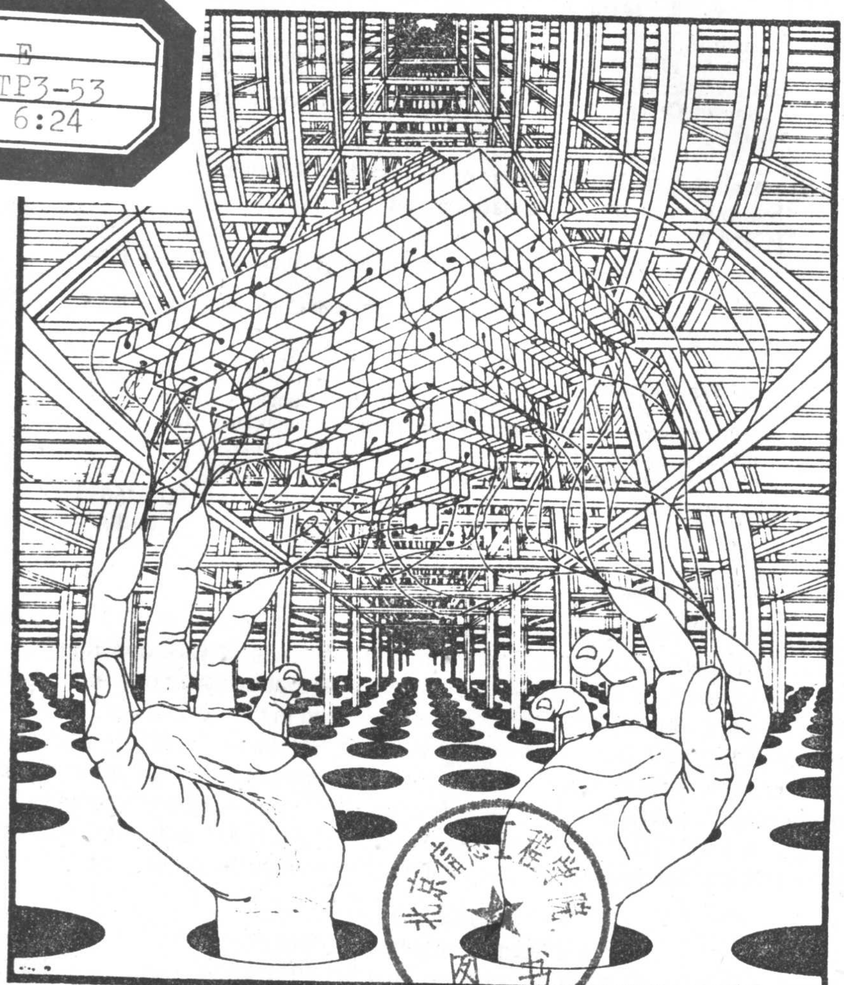
24th Annual Symposium on Foundations of Computer Sciecne

1983

24th Annual

Symposium on Foundations of Computer Science

(Formerly called the Annual Symposium on Switching and Automata Theory)



NOVEMBER 7-9, 1983 IEEE 83CH1938-0

sponsored by
the IEEE Computer Society's Technical Committee on
Mathematical Foundations of Computing

ISSN 0272-5428
IEEE CATALOG NUMBER 83CH1938-0
LIBRARY OF CONGRESS NUMBER 80-646634
IEEE COMPUTER SOCIETY ORDER NUMBER 508
ISBN 0-8186-0508-1

COMPUTER SOCIETY PRESS

Symposium on Foundations of Computer Science.
Annual Symposium on Foundations of Computer Science [papers]. 16th- 1975-
[New York, Institute of Electrical and Electronics Engineers]

v. ill. 28 cm.

Annual.
Vols. for 1975- sponsored by IEEE Computer Society, Technical Committee on Mathematical Foundations of Computing and the ACM Special Interest Group for Automata and Computability Theory, and various universities.

Symposium on Foundations of Computer Science. Annual Symposium on Foundations of Computer Science ... (Card 2)

Continues: Annual Symposium on Switching & Automata Theory, ISSN 0272-4847.
Key title: Annual Symposium on Foundations of Computer Science, ISSN 0272-5428.

1. Switching theory—Congresses. 2. Machine theory—Congresses. 3. Electronic data processing—Congresses. I. IEEE Computer Society. Technical Committee on Mathematical Foundations of Computing. II. ACM Special Interest Group for Automata and Computability Theory. III. California. University. Dept. of Electrical Engineering and Computer Sciences. IV. Title.

QA268.5.S9a 519.4 80-646634
MARC-S

Library of Congress

ISSN 0272-5428
IEEE Catalog Number 83CH1938-0
Library of Congress Number 80-646634
IEEE Computer Society Order Number 508
ISBN 0-8186-0508-1 (paper)
ISBN 0-8186-4508-3 (microfiche)
ISBN 0-8186-8508-5 (casebound)

Order from: IEEE Computer Society
Post Office Box 80452
Worldway Postal Center
Los Angeles, CA 90080

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

1884 1984 The Institute of Electrical and Electronics Engineers, Inc.

Foreword

The papers in this volume were presented at the 24th Annual Symposium on Foundations of Computer Science, held on November 7-9, 1983, in Tucson, Arizona. The symposium was sponsored by the IEEE Computer Society Technical Committee for Mathematical Foundations of Computing.

These 60 papers were selected on June 30–July 1, 1983, at a meeting of the full program committee, from 160 extended abstracts submitted in response to the call for papers. The selection was based on perceived originality, quality, and relevance to theoretical computer science. The submissions were not refereed, and many of them represent preliminary reports on continuing research. It is anticipated that most of these papers will appear, in more polished and complete form, in scientific journals.

The program committee wishes to thank all who submitted papers for consideration.

Manuel Blum
Zvi Galil
Oscar H. Ibarra
Dexter Kozen
Gary L. Miller
J. Ian Munro
W.L. Ruzzo
Lawrence Snyder, Chairman
Richard Statman
Robert E. Tarjan

Machtey Award

"The Program Complexity of Searching a Table" by Harry G. Mairson is the 1983 winner of the Machtey Award for the most outstanding paper written by a student or students, as judged by the program committee.

Technical Committee on Mathematical Foundations of Computer Science

Technical Committee Chairman

Arnold Rosenberg
Duke University

Program Chairman

Lawrence Snyder
University of Washington

Local Arrangements Chairman

Peter Downey
University of Arizona

Publicity Chairman

David Bray
Clarkson University

Symposium Coordinator

Maria Klawe
IBM, San Jose

Twenty-Fourth Annual Symposium on Foundations of Computer Science
Tucson, Arizona
November 7-9, 1983

Table of Contents

SOLVING LOW-DENSITY SUBSET SUM PROBLEMS

J. C. Lagarias
A. M. Odlyzko

Bell Laboratories
Murray Hill, New Jersey 07974

Abstract. The *subset sum problem* is to decide whether or not the 0-1 integer programming problem

$$\sum_{i=1}^{n} a_i x_i = M; \quad \text{all } x_i = 0 \text{ or } 1;$$

has a solution, where the a_i and M are given positive integers. This problem is *NP*-complete, and the difficulty of solving it is the basis of public key cryptosystems of knapsack type. We propose an algorithm which when given an instance of the subset sum problem searches for a solution. This algorithm always halts in polynomial time, but does not always find a solution when one exists. It converts the problem to one of finding a particular short vector **v** in a lattice, and then uses a lattice basis reduction algorithm due to A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász to attempt to find **v**. We analyze the performance of the proposed algorithm. Let the *density* d of a subset sum problem be defined by $d = \frac{n}{\log_2(\max_i a_i)}$. Then for "almost all" problems of density $d < .645$ the vector **v** we are searching for is the shortest nonzero vector in the lattice. We prove that for "almost all" problems of density $d < \frac{1}{n}$ the lattice basis reduction algorithm locates **v**. Extensive computational tests of the algorithm suggest that it works for densities $d < d_c(n)$, where $d_c(n)$ is a cutoff value that is substantially larger than $\frac{1}{n}$. This method gives a polynomial time attack on knapsack public key cryptosystems that can be expected to break them if they transmit information at rates below $d_c(n)$, as $n \to \infty$.

1. Introduction

The *subset sum problem* is a well-known *NP*-complete set recognition problem [8, p. 226], which is: given a set $A = \{a_i: 1 \leqslant i \leqslant n\}$ of positive integers and a positive integer M, recognize when some subset of A has sum equal to a given integer M. We consider the related *NP*-hard algorithmic problem: find a feasible solution to the 0-1 integer programming problem

$$\sum_{i=1}^{n} a_i x_i = M; \quad \text{all } x_i = 0 \text{ or } 1; \qquad (1.1)$$

when one exists.

Several proposed public key cryptosystems, called *knapsack public key cryptosystems*, are based on this problem [12,15,18]. Such cryptosystems give a set of weights $\{a_i: 1 \leqslant i \leqslant n\}$ as public information. A plaintext message consisting of a 0-1 vector $(e_1, \ldots, e_n)$ is encrypted using (1.1), the integer M being the ciphertext. The problem of decrypting an encrypted message M is thus an instance of (1.1). In such cryptosystems the weights $\{a_i: 1 \leqslant i \leqslant n\}$ are chosen in such a way that (1.1) can be easily solved if certain secret information, called a *trapdoor*, is known. In particular, the sets of weights $\{a_i: 1 \leqslant i \leqslant n\}$ used in such cryptosystems forms a very special subclass of subset sum problems (1.1). In 1982 Adi Shamir [18] announced a method for breaking the simplest such public key cryptosystem, the basic Merkle-Hellman cryptosystem. Since then several attacks on more complicated knapsack cryptosystems have been proposed [1,16]. These attacks are all based on the idea of recovering the trapdoor information concealed in the weights $\{a_i: 1 \leqslant i \leqslant n\}$.

In this paper we propose a simple method for directly locating a feasible solution to (1.1). Let $\mathbf{a} = (a_1, \ldots, a_n)$. The method consists of transforming (1.1) to the problem of finding a particular short vector **e** in an integer lattice $L = L(\mathbf{a}, M)$. Then we apply a lattice basis reduction algorithm to produce a reduced basis of the lattice. This algorithm is due to A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász [13]; we call it the L^3 algorithm. The method succeeds if $\pm\mathbf{e}$ appears in the reduced basis; a solution to (1.1) follows immediately from **e**.

Since the problem (1.1) is *NP*-hard, our method cannot be expected to always succeed. We analyze the circumstances under which it can be expected to work. We define the *density* $d(\mathbf{a})$ of a set of weights $\mathbf{a} = (a_1, \ldots, a_n)$ by

$$d(\mathbf{a}) = \frac{n}{\log_2(\max_i a_i)}.$$

In terms of knapsack public-key cryptosystems, $d(\mathbf{a})$ is an approximate measure of the *information rate* at which bits are transmitted, i.e.,

$$d(\mathbf{a}) \approx \frac{\#\text{ bits in plaintext message}}{\text{average }\#\text{ bits in ciphertext message}}.$$

0272-5428/83/0000/0001$01.00 © 1983 IEEE

Our main result is a performance analysis of our method which shows that it succeeds for "low-density" subset sum problems as follows.

(1) For "almost all" subset sum problems with $d(\mathbf{a}) < .645$, the vector e is the shortest non-zero vector in the lattice $L = L(\mathbf{a}, M)$. (Theorem 3.3)

(2) For "almost all" solvable subset sum problems with n weights having $d(\mathbf{a}) < (1-\epsilon)(\log_2 \frac{4}{3})^{-1} \frac{1}{n}$, for any fixed $\epsilon > 0$, the method finds a solution. (Theorem 3.5 and the remark following its proof.)

We believe that the first result is essentially best possible in the sense that it is no longer true when .645 is replaced by .646. (Our belief is based on heuristic arguments which we describe in Section 5.)

The second result is weaker than what we believe to be true. The reason for this is as follows. The L^3 algorithm is not guaranteed to produce the shortest nonzero vector $\mathbf{x}_{min}$ in a lattice $L \subseteq \mathbf{Z}^n$, but only a relatively short vector. To prove that the algorithm succeeds on "almost all" problems with n weights having density $d(a) < (1-\epsilon)(\log_2 \frac{4}{3})^{-1} \frac{1}{n}$ we use a worst-case bound on the length of the shortest vector found by the L^3 algorithm (Proposition 2.1). Empirical experience with the L^3 algorithm suggests that it usually finds considerably shorter vectors than those guaranteed by this bound. Computational evidence suggests that the algorithm succeeds on "almost all" problems with n items for which $d(a) < d_c(n)$ where $d_c(n)$ is a cutoff value that slowly tends to 0 as $n \to \infty$, and which is substantially larger than $(\log_2 \frac{4}{3})^{-1} \frac{1}{n}$. We do not have enough data to make a reasonable guess on the behavior of $d_c(n)$, but it seems likely that $d_c(n) \to 0$ as $n \to \infty$. See Section 4 for more details.

The algorithm we present uses the L^3 algorithm because it is currently the only known algorithm for finding short vectors in a lattice which has been rigorously proved both to have a polynomial running time and to find reasonably short vectors in a lattice. One could use instead in our algorithm modifications of other algorithms for finding short vectors in a lattice or for finding good multidimensional Diophantine approximations such as those described in [2,3,6,7]; these might perform well in practice.

What are the consequences of these results for breaking knapsack-type public key cryptosystems? First, the empirical evidence implies that this method will very likely break nearly all knapsack cryptosystems for which $d(\mathbf{a}) < d_c(n)$ in polynomial time. In particular, it may well break "almost all" *ultimate knapsack cryptosystems* of Shamir [18] since these cryptosystems have $d(\mathbf{a}) < \frac{1}{\log_2 n}$. Second, our method complements nicely the existing attacks on knapsack cryptosystems which are based on recovering trapdoor information. When the information rate is low, the method described here should succeed. When the information rate is high, the trapdoor information is more difficult to conceal, and attacks based on finding the trapdoor are more likely to succeed, see [11].

E. Brickell [5] has developed another method to solve general subset sum problems, which can be expected to break most "low density" problems. Although his method is superficially dissimilar to our method, its success seems to us to be based on the same basic principles. His method is more complicated and seems difficult to analyze in detail theoretically. Some further remarks on Brickell's algorithm are made in Section 5.

2. The Method

Before describing the method, we state the basic facts about integer lattices and the L^3 algorithm which we shall use.

We present the vector space $\mathbf{R}^n$ using row vectors, and define the *length* (i.e. *Euclidean norm*) $\|\mathbf{v}\|$ of a vector $\mathbf{v} = (v_1, \ldots, v_n)$ by

$$\|\mathbf{v}\|^2 = \sum_{i=1}^{n} v_i^2 . \tag{2.1}$$

An *integer lattice* L is an additive subgroup of $\mathbf{Z}^n$ which contains n linearly independent vectors over $\mathbf{R}^n$. An *(ordered) basis* $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ of a lattice L is a set of elements of L such that $L = \mathbf{Z}\mathbf{v}_1 \oplus \mathbf{Z}\mathbf{v}_2 \oplus \ldots \oplus \mathbf{Z}\mathbf{v}_n$. We represent an ordered basis of a lattice L by the $n \times n$ *basis matrix*

$$V = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix}$$

whose rows are the basis vectors. If V_1 and V_2 are basis matrices of the same lattice L, then there is a unimodular matrix $U \in GL(n, \mathbf{Z})$ such that

$$UV_1 = V_2 .$$

Conversely, if V is a basis matrix of L and $U \in GL(n, \mathbf{Z})$, then UV is a basis matrix of L. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász define the notion of a *reduced* (ordered) basis $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ of a lattice L. For the purpose of this paper we do not need to know the precise definition of a reduced basis (it is given in Appendix A); we need only know that any reduced basis contains a relatively short vector [13, Prop. 1.11].

Proposition 2.1 *Let* $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ *be a reduced basis of a lattice L. Then*

$$\|\mathbf{v}_1\|^2 \leq 2^{n-1} \min_{\substack{\mathbf{x} \in L \\ \mathbf{x} \neq 0}} \|\mathbf{x}\|^2 . \tag{2.2}$$

In fact, all of the basis vectors in a reduced basis tend to be short, cf. [13, Prop. 1.12]; we take advantage of this in our method. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász [13] present an algorithm, which we call the L^3-*algorithm*, which when given a basis $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ of a lattice

L as input produces a reduced basis $[\mathbf{w}_1, \ldots, \mathbf{w}_n]$ as output. They give the following polynomial worst-case running time bound for its performance [13, Prop. 1.26].

Proposition 2.2 *Let $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be a basis of an integer lattice L such that $\|\mathbf{v}_i\|^2 \leq B$ for $1 \leq i \leq n$. Then the L^3 algorithm produces a reduced basis $[\mathbf{w}_1, \ldots, \mathbf{w}_n]$ for L using at most $O(n^4 \log B)$ arithmetic operations, and the integers on which these operations are performed have binary length at most $O(n \log B)$.*

If we use the classical algorithms for addition, subtraction, multiplication and division, this algorithm has a guaranteed running time of $O(n^6 (\log B)^3)$ bit operations. There are some practical speed-ups possible for this algorithm so that it seems possible in practice to find a reduced basis in $O(n (\log B)^3)$ bit operations, cf. [9], [16], and Section 4.

Now we can describe the method. We suppose we are given a vector $\mathbf{a} = (a_1, \ldots, a_n)$ of positive integers and an integer M. Our object is to find a feasible solution to:

$$\sum_{i=1}^{n} a_i x_i = M; \quad \text{all } x_i = 0 \text{ or } 1\,. \tag{2.3}$$

We need only consider the case that $1 \leq M < \sum_{i=1}^{n} a_i$. We use the following algorithm.

Algorithm SV (SV = Short Vector).

(1) Take the following vectors as a basis $[\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}]$ for an $n+1$ dimensional integer lattice $L = L(\mathbf{a}, M)$:

$$\begin{aligned} \mathbf{b}_1 &= (1,0,\ldots,0,-a_1) \\ \mathbf{b}_2 &= (0,1,\ldots,0,-a_2) \\ &\ldots \\ \mathbf{b}_n &= (0,0,\ldots,1,-a_n) \\ \mathbf{b}_{n+1} &= (0,0,\ldots,0,M)\,. \end{aligned} \tag{2.4}$$

(2) Find a *reduced basis* $[\mathbf{b}_1^*, \ldots, \mathbf{b}_{n+1}^*]$ of L using the L^3-algorithm.

(3) Check if any $\mathbf{b}_i^* = (b_{i,1}^*, \ldots, b_{i,n+1}^*)$ has all $b_{ij}^* = 0$ or λ for some fixed λ for $1 \leq j \leq n$. For any such $\mathbf{b}_i^*$, check if $x_j = \frac{1}{\lambda} b_{i,j}^*$ for $1 \leq j \leq n$ gives a solution to (2.3), and if so, halt. Otherwise continue.

(4) Repeat steps 1 through 3 with M replaced by $M' = \sum_{i=1}^{n} a_i - M$. Then halt.

If Algorithm SV produces a solution to (1) we say it *succeeds*; otherwise it *fails*.

Since Algorithm SV is essentially two applications of the L^3 algorithm, we immediately obtain the following running time bound.

Lemma 2.3. *Let $\{a_i\colon 1 \leq i \leq n\}$ and $M < \sum_{i=1}^{n} a_i$ be given as input to Algorithm SV, and suppose $\max a_i \leq B$. Then Algorithm SV halts after at most $O(n^6 (\log nB)^3)$ bit operations.*

3. Performance Analysis

Our goal is to analyze the performance of Algorithm SV on a class of subset sum problems

$$\sum_{i=1}^{n} a_i x_i = M; \quad \text{all } x_i = 0 \text{ or } 1; \tag{3.1}$$

which are known to have a solution. To this end, we suppose that (3.1) has a particular distinguished 0-1 solution $(e_1, \ldots, e_n)$ which we treat as fixed, and that

$$1 \leq \sum_{i=1}^{n} e_i \leq n-1\,,$$

i.e. we exclude the trivial cases where $M = 0$ or $\sum_{i=1}^{n} a_i$. We set $\mathbf{e} = (e_1, \ldots, e_n, 0)$.

We analyze the performance of Algorithm SV over a sample space of lattices. We define this *sample space* $\Lambda(B, \mathbf{e})$ to consist of all lattices $L(\mathbf{a}, M)$ defined by (2.4) such that

(i) $\quad \mathbf{a} = (a_1, \ldots, a_n)$ has $1 \leq a_i \leq B$ for all i (3.2)

$$\text{(ii)} \quad M = M(\mathbf{a}, \mathbf{e}) = \sum_{i=1}^{n} a_i e_i\,. \tag{3.3}$$

In particular there is exactly one lattice $L(\mathbf{a}, M)$ in $\Lambda(B, \mathbf{e})$ for each $\mathbf{a}$ satisfying (3.2); hence $\Lambda(B, \mathbf{e})$ contains exactly B^n lattices. The distinguished vector $\mathbf{e}$ is in all the lattices in the sample space $\Lambda(B, \mathbf{e})$ since (2.4) and (3.2) give

$$\mathbf{e} = \sum_{i=1}^{n} e_i \mathbf{b}_i + \mathbf{b}_{n+1}\,. \tag{3.4}$$

The connection between the sample space $\Lambda(B, \mathbf{e})$ and the density $d(\mathbf{a})$ of its associated subset sum problems is as follows. All subset sum problems (3.1) associated to lattices in $\Lambda(B, \mathbf{e})$ have

$$d(\mathbf{a}) \geq \frac{n}{\log_2 B}\,, \tag{3.5}$$

and every $\mathbf{a}$ satisfying (3.5) contributes exactly one lattice to $\Lambda(B, \mathbf{e})$. Furthermore for any $\epsilon > 0$ the fraction of lattices in $\Lambda(B, \mathbf{e})$ with

$$d(\mathbf{a}) \leq \frac{n}{\log_2 (B(1-\epsilon))}$$

goes to 1 as $n \to \infty$ if $\log_2 B \sim cn$ for some $c > 0$. Consequently the sample space $\Lambda(B, \mathbf{e})$ may be regarded as sampling subset sum problems of density $\frac{n}{\log_2 B}$.

We can now formulate the problem we want to solve as follows: Determine how often Algorithm SV finds the distinguished vector $\mathbf{e}$, when applied to all the lattices in the sample space $\Lambda(B, \mathbf{e})$. This problem is intimately tied to the question: How short is $\mathbf{e}$ relative to other short vectors in the lattices in $\Lambda(B, \mathbf{e})$? We consider this question first.

The expected length of other short vectors in lattices in $\Lambda(B, \mathbf{e})$ other than the distinguished vector $\mathbf{e}$ can be determined using Theorem 3.1 below. The bound given by Theorem 3.1 involves the number of lattice points in spheres in n-dimensional space. We define $S_n(R)$ to be the number of integer solutions to the inequality

$$\sum_{i=1}^{n} x_i^2 \leq R , \tag{3.6}$$

i.e., the number of integer lattice points inside or on the n-dimensional sphere of radius $\sqrt{R}$ centered at the origin.

Theorem 3.1. *The number of lattices $L(\mathbf{a})$ in the sample space $\Lambda(B, \mathbf{e})$ which contain a vector $\mathbf{w}$ such that*

(i) $\mathbf{w} \neq k\mathbf{e}$ *for all integers* k,

(ii) $\|\mathbf{w}\|^2 \leq R$,

is

$$O(R\, S_n(R) B^{n-1} \log(BR)) . \tag{3.7}$$

Proof. Let $T = T(R, B, \mathbf{e})$ denote the number of such lattices. Let $\mathbf{w} = (w_1, \ldots, w_n, r) \in \mathbf{Z}^{n+1}$ be a fixed vector satisfying

$$\|\mathbf{w}\|^2 = \sum_{i=1}^{n} w_i^2 + r^2 \leq R \tag{3.8}$$

and suppose that $\mathbf{w} \neq k\mathbf{e}$ for every integer k. We count how many lattices $L(\mathbf{a})$ in $\Lambda(B, \mathbf{e})$ contain $\mathbf{w}$. If $\mathbf{w} \in L(\mathbf{a})$ then expressing $\mathbf{w}$ in terms of the basis vectors (2.4) of $L(\mathbf{a})$ gives

$$\mathbf{w} = \sum_{i=1}^{n} w_i \mathbf{b}_i + \lambda \mathbf{b}_{n+1} \tag{3.9}$$

for some integer λ. In particular, evaluating the last coordinate of (3.9) gives

$$r = \sum_{i=1}^{n} w_i a_i - \lambda M(\mathbf{a}) \tag{3.10}$$

and using (3.3) gives

$$r = \sum_{i=1}^{n} (w_i - \lambda e_i) a_i . \tag{3.11}$$

We can easily bound λ using (3.10); we obtain

$$|\lambda| M \leq |r| + \sum_{i=1}^{n} |w_i a_i| \leq B(|r| + \sum_{i=1}^{n} |w_i|) \leq RB \tag{3.12}$$

using (3.8), since r and the w_i are integers, so that $M \geq 1$ implies

$$|\lambda| \leq RB . \tag{3.13}$$

Next we note that since $\mathbf{e} \neq 0$, it has a nonzero coordinate, which we suppose to be e_1 for convenience in subsequent calculations. Then

$$M = M(\mathbf{a}, \mathbf{e}) \geq a_1 e_1 = a_1 \tag{3.14}$$

so that (3.12) gives

$$a_1 \leq \frac{RB}{|\lambda|} , \quad \text{if } \lambda \neq 0 . \tag{3.15}$$

Also we note that (3.8) implies

$$|r| \leq R^{1/2} . \tag{3.16}$$

Now we commence counting. Let $N(\mathbf{w}, \lambda)$ denote the number of lattices $L(\mathbf{a})$ in $\Lambda(B, \mathbf{e})$ for which $\mathbf{w}$ is in $L(\mathbf{a})$ and for which λ satisfies (3.9). Then (3.13) gives:

$$T \leq \sum_{\|\mathbf{w}\|^2 \leq R}' \left\{ \sum_{\lambda=-RB}^{RB} N(\mathbf{w}, \lambda) \right\} , \tag{3.17}$$

where the prime in the summation indicates that all $\mathbf{w}$ with

$$\mathbf{w} = k\mathbf{e}; \quad k \text{ an integer} ; \tag{3.18}$$

are excluded. To estimate this sum, we divide the sum of the right side of (3.17) into four sums, depending on the value of the auxiliary vector

$$\mathbf{z} = \mathbf{z}(\mathbf{w}, \lambda) = (w_1 - \lambda e_1, \ldots, w_n - \lambda e_n) \tag{3.19}$$

and the value of λ.

Case 1. $\mathbf{z} = 0$.

In this case (3.19) gives

$$\mathbf{w} = (\lambda e_1, \ldots, \lambda e_n, N) \tag{3.20}$$

for some $N \neq 0$. Then

$$\mathbf{w} - \lambda \mathbf{e} = (0, \ldots 0, N)$$

is in $L(\mathbf{a})$, so that necessarily $N = kM(\mathbf{a})$ for some integer k. If $k = 0$, then $\mathbf{w} = \lambda\mathbf{e}$, which is ruled out by hypothesis (i). Hence $|k| \geq 1$ and

$$\|\mathbf{w}\| \geq |M(a)| \geq a_1 ,$$

using (3.14). The condition $\|w\|^2 < R$ implies that

$$a_1 \leqslant R^{1/2}. \tag{3.21}$$

Consequently we obtain the bound

$$N(\mathbf{w}, \lambda) \leqslant R^{1/2}B^{n-1}.$$

Now there are no more than $S_n(R)$ choices of w, and each such w uniquely determines λ via (3.20), so that

$$\sum_{\text{Case 1}} N(\mathbf{w}, \lambda) = O(R^{1/2}S_n(R)B^{n-1}). \tag{3.22}$$

Case 2. $w_1 - \lambda e_1 \neq 0$ *and* $w_j - \lambda e_j = 0$ *for* $2 \leqslant j \leqslant n$.

In this case (3.11) gives

$$r = (w_1 - \lambda e_1)a_1. \tag{3.23}$$

Together with (3.15) this gives

$$1 \leqslant a_1 \leqslant R^{1/2}, \tag{3.24}$$

so that

$$N(\mathbf{w}, \lambda) \leqslant R^{1/2}B^{n-1} \tag{3.25}$$

for such pairs $(\mathbf{w}, \lambda)$.

How many such pairs $(\mathbf{w}, \lambda)$ can occur? We have the bound

$$|w_1| < R^{1/2} \tag{3.26}$$

from (3.8), while (3.23) and (3.16) yield

$$|w_1 - \lambda e_1| \leqslant \frac{r}{a_1} \leqslant R^{1/2}. \tag{3.27}$$

Combining (3.26) and (3.27) and using $e_1 = 1$ gives

$$|\lambda| \leqslant 2R^{1/2}. \tag{3.28}$$

The values of $(w_2, \ldots, w_n)$ are all determined by

$$w_j = \lambda e_j,$$

so that there are $O(R)$ choices of pairs $(\mathbf{w}, \lambda)$ in case 2. Hence

$$\sum_{\text{Case 2}} N(\mathbf{w}, \lambda) = O(R^{3/2}B^{n-1}). \tag{3.29}$$

Case 3. $w_j - \lambda e_j \neq 0$ *for some* $j \geqslant 2$, *and* $\lambda \neq 0$.

Consider w and λ as fixed. Now by (3.15) there are at most $\frac{RB}{\lambda}$ choices for a_1. Now choose all the other a_i arbitrarily, except for $i = j$. There are B^{n-2} such choices. For each such choice there is at most one possible choice for a_j, since a_j is determined by equation (3.11), since $w_j - \lambda e_j \neq 0$. Hence in this case

$$N(\mathbf{w},\lambda) \leqslant \frac{RB^{n-1}}{\lambda}. \tag{3.30}$$

Hence

$$\sum_{\text{Case 3}} N(\mathbf{w}, \lambda) \leqslant \sum_{\|w\|^2 \leqslant R} \sum_{\substack{\lambda = -RB \\ \lambda \neq 0}}^{RB} \frac{RB^{n-1}}{\lambda}$$

$$\leqslant 2RB^{n-1}S_n(R) \sum_{\lambda=1}^{RB} \frac{1}{\lambda}.$$

Since

$$\sum_{i=1}^{RB} \frac{1}{\lambda} = O(\log(RB)),$$

this yields

$$\sum_{\text{Case 3}} N(\mathbf{w}, \lambda) = O(RS_n(R)B^{n-1}\log(RB)). \tag{3.31}$$

Case 4. Some $w_j - \lambda e_j \neq 0$ *for* $j \geqslant 2$ *and* $\lambda = 0$.

Consider w as fixed. In this case we can pick all a_i except a_j arbitrarily, and there are B^{n-1} such choices. There are at most $2R^{1/2}+1$ choices for a_j, since it must satisfy (3.11) and there are at most $2R^{1/2}+1$ choices of r by (3.16). Hence in this case

$$N(\mathbf{w}, 0) \leqslant (2R^{1/2}+1)B^{n-1}.$$

Consequently summing over all w gives

$$\sum_{\text{Case 4}} N(\mathbf{w}, \lambda) \leqslant (2R^{1/2}+1)S_n(R)B^{n-1}. \tag{3.32}$$

Theorem 3.1 follows on combining the bounds (3.22), (3.29), (3.31) and (3.32), together with the trivial inequality $S_n(R) \geqslant R$. □

We remark that the dependence on B in Theorem 3.1 cannot be much improved, since all $L(\mathbf{a}, M)$ for which $a_1 = a_2$ contain the short vector $\mathbf{w} = (1,-1,0,0,\ldots,0)$ which satisfies the conditions of Theorem 3.1, and there are B^{n-1} such lattices in $\Lambda(B, \mathbf{e})$. It is an interesting question as to whether or not substantial improvement is possible in the dependence on R in (3.7).

To apply Theorem 3.1, we need explicit estimates for the number of lattice points in spheres. A general principle here is that $S_n(R)$ should be equal to the volume $V_n(R)$ of a sphere of radius $R^{1/2}$, with an error proportional to the surface area a $A_n(R)$ of such a sphere. Now

$$V_n(R) = c_n R^{n/2}, \tag{3.33}$$

$$A_n(R) = nc_n R^{(n-1)/2},$$

where

$$c_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \tag{3.34}$$

is the volume of an n-dimensional sphere of radius 1. For large R one has $V_n(R)$ much larger than $A_n(R)$, but for R small enough, say $R = \alpha n$, this is not true, and spheres of this radius centered at the origin contain many more lattice points than their volume would suggest. It turns out, furthermore, that for n-dimensional spheres of such small radius αn, the number of lattice points in the sphere depends strongly on the location of the center of the sphere, cf. [14]. For our application we need a good upper bound for $S_n(\frac{1}{2}n)$, and to obtain it we use the following simplified version of the proof in [14].

Theorem 3.2. *For all $n \geq 1$, $S_n(\frac{1}{2}n) \leq 2^{1.54725n}$.*

Proof. Let $\theta(z) = 1+2\sum_{i=1}^{\infty} z^{i^2}$. Let $r_n(k)$ count the number of solutions to

$$\sum_{i=1}^{n} x_i^2 = k .$$

Then

$$[\theta(z)]^n = \sum_{k=0}^{\infty} r_n(k) z^k .$$

Now for $x \geq 0$ we have

$$\begin{aligned} S_n(\alpha n) &= \sum_{k \leq \alpha n} r_n(k) \\ &\leq e^{n\alpha x} \sum_{k=0}^{\infty} r_n(k) e^{-kx} \\ &= e^{n\alpha x}[\theta(e^{-x})]^n \end{aligned} \tag{3.35}$$

since for $x \geq 0$ we have

$$e^{n\alpha x}e^{-kx} \geq 1 \quad \text{when} \quad k \leq n\alpha .$$

Now set

$$\delta(\alpha, x) = \alpha x + \ln \theta(e^{-x})$$

and observe that (3.35) gives

$$S_n(\alpha n) \leq e^{n\delta} = 2^{(\log_2 e)\delta(\alpha,x)n} . \tag{3.36}$$

We are interested in $\alpha = 1/2$ and choose $x \geq 0$ to optimize (3.36); the value $x = x_0 = 0.997994$ is a nearly optimal choice. Then

$$\delta(\tfrac{1}{2}, x_0) \leq 1.07247$$

and

$$(\log_2 e)\delta(\tfrac{1}{2}, x_0) \leq 1.54725 . \qquad \square$$

We remark that the constant 1.54725 in Lemma 3.2 is best possible to within one unit in the last decimal place (see [14]).

Now we prove a result about short vectors in lattices in the class $\Lambda(B, \mathbf{e})$ where $\mathbf{e}$ satisfies

$$\sum_{i=1}^{n} e_i \leq \frac{1}{2} n . \tag{3.37}$$

The reason we consider this extra condition is that Algorithm SV examines two lattice problems, one of which is a lattice $L(\mathbf{a}, \mathbf{e})$ and the other $L(\mathbf{a}, \mathbf{e}^*)$ where $\mathbf{e}^* = (e_1^*, \ldots, e_n^*)$ is the 0-1 vector complementary to $\mathbf{e}$, i.e. $e_i^* = 1-e_i$ for all i. Since

$$\min(\sum_{i=1}^{n} e_i, \sum_{i=1}^{n} e_i^*) \leq \frac{1}{2} n ,$$

the hypothesis (3.37) applies to at least one of these lattice problems.

Theorem 3.3. *Let $\mathbf{e}$ be a 0-1 vector for which $\sum_{i=1}^{n} e_i \leq \frac{n}{2}$. Then if $B = 2^{\beta n}$ for any constant $\beta > 1.54725$, the number of lattices L in $\Lambda(B, \mathbf{e})$ for which $\mathbf{e}$ is the nonzero vector of shortest Euclidean norm in L is*

$$B^n + O(B^{n-c_1(\beta)}(\log B)^2)$$

where $c_1(\beta) = 1 - \dfrac{1.54725}{\beta} > 0$.

This theorem asserts that, under the stated hypotheses, "almost all" the lattices in $\Lambda(B, \mathbf{e})$ have $\mathbf{e}$ as the shortest vector. In particular, for $B = 2^{\beta n}$ the density $d(\mathbf{a})$ of lattices in $\Lambda(B, \mathbf{e})$ is β^{-1}, so that this theorem applies to sets of lattices with density less than $(1.54725)^{-1} \cong .645$.

Proof of Theorem 3.3. Theorem 3.1 estimates the number of such lattices by

$$B^n + O(n\, S_n(\tfrac{1}{2} n) B^{n-1} \log(B_n)) .$$

Applying Theorem 3.2 gives

$$S_n(\tfrac{1}{2} n) \leq 2^{1.54725n} \leq B^{1-c_1(\beta)} ,$$

where $B \geq 2^{\beta n}$. Finally $n \log B_n = O((\log B)^2)$ for $B \geq 2^{\beta n}$, and the theorem follows. $\square$

Theorem 3.3 gave a result when the vector $\mathbf{e}$ is fixed. We can immediately derive a result where $\mathbf{e}$ varies.

Theorem 3.4. *Let $B = 2^{\beta n}$ for any $\beta > 2.54725$. The number of vectors $\mathbf{a} = (a_1, \ldots, a_n)$ with $1 \leqslant a_i \leqslant B$ for $1 \leqslant i \leqslant n$ for which* **e** *is the shortest vector in $L(\mathbf{a}, \mathbf{e})$ for all 0-1 vectors* **e** *for which*

$$1 \leqslant \sum_{i=1}^{n} e_i \leqslant \frac{n}{2} \tag{3.38}$$

is

$$B^n + O(B^{n-c_2(\beta)}(\log B)^2) \tag{3.39}$$

where $c_2(\beta) = 1 - \dfrac{2.54725}{\beta} > 0$.

Proof. Sum the result of Theorem 3.1 over all $2^{n-1}-1$ vectors **e** satisfying (3.38). The resulting bound is

$$O(n\, 2^n S_n(\tfrac{n}{2}) B^{n-1} \log(nB)) .$$

This is certainly an upper bound for the error term in (3.39). Now use

$$2^n S_n(\tfrac{n}{2}) \leqslant 2^{2.54725 n} \leqslant B^{1-c_2(\beta)} .$$

and the result follows. □

Theorem 3.4 makes an assertion about lattices of density $d(\mathbf{a}) \leqslant .393 < (2.54725)^{-1}$.

Theorem 3.5. *Let $B \geqslant 2^{(1+\beta)n^2}$ for some fixed $\beta > 0$. Then the number of vectors $\mathbf{a} = (a_1, \ldots, a_n)$ with $1 \leqslant a_i \leqslant B$ for all i for which Algorithm SV will succeed for all 0-1 vectors* **e** *is*

$$B^n + O((1+\beta) B^{n - c_3(\beta) + 3 \frac{\log n}{n}}) ,$$

where $c_3(\beta) = 1 - (1+\beta)^{-1} > 0$.

This theorem asserts then that for any fixed $\beta > 0$ one can solve the subset sum problem for "almost all" $\mathbf{a} = (a_1, \ldots, a_n)$ for which $d(\mathbf{a}) < (1+\beta)^{-1} \dfrac{1}{n}$, provided $n \geqslant n_0(\beta)$.

Proof of Theorem 3.5. At least one of the two lattice problems Algorithm SV considers has an associated **e** satisfying

$$\sum_{i=1}^{n} e_i \leqslant \frac{1}{2} n . \tag{3.40}$$

Now suppose for this lattice problem that the lattice $L(\mathbf{a}, \mathbf{e})$ is a lattice with the property that all vectors **w** in $L(\mathbf{a}, \mathbf{e})$ which are not a scalar multiple of **e** satisfy

$$\|\mathbf{w}\| > n\, 2^{n-2} \geqslant 2^{n-1} \|\mathbf{e}\|$$

using (3.40). Then Proposition 2.1 guarantees that some vector $\lambda\mathbf{e}$ must appear in the reduced basis produced by the L^3 algorithm applied to $L(\mathbf{a}, \mathbf{e})$. Hence Algorithm SV succeeds in this case. (We remark that if $\lambda\mathbf{e}$ appears in a reduced basis, then necessarily $\lambda = \pm 1$.)

It remains to bound the exceptional cases where this does not occur. We use the bound of Theorem 3.1 with $R = n\, 2^{n-2}$, summing over all **e** satisfying (3.40), to obtain the upper bound

$$O(n\, 2^n s_n(n\, 2^{n-2}) B^{n-1} \log(n\, 2^{n-2} B)) , \tag{3.41}$$

for the exceptional cases. Then using the trivial bound

$$S_n(R) \leqslant (2R+1)^n \leqslant 3nR^n$$

we can easily obtain an upper bound for (3.41) of

$$O((1+\beta) 2^{n^2 + 3 \log_2 n} B^{n-1}) .$$

Taking $B = 2^{(1+\beta)n^2}$, we find that

$$2^{n^2 + 3 \log n} = O(B^{1 - c_3(\beta) + 3 \frac{\log n}{n}})$$

where $c_3(\beta) = 1 - (1+\beta)^{-1}$. □

Theorem 3.5 can be sharpened by using an improved form of the L^3-algorithm. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász [13] actually defined a notion of *y-reduced basis,* which depends on a parameter y satisfying $\frac{1}{4} \leqslant y < 1$. The notion of reduced basis corresponds to choosing $y = 3/4$; the general definition is given in Appendix A. For a y-reduced basis the bound (2.2) of Proposition 2.1 is replaced by:

$$\|\mathbf{v}_i\|^2 \leqslant \left(\frac{4}{4y-1}\right)^{n-1} \min_{\substack{\mathbf{x} \in L \\ \mathbf{x} \neq 0}} \|\mathbf{x}\|^2 .$$

They gave an algorithm, which we may call the $L^3(y)$-algorithm, which produces a y-reduced basis. An analogue of Proposition 2.2 holds for this algorithm, in which the constants implied by the O-symbols depend on the choice of y. We can modify Algorithm SV to use the $L^3(y)$-algorithm and obtain Algorithm SV(y). Then we may prove Theorem 3.5 for Algorithm SV(y), obtaining a similar bound for $B = 2^{(1+\beta)c(y)n^2}$ where $c(y) = \log_2 \dfrac{4}{4y-1}$. With this bound, letting $y \to 1$, we get a result which asserts that we can solve the subset sum problem for "almost all" problems of density $d(\mathbf{a}) < (1-\epsilon)\left(\log_2 \dfrac{4}{3}\right)^{-1} \dfrac{1}{n}$.

4. Computational Results

We performed extensive computational tests using Algorithm SV. We tested several variants of Algorithm SV obtained by modifying the L^3 algorithm in ways designed to improve its chance of finding the shortest vector in a lattice. We considered two such modifications.