made of ℓ_i that are all in the same column (or row), then they are still independent since the b_j are independent. Generalizing this to row distinct maintains the same property.

We next present Algorithm ListMerge, which instantiates the join operator $\bowtie$ for subset sum. ListMerge keeps more than one per row despite our previous comments, since in the analysis we only need the existence of a row distinct sublist. In order for

ListMerge to match our analysis we need the addition to act like it is in $\mathbb{Z}/m\mathbb{Z}$. This is satisfied by wrapping the interval $[-a - \frac{mp}{2}, -a + \frac{mp}{2})$ modulo m.

Algorithm 2 4-set ListMerge

Input: two lists L_1, L_2 of numbers in $\mathbb{Z}/m\mathbb{Z}$, parameter p
Output: list L_{12} of numbers $b + c \in [-\frac{mp}{2}, \frac{mp}{2})$ where $b \in L_1$, $c \in L_2$
sort L_1, L_2
for $b \in L_1$ **do**
  for $c \in L_2$ in interval $[-b - \frac{mp}{2}, -b + \frac{mp}{2})$ **do**
    $L_{12} \leftarrow L_{12} \cup \{b + c\}$
  end for
end for
output L_{12} { cut list down to N elements with at least one per $b \in L_1$}

Note that the resource usage of ListMerge is dominated by the size of L_{12} and having to sort L_1 and L_2. Assuming that the later have size N, this algorithm takes time and space $O(N \log N)$.

Algorithm 3 presents pseudocode for the 4-set Algorithm for RMSS. Note that $-t$ is set as an extra a_i in the fourth set. Its inclusion allows us to solve the problem as if it had target 0, and does not disturb our conclusions about well-distributiveness or uniformity. In proving the correctness of Algorithm 3 we need $\log m < n/4$ in order to apply Proposition 3.11, but as with the 2-set Algorithm we require a stricter condition due to technical reasons.

Theorem 4.7. *Assume $m = 2^{cn/4}$ where $c < 3/5$ (so density is greater than $20/3$). Then Algorithm 3 runs in time $O(n^4 m^{1/3})$ and space $O(n^3 m^{1/3})$, both of which are $\widetilde{O}(m^{1/3})$.*

With the a_i chosen uniformly at random from $\mathbb{Z}/m\mathbb{Z}$, the algorithm outputs a solution with probability at least

$$(1 - 4 \cdot 2^{-\frac{(3-5c)n}{48}})(1 - (2n \cdot 2^{-\frac{(6-c)n}{12}} + \beta + 4 \cdot 2^{-\frac{(3-5c)n}{48}})^n) = 1 - e^{-\Omega(n)} \ .$$

Algorithm 3 4-set algorithm for dense modular subset sum

Input: modulus m, target t, $a_1, \ldots, a_n \in \mathbb{Z}/m\mathbb{Z}$, parameter p, α
Require: $\log m < 3n/20$
Output: bit vector $\boldsymbol{x}$ such that $\sum_{i=1}^{n} a_i x_i = t \mod m$ or *Probably No Solution*
Let $a_{n+1} = -t \mod m$. {Now solve new problem with target 0}
Divide a_i into four sets indexed by I_j, $1 \leq j \leq 4$.
for $j = 1$ to 4 **do**
  for $k = 1$ to α/p **do**
    generate random bits x_i, $i \in I_j$
    store $\sum_{i \in I_j} a_i x_i$ along with the bits in L_j { set $x_{n+1} = 1$ so that $-t$ is included in all elements of I_4}
  end for
end for
$L_{12} \leftarrow \text{ListMerge}(L_1, L_2)$
$L_{34} \leftarrow \text{ListMerge}(L_3, L_4)$
sort L_{12}
for $z \in L_{34}$ **do**
  search for $-z$ in L_{12} using binary search
  if collision found **then**
    output $\boldsymbol{x} = (x_1, \ldots, x_n)$
    break; { that is, end the algorithm}
  end if
end for
repeat steps above n times
output *Probably No Solution*

Proof. We first work to prove correctness. As with the 2-set algorithm, the probability of success is the probability that all four subsets of the a_i are well-distributed, times the probability of success given that the subsets are well-distributed. By replacing n with $n/4$ in Proposition 3.11 and making a parameter choice of $\gamma = -\frac{(3-5c)n}{48}$ we see that the probability that all four subsets are well-distributed is greater than

$$(1 - 2^{-\frac{(3-5c)n}{48}})^4 > 1 - 4 \cdot 2^{-\frac{(3-5c)n}{48}}$$

and the distance between these distributions and uniform ones is less than

$$2^{-\frac{(1-c)n}{8} + \frac{(3-5c)n}{48}} = 2^{-\frac{(3-c)n}{48}} \ .$$

Now assume that all elements of each of the L_j are drawn from uniform distributions. Success in the algorithm after one trial means that the lists L_{12} and L_{34} each contain a row distinct sublist in an interval of length $mp = m^{2/3}$, and that these two lists have a collision. It might be the case that L_{12} and L_{34} have more than just a row distinct sublist, but that only increases the probability of a collision.

By Lemma 4.4, the probability that L_{12} and L_{34} both contain a row distinct sublist with elements contained in $I = [-\frac{mp}{2}, \frac{mp}{2})$ is at least $(1 - (\alpha/p)e^{-\alpha/2})^2$. Call these sublists S_{12} and S_{34}. By Lemma 4.6, given that the elements all fall in I, we know that the elements of S_{12} and S_{34} are independent and uniformly generated. Each list contains $m^{1/3}$ elements and the interval is of size $m^{2/3}$, and so the probability of a collision is at least $1 - \beta$ by Lemma 4.1.

Thus the probability of success assuming the L_j contain uniform elements is at least

$$(1 - (\alpha/p)e^{-\alpha/2})^2(1 - \beta) \geq 1 - 2\alpha 2^{cn/12}e^{-\alpha/2} - \beta \ .$$

We next account for the fact that the elements of L_j are only close to uniform. One trial requires a total of $4m^{1/3}$ elements of $\mathbb{Z}/m\mathbb{Z}$. Define $X_{\boldsymbol{a}_j}$, $1 \leq j \leq 4$ by 3.2 as we did in the 2-set case. Define random variables X_i and U_i, $1 \leq i \leq 4m^{1/3}$ so that U_i is the uniform distribution on $\mathbb{Z}/m\mathbb{Z}$ and a quarter of the X_i are copies of $X_{\boldsymbol{a}_j}$ for each j.

Let the predicate f take value 1 if Algorithm 3 fails with lists L_j filled with the X_i. Then by Proposition 3.5 and Proposition 3.6 we have

$$|\Pr[f(X_1, \ldots, X_{4m^{1/3}}) = 1] - \Pr[f(U_1, \ldots, U_{4m^{1/3}}) = 1]| \leq \sum_{i=1}^{4m^{1/3}} \Delta(X_i, U_i) \ .$$

Applying Proposition 3.11, this quantity is less than $4m^{1/3} \cdot 2^{-\frac{(3-c)n}{48}} = 4 \cdot 2^{-\frac{(3-5c)n}{48}}$.

Given well-distribution, the probability of failure after one trial is at most

$$2\alpha 2^{cn/12} e^{-\alpha/2} + \beta + 4 \cdot 2^{-\frac{(3-5c)n}{48}} \ .$$

Given $c < 3/5$, $\alpha = n$, and n large enough this probability is bounded away from 1. So the probability of failure after n trials is at most

$$(2n2^{-\frac{(6-c)n}{12}} + \beta + 4 \cdot 2^{-\frac{(3-5c)n}{48}})^n$$

and hence the probability of success for the entire algorithm is at least

$$(1 - 4 \cdot 2^{-\frac{(3-5c)n}{48}})(1 - (2n2^{-\frac{(6-c)n}{12}} + \beta + 4 \cdot 2^{-\frac{(3-5c)n}{48}})^n) \ .$$

If $c < 3/5$ we can rewrite this as $1 - e^{-\Omega(n)}$.

Finally, we analyze the resource usage given our parameter choice of $p = 1/m^{1/3}$, $\alpha = n$. The storage is $4\alpha m^{1/3}$ entries, each of which is $n/4$ bits and an element of $\mathbb{Z}/m\mathbb{Z}$. For time complexity, computing the sets L_j takes $n \log m$ bit operations for each of the $4\alpha m^{1/3}$ elements. Next is the cost of ListMerge applied twice which is

$O(nm^{1/3} \log (nm^{1/3}))$. The final step is α/p searches for a cost of $O(nm^{1/3} \log (nm^{1/3}))$. The total running time is thus dominated by the cost of making the initial sets, which is $O(n \cdot n^3 \cdot m^{1/3})$ bit operations. □

Chapter 5

The k-set Algorithm

The goal of this chapter is to provide a new algorithm that applies the k-set birthday problem to RMSS, one that improves the $\widetilde{O}(m^{2/\log k})$ time and space algorithm of [20] for RMSS problems of high density.

The key step is to generalize the 4-set list merging lemma so that it works even when the random variables involved are no longer independent or uniform. We instead show that the random variables are close to uniform and close to pairwise independent, and this will be enough to apply a tail bound. The algorithm itself is essentially Wagner's algorithm for the k-set birthday problem. See Section 3.3 for a general introduction.

5.1 Symmetric Unimodal Distributions

In this section we develop tools to describe the distributions that arise in the k-set algorithm. We again assume all distributions are discrete, though the definitions transfer easily to the continuous case.

Definition 5.1. *Let independent random variables X and Y have distributions F and G with probability mass functions f and g.*

1. *The distribution F is* symmetric *about the origin if $f(-x) = f(x)$ for all x.*

2. *The distribution F is* unimodal *at a if f is nondecreasing on $(-\infty, a]$ and nonincreasing on $[a, \infty)$.*

3. *The* convolution *of F and G, denoted $F * G$, is defined by*

$$f * g(s) = \sum_x f(x)g(s - x)$$

where the sum is over the probability space (for ease of notation, this is extended to $(-\infty, \infty)$).

From now on we take symmetric, unimodal to mean symmetric and unimodal about the origin. The following are standard facts from probability theory.

Proposition 5.2. *Let X and Y be independent random variables with discrete distributions F and G given by functions f and g.*

1. *The random variable $S = X + Y$ has distribution $F * G$.*

2. *If X and Y are symmetric, $F * G$ is symmetric.*

3. *If X and Y are symmetric and unimodal, $F * G$ is unimodal.*

Proof. 1. We need to find the probability that S has value s, i.e. $f_S(s)$. We have that $f_S(s) = \sum_x f(x)g(s-x)$ since X and Y are independent. Thus $f_S(s) = f * g$ and so the distribution of S is $F * G$.

2. Consider $f * g(-s) = \sum_x f(x)g(-s - x)$. The sum stays fixed if we sum in the opposite direction, so this equals $\sum_x f(-x)g(-s + x)$. But F and G are symmetric, so we have $\sum_x f(x)g(s - x)$. Thus $f * g(-s) = f * g(s)$, and so $F * G$ is symmetric.

3. It appears that the standard proof is quite complicated. Here we give a discrete version of a simpler proof by Purkayastha [24].

It suffices to show that $f * g$ is nonincreasing on $[0, \infty)$ since we know that $f * g$ is symmetric. Let X be a random variable with density f. We have that

$$f * g(x) = E[g(x - X)] = E[g(X - x)]$$

since g is symmetric. Define random variables $Y_x = g(X-x)$ and note that $E[g(X-x)] = \sum_{u \geq 0} \Pr[Y_x > u]$.

Now, the set $\{s : \ g(s) > u\}$ is an interval since g is unimodal. Therefore

$$\Pr[Y_x > u] = \Pr[x - t < X < x + t]$$

for some t depending on u, where here we rely on f being symmetric. But as a function of x, $\Pr[x-t < X < x+t]$ is nonincreasing on $[0, \infty)$ since X has a unimodal distribution (this is best seen graphically by comparing areas). Therefore $\Pr[Y_x > u]$ is nonincreasing for all u and since $f * g(x) = \sum_{u \geq 0} \Pr[Y_x > u]$, the result follows.

□

The following simple result is surprisingly useful.

Proposition 5.3. *Let f be symmetric unimodal. Then $f * f(0) \leq f(0)$.*

Proof. By the definitions of the terms used and the fact that f is a probability mass function we have

$$f * f(0) = \sum_x f(x)f(-x) \leq \sum_x f(0)f(x) = f(0) \sum_x f(x) = f(0) \ .$$

□

While the distributions we will study are symmetric unimodal, they are constructed by a particular process. The process starts with uniform distributions. At each step two

copies of the distribution are convolved, the tails are thrown out, and the remainder is normalized to make a new probability distribution.

In particular, let D_0 have the uniform distribution on $[-\frac{m}{2}, \frac{m}{2})$ with probability mass function $f_0 : x \mapsto \frac{1}{m}$. Set a parameter $p < 1$ such that $1/p = m^{1/\log k}$. For $0 \leq \lambda \leq \log k - 1$ let D_λ have probability mass function f_λ which is supported on $[-\frac{R}{2}, \frac{R}{2}) = [-\frac{mp^\lambda}{2}, \frac{mp^\lambda}{2})$. Then f_λ is supported on $[-\frac{Rp}{2}, \frac{Rp}{2})$ and is defined by

$$f_\lambda(x) = \frac{1}{A_{\lambda-1}} f_{\lambda-1} * f_{\lambda-1}(x) = \frac{1}{\sum_{a=-Rp/2}^{Rp/2} f_{\lambda-1} * f_{\lambda-1}(a)} \cdot f_{\lambda-1} * f_{\lambda-1}(x)$$

where summing over an interval will always mean summing over the integers in the interval.

Note that $\sum_{x=-Rp/2}^{Rp/2} f_\lambda(x) = 1$ and so f_λ really is a probability distribution. By Proposition 5.2 the f_λ are all symmetric unimodal. Also note that $f_{\log k}$ is supported on $[-\frac{1}{2}, \frac{1}{2})$.

The next result will be used often in technical results that follow. Symmetric unimodal distributions have the nice property that they come with an easy lower bound for area about the origin.

Proposition 5.4. *Let X be a random variable with a symmetric unimodal distribution on the interval $[-R, R)$. Then* $\Pr[X \in [-\frac{Rp}{2}, \frac{Rp}{2})] \geq p/4$.

Proof. Since the distribution is symmetric unimodal, $\Pr[X \in [-\frac{Rp}{2}, \frac{Rp}{2})] \cdot a \geq 1$, where a is the proportion of the number of integers in $[-\frac{Rp}{2}, \frac{Rp}{2})$ to the number of integers in $[-\frac{R}{2}, \frac{R}{2})$. We have

$$\frac{1}{a} = \frac{\lfloor Rp \rfloor}{\lfloor 2R \rfloor} \geq \frac{Rp - \epsilon}{2R} \geq \frac{p}{2} - \frac{1}{2R}$$

for some $\epsilon < 1$. This will be greater than $\frac{p}{4}$ as long as $\frac{1}{2R} \leq \frac{p}{4}$ which means $Rp \geq 2$. We might have $Rp = 1$, but in this case $R = 1/p$ and the result still holds. □

A very surprising and useful fact is that since λ is small, f_λ is always close to a uniform distribution . The following lemma supports this claim if we assign the parameter values used in the analysis of Algorithm k-set. With $p = 2^{-cn/(k \log k)}$, $n > O(k(\log k)^2)$, we see that $p \leq 1/(24k^3)$ as needed. Since $\lambda < \log k$ this assumption allows us to conclude that for all λ, a term such as $6^\lambda p$ is exponentially small and $6^\lambda p + 6^{2\lambda} p^2$ is bounded by $2 \cdot 6^\lambda p$.

Lemma 5.5. *Let U be the uniform distribution on $[-\frac{mp^\lambda}{2}, \frac{mp^\lambda}{2})$, and assume that $p \leq 1/(24k^3)$. Then*

$$|f_\lambda(x) - U(x)| \leq \frac{6^\lambda p}{mp^\lambda} \ .$$

Proof. We will prove the theorem by induction on λ. For the base case, $f_0(x) = 1/m$ for all x and thus $|f_0(x) - 1/m| = 0$ and the claim holds. Assume by induction that $|f_{\lambda-1}(x) - 1/mp^{\lambda-1}| \leq 6^{\lambda-1}/(mp^{\lambda-2})$.

Now for the inductive step. Consider

$$A_{\lambda-1} = \sum_{a=-mp^\lambda/2}^{mp^\lambda/2} f_{\lambda-1} * f_{\lambda-1}(a) = \sum_{a=-mp^\lambda/2}^{mp^\lambda/2} \sum_{y=-mp^{\lambda-1}/2}^{mp^{\lambda-1}/2} f_{\lambda-1}(y) f_{\lambda-1}(a-y) \ .$$

Since f_λ is close to uniform, the term $\sum_y f_{\lambda-1}(y) f_{\lambda-1}(a-y)$ is close to the triangle distribution, and so $f_{\lambda-1} * f_{\lambda-1}(a)$ is close to $\frac{mp^{\lambda-1}-a}{(mp^{\lambda-1})^2}$. Rigorously we have by induction that

$$\frac{1 - 6^{\lambda-1}p}{mp^{\lambda-1}} \leq f_{\lambda-1}(x) \leq \frac{1 + 6^{\lambda-1}p}{mp^{\lambda-1}} \quad \text{which implies}$$

$$\frac{(mp^{\lambda-1} - |a|)(1 - 6^{\lambda-1}p)^2}{(mp^{\lambda-1})^2} \leq \sum_{y=-mp^{\lambda-1}}^{mp^{\lambda-1}} f_{\lambda-1}(y) f_{\lambda-1}(a-y) \quad \text{and}$$

$$\sum_{y=-mp^{\lambda-1}}^{mp^{\lambda-1}} f_{\lambda-1}(y) f_{\lambda-1}(a-y) \leq \frac{(mp^{\lambda-1} - |a|)(1 + 6^{\lambda-1}p)^2}{(mp^{\lambda-1})^2} \leq \frac{(1 + 6^{\lambda-1}p)^2}{mp^{\lambda-1}}$$

and so summing over a yields

$$A_{\lambda-1} \geq \frac{mp^{\lambda-1}(1-p/2+mp^{\lambda}(1-p/4))(1-6^{\lambda-1}p)^2}{(mp^{\lambda-1})^2} \geq p(1-p)(1-6^{\lambda-1}p)^2$$
$$\text{and} \quad A_{\lambda-1} \leq mp^{\lambda} \cdot \frac{(1+6^{\lambda-1}p)^2}{mp^{\lambda-1}} = p(1+6^{\lambda-1}p)^2 \ .$$

Now, since $f_{\lambda-1} * f_{\lambda-1}$ is symmetric unimodal, the quantity $|\frac{1}{A_{\lambda-1}} f_{\lambda-1} * f_{\lambda-1}(x) - U(x)|$ is greatest at $x = 0$ and $x = \pm mp^{\lambda}/2$. For $x = 0$ we maximize it by giving a lower bound for $A_{\lambda-1}$ and upper bound for $f_{\lambda-1} * f_{\lambda-1}(0)$ to achieve

$$\begin{aligned}\left|\frac{1}{A_{\lambda-1}} f_{\lambda-1} * f_{\lambda-1}(0) - \frac{1}{mp^{\lambda}}\right| &\leq \left|\frac{1}{p(1-p)(1-6^{\lambda-1}p)^2} \cdot \frac{(1+6^{\lambda-1}p)^2}{mp^{\lambda-1}} - \frac{1}{mp^{\lambda}}\right| \\ &\leq \left|\frac{1+6^{\lambda}p}{mp^{\lambda}} - \frac{1}{mp^{\lambda}}\right| \\ &\leq \frac{6^{\lambda}p}{mp^{\lambda}}\end{aligned}$$

where we have used the fact that $p \leq \frac{1}{24k^3} \leq \frac{1}{24 \cdot 6^{\log_2 k}}$ implies

$$\frac{(1+6^{\lambda-1}p)^2}{(1-p)(1-6^{\lambda-1}p)^2} \leq 1 + 6 \cdot 6^{\lambda-1}p \ .$$

For $x = mp^{\lambda}/2$ we give an upper bound for A and a lower bound for $f_{\lambda-1} * f_{\lambda-1}(mp^{\lambda}/2)$ to make $\frac{1}{A} f_{\lambda-1} * f_{\lambda-1}(mp^{\lambda}/2)$ as small as possible.

$$\begin{aligned}&\left|\frac{1}{A} f_{\lambda-1} * f_{\lambda-1}(mp^{\lambda}/2) - \frac{1}{mp^{\lambda}}\right| \\ &\leq \left|\frac{1}{p(1+6^{\lambda-1}p)^2} \cdot \frac{(mp^{\lambda-1} - mp^{\lambda}/2)(1-6^{\lambda-1}p)^2}{(mp^{\lambda-1})^2} - \frac{1}{mp^{\lambda}}\right| \\ &= \left|\frac{(1-p/2)(1-6^{\lambda-1}p)^2}{(1+6^{\lambda-1}p)^2 mp^{\lambda}} - \frac{1}{mp^{\lambda}}\right| \\ &\leq \left|\frac{1-6^{\lambda}p}{mp^{\lambda}} - \frac{1}{mp^{\lambda}}\right| \\ &= \frac{6^{\lambda}p}{mp^{\lambda}}\end{aligned}$$

where we have used the fact that $p \leq 1$ implies

$$\frac{(1-p/2)(1-6^{\lambda-1}p)^2}{(1+6^{\lambda-1}p)^2} \geq 1 - 6 \cdot 6^{\lambda-1}p \ .$$

This finishes the induction.

□

As an easy corollary we conclude that

$$\Delta(f_i, U) \leq \frac{mp^\lambda \cdot 6^\lambda}{2mp^{\lambda-1}} \leq 6^\lambda \cdot p \ .$$

We also get as a corollary the following result which complements Proposition 5.4.

Proposition 5.6. *Let X_λ be a random variable with distribution $f_\lambda * f_\lambda$ (defined above) which is supported on the interval $[-R, R) = [-mp^\lambda, mp^\lambda)$. Assume that $p \leq 1/(4k^3)$. Then $\Pr[X_\lambda \in [-\frac{Rp}{2}, \frac{Rp}{2})] \leq 2p$.*

Proof. Using the fact that f_λ is symmetric, we have that

$$f_\lambda * f_\lambda(0) = \sum_{x=-R/2}^{R/2} f_\lambda(x) f_\lambda(-x) = \sum_{x=-R/2}^{R/2} f_\lambda(x)^2 \ .$$

By Lemma 5.5 we know that $f_\lambda(x)$ is close to $\frac{1}{mp^\lambda}$, and in fact we have a good upper bound. Thus

$$\sum_{x=-R/2}^{R/2} f_\lambda(x)^2 \leq mp^\lambda \cdot \left(\frac{1+6^\lambda p}{mp^\lambda}\right)^2 = \frac{1}{mp^\lambda} + \frac{2 \cdot 6^\lambda p}{mp^\lambda} + \frac{6^{2\lambda}p^2}{mp^\lambda} \ .$$

Since $f_\lambda * f_\lambda$ is unimodal and $p \leq \frac{1}{4k^3}$, we have

$$\begin{aligned} \Pr\Big[X_\lambda \in \Big[-\frac{Rp}{2}, \frac{Rp}{2}\Big)\Big] &\leq mp^{\lambda+1} \cdot f_\lambda * f_\lambda(0) \\ &\leq p + 2 \cdot 6^\lambda p^2 + 6^{2\lambda}p^3 \leq 2p \ . \end{aligned}$$

□

5.2 Almost Pairwise Independent

In the previous section we showed that the distributions f_λ that arise in the k-set algorithm are close to uniform. In this section we show that the elements generated are close to pairwise independent, which will enable us to apply the Chebyshev inequality.

First we fix notation. Choose $k < n$ so that $n > O(k(\log k)^2)$. Let the modulus $m = 2^{cn/k}$ for $c < 1$, so that m is exponential in n (in fact c will need to be smaller for the same technical reasons that arose in the 2-set and 4-set algorithms). We choose parameter $p = m^{-1/\log k}$ and $\alpha = O(n)$. At level λ of the algorithm (note that $\lambda < \log k$), we have lists L_1 and L_2 of integers in the interval $[-\frac{mp^\lambda}{2}, \frac{mp^\lambda}{2})$ with $|L_1| = |L_2| = N = \frac{\alpha}{p}$. Let b be the elements of L_1, c the elements of L_2, and ℓ be the elements of $L_1 + L_2$. Let I be the interval $[-\frac{mp^{\lambda+1}}{2}, \frac{mp^{\lambda+1}}{2})$, and define bernoulli random variables X_i where $X_i = 1$ if $\ell_i \in I$ and 0 otherwise. Let $S = \sum_{i=1}^{N^2} X_i$.

As a guide to intuition we will often use the idea of relating the b_i, c_i, and ℓ_i in a table as was done in Section 4.2. We noted there that the ℓ_i had no hope of being independent due to functional dependencies built into the table. However, if the b_i and c_i are independent then the ℓ_i are pairwise independent. If we could guarantee that the ℓ_i that pass on to the next level never share a row or column, then the elements at every level would remain pairwise independent. In the next lemma we quantify how much is lost due to the nonuniformity of the ℓ_i and the fact that they might share a row or column.

Lemma 5.7. *Assume we are at level λ, with the notation defined above. For all i, j between 1 and N^2 we have*

$$\frac{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}} \le \frac{\Pr[\ell_i, \ell_j \mid X_i = 1, X_j = 1]}{\Pr[\ell_i \mid X_i = 1]\Pr[\ell_j \mid X_j = 1]} \le \frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}$$

assuming that both numerator and denominator are nonzero, and that $p \leq 1/(4k^3)$.

Proof. We will prove this by induction on the level, so that for the base case we assume the elements of L_1 and L_2 are independent, uniformly generated.

If ℓ_i and ℓ_j share neither a row nor a column, they are independent and we are done. Assume then that (without loss of generality) ℓ_i and ℓ_j share a row. Let $\ell_i = b + c_i$ and $\ell_j = b + c_j$. Also assume that $\ell_i, \ell_j \in I$ since otherwise both terms are zero. We have

$$\begin{aligned}\Pr[\ell_i, \ell_j] &= \sum_{z=-m/2}^{m/2} \Pr[b = z \wedge c_i = \ell_i - z \wedge c_j = \ell_j - z] \\ &\leq \frac{1}{m} \sum_{z=-m/2}^{m/2} \Pr[b = z] \Pr[c_i = \ell_i - z] = \frac{1}{m} \Pr[\ell_i]\end{aligned}$$

since the first sum has more restrictions, and hence more terms that are zero. However, more extra zero terms occur the farther ℓ_j is from 0, and this is limited by assumption. In particular, $\ell_j \in I$ gives us that $\Pr[\ell_i, \ell_j] \geq \frac{1-p}{m} \Pr[\ell_i]$.

Again using the fact that $\ell_i \in I$, computing the convolution reveals that

$$\begin{gathered}\frac{1-p}{m} \leq \Pr[\ell_i] \leq \frac{1}{m} \quad \text{and thus} \\ (1-p) \Pr[\ell_i] \Pr[\ell_j] \leq \Pr[\ell_i, \ell_j] \leq \frac{1}{1-p} \Pr[\ell_i] \Pr[\ell_j] \ .\end{gathered}$$

It easily follows from summing terms of this form that

$$(1-p) \Pr[X_i = 1] \Pr[X_j = 1] \leq \Pr[X_i = 1 \wedge X_j = 1] \leq \frac{\Pr[X_i = 1] \Pr[X_j = 1]}{1-p} \ .$$

Note that as long as $\ell_i, \ell_j \in I$ we have

$$\frac{\Pr[\ell_i, \ell_j \mid X_i = 1, X_j = 1]}{\Pr[\ell_i \mid X_i = 1] \Pr[\ell_j \mid X_j = 1]} = \frac{\Pr[\ell_i, \ell_j]}{\Pr[\ell_i] \Pr[\ell_j]} \cdot \frac{\Pr[X_i = 1] \Pr[X_j = 1]}{\Pr[X_i = 1, X_j = 1]}$$

from which we conclude that

$$(1-p)^2 \leq \frac{\Pr[\ell_i, \ell_j \mid X_i = 1, X_j = 1]}{\Pr[\ell_i \mid X_i = 1] \Pr[\ell_j \mid X_j = 1]} \leq \frac{1}{(1-p)^2} \ .$$

For the inductive step we use the same calculation, but we must account for the fact that the b_j and c_j have distributions which are only close to uniform, and splitting apart c_j terms introduces a factor close to 1 by induction. Let the inductive lower and upper bounds be

$$L_{\lambda-1} = \frac{(1-p)^{4^{\lambda-1}}(1-3\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}}{(1+4\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}} \quad \text{and} \quad U_{\lambda-1} = \frac{(1+4\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}}{(1-p)^{4^{\lambda-1}}(1-3\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}} \ .$$

Assume first that ℓ_i and ℓ_j do not share a row or column. We have

$$\begin{aligned}\Pr[\ell_i,\ell_j] &= \sum_{z_1}\sum_{z_2}\Pr[b_i = z_1 \wedge c_i = \ell_i - z_1 \wedge b_j = z_2 \wedge c_j = \ell_j - z_2] \\ &= \sum_{z_1}\sum_{z_2}\Pr[b_i = z_1 \wedge b_j = z_2]\Pr[c_i = \ell_i - z_1 \wedge c_j = \ell_j - z_2]\end{aligned}$$

since the b_j and c_j are independent of each other. By induction we have upper bounds and lower bounds for how close b_i, b_j are to pairwise independent. Thus

$$L_{\lambda-1}^2 \Pr[\ell_i]\Pr[\ell_j] \le \Pr[\ell_i,\ell_j] \le U_{\lambda-1}^2 \Pr[\ell_i]\Pr[\ell_j]$$

and similar reasoning applied to sums of such terms allows us to conclude that

$$\frac{L_{\lambda-1}^2}{U_{\lambda-1}^2} \le \frac{\Pr[\ell_i,\ell_j \mid X_i = 1, X_j = 1]}{\Pr[\ell_i \mid X_i = 1]\Pr[\ell_j \mid X_j = 1]} \le \frac{U_{\lambda-1}^2}{L_{\lambda-1}^2} \ .$$

Now assume that ℓ_i and ℓ_j do share a row. As in the base case we have

$$\Pr[\ell_i,\ell_j] = \sum_z \Pr[b = z \wedge c_i = \ell_i - z \wedge c_j = \ell_j - z]$$

and applying induction to break off the $c_j = \ell_j - z$ term (taking into account the possibly different number of terms in the two expressions) yields

$$L_{\lambda-1}(1-p)\Pr[c_j = \ell_j - z]\Pr[\ell_i] \le \Pr[\ell_i,\ell_j] \le U_{\lambda-1}\Pr[c_j = \ell_j - z]\Pr[\ell_i] \ .$$

Lemma 5.5 gives bounds on how close $\Pr[c_j = \ell_j - z]$ is to $\Pr[b_j + c_j = \ell_j]$, namely

$$\frac{1}{mp^\lambda} - \frac{6^\lambda p}{mp^\lambda} \le \Pr[c_j = \ell_j - z] \le \frac{1}{mp^\lambda} + \frac{6^\lambda p}{mp^\lambda} \quad \text{and since } \ell_j \in I$$
$$mp^\lambda(1-p)\left(\frac{1}{mp^\lambda} - \frac{6^\lambda p}{mp^\lambda}\right)^2 \le \Pr[\ell_j] \le mp^\lambda\left(\frac{1}{mp^\lambda} + \frac{6^\lambda p}{mp^\lambda}\right)^2$$

where the bound on $\Pr[\ell_j]$ comes from counting the number of nonzero terms $\Pr[b_j = z]\Pr[c_j = \ell_j - z]$ and applying the uniform bound from Lemma 5.5.

It is a fact that $(\frac{1}{x} + \epsilon)(1 - 3\epsilon x) \le x(\frac{1}{x} - \epsilon)^2$ assuming x and ϵ are positive. With the additional assumption that $\epsilon > 4\epsilon^2 x$ we also have $x(\frac{1}{x} + \epsilon)^2 \le (\frac{1}{x} - \epsilon)(1 + 4\epsilon x)$. Using these facts with $x = mp^\lambda$, $\epsilon = \frac{6^\lambda p}{mp^\lambda}$ we see that

$$\Pr[\ell_j] \le \Pr[c_j = \ell_j - z]\left(1 + 4 \cdot 6^\lambda p\right) \quad \text{and} \quad \Pr[c_j = \ell_j - z]\left(1 - 3 \cdot 6^\lambda p\right) \le \frac{\Pr[\ell_j]}{1-p}$$
$$\text{and thus} \quad \frac{\Pr[\ell_j]}{1 + 4 \cdot 6^\lambda p} \le \Pr[c_j = \ell_j - z] \le \frac{\Pr[\ell_j]}{(1-p)(1 - 3 \cdot 6^\lambda p)} \ .$$

We now plug this new information into the previous expression to get

$$L_{\lambda-1} \cdot \frac{1-p}{1 + 4 \cdot 6^\lambda p} \le \frac{\Pr[\ell_i, \ell_j]}{\Pr[\ell_i]\Pr[\ell_j]} \le \frac{1}{(1-p)(1 - 3 \cdot 6^\lambda p)} \cdot U_{\lambda-1}$$

and similar bounds for $\Pr[X_i = 1, X_j = 1]/(\Pr[X_i = 1]\Pr[X_j = 1])$ allows us to conclude

$$\frac{L_{\lambda-1}}{U_{\lambda-1}} \frac{(1-p)^2(1 - 3 \cdot 6^\lambda p)}{(1 + 4 \cdot 6^\lambda p)} \le \frac{\Pr[\ell_i, \ell_j \mid X_i = 1, X_j = 1]}{\Pr[\ell_i \mid X_i = 1]\Pr[\ell_j \mid X_j = 1]} \le \frac{1 + 4 \cdot 6^\lambda p}{(1-p)^2(1 - 3 \cdot 6^\lambda p)} \frac{U_{\lambda-1}}{L_{\lambda-1}}$$

which finishes the induction. □

It requires very little additional work to get a similar statement with expected values, which is necessary since we wish to find an upper bound for the variance of S.

Lemma 5.8. *Assume that $p \le 1/(864k^5)$. For all i, j from 1 to N^2 we have*

$$\mathrm{E}[X_i X_j] \le \delta \mathrm{E}[X_i]\mathrm{E}[X_j]$$

where $\delta = 1 + 2 \cdot 24^\lambda p$.

Proof. Since X_iX_j is zero unless both ℓ_i and ℓ_j are in I, we have

$$\begin{aligned}
\mathrm{E}[X_iX_j] &= \sum_{\ell_i\in I}\sum_{\ell_j\in I}\Pr[\ell_i\wedge\ell_j] \\
&\le \sum_{\ell_i\in I}\sum_{\ell_j\in I}\frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}\Pr[\ell_i]\Pr[\ell_j] \\
&= \frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}\mathrm{E}[X_i]\mathrm{E}[X_j]\ .
\end{aligned}$$

It is important to note that Lemma 5.7 gives us that the components of ℓ_i, ℓ_j are pairwise independent. Using the fact that the values of ℓ_i, ℓ_j are in I, we then use the same argument as Lemma 5.7 to prove that ℓ_i, ℓ_j are close to pairwise independent (without conditioning on the fact that $\ell_i, \ell_j \in I$).

To estimate the constant we use the first terms of the respective Taylor expansions and the fact that $p \le 1/N$ implies $(1-p)^N \ge 1-Np$ to get

$$\begin{aligned}
&\frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}} \\
&\le \frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-4^\lambda p)(1-18\cdot 24^{\lambda-1}p)} \\
&\le (1+24\cdot 24^{\lambda-1}p+(24\cdot 24^{\lambda-1}p)^2+\cdots)(1+4^\lambda p+(4^\lambda p)^2+\cdots)\times \\
&\qquad\qquad (1+18\cdot 24^{\lambda-1}p+(18\cdot 24^{\lambda-1}p)^2+\cdots) \\
&\le (1+24^\lambda p+2\cdot(24^\lambda p)^2)(1+4^\lambda p+2\cdot(4^\lambda p)^2)(1+18\cdot 24^{\lambda-1}p+2\cdot(18\cdot 24^{\lambda-1}p)^2)
\end{aligned}$$

where for the last step our assumption that $p \le \frac{1}{2\cdot 24^\lambda p}$ gives us

$$(24^\lambda p)^2+(24^\lambda p)^3+(24^\lambda p)^4+\cdots \le (24^\lambda p)^2\Big(1+\frac{1}{2}+\Big(\frac{1}{2}\Big)^2+\cdots\Big) = 2\cdot(24^\lambda p)^2$$

and similarly for the other two terms under the same assumption.

A good bound for δ is now

$$
\begin{aligned}
&(1+24^\lambda p+2\cdot(24^\lambda p)^2)(1+4^\lambda p+2\cdot(24^\lambda p)^2)(1+18\cdot 24^{\lambda-1}p+2\cdot(24^\lambda p)^2)\\
&\le 1+(24^\lambda+4^\lambda+18\cdot 24^{\lambda-1})p+9(24^\lambda p)^2+13(24^\lambda p)^3+18(24^\lambda p)^4+12(24^\lambda p)^5+8(24^\lambda p)^6\\
&\le 1+(24^\lambda+4^\lambda+18\cdot 24^{\lambda-1})p+24^{\lambda-1}p\\
&\le 1+2\cdot 24^\lambda p\ .
\end{aligned}
$$

Here we require that $p \le \frac{1}{2\cdot 18\cdot 24\cdot 24^\lambda}$, which is satisfied by our assumption that $p \le \frac{1}{864k^5}$. □

5.3 The k-set Algorithm

This section offers a generalization of the algorithms in the previous chapter, and continues the trend of attaining faster speeds at the cost of a more restrictive assumption on the density. For problems of density greater than $k(1+\frac{4}{\log k})$, this new algorithm splits the input vector $\boldsymbol{a}$ into k sets in order to attain time and space bounds of $\widetilde{O}(m^{1/\log k})$. This is a significant improvement over the algorithm in [20], which for problems of the same density has a resource bound of $\widetilde{O}(m^{2/\log k})$.

We will continue to use the notation of the previous section, though sometimes the interval $I=[-\frac{mp^\lambda}{2},\frac{mp^\lambda}{2})$ at level λ will be shortened to $[-\frac{R}{2},\frac{R}{2})$.

Lemma 5.9. *Suppose that L_1 and L_2 are lists of integers in the range $[-\frac{R}{2},\frac{R}{2})$, drawn from symmetric unimodal distributions, with $|L_1|=|L_2|=\frac{\alpha}{p}$ for $p<1$. Then the elements of $L_3=L_1+L_2$ also follow a symmetric unimodal distribution, and the expected number of ℓ_i in the restricted interval $[-\frac{Rp}{2},\frac{Rp}{2})$ satisfies*

$$\frac{\alpha^2}{4p} \le \mathrm{E}[S] \le \frac{2\alpha^2}{p}\ .$$

Proof. The distribution of the elements of L_3 is the convolution of the distributions of the elements of L_1 and L_2. The fact that this new distribution is also symmetric unimodal follows from Proposition 5.2. By Propositions 5.4, 5.6 the probability that an element ℓ of L_3 is in $[-\frac{Rp}{2}, \frac{Rp}{2})$ is at least $p/4$ and at most $2p$.

Now let $\ell_1, \ldots, \ell_{N^2}$ be all $(\alpha/p)^2$ elements of $L_1 + L_2$. Then the expected number of elements in the restricted set is

$$\mathrm{E}[S] = \mathrm{E}\left[\sum_{i=1}^{N^2} X_i\right] = \sum_{i=1}^{N^2} \mathrm{E}[X_i] \quad \text{which gives}$$
$$\left(\frac{\alpha}{p}\right)^2 \cdot \frac{p}{4} \leq \mathrm{E}[S] \leq \left(\frac{\alpha}{p}\right)^2 \cdot 2p$$

since $\mathrm{E}[X_i] = \Pr[\ell_i \in [-\frac{Rp}{2}, \frac{Rp}{2})]$. □

A key thing to note about this lemma is that it works regardless of the dependencies between the ℓ_i. This is the reason why the expected success rate of the algorithm is so much easier to analyze. Intuitively, this expectation should be reached in the vast majority of cases. However, this can be fiendish to nail down.

Note that the 4-set list merging lemma required that the merged list contained a row distinct subset. This is not necessary now since the nearly pairwise independent result holds for any two ℓ_i in the table.

Lemma 5.10 (k-set ListMerge)**.** *At least $N = \alpha/p$ elements of $L_1 + L_2$ fall in I with probability at least*

$$1 - 576 \cdot 24^{\lambda} \cdot 2^{-cn/(k \log k)} \quad .$$

Proof. We first calculate bounds on the expected value and variance of S. By Lemma 5.9 we know that $\frac{p}{4} \leq \mathrm{E}[X] \leq 2p$ and $\frac{\alpha^2}{4p} \leq \mathrm{E}[S] \leq \frac{2\alpha^2}{p}$. It remains to calculate an upper

bound on the variance, for which Lemma 5.8 will prove essential. By definition,

$$\begin{aligned}
\sigma^2 &= \mathrm{E}[S^2] - \mathrm{E}[S]^2 \\
&= \sum_{i=1}^{N^2} \sum_{j \neq i} \mathrm{E}[X_i X_j] + \sum_{i=1}^{N^2} \mathrm{E}[X_i^2] - N^4 \mathrm{E}[X]^2 \\
&\leq \delta(N^4 - N^2)\mathrm{E}[X]^2 + N^2 \mathrm{E}[X] - N^4 \mathrm{E}[X]^2 \\
&= (\delta - 1) N^4 \mathrm{E}[X]^2 - \delta N^2 \mathrm{E}[X]^2 + N^2 \mathrm{E}[X] \ .
\end{aligned}$$

Now substituting α/p for N and the proper bounds for $\mathrm{E}[X]$,

$$\begin{aligned}
\sigma^2 &\leq (\delta - 1)\frac{\alpha^4}{p^4} \cdot 4p^2 - \delta \frac{\alpha^2}{p^2} \cdot \frac{p^2}{16} + \frac{\alpha^2}{p^2} \cdot 2p \\
&\leq 8 \cdot 24^{\lambda} \frac{\alpha^4}{p} + \frac{2\alpha^2}{p} \leq 9 \cdot 24^{\lambda} \frac{\alpha^4}{p} \ .
\end{aligned}$$

Applying Chebyshev's inequality, we see that

$$\Pr\left[S \leq \frac{\alpha^2}{4p} - t\sqrt{9 \cdot 24^{\lambda}} \frac{\alpha^2}{\sqrt{p}}\right] \leq \Pr[S \leq \mu - t\sigma] \leq \frac{1}{t^2}$$

where we have used our lower bound for $E[S]$ and our upper bound for the variance of S. We choose

$$t = \frac{1}{\sqrt{p}} \cdot \frac{\alpha^2 - 4\alpha}{\alpha^2 \cdot 4\sqrt{9 \cdot 24^{\lambda}}}$$

so that $\frac{\alpha^2}{4p} - t\sqrt{9 \cdot 24^{\lambda}} \frac{\alpha^2}{\sqrt{p}} = \frac{\alpha}{p}$. Thus we have

$$\Pr[S \leq \alpha/p] \leq \frac{9 \cdot 24^{\lambda}}{(\frac{1}{4} - \frac{1}{\alpha})^2} \cdot p \leq \frac{9 \cdot 24^{\lambda}}{(\frac{1}{4} - \frac{1}{8})^2} \cdot p = 576 \cdot 24^{\lambda} p \ .$$

□

The following theorem gives the correctness proof for Algorithm 5. It has a significant weakness as compared with the similar results before it, as well as the similar result of Lyubashevsky [20]. Namely, that the bound on the probability of success gets weaker

Algorithm 4 k-set ListMerge

Input: two lists L_1, L_2 of integers in the interval $[-\frac{R}{2}, \frac{R}{2})$, parameter $p < 1$, parameter α
Output: list L_{12} of integers $b + c \in [-\frac{Rp}{2}, \frac{Rp}{2})$ where $b \in L_1$, $c \in L_2$
sort L_1, L_2
for $b \in L_1$ **do**
  for $c \in L_2$ in interval $[-b - \frac{Rp}{2}, -b + \frac{Rp}{2})$ **do**
    $L_{12} \leftarrow L_{12} \cup \{b + c\}$
  end for
end for
output L_{12} { cut the list down to α/p elements if necessary}

Algorithm 5 k-set algorithm for dense modular subset sum

Input: modulus m, target t, $a_1, \ldots, a_n \in \mathbb{Z}/m\mathbb{Z}$, parameter $p = m^{-1/\log k}$, parameter α
Require: $\log m < n/k$
Output: bit vector $\boldsymbol{x}$ such that $\sum_{i=1}^{n} a_i x_i = t \mod m$ or *Probably No Solution*

Let $a_{n+1} = -t \mod m$ {now solve new problem with target 0}
Divide a_i into k sets indexed by I_j, $1 \le j \le k$

for $j = 1$ to k **do**
  for $s = 1$ to α/p **do**
    generate random bits x_i, $i \in I_j$
    store $\sum_{i \in I_j} a_i x_i$ along with the bits in list L_j { set $x_{n+1} = 1$ so that $-t$ is included in all elements of I_k}
  end for
end for

for level $\lambda = 0$ to $\log k - 1$ **do**
  for the $2^{\log k - \lambda}$ pairs $(L_{j,1}^{(\lambda)}, L_{j,2}^{(\lambda)})$ **do**
    $L_j^{(\lambda+1)} = \text{ListMerge}(L_{j,1}^{(\lambda)}, L_{j,2}^{(\lambda)})$
  end for
end for

if $L^{(\log k)} \neq \emptyset$ **then**
  output $\boldsymbol{x}$ { since $L^{(\log k)}$ has integers in the range $[-\frac{1}{2}, \frac{1}{2})$}
else
  output *Probably No Solution*
end if

with increasing density. Thus in order to claim a failure bound of $e^{-\Omega(n)}$ we need to assume that c is constant as well as k. This runs counter to our intuition about dense subset sum problems, but nevertheless gives the exponential bound we require.

Theorem 5.11. *Choose $k < n$ so that $n > O(k(\log k)^2)$ and assume $m = 2^{cn/k}$ for $c < \frac{\log k}{\log k+4}$ (so the density is greater than $k(1 + \frac{4}{\log k})$). Then Algorithm 5 runs in time and space $O(n^3 \cdot m^{1/\log k}) = \widetilde{O}(m^{1/\log k})$.*

With the a_i chosen uniformly at random from $\mathbb{Z}/m\mathbb{Z}$, the algorithm outputs a solution with probability at least

$$\left(1 - k2^{-\frac{(1-c)n}{4k}}\right)\left(1 - 1152k^6 \cdot 2^{-cn/(k\log k)} - k2^{-\frac{(\log k-(\log k+4)c)n}{4k\log k}}\right) \quad .$$

Choosing constant c, k gives the conceptually easier bound of $1 - e^{-\Omega(n)}$.

Proof. We first prove correctness. The probability of success is greater than the probability that all k subsets of $\boldsymbol{a}$ are well-distributed, times the probability that the algorithm succeeds given that all subsets are well-distributed. By applying Proposition 3.11 with $n = n/k$ and $\gamma = \frac{(1-c)n}{4k}$, the probability that one of the subsets is well-distributed is greater than $1 - 2^{-\frac{(1-c)n}{4k}}$ and thus the probability that all are well-distributed is greater than

$$(1 - 2^{-\frac{(1-c)n}{4k}})^k > 1 - k \cdot 2^{-\frac{(1-c)n}{4k}} \quad .$$

In addition, the distance between these distributions and uniform ones is less than $2^{-\frac{(1-c)n}{4k}}$.

Now, assume that all elements from all initial lists are drawn independently from uniform distributions. Then the algorithm succeeds if every application of ListMerge results in a new list with at least α/p elements, since in particular this means there is an element in the final list that falls in the interval $[-\frac{1}{2}, \frac{1}{2})$.

Since by Lemma 5.10 the probability of success for one one application of ListMerge is at least $1 - 576k^5 \cdot 2^{-cn/(k \log k)}$, the probability of success for all $2k$ applications is at least

$$(1 - 576k^5 \cdot 2^{-cn/(k \log k)})^{2k} \geq 1 - 1152k^6 \cdot 2^{-cn/(k \log k)}$$

since $p \leq 1/(2k)$.

We next account for the fact that the elements of the initial lists are only close to uniform. The k initial lists each have $m^{1/\log k}$ elements of $\mathbb{Z}/m\mathbb{Z}$. Define $X_{\boldsymbol{a}_j}$, $1 \leq j \leq k$, by 3.2 as was done in the 2-set and 4-set cases. Define random variables X_i and U_i, $1 \leq i \leq km^{1/\log k}$ so that U_i is the uniform distribution on $\mathbb{Z}/m\mathbb{Z}$ and the X_i are evenly divided among the k choices of $X_{\boldsymbol{a}_j}$.

Let the predicate f take value 1 if Algorithm 5 fails with initial input X_i. Then applying Propositions 3.5 and 3.6 yields

$$|\Pr[f(X_1, \ldots, X_{km^{1/\log k}}) = 1] - \Pr[f(U_1, \ldots, U_{km^{1/\log k}}) = 1]| \leq \sum_{i=1}^{km^{1/\log k}} \Delta(X_i, U_i) \ .$$

This quantity is bounded by

$$km^{\frac{1}{\log k}} 2^{-\frac{(1-c)n}{4k}} \leq k2^{\frac{cn}{k \log k}} 2^{-\frac{(1-c)n}{4k}} \leq k2^{-\frac{(\log k - (\log k + 4)c)n}{4k \log k}} \ .$$

Thus the probability of success is greater than

$$\left(1 - k2^{-\frac{(1-c)n}{4k}}\right)\left(1 - 1152k^6 2^{-cn/(k \log k)} - k2^{-\frac{(\log k - (\log k + 4)c)n}{4k \log k}}\right) \ .$$

Finally, we analyze the resource usage. The storage requirement is maximized at level 0, where we have k lists each with $\alpha m^{1/\log k}$ elements consisting of an element of $\mathbb{Z}/m\mathbb{Z}$ and a bit vector with n/k bits. Thus the algorithm requires $O(n^3 \cdot m^{1/\log k})$ space.

The time taken by the algorithm is in two components, namely initializing the lists L_j and then looping through a series of calls to ListMerge. Initializing involves doing a sum

that takes time $O(\frac{n}{k}\log m)$ and doing it $\alpha k m^{1/\log k}$ times for a cost of $O(\alpha n \log m \cdot m^{1/\log k})$ bit operations. ListMerge, as it did in the 4-set case, costs $O(\alpha m^{1/\log k} \cdot \log(\alpha m^{1/\log k}))$ bit operations and we do fewer than $2k$ of them. Since $k/\log k < n$, the total running time is thus $O(\alpha n \log m \cdot m^{1/\log k}) = O(n^3 \cdot m^{1/\log k})$.

□

5.4 Martingale ListMerge

We noted that the k-set list merging lemma in the previous section is less than ideal since the lower bound on the probability that the algorithm succeeds shrinks with increasing density. In this section we restore our intuition that finding a solution is easier for MSS problems of greater density.

Recall that in the 4-set algorithm we needed the output of ListMerge to contain a row distinct list so that the elements would be independent. We will adapt that argument to the k-set case, but proving that a row distinct list exists will require the theory of martingales since at higher levels the elements are no longer independent.

Recall our previous notation at level λ of the k-set algorithm. Choose $k < n$ so that $n > O(k(\log k)^2)$, $m = 2^{cn/k}$ with $c < 1$, parameter $p = m^{1/\log k}$ and $\alpha = O(n)$. Lists L_1 and L_2 contain elements generated with probability mass function f_λ, where $|L_1| = |L_2| = N = \frac{\alpha}{p}$. Let b_j be the elements of L_1, c_j the elements of L_2, and ℓ_j the elements of $L_1 + L_2$. Treat the ℓ_j as arranged in a table with the rows indexed by b_j and the columns indexed by c_j.

The b_j, c_j are integers from the interval $[-\frac{mp^\lambda}{2}, \frac{mp^\lambda}{2})$. We are interesting in proving that at least one ℓ_j per row is in the restricted interval $[-\frac{mp^{\lambda+1}}{2}, \frac{mp^{\lambda+1}}{2})$. Towards this

end we fix $b \in L_1$. Relabel indices so that $\ell_i = b + c_i$ for $1 \leq i \leq N$. Let $I_b = [-\frac{mp^{\lambda+1}}{2} - b, \frac{mp^{\lambda+1}}{2} - b)$, and let the random variable X_i take the value 1 if $c_i \in I_b$ and 0 otherwise.

The next set of notation follows a survey paper by McDiarmid [21, Section 3.2] that covers numerous concentration inequalities and their applications to problems in combinatorics and computer science.

Let $f(\boldsymbol{X})$ be a bounded real valued function on $X_1, \ldots, X_N$, which for our purposes will be $S = \sum_{i=1}^{N} X_i$. Let B denote the event that $X_i = x_i$ for $i = 1, \ldots, j-1$ where x_i is either 0 or 1. For $x = 0, 1$ let

$$K_j(x) = \mathrm{E}[f(\boldsymbol{X}) \mid B, X_j = x] - \mathrm{E}[f(\boldsymbol{X}) \mid B] \ .$$

Define $dev(x_1, \ldots, x_{j-1})$ to be $\sup\{|K_j(0)|, |K_j(1)|\}$, and define $ran(x_1, \ldots, x_{j-1})$ to be $|K_j(0) - K_j(1)|$.

Let the *sum of squared ranges* be $R^2(\boldsymbol{x}) = \sum_{j=1}^{N} ran(x_1, \ldots, x_{j-1})^2$ and let $\hat{r}^2$, the *maximum sum of squared ranges*, be the supremum of R^2 over all choices of $x_1, \ldots, x_N$. Let *maxdev* be the maximum of $dev(x_1, \ldots, x_{j-1})$ over all choices of j and all choices of x_i.

The context of all this notation is the theory of martingales. By the Doob construction, $Y_j = \mathrm{E}[f(\boldsymbol{X}) \mid X_1, \ldots, X_j]$, $1 \leq j \leq N$ forms a martingale sequence since by the tower lemma, $\mathrm{E}[Y_j \mid X_1, \ldots, X_{j-1}] = Y_{j-1}$. Thus the martingale differences $Y_j - Y_{j-1}$ form a sequence of random variables with mean 0, ideal for applying convexity bounds. The standard technique is to find uniform bounds on the martingale differences and then apply the Azuma-Hoeffding theorem to get an exponential tail bound. Unfortunately, in our case the bound is not tight enough to be meaningful. For further information

regarding martingales, see [12] or [33].

The theorem we will use in place of Azuma-Hoeffding is the following martingale version of Bernstein's inequality, proven by McDiarmid [21, Section 3.2].

Theorem 5.12. *Let $X_1, \ldots, X_N$ be a family of random variables with X_i taking values in $\{0,1\}$, and let f be a bounded real-valued function defined on $\{0,1\}^N$. Let μ denote the mean of $f(\boldsymbol{X})$, let b denote the maximum deviation maxdev, and let $\hat{r}^2$ denote the maximum sum of squared ranges. Suppose that X_i takes two values with the smaller probability being $p < \frac{1}{2}$. Then for any $t \geq 0$,*

$$\Pr[|f(\boldsymbol{X}) - \mu| \geq t) \leq 2 \exp\left(-\frac{t^2}{2p\hat{r}^2(1 + (bt/3p\hat{r}^2))}\right) \quad .$$

The next lemma is crucial for finding good bounds of $\hat{r}^2$ and *maxdev*. It relies on a technical result proven in Section 5.5, which is a generalization of Lemma 5.7.

Lemma 5.13. *Use the notation for level λ, with X_i being the indicator event for $c_i \in I_b$. Assume that $c_1, \ldots, c_N$ are row distinct when treated as ℓ_i from level $\lambda - 1$, and independent if $\lambda = 0$. Then for any $i > j$,*

$$|\mathrm{E}[X_i \mid X_1, \ldots, X_j] - \mathrm{E}[X_i]| \leq 4 \cdot 24^{\lambda} p^2$$

assuming that $p \leq 1/(864k^5)$.

Proof. We will reduce the result to finding a uniform bound for $|\Pr[c_i \mid X_1, \ldots, X_j] - \Pr[c_i]|$. Note that

$$\begin{aligned}\Pr[c_i \mid X_1, \ldots, X_j] &= \frac{\Pr[c_i \wedge X_1 \wedge \cdots \wedge X_j]}{\Pr[X_1 \wedge \cdots \wedge X_j]} \\ &= \frac{\sum_{d_1} \cdots \sum_{d_j} \Pr[c_i \wedge c_1 = d_1 \wedge \cdots \wedge c_j = d_j]}{\sum_{d_1} \cdots \sum_{d_j} \Pr[c_1 = d_1 \wedge \cdots \wedge c_j = d_j]}\end{aligned}$$

where each sum in the numerator and denominator ranges over $d \in I_b$ if $X = 1$ and $d \notin I_b$ if $X = 0$.

If the level λ is 0, then $c_1, \ldots, c_j$, c_i are all fully independent and hence

$$|\Pr[c_i \mid X_1, \ldots, X_j] - \Pr[c_i]| = 0 \ .$$

If $\lambda \neq 0$, each variable c in consideration has arrived at level λ, and thus was some ℓ at level $\lambda - 1$ that fell in the restricted interval $[-\frac{mp^\lambda}{p}, \frac{mp^\lambda}{p})$. We apply Lemma 5.15 (proven in the next section) with parameters from level $\lambda - 1$ to break off the c_i term, after which the rest of the numerator cancels with the denominator. For the rest of the proof we instead use parameters from level λ for ease of exposition. Thus

$$\frac{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}\Pr[c_i] \leq \Pr[c_i \mid X_1, \ldots, X_j] \leq \frac{(1+4\cdot 3^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-6\cdot 6^\lambda p)^{4^{\lambda-1}}}\Pr[c_i].$$

We saw in the proof of Lemma 5.8 that for $p \leq \frac{1}{864k^5} \leq \frac{1}{864\cdot 24^\lambda}$,

$$\frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}} \leq 1 + 2\cdot 24^\lambda p \ .$$

Using Taylor series we get a similar result for the constant in the lower bound, namely

$$\begin{aligned}
\frac{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}} &\geq (1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}(1-4\cdot 6^\lambda p)^{4^{\lambda-1}} \\
&\geq (1-4^\lambda p)(1-18\cdot 24^{\lambda-1}p)(1-24\cdot 24^{\lambda-1}p) \\
&\geq 1 - 47\cdot 24^{\lambda-1}p \geq 1 - 2\cdot 24^\lambda p \ .
\end{aligned}$$

We now have

$$\begin{aligned}
|\Pr[c_i \mid X_1, \ldots, X_j] - \Pr[c_i]| &\leq |\Pr[c_i](1 \pm 2\cdot 24^\lambda p) - \Pr[c_i]| \\
&= \Pr[c_i]\cdot 2\cdot 24^\lambda p \\
&\leq 2\cdot 24^\lambda p\Big(\frac{1}{mp^\lambda} + \frac{6^\lambda p}{mp^\lambda}\Big)
\end{aligned}$$

and conclude that

$$\begin{aligned}|\mathrm{E}[X_i \mid X_1, \ldots, X_j] - \mathrm{E}[X_i]| &\leq \sum_{c_i \in I_b} |\Pr[c_i \mid X_1, \ldots, X_j] - \Pr[c_i]| \\ &\leq mp^{\lambda+1} \cdot 2 \cdot 24^{\lambda} p\Big(\frac{1}{mp^{\lambda}} + \frac{6^{\lambda} p}{mp^{\lambda}}\Big) \\ &\leq 2 \cdot 24^{\lambda} p^2 + 2 \cdot 24^{\lambda} \cdot 6^{\lambda} p^3 \\ &\leq 4 \cdot 24^{\lambda} p^2\end{aligned}$$

since our assumption on p gives us $p \leq \frac{1}{6^{\lambda}}$. □

With ingredients in hand, we next present an improved list merging lemma, one with a lower bound on the success probability that increases with increasing density of the RMSS instance. Note that Algorithm ListMerge preserves the property that each list contains a row distinct sublist. Also note that we have a slightly more stringent requirement on p, and thus need to change our assumption on the relationship between n and k so that $n > O(k(\log n)(\log k))$.

Lemma 5.14 (k-set ListMerge)**.** *Use the notation for level λ. Let A be the following event: for every $b \in L_1$, there exists $c \in L_2$ such that $b + c \in [-\frac{mp^{\lambda+1}}{2}, \frac{mp^{\lambda+1}}{2})$. Then*

$$\Pr[A] \geq 1 - (\alpha/p)e^{-\alpha/1024}$$

assuming that $p \leq 1/(128\alpha k^5)$.

Proof. First consider one row of the table. Fix $b \in L_1$, and let X_i be indicator variables for $c_i \in I_b$, $1 \leq i \leq N$.

We first find bounds on $\mathrm{E}[X_i] = \sum_{c_i \in I_b} \Pr[c_i]$. By Lemma 5.5 the distribution of c_i is close to uniform. Thus $\mathrm{E}[X_i]$ is affected most by the length of $I_b \cap [-\frac{mp^{\lambda}}{2}, \frac{mp^{\lambda}}{2})$, which

falls between $mp^{\lambda+1}/2$ and $mp^{\lambda+1}$. Taking discretization into account, we have

$$\frac{p}{8} \leq \mathrm{E}[X_i] \leq 2p \ .$$

Our main goal now is to find upper bounds for *maxdev* and $\hat{r}^2$.

Consider $dev(x_1, \ldots, x_{j-1})$ for any $j \leq N$, any choice of $x_1, \ldots, x_{j-1}$, and $x_j = 0$ or 1. Note that

$$\begin{aligned} |K_j(x_j)| &= |\mathrm{E}[S \mid X_1, \ldots, X_j] - \mathrm{E}[S \mid X_1, \ldots, X_{j-1}]| \\ &\leq \sum_{i=1}^{N} |\mathrm{E}[X_i \mid X_1, \ldots, X_j] - \mathrm{E}[X_i \mid X_1, \ldots, X_{j-1}]| \ . \end{aligned}$$

If $i < j$ the corresponding term is 0 since its value has already been fixed. If $i = j$ the term can be at most 1 since that is the range of X_i. If $i > j$ then we apply Lemma 5.13 to see that

$$\begin{aligned} &|\mathrm{E}[X_i \mid X_1, \ldots, X_j] - \mathrm{E}[X_i \mid X_1, \ldots, X_{j-1}]| \\ &= |\mathrm{E}[X_i \mid X_1, \ldots, X_j] - \mathrm{E}[X_i] + \mathrm{E}[X_i] - \mathrm{E}[X_i \mid X_1, \ldots, X_{j-1}]| \\ &\leq 8 \cdot 24^{\lambda} p^2 \end{aligned}$$

Putting this all together we see that *maxdev* is no more than

$$1 + \frac{\alpha}{p} \cdot 8 \cdot 24^{\lambda} p^2 = 1 + 8 \cdot 24^{\lambda} \alpha p \ .$$

By definition

$$\begin{aligned} ran(x_1, \ldots, x_{j-1}) &= |K_j(0) - K_j(1)| \\ &= |\mathrm{E}[S \mid B, X_j = 1] - \mathrm{E}[S \mid B, X_j = 0]| \\ &\leq \sum_{i=1}^{N} |\mathrm{E}[X_i \mid B, X_j = 1] - \mathrm{E}[X_i \mid B, X_j = 0]| \ . \end{aligned}$$

Following the same reasoning as we did for $maxdev$, if $i < j$ the corresponding term is 0, if $i = j$ the corresponding term is at most 1 since X_i is an indicator variable, while if $i > j$ the term is at most $8 \cdot 24^\lambda p^2$ by Lemma 5.13.

So $ran(x_1, \ldots, x_{j-1})$ has a uniform upper bound of $1 + 8 \cdot 24^\lambda \alpha p$ and thus $\hat{r}^2 \leq \frac{\alpha}{p}(1 + 8 \cdot 24^\lambda \alpha p)^2 \leq \frac{\alpha}{p} + 32 \cdot 24^\lambda \alpha^2$, assuming that $p \leq \frac{1}{4\alpha 24^\lambda}$.

We now conclude from Theorem 5.12 that

$$\begin{aligned}
&\Pr\left[S \leq \frac{\alpha}{8} - \frac{\alpha}{16}\right] \\
&\leq \Pr\left[S \leq \mu - \frac{\alpha}{16}\right] \\
&\leq \exp\left(-\frac{\alpha^2/256}{2(\alpha + 32 \cdot 24^\lambda \alpha^2 p) + \frac{2}{3}\frac{\alpha}{16}(1 + 8 \cdot 24^\lambda \alpha p)}\right) \\
&\leq \exp\left(-\frac{\alpha^2/256}{2\alpha + \frac{\alpha}{2} + \frac{2}{3}\alpha + \frac{\alpha}{2}}\right) \\
&\leq e^{-\alpha/1024}
\end{aligned}$$

assuming that $p \leq \frac{1}{128\alpha 24^\lambda}$.

Finally, by using the first two terms of the inclusion-exclusion principle the probability that some row fails to have at least $\alpha/16$ elements fall in I_b is smaller than $(\alpha/p)e^{-\alpha/1024}$ and from this the bound on the probability of event A follows. □

Note that with a parameter choice of $\alpha = n$ we have

$$(\alpha/p)e^{-\alpha/1024} = n2^{cn/(k \log k)}e^{-n/1024} \leq ne^{-\frac{(k \log k - 1024c)n}{1024k \log k}}$$

which decreases with decreasing c (and hence increasing density).

For fixed c and k we can express the success of Algorithm ListMerge as $1 - e^{-\Omega(n)}$.

5.5 Bounding Dependency

A crucial element in our analysis of the 4-set algorithm was Lemma 4.6, which proved that a list of elements coming out of the ListMerge algorithm are independent as long as they are row distinct and their components are independent. With the k-set algorithm the components are no longer necessarily independent; as soon as two ℓ_i that share a row or column both fall in the restricted interval, all the elements at the next level of which those ℓ_i are components will have dependencies. The key result was Lemma 5.7, which proved that two ℓ_i are close to pairwise independent, given that both advance to the next level and thus have values close to 0.

In this section we combine these ideas. Assuming that $\ell_1, \ldots, \ell_r$ are row distinct and given the event that all advance to the next level, we bound the amount of dependency that results from breaking one ℓ_i out of the joint distribution. Here we return to the previous notation where $I = [-\frac{mp^{\lambda+1}}{2}, \frac{mp^{\lambda+1}}{2})$ and X_i is the indicator variable for ℓ_i being in I.

Lemma 5.15. *Let the current level of the algorithm be λ, and let X' be the event $X_1 = 1 \wedge \cdots \wedge X_{r-1} = 1$. Assume that $\ell_1, \ldots, \ell_r$ are row distinct. Then*

$$\frac{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}} \leq \frac{\Pr[\ell_1, \ldots, \ell_r \mid X_r = 1, X']}{\Pr[\ell_r \mid X_r = 1]\Pr[\ell_1, \ldots, \ell_{r-1} \mid X']} \quad \textit{and}$$
$$\frac{\Pr[\ell_1, \ldots, \ell_r \mid X_r = 1, X']}{\Pr[\ell_r \mid X_r = 1]\Pr[\ell_1, \ldots, \ell_{r-1} \mid X']} \leq \frac{(1+4\cdot 6^\lambda p)^{4^{\lambda-1}}}{(1-p)^{4^\lambda}(1-3\cdot 6^\lambda p)^{4^{\lambda-1}}}$$

unless both numerator and denominator are 0.

Proof. We will prove this by induction, where for the base case each $\ell_r = b + c$ with b and c drawn independently from uniform distributions on $[-\frac{m}{2}, \frac{m}{2})$.

Since the ℓ_j are row distinct, ℓ_r either shares c with other ℓ_j, or shares neither b nor c. In writing our the joint distribution we assign free variables $z_1, \ldots, z_d$ to the c_j, where $\ell_1, \ldots, \ell_r$ fall into d columns. We write the probability that $b + c = \ell_r$ as $\sum_z \Pr[c = z \wedge b = \ell_r - z]$.

If b and c both appear only in ℓ_r and not in any other ℓ_j, then

$$\begin{aligned}
&\Pr[\ell_1, \ldots, \ell_r] \\
&= \sum_{z_1} \cdots \sum_{z_d} \Pr[c_1 = z_1 \wedge b_1 = \ell_1 - z_1 \wedge \cdots \wedge c = z_d \wedge b = \ell_r - z_d] \\
&= \sum_{z_1} \cdots \sum_{z_{d-1}} \Pr[c_1 = z_1 \wedge b_1 = \ell_1 - z_1 \wedge \ldots] \sum_{z_d} \Pr[c = z_d \wedge b = \ell_r - z_d] \\
&= \Pr[\ell_1, \ldots, \ell_{r-1}] \Pr[\ell_r]
\end{aligned}$$

where we have used the fact that the b_j and c_j are independent to separate the b and c terms. By summing $\Pr[\ell_1, \ldots, \ell_r]$ over all $\ell_1, \ldots, \ell_r \in I$ we infer the similar equality

$$\Pr[X_r = 1 \wedge X'] = \Pr[X'] \Pr[X_r = 1] \quad \text{and hence}$$
$$\frac{\Pr[\ell_1, \ldots, \ell_r \mid X_r = 1, X']}{\Pr[\ell_r \mid X_r = 1] \Pr[\ell_1, \ldots, \ell_{r-1} \mid X']} = 1 \ .$$

Next suppose that c appears in other ℓ_j but b appears only in ℓ_r. Then

$$\begin{aligned}
&\Pr[\ell_1, \ldots, \ell_r] \\
&= \sum_{z_1} \cdots \sum_{z_d} \Pr[c_1 = z_1 \wedge b_1 = \ell_1 - z_1 \wedge \cdots \wedge c = z_d \wedge b = \ell_r - z_d] \\
&\leq \sum_{z_1} \cdots \sum_{z_d} \Pr[c_1 = z_1 \wedge b_1 = \ell_1 - z_1 \wedge \cdots \wedge c = z_d] \frac{1}{m}
\end{aligned}$$

since the first sum has more restrictions on its free variables and hence more terms are zero (in fact, more terms could be zero the farther ℓ_r is from 0). However, since we

are concerned with the probability conditioned on ℓ_r being in I, the number of 0 terms caused by ℓ_r is at most mp. We thus conclude that

$$\frac{1-p}{m}\Pr[\ell_1,\ldots,\ell_{r-1}] \le \Pr[\ell_1,\ldots,\ell_r] \le \frac{1}{m}\Pr[\ell_1,\ldots,\ell_{r-1}]$$

Meanwhile, we have

$$\frac{1-p}{m} \le \Pr[b+c=\ell_r] \le \frac{1}{m} \quad \text{and so} \quad 1-p \le \frac{\Pr[\ell_1,\ldots,\ell_r]}{\Pr[\ell_r]\Pr[\ell_1,\ldots,\ell_{r-1}]} \le \frac{1}{1-p} \ .$$

By summing $\Pr[\ell_1,\ldots,\ell_r]$ over all $\ell_1,\ldots,\ell_r \in I$ we get the similar bounds

$$(1-p)\Pr[X_r=1]\Pr[X'] \le \Pr[X_r=1 \wedge X'] \le \frac{\Pr[X_r=1]\Pr[X']}{1-p} \ .$$

Note that as long as $\ell_1,\ldots,\ell_r \in I$ we have

$$\frac{\Pr[\ell_1,\ldots,\ell_r \mid X_r=1,X']}{\Pr[\ell_r \mid X_r=1]\Pr[\ell_1,\ldots,\ell_{r-1} \mid X']} = \frac{\Pr[\ell_1,\ldots,\ell_r]}{\Pr[\ell_r]\Pr[\ell_1,\ldots,\ell_{r-1}]} \cdot \frac{\Pr[X_r=1]\Pr[X']}{\Pr[X_r=1,X']}$$

from which we infer that

$$(1-p)^2 \le \frac{\Pr[\ell_1,\ldots,\ell_r \mid X_r=1,X']}{\Pr[\ell_r \mid X_r=1]\Pr[\ell_1,\ldots,\ell_{r-1} \mid X']} \le \frac{1}{(1-p)^2} \ .$$

We next prove the inductive step, considering the same two cases. Let the inductive lower and upper bounds be

$$L_{\lambda-1} = \frac{(1-p)^{4^{\lambda-1}}(1-3\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}}{(1+4\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}} \quad \text{and} \quad U_{\lambda-1} = \frac{(1+4\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}}{(1-p)^{4^{\lambda-1}}(1-3\cdot 6^{\lambda-1}p)^{4^{\lambda-2}}} \ .$$

If b and c both appear only in ℓ_r then

$$\begin{aligned}
&\Pr[\ell_1,\ldots,\ell_r] \\
&= \sum_{z_1}\cdots\sum_{z_d}\Pr[c_1=z_1 \wedge b_1=\ell_1-z_1 \wedge \cdots \wedge c=z_d \wedge b=\ell_r-z_d] \\
&= \sum_{z_1}\cdots\sum_{z_d}\Pr[c_1=z_1 \wedge \cdots \wedge c=z_d]\Pr[b_1=\ell_1-z_1 \wedge \cdots \wedge b=\ell_r-z_d]
\end{aligned}$$

since the b_j and c_j come from different lists and are thus independent. Using the induction hypothesis we see that

$$L_{\lambda-1}\Pr[c = z_d]\Pr[c_1 = z_1 \wedge \cdots \wedge c_{d-1} = z_{d-1}] \leq \Pr[c_1 = z_1 \wedge \cdots \wedge c = z_d]$$
$$\leq U_{\lambda-1}\Pr[c = z_d]\Pr[c_1 = z_1 \wedge \cdots \wedge c_{d-1} = z_{d-1}]$$

and similarly for $\Pr[b_1 = \ell_1 - z_1 \wedge \cdots \wedge b = \ell_r - z_d]$. Thus

$$L_{\lambda-1}^2\Pr[\ell_r]\Pr[\ell_1, \ldots, \ell_{r-1}] \leq \Pr[\ell_1, \ldots, \ell_r] \leq U_{\lambda-1}^2\Pr[\ell_r]\Pr[\ell_1, \ldots, \ell_{r-1}]$$

and since similar reasoning applies to sums of terms, we conclude that

$$\frac{L_{\lambda-1}^2}{U_{\lambda-1}^2} \leq \frac{\Pr[\ell_1, \ldots, \ell_r \mid X_r = 1, X']}{\Pr[\ell_r \mid X_r = 1]\Pr[\ell_1, \ldots, \ell_{r-1} \mid X']} \leq \frac{U_{\lambda-1}^2}{L_{\lambda-1}^2} \ .$$

Finally suppose that ℓ_r shares a row with ℓ_j (and possibly others), so that b appears in no other ℓ_j. Then

$$\begin{aligned}
&\Pr[\ell_1, \ldots, \ell_r] \\
&= \sum_{z_1} \cdots \sum_{z_d} \Pr[c_1 = z_1 \wedge \cdots \wedge c = z_d]\Pr[b_1 = \ell_1 - z_1 \wedge \cdots \wedge b = \ell_r - z_d] \\
&\leq U_{\lambda-1}\Pr[b = \ell_r - z_d]\Pr[\ell_1, \ldots, \ell_{r-1}]
\end{aligned}$$

by using induction and noting that the bounds on z_d are less restrictive after peeling off the $b = \ell_r - z_d$ term. However, since we assume that $\ell_r \in I$ (otherwise the conditional probability will be zero), including the $b = \ell_r - z_d$ term increases the number of 0 terms by at most a factor of $1 - p$. Thus there is a corresponding lower bound

$$L_{\lambda-1}(1-p)\Pr[b = \ell_r - z_d]\Pr[\ell_1, \ldots, \ell_{r-1}] \leq \Pr[\ell_1, \ldots, \ell_r] \ .$$

To bound $\Pr[b = \ell_r - z_d]$ in terms of $\Pr[b + c = \ell_r]$ we apply Lemma 5.5. We have

$$\frac{1}{mp^\lambda} - \frac{6^\lambda}{mp^{\lambda-1}} \le \Pr[b = \ell_r - z_d] \le \frac{1}{mp^\lambda} + \frac{6^\lambda}{mp^{\lambda-1}} \quad \text{and since } \ell_r \text{ advances}$$
$$mp^\lambda(1-p)\left(\frac{1}{mp^\lambda} - \frac{6^\lambda}{mp^{\lambda-1}}\right)^2 \le \Pr[b + c = \ell_r] \le mp^\lambda\left(\frac{1}{mp^\lambda} + \frac{6^\lambda}{mp^{\lambda-1}}\right)^2 .$$

It is a fact that $(\frac{1}{x} + \epsilon)(1 - 3\epsilon x) \le x(\frac{1}{x} - \epsilon)^2$ assuming x and ϵ are positive. With the additional assumption that $\epsilon > 4\epsilon^2 x$ we also have $x(\frac{1}{x} + \epsilon)^2 \le (\frac{1}{x} - \epsilon)(1 + 4\epsilon x)$. Using these facts with $x = mp^\lambda$, $\epsilon = \frac{6^\lambda}{mp^{\lambda-1}}$ we see that

$$\Pr[\ell_r] \le \Pr[b = \ell_r - z_d]\left(1 + 4 \cdot 6^\lambda p\right) \quad \text{and} \quad \Pr[b = \ell_r - z_d]\left(1 - 3 \cdot 6^\lambda p\right) \le \frac{\Pr[\ell_r]}{1-p}$$
$$\text{and thus} \quad \frac{\Pr[\ell_r]}{(1 + 4 \cdot 6^\lambda p)} \le \Pr[b = \ell_r - z_d] \le \frac{\Pr[\ell_r]}{(1-p)(1 - 3 \cdot 6^\lambda p)} .$$

We conclude that

$$L_{\lambda-1} \cdot \frac{1-p}{1 + 4 \cdot 6^\lambda p} \le \frac{\Pr[\ell_1, \ldots, \ell_r]}{\Pr[\ell_r]\Pr[\ell_1, \ldots, \ell_{r-1}]} \le \frac{1}{(1-p)(1 - 3 \cdot 6^\lambda p)} \cdot U_{\lambda-1}$$

and similar bounds for $\Pr[X_r = 1, X']/(\Pr[X_r = 1]\Pr[X'])$ yield

$$\frac{L_{\lambda-1}}{U_{\lambda-1}} \frac{(1-p)^2(1 - 3 \cdot 6^\lambda p)}{(1 + 4 \cdot 6^\lambda p)} \le \frac{\Pr[\ell_1, \ldots, \ell_r \mid X_r = 1, X']}{\Pr[\ell_r \mid X_r = 1]\Pr[\ell_1, \ldots, \ell_{r-1} \mid X']} \quad \text{and}$$
$$\frac{\Pr[\ell_1, \ldots, \ell_r \mid X_r = 1, X']}{\Pr[\ell_r \mid X_r = 1]\Pr[\ell_1, \ldots, \ell_{r-1} \mid X']} \le \frac{1 + 4 \cdot 6^\lambda p}{(1-p)^2(1 - 3 \cdot 6^\lambda p)} \frac{U_{\lambda-1}}{L_{\lambda-1}} .$$

□

Chapter 6

Future Work

The algorithms set forth in this thesis lend themselves to future research. First, there are a number of incremental improvements.

1. The algorithm for constructing points on elliptic curves could be generalized to work for curves of higher genus.

2. The rational map constructed in that algorithm has no algebraic properties. However, if a different one could be constructed that did, numerous applications would arise such as constructing points of a certain order.

3. The density bound in the 2-set and 4-set algorithms seems unnaturally large. The natural density bounds would be 2 and 4, respectively. It would be worthwhile to verify this experimentally. Proving it would require improving Proposition 3.6.

4. In Section 3.4 we noted that it would be useful to find a constructive condition for a set $a_1, \ldots, a_n$ to be well-distributed. With this we could eliminate the assumption that the MSS problem be random and instead assume the a_i are well-distributed.

5. One weakness of the RMSS algorithms presented in this thesis is that they do not output a random solution, but instead a solution that survives the culling process. Wagner [32] outlines a modification to his 4-set birthday algorithm that results

in a finding a random solution to the 4-set birthday problem. It would be worth investigating whether that method generalizes to the k-set RMSS algorithm.

6. In [27] a general theory of decomposable problems is developed, of which the subset-sum problem is one of them. Perhaps the algorithms in this paper could be modified to attack other decomposable problems.

7. The general $O(2^{n/2})$ time and space algorithm is parallelized in [25]. Is is possible that the algorithms in this thesis are also parallelizable, perhaps using similar ideas.

In addition, the subset-sum problem has a strong connection to Carmichael numbers and various generalizations. In fact, it was the search for a particular order two Carmichael number (as defined in [13]) that led to the work in this thesis. This number satisfies the following generalization of the Korselt condition: a composite squarefree integer n such that for every prime divisor p of n we have $(p-1)|(n-1)$ and $(p+1)|(n+1)$.

Although none have been found, nor have any been proven to exist, a heuristic argument of Erdős (found in [13] and [2]) suggests the following algorithm. First choose L and M with $\gcd(L, M) = 2$, and find many members of the set $P = \{\text{primes } p : (p-1)|L, (p+1)|M\}$. Now find a subset of P that products to the element $(1, -1)$ in $(\mathbb{Z}/L\mathbb{Z})^\times \times (\mathbb{Z}/M\mathbb{Z})^\times$. This product will have the requested properties, and should exist as long as L and M can be chosen to make $\frac{|P|}{\log(L \cdot M)}$ large enough. Improving subset-sum algorithms (and hence subset-product algorithms) will make this search more feasible.

A more ambitious project would be to prove that infinitely many such numbers exist. Most likely this would involve generalizing the proof of the existence of infinitely many Carmichael numbers by Alford, Granville, Pomerance [2].

Appendix A

Leftover Hash Lemma

In this chapter we prove Proposition 3.11, in essence reproducing the proof given in [14], which relies on a result called the Leftover Hash Lemma proven by Charlie Rackoff and appearing in [15]. There are several reasons why this proof is given. First, the result is the starting point for all the new subset sum algorithms in this work, and thus it is vital that it be correct and complete. Second, the result is split among a couple of different papers and it is worthwhile to have the entire proof in one location. Finally, some minor improvements are made. Different definitions of what it means for distributions to be close are used in [20] and [14]; these definitions are proven equivalent. In addition, the proposition is stated in a slightly more general form.

First we need a definition.

Definition A.1. *Let H be a family of functions mapping $\{0,1\}^n$ to $\{0,1\}^l$. H is a* universal family of hash functions *if, for h selected uniformly at random from H we have*

$$\Pr[h(x) = h(y)] = 1/2^l \quad \text{for every } x, y \in \{0,1\}^n,\ x \neq y.$$

H is almost universal *if instead we have* $\Pr[h(x) = h(y)] \leq 1/2^l + 1/2^n$.

Also recall the definitions of statistical indistinguishability and statistical distance from Section 3.4, both tools for measuring the closeness of distributions. The following

result proves that the two definitions are equivalent. This result is also found as an exercise in [9, Chapter 3].

Proposition A.2. *Two distributions D and D' are statistically indistinguishable within ϵ if and only if $\Delta(D, D') < \epsilon$.*

Proof. Suppose first that $\Delta(D, D') < \epsilon$ and let X be a subset of S. Let X_p be the subset of X where $D(x) > D'(x)$ for $x \in X_p$, and let X_n be the subset with $D(x) \leq D'(x)$. Then

$$|D(X) - D'(X)| = \Big| \sum_{x \in X_p} D(x) - D'(x) - \sum_{x \in X_n} D'(x) - D(x) \Big| \tag{A.1}$$

$$= \Big| \sum_{x \in X_p} |D(x) - D'(x)| - \sum_{x \in X_n} |D'(x) - D(x)| \Big| \tag{A.2}$$

If we use the same reasoning when $X = S$ we see that

$$0 = |D(S) - D'(S)| = \Big| \sum_{x \in S_p} |D(x) - D'(x)| - \sum_{x \in S_n} |D'(x) - D(x)| \Big| \quad \text{and thus}$$

$$\sum_{x \in S_p} |D(x) - D'(x)| = \sum_{x \in S_n} |D'(x) - D(x)| = \frac{1}{2} \sum_{x \in S} |D'(x) - D(x)| \ .$$

Assume without loss of generality that $\sum_{X_p} |D(x) - D'(x)| > \sum_{X_n} |D'(x) - D(x)|$. Then A.2 is maximized when $X_p = S_p$ and $X_n = \emptyset$. So

$$|D(X) - D'(X)| \leq \sum_{x \in S_p} |D(x) - D'(x)| - 0 = \frac{1}{2} \sum_{x \in S} |D(x) - D'(x)| < \epsilon$$

and so D and D' are statistically indistinguishable.

Conversely, suppose that $|D(X) - D'(X)| < \epsilon$ for all $X \subset S$ and consider $\Delta(D, D') =$

$\frac{1}{2}\sum_{x\in S}|D(x)-D'(x)|$. We have

$$\begin{aligned}\frac{1}{2}\sum_{x\in S}|D(x)-D'(x)| &= \frac{1}{2}\sum_{x\in S_p}|D(x)-D'(x)| + \frac{1}{2}\sum_{x\in S_n}|D(x)-D'(x)| \\ &= \frac{1}{2}\sum_{x\in S_p} D(x)-D'(x) + \frac{1}{2}\sum_{x\in S_n} D'(x)-D(x) \\ &= \frac{1}{2}\left|\sum_{x\in S_p} D(x)-D'(x)\right| + \frac{1}{2}\left|\sum_{x\in S_n} D'(x)-D(x)\right| \\ &< \frac{\epsilon}{2}+\frac{\epsilon}{2} = \epsilon\end{aligned}$$

which ends the proof. □

Definition A.3. *Let D be a discrete probability distribution on a finite set S. The* collision probability *of D is the probability that $x = y$ given that x and y are chosen independently according to D.*

A classical result is that collision probability is minimized when the distribution is uniform.

Proposition A.4. *Note that the collision probability of D is $\sum_{s\in S} D(s)^2$. We have*

$$\sum_{s\in S} D(s)^2 \geq \sum_{s\in S}\frac{1}{|S|^2} = \frac{1}{|S|} \ .$$

Proof. Consider

$$\sum_{s\in S} D(s)^2 - \sum_{s\in S}\frac{1}{|S|^2} = \sum_{s\in S}(D(s)-1/|S|)(D(s)+1/|S|) \ .$$

As in the previous proposition, let S_p be the subset of S where

$D(s) > 1/|S|$ and S_n the subset where $D(s) \leq 1/|S|$. Then

$$\begin{aligned}
&\sum_{s\in S}(D(s) - 1/|S|)(D(s) + 1/|S|) \\
=&\sum_{s\in S_p}(D(s) - 1/|S|)(D(s) + 1/|S|) - \sum_{s\in S_n}(1/|S| - D(s))(D(s) + 1/|S|) \\
\geq&\sum_{s\in S_p}(D(s) - 1/|S|)\frac{2}{|S|} - \sum_{s\in S_n}(1/|S| - D(s))\frac{2}{|S|}
\end{aligned}$$

where in the last line replacing $D(s)$ with $1/|S|$ makes the terms in S_p smaller and the terms in S_n bigger, so the whole expression is smaller.

Thus we have

$$\sum_{s\in S} D(s)^2 - \sum_{s\in S}\frac{1}{|S|^2} \geq \frac{2}{|S|}\sum_{s\in S}\left(D(s) - \frac{1}{|S|}\right) = 0$$

which finishes the proof. □

Next we show that a sufficient condition for being close to uniform is to have a low collision probability.

Proposition A.5. *Let D be a distribution on a finite set S, and let U be the uniform distribution on S. If the collision probability of D is at most $(1 + 2\epsilon^2)/|S|$ then D and U are statistically indistinguishable within ϵ.*

Proof. Assume not. Then there exists $X \subset S$ such that $|D(X) - |X|/|S|| \geq \epsilon$ and thus we either have $D(X) \geq |X|/|S| + \epsilon$ or $D(X) \leq |X|/|S| - \epsilon$. Equivalently assume that $D(X) = |X|/|S| \pm \delta$ where $\delta \geq \epsilon$. Let x_1 and x_2 be drawn independently from S using

the distribution D, so that $\Pr[x_1 = x_2]$ is the collision probability of D. We have

$$\begin{aligned}
&\Pr[x_1 = x_2] \\
&= \Pr[x_1, x_2 \in X] \Pr[x_1 = x_2 | x_1, x_2 \in X] + \Pr[x_1, x_2 \in S \setminus X] \Pr[x_1 = x_2 | x_1, x_2 \in S \setminus X] \\
&\geq \frac{D(X)^2}{|X|} + \frac{(1 - D(X))^2}{|S| - |X|}
\end{aligned}$$

where the conditional probabilities have been bounded using Proposition A.4. Replacing $D(X)$ using our assumption above yields

$$\frac{|X|}{|S|^2} + \frac{\pm 2\delta}{|S|} + \frac{\delta^2}{|X|} + \left[\left(1 - \frac{|X|}{|S|}\right)^2 + \delta^2 \mp 2\delta \left(1 - \frac{|X|}{|S|}\right) \right] \Big/ \left(|S| - |X|\right) \ .$$

Algebraic manipulation reveals

$$\frac{|X|}{|S|^2} + \frac{(1 - |X|/|S|)^2}{(|S| - |X|)} = \frac{1}{|S|} \quad \text{and} \quad \pm \frac{2\delta}{|S|} \mp \frac{2\delta(1 - |X|/|S|)}{(|S| - |X|)} = 0,$$

$$\text{so we conclude that} \quad \Pr[x_1 = x_2] \geq \frac{1}{|S|} + \frac{\delta^2}{|X|} + \frac{\delta^2}{|S| - |X|} \ .$$

This expression is minimized when $\delta = \epsilon$ and $|X| = |S|/2$, and so $\Pr[x_1 = x_2] \geq (1 + 4\epsilon^2)/|S|$ which is a contradiction. □

We are now ready to prove the Leftover Hash Lemma. Note that while the statement of the lemma regards hash functions on bit strings, the lemma applies to more general finite families of functions. This is important since we will be applying the result to functions that map bit strings of length n to $\mathbb{Z}/m\mathbb{Z}$ where $\log m < n$.

Lemma A.6 (Leftover Hash Lemma, [15])**.** *Let $X \subset \{0,1\}^n$ with $|X| \geq 2^l$. Let $e > 0$, and let H be an almost universal family of hash functions mapping n bits to $l - 2e$ bits. Let U be the uniform distribution on $H \times \{0,1\}^{l-2e}$ and D be the distribution $(h, h(x))$ with h chosen uniformly from H and x uniformly from X. Then D and U are statistically indistinguishable within 2^{-e}.*

Proof. Consider the collision probability of D, namely $\Pr[h_1 = h_2]\Pr[h_1(x_1) = h_2(x_2)]$ with h_1, h_2 drawn independently from H and x_1, x_2 drawn independently from X. We have that $\Pr[h_1(x_1) = h_2(x_2)]$ equals

$$\Pr[x_1 = x_2]\Pr[h_1(x_1) = h_2(x_2) \mid x_1 = x_2] + \Pr[x_1 \neq x_2]\Pr[h_1(x_1) = h_2(x_2) \mid x_1 \neq x_2] \quad .$$

By the definition of almost universal, $\Pr[h_1(x_1) = h_2(x_2) \mid x_1 \neq x_2] \leq 1/2^{l-2e} + 1/2^n$. Since h_1 and h_2 are functions, $\Pr[h_1(x_1) = h_2(x_2) \mid x_1 = x_2] = 1$. With the assumption that $|X| \geq 2^l$, $\Pr[x_1 = x_2] = 1/|X| \leq 1/2^l$. Combining these statements with the fact that $\Pr[x_1 \neq x_2] \neq 1$, the collision probability is strictly bounded above by

$$\frac{1}{|H|}\left(\frac{1}{2^l} + \frac{1}{2^{l-2e}} + \frac{1}{2^n}\right) \leq \frac{1 + 2/2^{2e}}{|H|2^{l-2e}} \quad .$$

Now let $\epsilon = 1/2^e$. Then since U is defined on a set of size $|H|2^{l-2e}$, we have by Proposition A.5 that D is statistically indistinguishable from U within ϵ. □

We are finally ready to prove Proposition 3.11. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ and let S index the 2^n subsets of $\boldsymbol{a}$. Let $m = 2^{cn}$ where $c < 1$. Define the hash function $f_{\boldsymbol{a}}(S) : \{0,1\}^n \to \{0,1\}^{cn}$ by $f_{\boldsymbol{a}}(S) = \sum_{i \in S} a_i \mod m$, and consider the family of hash functions $f_{\boldsymbol{a}}$ indexed by all $\boldsymbol{a} \in A$. We use A to notate the finite set of all possible $\boldsymbol{a}$ composed of a_i of length at most cn.

We restate the proposition here for convenience.

Proposition A.7. *Define a distribution given by $f_{\boldsymbol{a}}$ on random input, and let U be the uniform distribution on $\{0,1\}^{cn}$. For all but a $2^{-\gamma}$ fraction of the possible choices for $\boldsymbol{a} = (a_1, \ldots, a_n)$, $\Delta(f_{\boldsymbol{a}}, U) < 2^{-\frac{(1-c)n}{2} + \gamma}$.*

Proof. Our first task is to prove that the family of hash functions $f_{\boldsymbol{a}}$ is universal (and hence almost universal). Let $\boldsymbol{a}$ be chosen uniformly from A, so that each of the a_i is

chosen uniformly from $\mathbb{Z}/m\mathbb{Z}$. Let S and S' be subsets of $\boldsymbol{a}$ such that $S \neq S'$. Treat S, S' as elements of $\{0,1\}^n$ and let S_i denote the ith component.

At least one of the sets is nonempty, and S nonempty means that $S \neq \{0\}^n$ and so assume that $S_i \neq 0$. Then $f_{\boldsymbol{a}}(S) = a_i + \sum_{j \neq i} S_j a_j$ is uniformly distributed on $\{0,1\}^{cn}$ since the a_j are independent. A similar argument holds for S'. Since $S \neq S'$, there exists an index i such that $S_i \neq S'_i$, say $S_i = 1$ and $S'_i = 0$. Then $f_{\boldsymbol{a}}(S) = a_i + \sum_{j \neq i} S_j a_j$ is independent of $f_{\boldsymbol{a}}(S') = \sum_{j \neq i} S'_j a_j$ since fixing $\sum_{j \neq i} S_j a_j$ leaves $f_{\boldsymbol{a}}(S)$ completely undetermined.

We conclude that $f_{\boldsymbol{a}}(S)$ and $f_{\boldsymbol{a}}(S')$ are independent uniformly distributed variables on $\mathbb{Z}/m\mathbb{Z}$. As a result, for fixed $S \neq S'$ and $\boldsymbol{a}$ chosen uniformly at random from A, $\Pr[f_{\boldsymbol{a}}(S) = f_{\boldsymbol{a}}(S')] = 1/m = 1/2^{cn}$ and hence $f_{\boldsymbol{a}}$ is a universal family of hash functions.

Next we apply the Leftover Hash Lemma with parameters $X = \{0,1\}^n$, $l = n$, $l - 2e = cn$, $e = (1-c)n/2$ and conclude that if D is the distribution $(\boldsymbol{a}, f_{\boldsymbol{a}}(S))$ with $S \in X$ and U is the uniform distribution on $A \times \{0,1\}^{cn}$, then $\Delta(D,U) \leq 2^{-\frac{(1-c)n}{2}}$. Here we have used Proposition A.2 to replace statistical indistinguishability with statistical distance.

We now change notation to let $D = f_{\boldsymbol{a}}(S)$ be the distribution with fixed $\boldsymbol{a}$ and S drawn uniformly from $\{0,1\}^n$, and to let U be the uniform distribution on $\{0,1\}^{cn}$. The expected value over $\boldsymbol{a} \in A$, given by $E[\Delta(D,U)] = \sum_{\boldsymbol{a}} \frac{1}{|A|} \Delta(f_{\boldsymbol{a}}, U)$, is less than $2^{-\frac{(1-c)n}{2}}$ since $\boldsymbol{a}$ is drawn uniformly from A. We now apply Markov's inequality to conclude that

$$\Pr_{\boldsymbol{a}} \left[\Delta(f_{\boldsymbol{a}}, U) \geq 2^{-\frac{(1-c)n}{2}+\gamma} \right] \leq \frac{2^{-\frac{(1-c)n}{2}}}{2^{-\frac{(1-c)n}{2}+\gamma}} = 2^{-\gamma}$$

and thus the probability that $\boldsymbol{a}$ gives a D that is far from uniform is less than $2^{-\gamma}$. $\square$

Bibliography

[1] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Math.*, 160(2):781 – 793, 2004.

[2] W. Alford, A. Granville, and C. Pomerance. There are infinitely many Carmichael numbers. *Ann. of Math.*, 139(3):703 – 722, 1994.

[3] A. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203):29 – 68, 1993.

[4] E. Bach and J. Shallit. *Algorithmic Number Theory.* The MIT Press, Cambridge, 1996.

[5] E. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill Book Co., New York, 1968.

[6] B. Birch and H. P. F. Swinnerton-Dyer. Notes on elliptic curves. I. *J. Reine Angew. Math.*, 212:7 – 25, 1963.

[7] M. Chaimovich. New algorithm for dense subset-sum problem. *Astérisque*, 258:363 – 373, 1999.

[8] M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low–density subset sum algorithms. *Comput. Complexity*, 2(2):111 – 128, 1992.

[9] P. Diaconis. *Group Representations in Probability and Statistics*, volume 11 of

Lecture Notes - Monograph Series. Institute of Mathematical Statistics, 1988. Shanti S. Gupta, Series Editor.

[10] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, IT-22(6):644 – 654, 1976.

[11] A. Flaxman and B. Przydatek. Solving medium-density subset sum problems in expected polynomial time. In *STACS 2005*, volume 3404 of *Lecture Notes in Comput. Sci.*, pages 305 – 314. Springer, Berlin, 2005.

[12] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, New York, second edition, 1992.

[13] E. W. Howe. Higher-order Carmichael numbers. *Math. Comp.*, 69(232):1711 – 1719, 2000.

[14] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *J. of Cryptology*, 9(4):199 – 216, 1996.

[15] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *IEEE Symposium on Foundations of Computer Science*, pages 248 – 253, 1989.

[16] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85 – 103. Plenum Press, NY, 1972.

[17] N. Koblitz. *A Course in Number Theory and Cryptography*. Springer–Verlag, New York, second edition, 1994.

[18] J. Lagarias and A. Odlyzko. Solving low-density subset sum problems. *JACM: Journal of the ACM*, 32(1):229 – 246, 1985.

[19] H. W. Lenstra. Factoring integers with elliptic curves. *Ann. of Math.*, 126(2):649 – 673, 1987.

[20] V. Lyubashevsky. On random high density subset sums. *Electronic Colloquium on Computational Complexity (ECCC)*, 12, 2005. http://eccc.hpi-web.de/eccc-reports/2005/TR05-007/index.html.

[21] C. McDiarmid. Concentration. In *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16 of *Algorithms Combin.*, pages 195 – 248. Springer, Berlin, 1998.

[22] K. Nishimura and M. Sibuya. Occupancy with two types of balls. *Ann. Inst. Statist. Math.*, 40(1):77 – 91, 1988.

[23] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.

[24] S. Purkayastha. Simple proofs of two results on convolutions of unimodal distributions. *Statist. Prob. Lett.*, 39(2):97 – 100, 1998.

[25] C. A. A. Sanches, N. Y. Soma, and H. H. Yanasse. An optimal and scalable parallelization of the two-list algorithm for the subset-sum problem. *European J. Oper. Res.*, 176(2):870 – 879, 2007.

[26] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170):483 – 494, 1985.

[27] R. Schroeppel and A. Shamir. A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems. *SIAM J. Comput.*, 10(3):456 – 464, 1981.

[28] A. Shallue and C. E. van de Woestijne. Construction of rational points on elliptic curves over finite fields. In *ANTS VII: Algorithmic Number Theory Symposium*, volume 4076 of *Lecture Notes in Comput. Sci.*, pages 510 – 524. Springer, Berlin, 2006.

[29] J. Silverman. *The Arithmetic of Elliptic Curves*. Springer–Verlag, New York, 1992. Corrected reprint of the 1986 original.

[30] C. E. van de Woestijne. *Deterministic Equation Solving over Finite Fields*. PhD thesis, Universiteit Leiden, 2006.

[31] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, second edition, 2003.

[32] D. Wagner. A generalized birthday problem (extended abstract). In *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Comput. Sci.*, pages 288 – 303. Springer, Berlin, 2002.

[33] D. Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 1991.