Topics in Basis Reduction and Integer Programming

Mustafa Kemal Tural

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2009

Approved by:

Shu Lu

Gábor Pataki

Scott Provan

David Rubin

Jon Tolle

Abstract

MUSTAFA KEMAL TURAL: Topics in Basis Reduction and Integer Programming
(Under the direction of Gábor Pataki)

A basis reduction algorithm computes a reduced basis of a lattice consisting of short and nearly orthogonal vectors. The best known basis reduction method is due to Lenstra, Lenstra and Lovász (LLL): their algorithm has been extensively used in cryptography, experimental mathematics and integer programming. Lenstra used the LLL basis reduction algorithm to show that the integer programming problem can be solved in polynomial time when the number of variables is fixed.

In this thesis, we study some topics in basis reduction and integer programming. We make the following contributions.

We unify the fundamental inequalities in an LLL reduced basis, which express the shortness and near orthogonality of the basis.

We analyze two recent integer programming reformulation techniques which also rely on basis reduction. The reformulation methods are easy to describe. They are also successful in practice in solving several classes of hard integer programs.

First, we analyze the reformulation techniques on bounded knapsack problems. The only analyses so far are for knapsack problems with a constraint vector having a certain decomposable structure. Here we do not assume any a priori structure on the constraint vector.

We then analyze the reformulation techniques on bounded integer programs. We show that if the coefficients of the constraint matrix are drawn from a sufficiently large interval, then branch and bound creates at most one node at each level if applied to the reformulated instances.

On the practical side, we give some numerical values as to how large the numbers should be to make sure that for 90 and 99 percent of the reformulated instances, the number of subproblems that need to be enumerated by branch and bound is at most one at each level. These values turned out to be surprisingly small when the problem size is moderate.

We also analyze the solvability of the "majority" of the low density subset sum problems using the method of branch and bound when the coefficients are chosen from a large interval.

Acknowledgements

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends and support from my family and wife.

First of all, I would like to express my deepest gratitude to my advisor Dr. Gábor Pataki for his invaluable guidance, advice and inspiration during my graduate study. I would like to thank Dr. David Rubin for carefully going through my dissertation and suggesting corrections. I would also like to thank my other dissertation committee members Dr. Shu Lu, Dr. Scott Provan and Dr. Jon Tolle.

I want to thank Dr. Serhan Ziya for his support throughout my graduate study. I also want to thank Dr. Edcard Carlstein who has been a remarkable teaching advisor. Dr. Edcard Carlstein was always there to listen and to give advice.

I am deeply indebted to my parents, Makbule Tural and Veysi Tural. Their love and support provided me the energy to attain my study.

Finally, I would like to thank my wife Dilber for her continual support, patience and understanding. The completion of this dissertation would not have been possible without her endless love and unconditional support.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Algorithms based on geometry of numbers have been an essential part of the integer programming (IP) landscape starting with the work of H. W. Lenstra [36]. Typically, these algorithms reduce an IP feasibility problem to a provably small number of smaller dimensional ones and have strong theoretical properties. For instance, the algorithms of [27, 36, 39] have polynomial running time in fixed dimension; the algorithm of [14] has linear running time in dimension two. One essential tool in creating the subproblems is a "thin" branching direction, i.e., an integral (row-)vector c with the difference between the maximum and the minimum of cx over the underlying polyhedron being provably small. Basis reduction in lattices – in the Lenstra, Lenstra and Lovász (LLL) [35], or Korkine and Zolotarev (KZ) [27, 30] sense – is usually a key ingredient in the search for a thin direction. For implementations and computational results, we refer to [10, 18, 41].

A simple and experimentally very successful reformulation technique for integer programming was proposed by Aardal, Hurkens and A. K. Lenstra in [2] for equality constrained IP problems; see also [1]. For several classes of hard equality constrained integer programming problems – e.g., [11] – the reformulation turned out to be much easier to solve by commercial solvers than the original problem.

In [31] an experimentally just as effective reformulation method was introduced, which leaves the number of the variables the same and is applicable to both inequality or equality constrained problems.

These reformulation methods are very easy to describe (as opposed to say Lenstra's and Kannan's methods), but seem difficult to analyze. The only analyses are for knapsack problems, with the weight vector having a given "decomposable" structure. See [3, 31].

These reformulation methods also rely on basis reduction. A basis reduction algorithm computes a reduced basis of a lattice consisting of "short" and "nearly orthogonal" vectors. There are different

notions of reducedness. In this thesis, we will use LLL, KZ, and RKZ reduced bases. An LLL reduced basis of a lattice can be computed in polynomial time for rational lattices. The first vector of an LLL reduced basis of a lattice L is an approximation of a nonzero shortest vector in L. In an LLL reduced basis, as shown in [35], the norm of the first vector is bounded by a function of the norm of a nonzero shortest vector of L and also by a function of the determinant of L. The product of the norms of the basis vectors is also bounded by a function of the determinant of L. We call these three inequalities "the fundamental inequalities of an LLL reduced basis". KZ [27, 30] and RKZ [32] reduced bases have stronger reducedness properties, but are only computable in polynomial time when the dimension n of the lattice is fixed. Section 2.2 provides some details about basis reduction and different notions of reducedness.

This thesis studies some topics in geometry of numbers and integer programming. It makes the following contributions:

(1) It generalizes the fundamental inequalities for an LLL reduced basis.

(2) It provides an analysis of the IP reformulation techniques for knapsack problems without assuming any a priori structure on the constraint vector.

(3) It resolves the question of the solvability of an overwhelming majority of the subset sum (feasibility) problems (all but a vanishing proportion of the problems as n increases) in polynomial time using the method of branch and bound. We will assume that the coefficients of the subset sum problems are chosen from a sufficiently large interval of integers. In more detail, we have the following results. We show that an overwhelming majority of the subset sum problems are hard for ordinary branch and bound. On the other hand, an overwhelming majority of the subset sum problems are easy for generalized branch and bound. Moreover, if we reformulate the subset sum problem using the rangespace [31] or the nullspace [2] reformulation, then an overwhelming majority of the reformulated problems become easy for ordinary branch and bound. Here the word "easy" means the problem is solved in polynomial time and at most one branch and bound node is created at each level of the branch and bound tree in the process of solving it. A "hard" problem, however, can be solved only by creating an exponential number of nodes.

(4) It shows that for general bounded integer programs, if the coefficients are chosen from a suffi-

ciently large interval, then for almost all such instances the number of subproblems that need to be enumerated by branch and bound is at most one at each level of the branch and bound tree (when applied to reformulated instances).

(5) On the practical side, it provides numerical values of M which ensure that at least 90 and 99 percent of the reformulated (binary) instances (with coefficients chosen from $\{1, \ldots, M\}$) solve in at most n subproblems. These numbers are surprisingly small for moderate-size binary problems.

(6) It computationally confirms the somewhat counter-intuitive finding: the reformulations of random integer programs tend to get easier, as the coefficients become larger.

The rest of the thesis is organized as follows. In Chapter 2, we give notation, definitions and basic results that will be used throughout the proposal. Here, we introduce a modified version of Lenstra's algorithm which potentially uses a smaller number of rounding and basis reduction steps.

In Chapter 3, we unify and generalize the fundamental inequalities for an LLL reduced basis.

In Chapter 4, we analyze two integer programming reformulations of the knapsack problem, namely the rangespace and the nullspace reformulations. We first show that in a knapsack problem, branching on an integral vector which is "near parallel" to the constraint vector creates a small number of branch and bound nodes. A transference result proves an upper bound on the integer width along the last variable in the reformulated problems. This upper bound becomes 1 when the density is sufficiently small, i.e., when the Euclidean norm of the constraint vector is sufficiently large.

In Chapter 5, we show that for a low density subset sum problem, there is a polynomial time computable certificate of infeasibility for almost all integer right hand sides β. Using a transference result, we prove that for almost all right hand sides, the integer width along the last variable in the rangespace reformulation of a low density subset sum problem is zero.

In Chapter 6, we show that the classical branch and bound algorithm is surprisingly efficient on reformulations of bounded integer programs. We show that when the coefficients of the constraint matrix are chosen from a large interval, then branch and bound creates at most one branch and bound node at each level of the branch and bound tree if applied to reformulated instances. Our computational study confirms our theoretical finding that the reformulations of random integer programs become easier, as the coefficients grow.

In Chapter 7, we modify a result of Chvátal and show that an overwhelming majority of the subset sum (feasibility) problems are hard for ordinary branch and bound if the coefficients are chosen from a sufficiently large interval of integers.

CHAPTER 2

Notation, Definitions and Basic Results

2.1 Basics

Let $\langle ., . \rangle$ be the Euclidean scalar product on $\mathbb{R}^m$, i.e., for any $x, y \in \mathbb{R}^m$

$$\langle x, y \rangle = \sum_{i=1}^{m} x_i y_i,$$

where x_i and y_i are the ith components of x and y, respectively. We use $\| . \|$ or $\| . \|_2$ for the Euclidean norm, i.e. for any $x \in \mathbb{R}^m$

$$\| x \| = \| x \|_2 = \sqrt{\langle x, x \rangle}.$$

Two other norms will be important for our purposes: the ℓ_1 norm and the ℓ_∞ norm

$$\| x \|_1 = \sum_{i=1}^{m} |x_i|$$

$$\| x \|_\infty = \max_i |x_i|.$$

When we want to talk about the ℓ_1 or ℓ_∞ norms of a vector, we explicitly say so. When we just say "norm of x", we mean the Euclidean norm of x.

It is known that for all $x \in \mathbb{R}^m$, the following relations hold:

$$\| x \| \leq \| x \|_1 \leq \sqrt{m} \, \| x \|, \tag{2.1.1}$$

$$\| x \|_\infty \leq \| x \| \leq \sqrt{m} \, \| x \|_\infty, \tag{2.1.2}$$

$$\| x \|_\infty \leq \| x \|_1 \leq m \, \| x \|_\infty \, . \tag{2.1.3}$$

For any $x, y \in \mathbb{R}^m$, we have the Cauchy-Schwarz Inequality:

$$|\langle x, y \rangle| \leq \| x \| \| y \| \, . \tag{2.1.4}$$

Equality holds if and only x and y are linearly dependent.

For a matrix B, B_{ij} is the entry at the intersection of ith row and jth column of B. We let B^{T} denote the transpose of B. For an invertible matrix B, B^{-1} denotes the inverse of B and $B^{\text{-T}}$ denotes the transpose of the inverse of B.

For an m-by-m matrix $B = [b_1, \ldots, b_m]$, $\det(B)$ represents the determinant of B. B is called nonsingular if $\det(B) \neq 0$, otherwise it is singular. We have Hadamard's Inequality

$$|\det(B)| \leq \prod_{i=1}^{m} \| b_i \| \, . \tag{2.1.5}$$

Equality holds if and only if either both sides are zero or the vectors $b_1, \ldots, b_m$ are orthogonal.

For matrices (and vectors) A and B with appropriate dimensions, we write $(A; B)$ for $\begin{pmatrix} A \\ B \end{pmatrix}$; and we write (A, B) for $(A \ B)$.

2.2 Lattices and Basis Reduction

A lattice in $\mathbb{R}^m$ is a set of the form

$$L = \mathbb{L}(B) = \{ Bx \,|\, x \in \mathbb{Z}^n \}, \tag{2.2.6}$$

where B is a real matrix with m rows and n independent columns, called a *basis* of L. A lattice has infinitely many different bases when $n \geq 2$. Any basis B of a lattice L has the same number of columns, called the *dimension* of L. A square, integral matrix U is *unimodular* if $\det(U) = \pm 1$. It is well known that B_1 and B_2 are bases of the same lattice if and only if $B_2 = B_1 U$ for some unimodular U.

An elementary column operation performed on a matrix B is either

(1) exchanging two columns,

(2) multiplying a column by -1, or

(3) adding an integral multiple of a column to another column.

Multiplying a matrix B from the right by a unimodular U is equivalent to performing a sequence of elementary column operations on B.

The determinant of L is

$$\det L = (\det(B^T B))^{1/2}, \tag{2.2.7}$$

where $B = [b_1, \ldots, b_n]$ is a basis of L; it is easy to see that $\det L$ is well-defined. From Hadamard's Inequality, it follows that

$$\det L \leq \prod_{i=1}^{n} \| b_i \| .$$

The determinant of a lattice is the n-dimensional volume of the paralelepiped defined by any basis of the lattice (see Figure 2.1).

A lattice L in $\mathbb{R}^m$ is full dimensional if dimension of L is equal to m. Equivalently $L \subseteq \mathbb{R}^m$ is full dimensional if and only if the smallest subspace of $\mathbb{R}^m$ containing L is $\mathbb{R}^m$.

Example 1. *Let* $\Lambda = \mathbb{L}(B_1)$ *where*

$$B_1 = \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix}.$$

Lattice Λ *consists of all integral vectors* $x \in \mathbb{Z}^2$ *such that* x_2 *is even. The green area defined by the columns of* B_1 *is equal to* $\det \Lambda$ *which is* 2.

Lattice Λ *is also generated by the columns of*

$$B_2 = \begin{pmatrix} 1 & 5 \\ 0 & 2 \end{pmatrix},$$

since $B_2 = B_1 U$, *where*

$$U = \begin{pmatrix} 1 & 3 \\ -1 & -2 \end{pmatrix}$$

is a unimodular matrix. The pink area defined by the columns of B_2 *is also equal to* 2.

Note that in Example 1, Λ has an orthogonal basis. But not all lattices have an orthogonal basis.

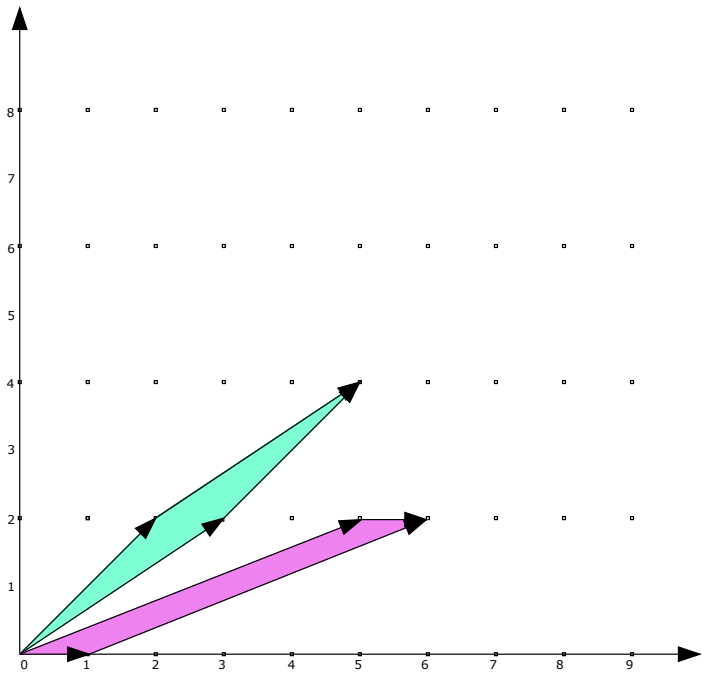

Figure 2.1: A Lattice in $\mathbb{R}^2$.

Example 2. *Let* $\Gamma = \mathbb{L}(B_2)$ *where*

$$B_2 = \begin{pmatrix} 1 & 3 \\ 3 & 2 \end{pmatrix}.$$

The lattice Γ *does not have any orthogonal basis. Note that both* Λ *and* Γ *are full dimensional lattices.*

Suppose that B has n independent columns

$$B = [b_1, \ldots, b_n], \tag{2.2.8}$$

and $b_1^*, \ldots, b_n^*$ form the Gram-Schmidt orthogonalization of $b_1, \ldots, b_n,$ that is $b_1 = b_1^*,$ and

$$b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{ij} b_j^* \text{ with } \mu_{ij} = b_i^T b_j^* / \| b_j^* \|^2 \quad (i = 2, \ldots, n;\ j \leq i-1). \tag{2.2.9}$$

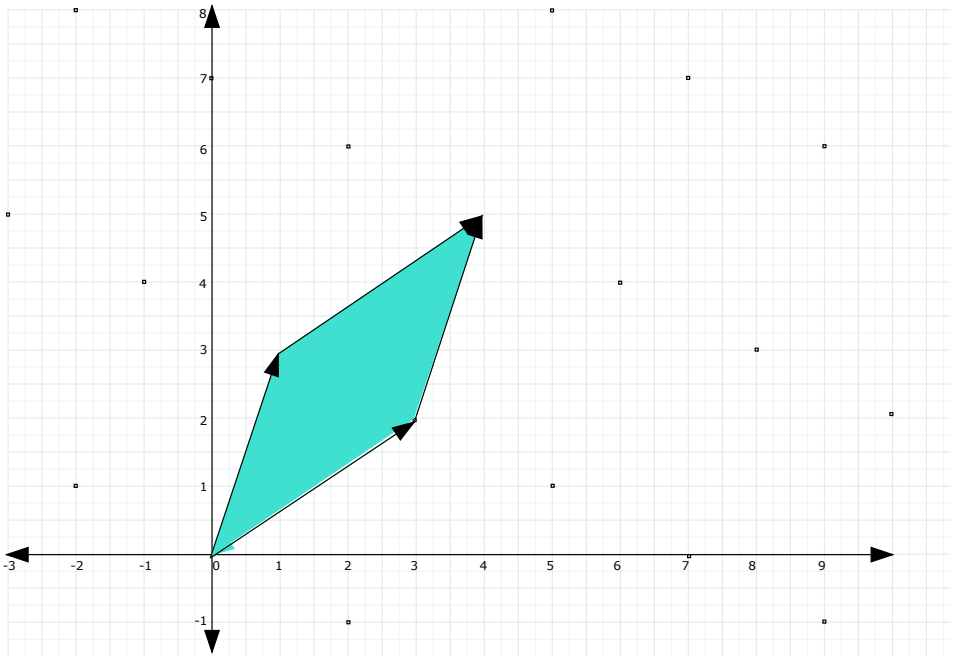

Figure 2.2: A Lattice with no Orthogonal Basis.

In terms of the Gram-Schmidt vectors,

$$\det \mathbb{L}(B) = \prod_{j=1}^{n} \| b_j^* \| \, . \tag{2.2.10}$$

Each lattice L contains a nonzero shortest vector. Let $\lambda_1(L)$ denote the norm of a nonzero shortest vector in L. Minkowski's convex body theorem implies that

$$\lambda_1(L) \leq \sqrt{n}(\det L)^{1/n}, \tag{2.2.11}$$

where n is the dimension of L. See for instance [27].

Turning back to our previous examples, we have $\lambda_1(\Lambda) = 1$ and $\lambda_1(\Gamma) = \sqrt{5}$.

Finding a short, nonzero vector in a lattice is a fundamental algorithmic problem with many uses in cryptography, optimization, and number theory. For surveys we refer to [20], [26], [47], and [42]. More generally, one may want to find a reduced basis consisting of short and nearly orthogonal vectors. Several different definitions of reduced basis have been suggested.

2.2.1 LLL Reduced Bases

The LLL basis reduction algorithm [35] was introduced in 1982 by Lenstra, Lenstra and Lovász; and has since been used in numerous applications in computational mathematics and computer science starting with factoring polynomials with rational coefficients and solving the integer linear programming problem in polynomial time in fixed dimensions. It computes a reduced basis of a lattice in polynomial time (for rational lattices). For simplicity, we use Schrijver's definition from [47].

We call $B = [b_1, \ldots, b_n]$ an *LLL reduced basis of* $\mathbb{L}(B)$, if

$$|\mu_{ij}| \leq 1/2 \quad (i = 2, \ldots, n;\ j = 1, \ldots, i-1), \text{ and} \tag{2.2.12}$$

$$\| b_i^* \|^2 \leq 2 \| b_{i+1}^* \|^2 \quad (i = 1, \ldots, n-1). \tag{2.2.13}$$

From (2.2.13) it immediately follows that

$$\| b_i^* \|^2 \leq 2^{j-i} \| b_j^* \|^2 \quad (1 \leq i \leq j \leq n). \tag{2.2.14}$$

As shown by Lenstra, Lenstra and Lovász, in an LLL reduced basis $B = [b_1, \ldots, b_n]$ of a lattice $L = \mathbb{L}(B)$, the norm of the first vector is bounded by a function of the norm of a nonzero shortest vector of L and also by a function of the determinant of L, namely

$$\| b_1 \| \leq 2^{(n-1)/4} (\det L)^{1/n}, \tag{2.2.15}$$

$$\| b_1 \| \leq 2^{(n-1)/2} \| d \| \text{ for any } d \in L \setminus \{0\}. \tag{2.2.16}$$

For an LLL-reduced basis $B = [b_1, \ldots, b_n]$ of a lattice L, they also show that

$$\| b_1 \| \cdots \| b_n \| \leq 2^{n(n-1)/4} \det L. \tag{2.2.17}$$

It is natural to ask, whether the three beautiful inequalities (2.2.15), (2.2.16), and (2.2.17) which we call as the fundamental inequalities can be generalized. In Chapter 3, we prove several inequalities generalizing and unifying the fundamental inequalities in an LLL reduced basis.

2.2.2 KZ Reduced Bases

Korkine-Zolotarev (KZ) reduced bases, which were described in [30] by Korkine and Zolotarev, and by Kannan in [27], have stronger reducedness properties than LLL reduced bases. For instance, the first vector in a KZ reduced basis is a shortest vector of the lattice. However, KZ reduced bases are computable in polynomial time only when n is fixed.

Given an m-by-n matrix $D = [d_1, \ldots, d_n]$ with rank r, $\mathrm{span}(D)$ (or $\mathrm{span}\{d_1, \ldots, d_n\}$) is defined as

$$\mathrm{span}(D) = \{Dx \,|\, x \in \mathbb{R}^n\}. \tag{2.2.18}$$

$\mathrm{span}(D)$ is an r-dimensional subspace of $\mathbb{R}^m$.

Let $L = \mathbb{L}(B)$ where $B = [b_1, \ldots, b_n]$ with n independent columns and for $j < i$ let $b_i(j)$ be the projection of b_i orthogonal to $\mathrm{span}\,\{b_1, b_2, \ldots, b_j\}$. Note that $b_i(i-1) = b_i^*$. Let

$$L(j) = \mathbb{L}([b_{j+1}(j), \ldots, b_n(j)])$$

be the projection of L orthogonal to $\mathrm{span}\,\{b_1, b_2, \ldots, b_j\}$. For convenience we define $b_i(0) = b_i$ and $L(0) = L$.

We say that a basis $B = [b_1, \ldots, b_n]$ is a KZ reduced basis of $\mathbb{L}(B)$ if

(1) $|\mu_{ij}| \leq 1/2$ $(i = 2, \ldots, n;\ j = 1, \ldots, i-1)$, and

(2) $b_i(i-1)$ is a shortest nonzero vector of $L(i-1)$ $(i = 1, \ldots, n)$.

Note that if $B = [b_1, \ldots, b_n]$ is a KZ reduced basis, then b_1 is a shortest nonzero vector in $\mathbb{L}(B)$.

For a KZ reduced basis $B = [b_1, \ldots, b_n]$ of a lattice $L = \mathbb{L}(B)$, from the definition of a KZ reduced basis and (2.2.11), it follows that

$$\| b_j^* \| \leq \sqrt{n-j+1} \prod_{i=j}^{n} \| b_i^* \|^{1/(n-j+1)}, \tag{2.2.19}$$

for any $j \in \{1, \ldots, n\}$. In particular, for $j = 1$ (2.2.19) becomes

$$\| b_1 \| \leq \sqrt{n} (\det L)^{1/n}. \tag{2.2.20}$$

It was also shown [32] that

$$\| b_i^* \| \geq \frac{\lambda_1(L)}{i^{(1+\log i)/2}} \tag{2.2.21}$$

holds for $i = 1, \ldots, n$.

Schnorr in [44] proposed several hierarchies of bases between LLL and KZ reduced ones: the semi block $2k$ bases among them are polynomial time computable when k is fixed; and both the "quality" of the basis, and the complexity of the reduction algorithm increases with k.

2.2.3 Hermite Normal Form

An integral m-by-n matrix with full row rank (i.e., with rank m) is in Hermite Normal Form (HNF) if it has the form $[B, 0]$, where B is a lower triangular, nonnegative matrix with each diagonal entry being the unique maximum in its row, and 0 is the matrix of all zeroes with appropriate size. Note that B is a nonsingular matrix. Any integral matrix A with full row rank can be brought into HNF by a series of elementary column operations [23] and this can be done in polynomial time as shown in [28]. In other words, there exists a polynomial time computable unimodular matrix U such that $AU = [B, 0]$ is in HNF. It is known that the HNF of A is unique and we write $\mathrm{HNF}(A) = [B, 0]$.

Let $\gcd(A)$ be the greatest common divisor of the m-by-m subdeterminants of A. Note that $\gcd(A)$ is invariant under elementary column operations. Therefore, we have that

$$\gcd(A) = \prod_{i=1}^{m} B_{ii}, \tag{2.2.22}$$

where $\mathrm{HNF}(A) = [B, 0]$.

Example 3. *Let*

$$A = \begin{pmatrix} 1 & 2 & 7 \\ 3 & 4 & 1 \end{pmatrix}.$$

The 2-by-2 subdeterminants of A are -2, -20, and -26. Therefore $\gcd(A) = 2$. We have

$$\mathrm{HNF}(A) \;= A \begin{pmatrix} -1 & 2 & 13 \\ 1 & -1 & -10 \\ 0 & 0 & 1 \end{pmatrix} \;= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \end{pmatrix}.$$

2.2.4 Null, Orthogonal, Dual and Complete Lattices

For an integral m by n matrix A, $m \leq n$, the null lattice of A is denoted by $\mathbb{N}(A)$ and is defined as

$$\mathbb{N}(A) = \{x \in \mathbb{Z}^n | Ax = 0\}. \tag{2.2.23}$$

For an integral lattice $L,$ its *orthogonal lattice* is defined as

$$L^{\perp} \;=\; \{\, y \in \mathbb{Z}^n \,|\, y^{\mathrm{T}} x = 0 \;\forall x \in L \,\}.$$

Note that $\mathbb{N}(A)$ is the same as $\mathbb{L}(A^T)^{\perp}$.

For a lattice L, the dual lattice L^* is

$$L^* = \{y \in \text{span } L \,|\, \langle x, y\rangle \in \mathbb{Z} \text{ for all } x \in L\}, \tag{2.2.24}$$

where $\text{span } L$ is $\text{span}(B)$ where B is a basis of L. It is known that $\det(L^*) = (\det(L))^{-1}$.

Let $B = [b_1, \ldots, b_n]$ be a basis of the lattice L. It is easy to see that $D = [d_1, \ldots, d_n] = B(B^{\mathrm{T}}B)^{-1}$ is a basis of L^*. We call $B^* = [d_n, \ldots, d_1]$ the dual basis (or the reciprocal basis) of B (note that the columns of D are reordered). One can check that B is the dual basis of B^* as well. If L is full dimensional, then ordering the columns of $B^{-\mathrm{T}}$ from highest index to smallest gives the dual basis of B.

Let B^* be the dual basis of B. And let $b_1^{**}, \ldots, b_n^{**}$ and $b_1^{\#}, \ldots, b_n^{\#}$ be the Gram-Schmidt orthogonalizations of columns of B^* and B, respectively. Then, it is easy to check that

$$\| b_i^{**} \| = 1/ \| b_{n-i+1}^{\#} \|. \tag{2.2.25}$$

A lattice $L \subseteq \mathbb{Z}^n$ is called *complete*, if

$$L = (\text{span } L) \cap \mathbb{Z}^n.$$

Each basis V of a complete lattice L can be completed to a unimodular matrix, i.e., there exists a matrix W such that $[V, W]$ is unimodular. Another useful characterization of complete lattices is that $\mathbb{L}(V)$ is complete if and only if $\text{HNF}(V^{\text{T}}) = [I, 0]$. For a proof see [43].

If $a \in \mathbb{Z}^n$, $\mathbb{L}(a^{\text{T}})$ is complete if and only if $\gcd(a_1, a_2, \ldots, a_n) = 1$ where gcd is the greatest common divisor. The following result relates the determinants of $\mathbb{N}(A)$ and $\mathbb{L}(A^{\text{T}})$ where A is an integral matrix.

Proposition 1. *Let A be an integral full row rank m-by-n matrix. Then*

$$\det \mathbb{N}(A) = \det \mathbb{L}(A^T)/\gcd(A). \tag{2.2.26}$$

Proof of Proposition 1 Let V be a basis for $\text{span}(A^{\text{T}}) \cap \mathbb{Z}^n$ and $\mathbb{L}(V) = \text{span}(A^{\text{T}}) \cap \mathbb{Z}^n$, which is an m dimensional complete lattice. We have that $A^{\text{T}} = VM$ for an invertible matrix M. Therefore $M^{\text{-T}}A = V^{\text{T}}$. Since $\mathbb{L}(V)$ is complete, $\text{HNF}(V^{\text{T}}) = [I, 0]$, which implies that $\text{HNF}(M^{\text{-T}}A) = [I, 0]$ as well. Since gcd is invariant under elementary column operations, $\gcd(M^{\text{-T}}A) = 1 = \det(M^{\text{-T}}) \gcd(A)$. This implies that $\det(M) = \gcd(A)$.

Now, we can write $\det \mathbb{L}(A^{\text{T}}) = \left[\det((VM)^{\text{T}}(VM))\right]^{1/2} = \det(M) \det \mathbb{L}(V)$. To finish the proof, we need to show that $\det \mathbb{L}(V) = \det \mathbb{N}(A)$.

Since $\mathbb{L}(V)$ is complete, V can be completed to a unimodular matrix, say U, i.e., there exists a matrix W such that $U = [V, W]$ is unimodular. Let $U^{-1} = [Y; Z]$, where the dimensions of Y and Z are the same as the dimensions of V^{T} and W^{T}, respectively. The rows of Z are a basis of $\mathbb{N}(A)$ and the projections of the columns of Y^{T} orthogonal to $\text{span}(Z^{\text{T}})$ are a basis of $\mathbb{L}(V)^*$. Furthermore $\det(U^{-1}) = (\det \mathbb{L}(V)^*)(\det \mathbb{N}(A)) = 1$, which implies that $\det \mathbb{L}(V)^* = 1/(\det \mathbb{N}(A))$ and therefore $\det \mathbb{L}(V) = \det \mathbb{N}(A)$ completing the proof. □

The following corollary of Proposition 1, has been used in some cryptographic applications. See for instance [43].

Corollary 1. $\det \mathbb{N}(A) \leq \det \mathbb{L}(A^T)$ *with equality holding if and only if* $\mathbb{L}(A^T)$ *is complete.*

The following lemma summarizes some basic results in lattice theory that we will use later on; for a complete proof, see for instance [40].

Lemma 1. *For an m by n integral matrix A with independent rows and $L = \mathbb{L}(A^T)$, the following are equivalent*

(1) *L is complete.*

(2) *The $\gcd$ of the determinants of the m by m submatrices of A is 1.*

(3) *$\mathrm{HNF}(A) = [I, 0]$.*

(4) *There exists a matrix V such that $[V; A]$ is unimodular.*

(5) *$\det L^{\perp} = \det L$.*

(6) *There is a unimodular matrix Z such that*

$$ZA^T = \begin{pmatrix} I_m \\ 0_{(n-m)\times m} \end{pmatrix}.$$

Furthermore, if Z is as in part (6), then the last $n - m$ rows of Z are a basis of $L^{\perp}$.

2.2.5 RKZ Reduced Bases

Hermite's constant C_i is defined as

$$C_i = sup\left\{ \left(\lambda_1(L)\right)^2 / \left(\det L\right)^{2/i} \mid L \text{ is a lattice of rank i} \right\}. \tag{2.2.27}$$

Its values are known exactly only for $i \leq 8$ and $i = 24$. It is known that [40]

$$C_i \leq 1 + i/4. \tag{2.2.28}$$

Sharper asymptotic bounds are known. In our analysis, for simplicity we will use 2.2.28, and for small values of i the Blichfeldt's upper bound [7]:

$$C_i \leq \frac{2}{\pi}\Gamma\left(\frac{i+4}{2}\right)^{2/i}, \tag{2.2.29}$$

where $\Gamma(.)$ is the gamma function.

A reciprocal Korkhine-Zolotarev (RKZ) basis is the dual (reciprocal) basis of a KZ reduced basis. Let $B = [b_1, \ldots, b_n]$ be an RKZ reduced basis of L and let $[b_1^*, \ldots, b_n^*]$ be the Gram-Schmidt orthogonalization of its columns. It can be shown that the Gram-Schmidt vectors of an RKZ reduced basis of a lattice are not too short. Combining 2.2.20 and 2.2.25 we get a lower bound on the norm of the last Gram-Schmidt vector in terms of the determinant of the lattice:

$$\| b_n^* \| \geq \frac{(\det L)^{1/n}}{\sqrt{n}}. \tag{2.2.30}$$

It was shown in [32] that

$$\| b_i^* \| \geq \frac{\lambda_1(L)}{C_i} \tag{2.2.31}$$

holds for $i = 1, \ldots, n$.

2.3 Integer Programming and Branch and Bound

Given a polyhedron Q, an integer programming (IP) feasibility problem is the problem of finding an integral vector in Q. In this thesis, we only consider feasibility problems. To solve an IP optimization problem, one needs to solve a sequence of feasibility problems using binary search.

Branch and bound, which we will abbreviate as B&B, was first studied by Land and Doig in [34] and is a classical method for IP feasibility (and optimization, more generally). It starts with Q as the sole subproblem. In a general step, one chooses a subproblem Q', an integral vector c, and creates new subproblems $Q' \cap \{x | cx = \gamma\}$, where γ ranges over all possible integer values that cx can take. This is repeated until all subproblems are found to be empty, or an integral point is found in one of them. Usually the vectors c are chosen to be the standard unit vectors e_i (i.e., we branch on the variable x_i). In this case, at each level of the B&B tree, one variable is fixed. This is called *ordinary B&B*. In a *generalized B&B* algorithm, the vectors c are allowed to be any integral vectors.

For a polyhedron Q and an integral vector c, the width and the integer width of Q along c are

$$\begin{aligned} \text{width}(c, Q) &= \max\{\, cx \,|\, x \in Q \,\} - \min\{\, cx \,|\, x \in Q \,\}, \text{ and} \\ \text{iwidth}(c, Q) &= \lfloor \max\{\, cx \,|\, x \in Q \,\} \rfloor - \lceil \min\{\, cx \,|\, x \in Q \,\} \rceil + 1. \end{aligned}$$

The integer width is the number of nodes generated by branch and bound when branching on the hyperplane cx; in particular, $\mathrm{iwidth}(e_i, Q)$ is the number of nodes generated when branching on x_i. It is easy to show that

$$\mathrm{iwidth}(c, Q) \leq \lfloor \mathrm{width}(c, Q) \rfloor + 1. \tag{2.3.32}$$

If the integer width along any integral vector is zero, then Q has no integral points. Given an integer program labeled by (P), and c an integral vector, we also write $\mathrm{width}(c, (\mathrm{P}))$, and $\mathrm{iwidth}(c, (\mathrm{P}))$ for the width and the integer width of the LP relaxation of (P) along c, respectively. Here, the LP relaxation of (P) is the underlying polyhedron describing the problem (P).

Given a lattice L with basis $B = [b_1, \ldots, b_n]$ and a polyhedron Q, the problem of determining whether Q contains a lattice point of L is a generalization of the IP feasibility problem. Let $b_1^*, \ldots, b_n^*$ be the Gram-Schmidt orthogonalization of $b_1, \ldots, b_n$. A lattice point $x \in L \cap Q$ is of the form

$$x = \sum_{j=1}^{n} \lambda_j b_j, \tag{2.3.33}$$

where λ_j are integers. Assume that Q is contained in a sphere of radius r. Then λ_n can take at most $(2r/ \parallel b_n^* \parallel) + 1$ different integer values. Similarly, having fixed $\lambda_{i+1}, \ldots, \lambda_n$; λ_i can take at most

$$2r/ \parallel b_i^* \parallel +1 \tag{2.3.34}$$

different integer values. Note that here the vectors b_i do not need to be integral vectors! This enumeration process is similar to branch and bound. In this enumeration process, the total number of nodes created on the level of b_i (i.e., on the $(n - i + 1)$st level) is at most

$$\prod_{j=i}^{n} \left(2r/ \parallel b_j^* \parallel +1\right). \tag{2.3.35}$$

The IP feasibility problem is NP-complete [9]. In 1983, H. W. Lenstra [36] devised a polynomial time algorithm for the IP feasibility problem in a fixed number of variables. Assume that the problem is described by the polyhedron Q. His algorithm, after some preprocessing steps, using the LLL basis reduction algorithm, either finds an integral point in Q, or finds a branching direction along which the polyhedron is thin, so that at most $O(2^{n^2})$ nodes are created, which is a constant when n is fixed.

The algorithm is repeated for each subproblem created until an integral point is found in any of the subproblems, which implies the integer feasibility of Q, or all the subproblems become the empty set, in which case the problem is integer infeasible. The upper bound on the number of B&B nodes created per level was later improved to $O(2^n)$ [5, 38].

Kannan [27] introduced a variant of Lenstra's algorithm which uses the KZ basis reduction algorithm instead. He showed that at the ith ($1 \leq i \leq n$) level of the branch and bound tree, there are at most $(2n)^{5i/2}$ nodes (where the value of i is determined by the algorithm), which implies a polynomial number of nodes $O(n^{5/2})$ per level ($O(n^{5/2})$ is not an upper bound on the number of nodes created for each subproblem at each level!). Note that his basis reduction algorithm does not run in polynomial time for varying n, but runs in polynomial time only when n is fixed.

In Section 2.3.1, we will briefly describe the algorithms of Lenstra and Kannan. In Section 2.3.2 we will introduce two experimentally very successful reformulation techniques for IP feasibility problem, namely the rangespace reformulation introduced in [31] for general IP feasibility problems and the nullspace reformulation introduced by Aardal, Hurkens and A. K. Lenstra in [2] for equality constrained IP feasibility problems; see also [1].

2.3.1 The Algorithms of Lenstra and Kannan for Integer Programming

In this section, we will briefly describe Lenstra's (a modified version) and Kannan's algorithms for integer programming. This exposition is mainly based on Kannan's survey on Algorithmic Geometry of Numbers [26].

Given an IP feasibility problem described by the polyhedron Q, these algorithms find an integral point in Q if there is any or prove that Q does not contain any integral point. Both algorithms run in polynomial time for fixed n.

We start with making Q a full dimensional polytope in $\mathbb{R}^n$ if it is not already; for the details see [36]. Lovász in [38] developed an algorithm to transform a polytope into a "rounded" one. He showed that there exists an invertible linear transformation ϕ such that $S_1 \subseteq P \subseteq S_2$ for two concentric spheres S_1 and S_2 where $P = \phi Q$ and $r_2/r_1 \leq (n+1)\sqrt{n}$ with r_i being the radius of S_i.

Therefore the problem of finding an integral point in Q is equivalent to the problem of finding a point of the lattice $L = \phi\mathbb{Z}^n$ in P. Let $B = [b_1, \ldots, b_n]$ be a reduced basis of L (in Lenstra's algorithm, we assume that B is LLL reduced; on the other hand in Kannan's algorithm, we assume that B is KZ

reduced). Let $\phi^{-1}B = [\phi^{-1}b_1, \ldots, \phi^{-1}b_n]$ and let $D = \left(\phi^{-1}B\right)^{-\mathrm{T}} = [d_1, \ldots, d_n]$. Both $\phi^{-1}B$ and D are bases of $\mathbb{Z}^n$ (i.e., they are unimodular), since D is a basis of the dual lattice of $\mathbb{L}(\phi^{-1}B) = \mathbb{Z}^n$.

Let j be the index such that $\| b_j^* \| \geq \| b_i^* \|$ for $i \in \{1, \ldots, n\}$. It is easy to show that if

$$r_1 \geq \sqrt{n} \, \| b_j^* \| \, /2, \tag{2.3.36}$$

then P contains a point of L, say ℓ which means that $\phi^{-1}\ell$ is an integral point in Q.

We modify Lenstra's algorithm, using ideas from [26]. Below are the main steps of both of the algorithms. We assume that we start with a polytope Q.

Algorithms

(1) Start with a polytope Q.

(2) Make it full dimensional and let n be the dimension of the full dimensional polytope Q.

(3) Round Q: find an invertible linear transformation ϕ such that $P = \phi Q$ is rounded. (Find r_1 and r_2 as well).

(4) Find a reduced basis $B = [b_1, \ldots, b_n]$ of $L = \phi\mathbb{Z}^n$ and let $b_1^*, \ldots, b_n^*$ be the Gram-Schmidt orthogonalization of $b_1, \ldots, b_n$.

(5) Let j be index such that $\| b_j^* \| \geq \| b_i^* \|$ for all $i \in \{1, \ldots, n\}$.

(6) If $r_1 \geq \sqrt{n} \, \| b_j^* \| \, /2$, then P contains a lattice point of L. STOP, Q is integer feasible.

(7) Otherwise, using the basis $D = [d_1, \ldots, d_n]$ of $\mathbb{Z}^n$, apply *backward B&B* for $n - j + 1$ levels (i.e., branch on $d_n x, \ldots, d_j x$ in the original space in this order). Then for each nonempty subproblem created if its dimension is 0 (i.e., if its a single integer point), STOP, Q is integer feasible; otherwise go to step 1.

(8) If the algorithm never stops and all subproblems become the empty set, then Q is integer infeasible.

Note that at each level of the branch and bound tree, the dimension of the subproblems is reduced at least by 1. Therefore the algorithm terminates in at most n levels.

Any integer point $y \in Q \cap \mathbb{Z}^n$ is of the form $\sum_{j=1}^{n} \left(\lambda_j(\phi^{-1}b_j)\right)$, where λ_j are integers, and any point $x \in Q$ is of the same form where λ_j are reals. Note that $d_j x = \lambda_j$, therefore fixing the value of $d_j x$ to an integer is the same as fixing the value of λ_j to the same integer.

In the original algorithm of Lenstra and in the follow-up papers [5, 18, 38], B&B is applied for one level, and all the steps are repeated for each subproblem created, i.e., the underlying polytope is rounded and basis reduction is used to find a new thin direction. In our version, these steps are repeated for the subproblems at the $(n-j+1)$st level. Therefore, the total time spent on rounding and basis reduction might be reduced. In [18] which is the only implementation of Lenstra's algorithm so far, it was stated that basis reduction is the bottleneck of the Lenstra's algorithm (i.e., most of the execution time was used by basis reduction).

Number of B&B Nodes in Lenstra's Algorithm

Note that, from (2.2.14), for any $\ell \in \{j, \ldots, n\}$ we have

$$\| b_\ell^* \| \geq \frac{\| b_j^* \|}{2^{(\ell-j)/2}}. \tag{2.3.37}$$

If at step 6, $r_1 \leq \sqrt{n} \| b_j^* \| /2$, then we have the following sequence of bounds on the number of B&B nodes created after branching on $d_n x, \ldots, d_j x$. Here the first expression follows from (2.3.35).

$$\begin{aligned}
\prod_{\ell=j}^{n} \left(\frac{2r_2}{\| b_\ell^* \|} + 1 \right) &\leq \prod_{\ell=j}^{n} \left(\frac{2(n+1)\sqrt{n} r_1}{\| b_\ell^* \|} + 1 \right) \\
&\leq \prod_{\ell=j}^{n} \left(\frac{(n+1)n \| b_j^* \|}{\| b_\ell^* \|} + 1 \right) \\
&\leq \prod_{\ell=j}^{n} \left((n+1)n 2^{(\ell-j)/2} + 1 \right) \\
&\leq \prod_{\ell=j}^{n} \left((n+1)n 2^{(\ell-j+1)/2} \right) \\
&\leq \left[(n+1)n 2^{(n-j+2)/4} \right]^{n-j+1},
\end{aligned}$$

where the first inequality follows from the fact that r_2 is not too large compared to r_1, the second from

$\| b_j^* \|$ being large and the third from (2.3.37).

Therefore, we get a factor of

$$(n+1)n2^{(n-j+2)/4} \tag{2.3.38}$$

B&B nodes per level in the B&B tree. We will not go into the details of the proof that this algorithm runs in polynomial time for fixed n.

Note that when j is large, i.e., close to n, the upper bound in (2.3.38) is small, therefore small number of B&B nodes are created per level. On the other hand, when j is smaller, the algorithm uses rounding and basis reduction less frequently than in the case with a larger j.

Number of B&B Nodes in Kannan's Algorithm

Assuming that $r_1 \leq \sqrt{n} \parallel b_j^* \parallel /2$, the total number of B&B nodes created after branching on $d_n x, \ldots, d_j x$ is bounded above by

$$\begin{aligned}
\prod_{i=j}^{n} \left(\frac{2r_2}{\| b_i^* \|} + 1 \right) &\leq \prod_{\ell=j}^{n} \left(\frac{2(n+1)\sqrt{n} r_1}{\| b_\ell^* \|} + 1 \right) \\
&\leq \prod_{\ell=j}^{n} \left(\frac{(n+1)n \, \| b_j^* \|}{\| b_\ell^* \|} + 1 \right) \\
&\leq \prod_{\ell=j}^{n} \left(((n+1)n+1) \frac{\| b_j^* \|}{\| b_\ell^* \|} \right) \\
&\leq ((n+1)n+1)^{n-j+1} \prod_{\ell=j}^{n} \frac{\| b_j^* \|}{\| b_\ell^* \|} \\
&\leq \left[\sqrt{n-j+1} \, ((n+1)n+1) \right]^{n-j+1},
\end{aligned}$$

where the last inequality follows from (2.2.19). Therefore, there is a factor of

$$(n^2+n+1)\sqrt{n-j+1} \tag{2.3.39}$$

B&B nodes per level. This improves the upper bound on the number of B&B nodes created in the algorithm of Lenstra.

In the next section, we describe two IP reformulation techniques which are used to improve the

performance of B&B. These reformulations also use basis reduction, but only once to preprocess the problem. Although, they do not result in polynomial time algorithms in fixed dimension in the worst case, they are very efficient in practice.

2.3.2 Two Integer Programming Reformulation Techniques

A simple and experimentally very successful technique for integer programming based on LLL reduction was proposed by Aardal, Hurkens and A. K. Lenstra in [2] for equality constrained IP problems. Consider the problem

$$\begin{array}{rcl} Ax & = & b \\ 0 \leq & x & \leq v \\ x & \in & \mathbb{Z}^n, \end{array} \tag{IP-EQ}$$

where A is an integral matrix with m independent rows.

The full-dimensional reformulation proposed in [2] is

$$\begin{array}{rcl} -x_b \leq & V\lambda & \leq v - x_b \\ \lambda & \in & \mathbb{Z}^{n-m}. \end{array} \tag{IP-EQ-N}$$

Here V and x_b satisfy

$$\{\, V\lambda \,|\, \lambda \in \mathbb{Z}^{n-m} \,\} \;=\; \mathbb{N}(A),\; x_b \in \mathbb{Z}^n,\; Ax_b = b,$$

the columns of V are reduced in the LLL-sense (one can also use other reduced bases, such as KZ or RKZ). For several classes of hard equality constrained IP problems – cf. [11] – the reformulation turned out to be much easier to solve by commercial solvers than the original problem.

In [31] an even simpler and experimentally just as effective reformulation method was introduced. It replaces

$$\begin{array}{rcl} b' \leq & Ax & \leq b \\ & x \in \mathbb{Z}^n & \end{array} \tag{IP}$$

with

$$\begin{array}{rcl} b' \leq & (AU)y & \leq b \\ & y \in \mathbb{Z}^n, & \end{array} \tag{IP-R}$$

where U is a unimodular matrix that makes the columns of AU reduced (in the LLL-, KZ-, or RKZ-sense). It applies the same way, even if some of the inequalities in the IP feasibility problem are actually equalities. In [31] the authors also introduced a simplified method to compute a reformulation which is essentially equivalent to (IP-EQ-N).

We call (IP-R) the *rangespace reformulation* of (IP); and (IP-EQ-N) the *nullspace reformulation* of (IP-EQ).

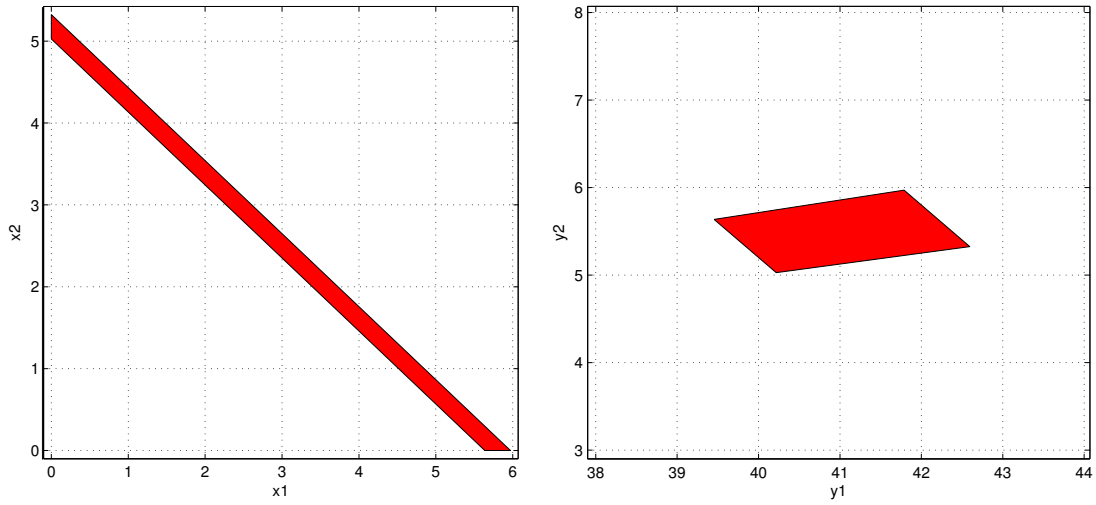

Figure 2.3: LP Relaxations of the Problem in Example 4 and its Rangespace Reformulation

Example 4. *Consider the following infeasible IP problem.*

$$\begin{aligned} 186 \leq \quad & 33x_1 + 37x_2 \quad && \leq 197 \\ 0 \leq \quad & x_1, x_2, && \leq 6 \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned} \tag{2.3.40}$$

Its LP relaxation is depicted on the first picture in Figure 2.3. Branching on x_i creates 6 branch and bound nodes $x_i = 0, \ldots, 5$ for $i = 1, 2$. On the other hand, branching on $x_1 + x_2$ proves the infeasibility of the problem at the root node; since the minimum and the maximum of $x_1 + x_2$ over the LP relaxation of 2.3.40 are 5.027 and 5.970, respectively.

When the rangespace reformulation is applied to 2.3.40 using LLL reduction, we get the following

problem:

$$\begin{aligned} 186 \leq \quad & 4y_1 + 5y_2 \quad && \leq 197 \\ 0 \leq \quad & -y_1 + 8y_2 \quad && \leq 6 \\ 0 \leq \quad & y_1 - 7y_2 \quad && \leq 6 \\ & y_1, y_2 \in \mathbb{Z}. \end{aligned} \tag{2.3.41}$$

Here

$$A = \begin{pmatrix} 33 & 37 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} -1 & 8 \\ 1 & -7 \end{pmatrix}, \quad AU = \begin{pmatrix} 4 & 5 \\ -1 & 8 \\ 1 & -7 \end{pmatrix}.$$

The LP relaxation of the reformulated problem 2.3.41 is depicted in the second picture in Figure 2.3. The second picture clearly shows that branching on y_2 immmediately proves the infeasibility of the problem. The minimum and the maximum of y_2 over the LP relaxation of 2.3.41 are again 5.027 and 5.970, respectively.

Let $ui_1, \ldots, ui_n$ be the rows of U^{-1}. It can be shown that branching on $y_n, \ldots, y_1$ in this order in (IP-R) is equivalent to branching on $ui_n x, \ldots, ui_1 x$ in this order in (IP) (i.e., the two B&B trees are isomorphic).

In Example 4, we have

$$U^{-1} = \begin{pmatrix} 7 & 8 \\ 1 & 1 \end{pmatrix},$$

therefore branching on y_2 in 2.3.41 is equivalent to branching on $x_1 + x_2$ in 2.3.40.

CHAPTER 3

Unifying LLL Inequalities

Several concepts of reducedness of a lattice basis are known. The most widely used one is LLL reducedness (for details, see Section 2.2.1), developed in the seminal paper [35] of Lenstra, Lenstra and Lovász. The quality of an LLL basis is expressed by three fundamental inequalities, (2.2.15)-(2.2.17). Surveys and textbook treatments of lattice basis reduction can be found in [20], [26], [47], and [42].

Improvements of the running time of the LLL algorithm were given, see for example Schnorr [45].

It is natural to ask, whether the three beautiful inequalities (2.2.15)-(2.2.17) can be unified and generalized: for instance, whether the product of the norms of the first few basis vectors can be bounded in terms of $\det L$, or if the norm of the first basis vector can be bounded by other parameters of L.

In this chapter we find unifying inequalities.

3.1 Generalizations of the Fundamental Inequalities in LLL Reduced Bases

Our Theorems 1 and 2 generalize inequalities (2.2.15) through (2.2.17).

Theorem 1. *Let $b_1, \dots, b_n \in \mathbb{R}^m$ be an LLL-reduced basis of the lattice L, and $d_1, \dots, d_k$ arbitrary linearly independent vectors in L. Then*

$$\| b_1 \| \leq 2^{(n-k)/2+(k-1)/4} (\det \mathbb{L}(d_1, \dots, d_k))^{1/k}, \tag{3.1.1}$$

$$\det \mathbb{L}(b_1, \dots, b_k) \leq 2^{k(n-k)/2} \det \mathbb{L}(d_1, \dots, d_k), \tag{3.1.2}$$

$$\det \mathbb{L}(b_1, \dots, b_k) \leq 2^{k(n-k)/4} (\det L)^{k/n}, \tag{3.1.3}$$

$$\| b_1 \| \cdots \| b_k \| \leq 2^{k(n-k)/2+k(k-1)/4} \det \mathbb{L}(d_1, \dots, d_k), \tag{3.1.4}$$

$$\| b_1 \| \cdots \| b_k \| \quad \leq \quad 2^{k(n-1)/4} (\det L)^{k/n}. \tag{3.1.5}$$

□

In the most general setting, we prove:

Theorem 2. *Let $b_1, \ldots, b_n \in \mathbb{R}^m$ be an LLL-reduced basis of the lattice L, $1 \leq k \leq j \leq n$, and $d_1, \ldots, d_j$ arbitrary linearly independent vectors in L. Then*

$$\det \mathbb{L}(b_1, \ldots, b_k) \quad \leq \quad 2^{k(n-j)/2 + k(j-k)/4} (\det \mathbb{L}(d_1, \ldots, d_j))^{k/j}, \tag{3.1.6}$$

$$\| b_1 \| \cdots \| b_k \| \quad \leq \quad 2^{k(n-j)/2 + k(j-1)/4} (\det \mathbb{L}(d_1, \ldots, d_j))^{k/j}. \tag{3.1.7}$$

By setting k and j to either 1 or n, from (3.1.6) we can recover the first two LLL inequalities, and from (3.1.7) we can recover all three.

The main tool is Lemma 3.1.8, which may be of independent interest. For $k = 1$ we can recover from it Lemma (5.3.11) in [20] (proven as part of Proposition (1.11) in [35]). First, note that if $b_1, \ldots, b_n$ are linearly independent vectors, then

$$\det \mathbb{L}(b_1, \ldots, b_n) \; = \; \det \mathbb{L}(b_1, \ldots, b_{n-1}) \, \| b' \|, \tag{3.1.8}$$

where b' is the projection of b_n on the orthogonal complement of the linear span of $b_1, \ldots, b_{n-1}$.

Lemma 2. *Let $d_1, \ldots, d_k$ be linearly independent vectors from the lattice L, and $b_1^*, \ldots, b_n^*$ the Gram Schmidt orthogonalization of an arbitary basis. Then*

$$\det \mathbb{L}(d_1, \ldots, d_k) \geq \min_{1 \leq i_1 < \cdots < i_k \leq n} \left\{ \| b_{i_1}^* \| \ldots \| b_{i_k}^* \| \right\}. \tag{3.1.9}$$

Proof of Lemma 2

We need the following

Claim There are elementary column operations performed on $d_1, \ldots, d_k$ that yield $\bar{d}_1, \ldots, \bar{d}_k$ with

$$\bar{d}_i = \sum_{j=1}^{t_i} \lambda_{ij} b_j \text{ for } i = 1, \ldots, k, \tag{3.1.10}$$

where $\lambda_{ij} \in \mathbb{Z}$, $\lambda_{i,t_i} \neq 0$, and

$$t_k > t_{k-1} > \cdots > t_1. \tag{3.1.11}$$

Proof of Claim Let us write

$$BV \;=\; [d_1, \ldots, d_k], \tag{3.1.12}$$

with V an integral matrix. Analogously to how the Hermite Normal Form of an integral matrix is computed, we can do elementary column operations on V to obtain $\bar{V}$ with

$$t_k := \max\{\, i \,|\, \bar{v}_{ik} \neq 0 \,\} \;>\; t_{k-1} := \max\{\, i \,|\, \bar{v}_{i,k-1} \neq 0 \,\} \;>\; \ldots \;>\; t_1 := \max\{\, i \,|\, \bar{v}_{i1} \neq 0 \,\}. \tag{3.1.13}$$

Performing the same elementary column operations on $d_1, \ldots, d_k$ yield $\bar{d}_1, \ldots, \bar{d}_k$ which satisfy

$$B\bar{V} \;=\; [\bar{d}_1, \ldots, \bar{d}_k], \tag{3.1.14}$$

so they satisfy (3.1.10).

End of proof of Claim

Obviously

$$\det\ \mathbb{L}(\bar{d}_1, \ldots, \bar{d}_k) \;=\; \det\ \mathbb{L}(d_1, \ldots, d_k). \tag{3.1.15}$$

Substituting from (2.2.9) for b_i we can rewrite (3.1.10) as

$$\bar{d}_i = \sum_{j=1}^{t_i} \lambda^*_{ij} b^*_j \text{ for } i = 1, \ldots, k, \tag{3.1.16}$$

where the λ^*_{ij} are now reals, but $\lambda^*_{i,t_i} = \lambda_{i,t_i}$ nonzero integers.

For all i we have

$$\text{span}\,\{\, \bar{d}_1, \ldots, \bar{d}_{i-1} \,\} \;\subseteq\; \text{span}\{\, b^*_1, \ldots, b^*_{t_{i-1}} \,\}. \tag{3.1.17}$$

Therefore

$$\|\,\text{Proj}\,\{\, \bar{d}_i \,|\, \{\, \bar{d}_1, \ldots, \bar{d}_{i-1} \,\}^{\perp} \,\}\,\| \geq \|\,\text{Proj}\,\{\, \bar{d}_i \,|\, \{\, b^*_1, \ldots, b^*_{t_{i-1}} \,\}^{\perp} \,\}\,\| \geq \|\,\lambda_{i,t_i} b^*_{t_i}\,\| \geq \|\,b^*_{t_i}\,\| \tag{3.1.18}$$

holds, with the second inequality coming from (3.1.11). Here $\mathrm{Proj}\ \{\, \bar{d}_i \,|\, \{\, \bar{d}_1, \ldots, \bar{d}_{i-1} \,\}^{\perp} \,\}$ is the projection of $\bar{d}_i$ orthogonal to $\mathrm{span}\{\bar{d}_1, \ldots, \bar{d}_{i-1}\}$. So applying (3.1.8) repeatedly we get

$$\begin{aligned} \det\ \mathbb{L}(\bar{d}_1, \ldots, \bar{d}_k) &\geq \det \mathbb{L}(\bar{d}_1, \ldots, \bar{d}_{k-1})\, \| b^*_{t_k} \| \\ &\ldots \\ &\geq \| b^*_{t_1} \| \| b^*_{t_2} \| \ldots \| b^*_{t_k} \|, \end{aligned} \tag{3.1.19}$$

which together with (3.1.15) completes the proof. □

3.2 Proofs of Theorem 1 and Theorem 2

The plan of the proof is as follows: we first prove (3.1.1) through (3.1.3) in Theorem 1. Then we prove Theorem 2. Finally, (3.1.4) follows as a special case of (3.1.7) with $j = k$; and (3.1.5) as a special case of (3.1.7) with $j = n$.

Proof of (3.1.1) and (3.1.2) Lemma 2 implies

$$\det\ \mathbb{L}(d_1, \ldots, d_k) \geq \| b^*_{t_1} \| \| b^*_{t_2} \| \ldots \| b^*_{t_k} \| \tag{3.2.20}$$

for some $t_1, \ldots, t_k \in \{1, \ldots, n\}$ distinct indices. Clearly

$$t_1 + \cdots + t_k \leq kn - k(k-1)/2 \tag{3.2.21}$$

holds. Applying first (2.2.14), then (3.2.21) yields

$$\begin{aligned} (\det\ \mathbb{L}(d_1, \ldots, d_k))^2 &\geq \| b^*_1 \|^2\, 2^{(1-t_1)} \ldots \| b^*_1 \|^2\, 2^{(1-t_k)} \\ &= \| b^*_1 \|^{2k}\, 2^{k-(t_1+\cdots+t_k)} \\ &\geq \| b_1 \|^{2k}\, 2^{k(k+1)/2-kn}, \end{aligned} \tag{3.2.22}$$

which is equivalent to (3.1.1). Similarly,

$$\begin{aligned}(\det\ \mathbb{L}(d_1,\ldots,d_k))^2 &\geq \|b_1^*\|^2\ 2^{(1-t_1)}\ \|b_2^*\|^2\ 2^{(2-t_2)}\ldots\|b_k^*\|^2\ 2^{(k-t_k)}\\ &= \|b_1^*\|^2\ldots\|b_k^*\|^2\ 2^{(1+\cdots+k)-(t_1+\cdots+t_k)} \qquad (3.2.23)\\ &\geq \|b_1^*\|^2\ldots\|b_k^*\|^2\ 2^{k(k-n)},\end{aligned}$$

which is equivalent to (3.1.2).

□

Proof of (3.1.3) The proof is by induction. Let us write $D_k = (\det \mathbb{L}(b_1,\ldots,b_k))^2$. For $k = n-1$, multiplying the inequalities

$$\|b_i^*\|^2 \leq 2^{n-i}\ \|b_n^*\|^2\ \ (\ i = 1,\ldots,n-1) \qquad (3.2.24)$$

gives

$$\begin{aligned}D_{n-1} &\leq 2^{n(n-1)/2}(\|b_n^*\|^2)^{n-1} \qquad (3.2.25)\\ &= 2^{n(n-1)/2}\left(\frac{D_n}{D_{n-1}}\right)^{n-1}, \qquad (3.2.26)\end{aligned}$$

and after simplifying, we get

$$D_{n-1} \leq 2^{(n-1)/2}(D_n)^{1-1/n}. \qquad (3.2.27)$$

Suppose that (3.1.3) is true for $k \leq n-1$; we will prove it for $k-1$. Since $b_1,\ldots,b_k$ forms an LLL-reduced basis of $\mathbb{L}(b_1,\ldots,b_k)$ we can replace n by k in (3.2.27) to get

$$D_{k-1} \leq 2^{(k-1)/2}(D_k)^{(k-1)/k}. \qquad (3.2.28)$$

By the induction hypothesis,

$$D_k \leq 2^{k(n-k)/2}(D_n)^{k/n}, \qquad (3.2.29)$$

from which we obtain

$$(D_k)^{(k-1)/k} \leq 2^{(k-1)(n-k)/2}(D_n)^{(k-1)/n}. \tag{3.2.30}$$

Using the upper bound on $(D_k)^{(k-1)/k}$ from (3.2.30) in (3.2.28) yields

$$D_{k-1} \leq 2^{(k-1)/2}2^{(k-1)(n-k)/2}(D_n)^{(k-1)/k} \tag{3.2.31}$$

$$= 2^{(k-1)(n-(k-1))/2}(D_n)^{(k-1)/n}, \tag{3.2.32}$$

as required.

□

Proof of Theorem 2 From (3.1.3) and (3.1.2) we have

$$\det \mathbb{L}(b_1, \ldots, b_k) \leq 2^{k(j-k)/4}(\det \mathbb{L}(b_1, \ldots, b_j))^{k/j}, \tag{3.2.33}$$

$$\det \mathbb{L}(b_1, \ldots, b_j) \leq 2^{j(n-j)/2} \det \mathbb{L}(d_1, \ldots, d_j). \tag{3.2.34}$$

Raising (3.2.34) to the power of k/j gives

$$(\det \mathbb{L}(b_1, \ldots, b_j))^{k/j} \leq 2^{k(n-j)/2} \det(\mathbb{L}(d_1, \ldots, d_j))^{k/j}, \tag{3.2.35}$$

and plugging (3.2.35) into (3.2.33) proves (3.1.6).

It is shown in [35] that

$$\| b_i \|^2 \leq 2^{i-1} \| b_i^* \|^2 \text{ for } i = 1, \ldots, n. \tag{3.2.36}$$

Multiplying these inequalities for $i = 1, \ldots, k$ yields

$$\| b_1 \| \cdots \| b_k \| \leq 2^{k(n-1)/4} \det \mathbb{L}(b_1, \ldots, b_k), \tag{3.2.37}$$

and using (3.2.37) with (3.1.6) yields (3.1.7).

□

3.3 Discussion

The kth successive minimum of L is the smallest real number t, such that there are k linearly independent vectors in L with length bounded by t. It is denoted by $\lambda_k(L)$. With the same setup as for (2.2.15)-(2.2.17) it is shown in [35] that

$$\| b_i \| \quad \leq \quad 2^{n-1}\lambda_i(L) \text{ for } i = 1, \ldots, n. \tag{3.3.38}$$

For KZ and block KZ bases similar results were shown in [32] and [46], respectively.

The successive minimum results (3.3.38) give a more global view of the lattice and the reduced basis, than (2.2.15) through (2.2.17). Our Theorem 2 is similar in this respect, but it seems to be independent of (3.3.38). Of course, multiplying the latter for $i = 1, \ldots, k$ gives an upper bound on $\| b_1 \| \cdots \| b_k \|$, but in different terms.

The quantities $\det \mathbb{L}(b_1, \ldots, b_k)$ and $\| b_1 \| \ldots \| b_k \|$ are also connected by

$$\det \mathbb{L}(b_1, \ldots, b_k) \quad = \quad \| b_1 \| \ldots \| b_k \| \sin\theta_2 \ldots \sin\theta_k, \tag{3.3.39}$$

where θ_i is the angle of b_i with the subspace spanned by $b_1, \ldots, b_{i-1}$. In [5] Babai showed that the sine of the angle of *any* basis vector with the subspace spanned by the other basis vectors in a d-dimensional lattice is at least $(\sqrt{2}/3)^d$. One could combine the lower bounds on $\sin\theta_i$ with the upper bounds on $\det \mathbb{L}(b_1, \ldots, b_k)$ to find an upper bound on $\| b_1 \| \ldots \| b_k \|$. However, the result would be weaker than (3.1.4) and (3.1.5).

CHAPTER 4

Branching on a Near Parallel Integral Vector in a Knapsack Problem

The knapsack problem is one of the most studied problems in combinatorial optimization and has many real life applications. In this chapter, we show that in a knapsack feasibility problem an integral vector p which is near parallel to the constraint vector a gives a branching direction with small integer width. This result is used to analyze the rangespace and the nullspace reformulations of the knapsack problem. We prove an upper bound on the integer width along the last variable in the reformulated problems, which becomes 1 when the density is sufficiently small, i.e., when $\| a \|$ is sufficiently large (for a formal definition of the density of a knapsack problem, see Section 5.2). The proof ingredients may be of independent interest. We extract, from the transformation matrices, an integral vector which is near parallel to the constraint vector a. The near parallel vector is a good branching direction in the original problem and a transference result shows that the last variable is a good branching direction in the reformulations.

4.1 Reformulations of the Knapsack Problem

The reformulation methods explained in Section 2.3.2 are very easy to describe (as opposed to say Lenstra's or Kannan's method), but seem difficult to analyze. The only analyses are for knapsack problems, with the weight vector having a given "decomposable" structure, i.e., $a = \lambda p + r,$ with $p, r,$ and λ integral, and λ large with respect to $\| p \|$ and $\| r \|$ – see [3, 31].

The goal of this chapter is to analyze these reformulations on the knapsack feasibility problem

$$\begin{array}{rcl} \beta_1 & \leq ax \leq & \beta_2 \\ 0 & \leq x \leq & v \\ & x \in \mathbb{Z}^n, & \end{array} \tag{KP}$$

where a is a positive, integral row vector, β_1 and β_2 are integers, without assuming any structure on the constraint vector *a priori*. We will assume only that $\| a \|$ is large – in fact, a key point will be that the large norm *implies* a decomposable structure, and this structure is automatically "discovered" by the reformulations.

The rangespace reformulation of (KP) is

$$\begin{array}{rcl} \beta_1 & \leq aUy \leq & \beta_2 \\ 0 & \leq Uy \leq & v \\ & y \in \mathbb{Z}^n, & \end{array} \tag{KP-R}$$

where U is a unimodular matrix that makes the columns of $\begin{pmatrix} a \\ I \end{pmatrix} U$ reduced in the LLL-sense (we do not analyze it with KZ reduction). The nullspace reformulation is

$$\begin{array}{rcl} -x_\beta \leq & V\lambda & \leq v - x_\beta \\ \lambda & \in & \mathbb{Z}^{n-m}, \end{array} \tag{KP-N}$$

where $x_\beta \in \mathbb{Z}^n$, $ax_\beta = \beta$, $\{ V\lambda \,|\, \lambda \in \mathbb{Z}^{n-m} \} = \mathbb{N}(a)$ and the columns of V are reduced in the LLL-sense.

Throughtout the chapter, we will assume $0 \leq \beta_1 \leq \beta_2 \leq av$, and that the gcd of the components of a is 1. For a rational vector b we denote by $\mathrm{round}(b)$ the vector obtained by rounding the components of b.

For an n-vector a, we will write

$$\begin{array}{rcl} f(a) & = & 2^{n/4} / \| a \|^{1/n}, \\ g(a) & = & 2^{(n-2)/4} / \| a \|^{1/(n-1)} . \end{array} \tag{4.1.1}$$

4.2 Main Results

In this section, we will review the main results of the chapter, give some examples, explanations, and some proofs that show their connection.

The main purpose of this section is an analysis of the reformulation methods. This is done in Theorem 3, which proves an upper bound on the number of B&B nodes, when branching on the last variable in the reformulations.

Theorems 4 and 5 show that an integral vector p, which is "near parallel" to a can be extracted from the transformation matrices of the reformulations. The notion of near parallelness that we use is stronger than just requiring $\sin(a, p)$ to be small. The relationship of the two parallelness concepts is clarified in Proposition 2.

Theorem 6 proves an upper bound on $\mathrm{iwidth}(p, (\mathrm{KP}))$, where p is an integral vector. A novelty of the bound is that it does not depend on β_1 and β_2, only on their difference. We show through examples that this bound is quite useful when p is a near parallel vector found according to Theorems 4 and 5.

In the end, a transference result between branching directions in the original, and reformulated problems completes the proof of Theorem 3.

Theorem 3. *Suppose* $\| a \| \geq 2^{(n/2+1)n}$. *Then*

(1) $\mathrm{iwidth}(e_n, (\mathrm{KP\text{-}R})) \leq \lfloor f(a)(2 \| v \| + (\beta_2 - \beta_1)) \rfloor + 1.$

(2) $\mathrm{iwidth}(e_{n-1}, (\mathrm{KP\text{-}N})) \leq \lfloor 2g(a) \| v \| \rfloor + 1.$

Given a and p integral vectors, we will need the notion of their near parallelness. The obvious thing would be to require that $|\sin(a, p)|$ is small. Instead, we will write a decomposition

$$a = \lambda p + r, \text{ with } \lambda \in \mathbb{Q},\ r \in \mathbb{Q}^n,\ r \perp p, \tag{DECOMP}$$

and ask for $\| r \| / \lambda$ to be small. The following proposition clarifies the connection of the two near parallelness concepts and shows two useful consequences of the latter one.

Proposition 2. *Suppose that* $a, p \in \mathbb{Z}^n$, *and* r *and* λ *are defined to satisfy (DECOMP). Assume w.l.o.g.* $\lambda > 0$. *Then*

(1) $\sin(a, p) \leq \| r \| / \lambda.$

(2) For any M there exists a, p with $\|a\| \geq M$ such that the inequality in (1) is strict.

(3) Denote by p_i and a_i the ith component of p and a. If $\|r\|/\lambda < 1$, and $p_i \neq 0$, then the signs of p_i and a_i agree. Also, if $\|r\|/\lambda < 1/2$, then $\lfloor a_i/\lambda \rceil = p_i$.

Proof Statement (1) follows from

$$\sin(a,p) \quad = \quad \|r\| \,/\, \|a\| \leq \|r\| \,/\, \|\lambda p\| \leq \|r\| \,/\lambda, \tag{4.2.2}$$

where in the last inequality we used the integrality of p.

To see (2), consider the family of a and p vectors

$$\begin{aligned} a &= \begin{pmatrix} m^2+1, & m^2 \end{pmatrix}, \\ p &= \begin{pmatrix} m+1, & m \end{pmatrix} \end{aligned} \tag{4.2.3}$$

with m an integer. Letting λ and r be defined as in the statement of the proposition, a straightforward computation (or experimentation) shows that as $m \to \infty$

$$\begin{aligned} \sin(a,p) &\to 0, \\ \|r\|/\lambda &\to 1/\sqrt{2}. \end{aligned}$$

Statement (3) is straightforward from

$$a_i/\lambda \quad = \quad p_i + r_i/\lambda. \tag{4.2.4}$$

□

The next two theorems show how the near parallel vectors can be found from the transformation matrices of the reformulations.

Theorem 4. *Suppose $\|a\| \geq 2^{(n/2+1)n}$. Let U be a unimodular matrix such that the columns of*

$$\begin{pmatrix} a \\ I \end{pmatrix} U$$

are LLL-reduced and p the last row of U^{-1}. Define r and λ to satisfy (DECOMP), and assume w.l.o.g. $\lambda > 0$.

Then

(1) $\| p \| \, (1+ \| r \|^2)^{1/2} \leq \| a \| \, f(a)$*;*

(2) $\lambda \geq 1/f(a)$*;*

(3) $\| r \| \, / \lambda \leq 2 f(a)$.

□

Theorem 5. *Suppose* $\| a \| \geq 2^{(n/2+1)n}$. *Let V be a matrix whose columns are an LLL-reduced basis of $\mathbb{N}(a)$, b an integral column vector with $ab = 1$, and p the $(n-1)st$ row of $(V, b)^{-1}$. Define r and λ to satisfy (DECOMP), and assume w.l.o.g. $\lambda > 0$.*

Then $r \neq 0$, *and*

(1) $\| p \| \| r \| \leq \| a \| \, g(a)$*;*

(2) $\| r \| \, / \lambda \leq 2 g(a)$.

□

It is important to note that p is integral, but λ and r may not be. Also, the measure of parallelness to a, i.e., the upper bound on $\| r \| \, / \lambda$ is quite similar for the p vectors found in Theorems 4 and 5, but their length can be quite different. When $\| a \|$ is large, the p vector in Theorem 4 is guaranteed to be much shorter than a by $\lambda \geq 1/f(a)$. On the other hand, the p vector from Theorem 5 may be much *longer* than a : the upper bound on $\| p \| \| r \|$ does not guarantee any bound on $\| p \|$, since r can be fractional.

The following example illustrates this:

Example 5. *Consider the vector*

$$a = \left(3488, \quad 451, \quad 1231, \quad 6415, \quad 2191 \right). \tag{4.2.5}$$

We computed $p_1,\ r_1,\ \lambda_1$ *according to Theorem 4:*

$$\begin{aligned} p_1 &= \Big(62,\quad 8,\quad 22,\quad 114,\quad 39\Big), \\ r_1 &= \Big(0.2582,\quad 0.9688,\quad -6.5858,\quad 2.0554,\quad -2.9021\Big), \\ \lambda_1 &= 56.2539, \\ \|\, r_1 \,\| \,/\lambda_1 &= 0.1342. \end{aligned} \tag{4.2.6}$$

We also computed $p_2,\ r_2,\ \lambda_2$ *according to Theorem 5; note* $\|\, p_2 \,\| > \|\, a \,\|$*:*

$$\begin{aligned} p_2 &= \Big(12204,\quad 1578,\quad 4307,\quad 22445,\quad 7666\Big) \\ r_2 &= \Big(-0.0165,\quad -0.0071,\quad 0.0194,\quad 0.0105,\quad -0.0140\Big) \\ \lambda_2 &= 0.2858 \\ \|\, r_2 \,\| \,/\lambda_2 &= 0.1110. \end{aligned} \tag{4.2.7}$$

□

Theorem 6 below gives an upper bound on the number of B&B nodes when branching on a hyperplane in (KP).

Theorem 6. *Suppose that* $a = \lambda p + r,$ *with* $p \geq 0$*. Then*

$$\text{iwidth}(p, (KP)) \ \leq\ \left\lfloor \frac{\|\, r \,\| \|\, v \,\|}{\lambda} + \frac{\beta_2 - \beta_1}{\lambda} \right\rfloor + 1. \tag{4.2.8}$$

This bound is quite strong for near parallel vectors computed from Theorems 4 and 5. For instance, let $a,\ p_1,\ r_1, \lambda_1$ be as in Example 5. If $\beta_1 = \beta_2$ in a knapsack problem with weight vector a and each x_i is bounded between 0 and $3,$ then Theorem 6 implies that the integer width is at most one. At the other extreme, it also implies that the integer width is at most one, if each x_i is bounded between 0 and 1, and $\beta_2 - \beta_1 \leq 39$. However, this bound does not seem as useful, when p is a "simple" vector, say a unit vector. Note that the assumption that $p \geq 0$ is only to simplify the proofs.

We now complete the proof of Theorem 3, based on a simple transference result between branching directions, taken from [31].

Proof of Theorem 3

Let us denote by $Q,\ Q_R,$ and Q_N the feasible sets of the LP relaxations of (KP), of (KP-R), and of (KP-N), respectively.

First, let U and p be the transformation matrix, and the near parallel vector from Theorem 4. It was shown in [31] that $\mathrm{iwidth}(p, Q) = \mathrm{iwidth}(pU, Q_R)$. But $pU = \pm e_n,$ so

$$\mathrm{iwidth}(p, Q) \quad = \quad \mathrm{iwidth}(e_n, Q_R). \tag{4.2.9}$$

On the other hand,

$$\begin{aligned} \mathrm{iwidth}(p, Q) &\leq \left\lfloor \frac{\|r\|\|v\|}{\lambda} + \frac{\beta_2 - \beta_1}{\lambda} \right\rfloor + 1 \\ &\leq \lfloor f(a)(2\,\|v\| + (\beta_2 - \beta_1)) \rfloor + 1 \end{aligned} \tag{4.2.10}$$

with the first inequality coming from Theorem 6 and the second from using the bounds on $1/\lambda$ and $\|r\|\ /\lambda$ from Theorem 4. Combining (4.2.9) and (4.2.10) yields (1) in Theorem 3.

Now let V and p be the transformation matrix, and the near parallel vector from Theorem 5. It was shown in [31] that $\mathrm{iwidth}(p, Q) = \mathrm{iwidth}(pV, Q_N)$. But $pV = \pm e_{n-1},$ so

$$\mathrm{iwidth}(e_{n-1}, Q_N) \quad = \quad \mathrm{iwidth}(p, Q). \tag{4.2.11}$$

On the other hand,

$$\begin{aligned} \mathrm{iwidth}(p, Q) &\leq \left\lfloor \frac{\|r\|\|v\|}{\lambda} \right\rfloor + 1 \\ &\leq \lfloor g(a)(2\,\|v\|) \rfloor + 1. \end{aligned} \tag{4.2.12}$$

with the first inequality coming from Theorem 6 and the second from using the bound on $\|r\|\ /\lambda$ in Theorem 5. Combining (4.2.11) and (4.2.12) yields (2) in Theorem 3.

4.3 Near Parallel Vectors: Intuition and Proofs of Theorems 4 and 5

Proof of Theorem 4 First note that the lower bound on $\|a\|$ implies

$$f(a) \leq \sqrt{3}/2. \tag{4.3.13}$$

Let L_ℓ be the lattice generated by the first ℓ columns of $\begin{pmatrix} a \\ I \end{pmatrix} U$, and

$$Z = \begin{pmatrix} 0 & U^{-1} \\ 1 & -a \end{pmatrix}.$$

Clearly, Z is unimodular and

$$Z \begin{pmatrix} aU \\ U \end{pmatrix} = \begin{pmatrix} I_n \\ 0_{1\times n} \end{pmatrix}. \tag{4.3.14}$$

So Lemma 1 implies that L_ℓ is complete and the last $n+1-\ell$ rows of Z generate $L_\ell^\perp$. The last row of Z is $(1, -a)$ and the next-to-last is $(0, p)$, so we get

$$\begin{aligned} \det L_n &= \det L_n^\perp = (\|a\|^2 + 1)^{1/2}, \\ \det L_{n-1} &= \det L_{n-1}^\perp = \|p\| \, (1 + \|r\|^2)^{1/2}. \end{aligned} \tag{4.3.15}$$

(3.1.3) of Theorem 1 implies

$$\det L_{n-1} \leq 2^{(n-1)/4} (\det L_n)^{1-1/n}. \tag{4.3.16}$$

Substituting into (4.3.16) from (4.3.15) gives

$$\begin{aligned} \|p\| \, (1 + \|r\|^2)^{1/2} &\leq 2^{(n-1)/4} (\sqrt{\|a\|^2 + 1})^{1-1/n} \\ &\leq 2^{n/4} \, \|a\|^{1-1/n} \\ &= \|a\| \, f(a), \end{aligned} \tag{4.3.17}$$

with the second inequality coming the lower bound on $\|a\|$. This shows (1).

Proof of (2) From (1) we directly obtain

$$\begin{aligned} \frac{f(a)^2 \,\|a\|^2 - \|r\|^2}{\|p\|^2} &\geq \frac{f(a)^2 \,\|a\|^2 - \|p\|^2\|r\|^2}{\|p\|^2} \\ &\geq 1 \\ &= \frac{f(a)^2 \,\|a\|^2}{f(a)^2 \,\|a\|^2}, \end{aligned} \tag{4.3.18}$$

where in the first inequality we used $\|p\| \geq 1$. Now note

$$\|p\|^2 \leq f(a)^2 \,\|a\|^2,$$

i.e., the the denominator of the first expression in (4.3.18) is not larger than the denominator of the last expression. So if we replace $f(a)^2$ by 1 in the *numerator* of both, the inequality will remain valid. The result is

$$\frac{\|a\|^2 - \|r\|^2}{\|p\|^2} \geq \frac{1}{f(a)^2}, \tag{4.3.19}$$

which is the square of the required inequality.

Proof of (3) We have

$$\begin{aligned} \frac{\|r\|^2}{\lambda^2} &\leq \frac{\|p\|^2\|r\|^2}{\|\lambda p\|^2} \\ &= \frac{\|p\|^2\|r\|^2}{\|a\|^2 - \|r\|^2} \\ &\leq \frac{f(a)^2 \,\|a\|^2}{\|a\|^2 - \|r\|^2} \\ &\leq \frac{f(a)^2 \,\|a\|^2}{\|a\|^2 - f(a)^2 \,\|a\|^2} \\ &= \frac{f(a)^2}{1 - f(a)^2} \\ &\leq 4f(a)^2, \end{aligned} \tag{4.3.20}$$

where the last inequality comes from (4.3.13) and the others are straightforward.

□

Proof of Theorem 5 The lower bound on $\| a \|$ implies

$$g(a) \leq \sqrt{3}/2. \tag{4.3.21}$$

Let L_ℓ be the lattice generated by the first ℓ columns of V. We have

$$(V, b)^{-1} V \quad = \quad \begin{pmatrix} I_{n-1} \\ 0 \end{pmatrix}. \tag{4.3.22}$$

So Lemma 1 implies that L_ℓ is complete and the last $n-\ell$ rows of $(V, b)^{-1}$ generate $L_\ell^\perp$. It is elementary to see that the last row of $(V, b)^{-1}$ is a and by definition the next-to-last row is p, and these rows are independent, so $r \neq 0$. Also,

$$\begin{aligned} \det L_{n-1} &= \det L_{n-1}^\perp = \| a \|, \\ \det L_{n-2} &= \det L_{n-2}^\perp = \| p \| \| r \| . \end{aligned} \tag{4.3.23}$$

(3.1.3) of Theorem 1 with $n-1$ in place of n and $n-2$ in place of k implies

$$\det\ L_{n-2} \quad \leq \quad 2^{(n-2)/4} (\det L_{n-1})^{1-1/(n-1)}. \tag{4.3.24}$$

Substituting into (4.3.24) from (4.3.23) gives

$$\begin{aligned} \| p \| \| r \| &\leq 2^{(n-2)/4} \| a \|^{1-1/(n-1)} \\ &= \| a \| \, g(a), \end{aligned} \tag{4.3.25}$$

as required.

Proof of (2) It is enough to note that in proof of (3) in Theorem 4 we only used the inequality $\| p \|^2 \| r \|^2 \leq f(a)^2 \| a \|^2$. So the exact same argument works here as well with $g(a)$ instead of $f(a)$, and invoking (4.3.21) as well.

□

4.4 Branching on a Near Parallel Vector: Proof of Theorem 6

This proof is somewhat technical, so we state and prove some intermediate claims, to improve readability. Let us fix a, p, β_1, β_2, and v. For a row-vector w and an integer ℓ we write

$$\begin{aligned} \max(w,\ell) &= \max\{\, wx \,|\, px \le \ell,\ 0 \le x \le v \,\} \\ \min(w,\ell) &= \min\{\, wx \,|\, px \ge \ell,\ 0 \le x \le v \,\}. \end{aligned} \tag{4.4.26}$$

The dependence on p, on v and on the sense of the constraint (i.e., $\le$ or $\ge$) is not shown by this notation; however, we always use $px \le \ell$ with "max" and $px \ge \ell$ with "min", and p and v are fixed. Note that as a is a row-vector and v a column-vector, av is their inner product, and the meaning of pv is similar.

Claim 1. *Suppose that ℓ_1 and ℓ_2 are integers in $\{0,\ldots,pv\}$. Then*

$$\min(a,\ell_2) - \max(a,\ell_1) \;\ge\; -\|r\|\|v\| + \lambda(\ell_2 - \ell_1). \tag{4.4.27}$$

Proof The decomposition of a shows

$$\begin{aligned} \max(a,\ell_1) &\le \max(r,\ell_1) + \lambda\ell_1, \text{ and} \\ \min(a,\ell_2) &\ge \min(r,\ell_2) + \lambda\ell_2. \end{aligned} \tag{4.4.28}$$

So we get the following chain of inequalities, with ensuing explanation:

$$\begin{aligned} \min(a,\ell_2) - \max(a,\ell_1) &\ge \min(r,\ell_2) - \max(r,\ell_1) + \lambda(\ell_2 - \ell_1) \\ &\ge rx_2 - rx_1 + \lambda(\ell_2 - \ell_1) \\ &= r(x_2 - x_1) + \lambda(\ell_2 - \ell_1) \\ &\ge -\|r\|\|v\| + \lambda(\ell_2 - \ell_1). \end{aligned} \tag{4.4.29}$$

Here x_2 and x_1 are the solutions that attain the maximum and the minimum in $\min(r,\ell_2)$ and $\max(r,\ell_1)$, respectively. The last inequality follows from the fact that the ith component of $x_2 - x_1$ is at most v_i in absolute value and the Cauchy-Schwartz inequality.

End of proof of Claim 1

Next, let us note

$$\min(a,k) \quad \leq \quad \max(a,k) \text{ for } k \in \{0,\ldots,pv\}. \tag{4.4.30}$$

Indeed, (4.4.30) holds, since the feasible sets of the optimization problems defining $\min(a,k)$ and $\max(a,k)$ contain $\{\, x \,|\, px = k,\ 0 \leq x \leq v \,\}$.

The nonnegativity of p and of a imply $\min(a,0) = 0$ and $\max(a,pe) = av$. The proof of the following claim is trivial, hence omitted.

Claim 2. *Suppose that ℓ_1 and ℓ_2 are integers in $\{0,\ldots,pv\}$ with $\ell_1 + 1 \leq \ell_2$ and*

$$\max(a,\ell_1) < \beta_1 \leq \beta_2 < \min(a,\ell_2). \tag{4.4.31}$$

Then for all x with $\beta_1 \leq ax \leq \beta_2,\ 0 \leq x \leq v$

$$\ell_1 < px < \ell_2 \tag{4.4.32}$$

holds.

We assume for simplicity

$$\max(a,0) < \beta_1 \leq \beta_2 < \min(a,pe); \tag{4.4.33}$$

the cases when this fails to hold are easy to handle separately. Let ℓ_1 be the largest and ℓ_2 the smallest integer such that

$$\max(a,\ell_1) < \beta_1 \leq \beta_2 < \min(a,\ell_2). \tag{4.4.34}$$

From (4.4.30) $\ell_2 \geq \ell_1 + 1$ follows and Claim 2 yields

$$\text{iwidth}(p,\ (SUB)) \ \leq\ \ell_2 - \ell_1 - 1. \tag{4.4.35}$$

By the choices of ℓ_1 and ℓ_2 we have

$$\beta_1 \leq \max(a,\ell_1+1) \text{ and } \beta_2 \geq \min(a,\ell_2-1), \tag{4.4.36}$$

hence Claim 1 leads to

$$\begin{aligned} \beta_2 - \beta_1 &\geq \min(a, \ell_2 - 1) - \max(a, \ell_1 + 1) \\ &\geq -\|r\|\|v\| + \lambda(\ell_2 - \ell_1 - 2), \end{aligned} \tag{4.4.37}$$

that is

$$\ell_2 - \ell_1 - 2 \leq \frac{\beta_2 - \beta_1}{\lambda} + \frac{\|r\|\|v\|}{\lambda}. \tag{4.4.38}$$

Comparing (4.4.35) and (4.4.38) completes the proof.

□

4.5 Successive Approximation

Theorems 4 and 5 approximate a by a single vector. It is natural to ask: if one row of U^{-1}, or of $(V, b)^{-1}$ is a good approximation of a, can we construct a better approximation from $2, 3, \ldots, k$ rows?

The answer is yes and we outline the corresponding results below, and their proofs, which are slight modifications of the proofs of Theorems 4 and 5. As of now, we don't know how to use the general results for a better analysis of the reformulations than what is already given in Theorem 3.

So we mainly state the successive approximation results for the interesting geometric intuition they give. Let us define

$$\begin{aligned} f(a, k) &= 2^{(k(n-k)+1)/4} / \|a\|^{k/n}, \\ g(a, k) &= 2^{k(n-1-k)/4} / \|a\|^{(k-1)/n}. \end{aligned} \tag{4.5.39}$$

The successive version of Theorem 4 is given below:

Theorem 7. *Let $a \in \mathbb{Z}^n$ be a row-vector, with $\|a\| \geq 2^{(n/2+1)n}$, U a unimodular matrix such that the columns of*

$$\begin{pmatrix} a \\ I \end{pmatrix} U$$

are LLL-reduced and P_k the (integral) submatrix of U^{-1} consisting of the last k rows. Furthermore, let $a(k)$ be the projection of a onto the subspace spanned by the rows of P_k, $r = a - a(k)$ and

$$\lambda_k := \|a(k)\| / \det(P_k P_k^T)^{1/2}.$$

Then

(1) $(\det(P_k P_k^T))^{1/2}(1+\|r\|^2)^{1/2} \le \|a\| f(a,k)$;

(2) $\lambda_k \ge 1/f(a,k)$;

(3) $|\sin(a,a(k))| \le \|r\|/\lambda_k \le 2f(a,k)$.

Proof sketch We will use the notation of Theorem 4. In its proof we simply change (4.3.15) (we copy the first expression for $\det L_n$ for easy reference) to

$$\begin{aligned} \det L_n &= \det L_n^{\perp} = (\|a\|^2+1)^{1/2}, \\ \det L_{n-k} &= \det L_{n-k}^{\perp} = (\det(P_k P_k^T))^{1/2}(1+\|r\|^2)^{1/2}, \end{aligned} \tag{4.5.40}$$

and (4.3.16) to

$$\det L_{n-k} \le 2^{k(n-k)/4}(\det L_n)^{1-k/n}. \tag{4.5.41}$$

Then substituting into (4.5.41) from (4.5.40) gives

$$\begin{aligned} (\det(P_k P_k^T))^{1/2}(1+\|r\|^2)^{1/2} &\le 2^{(k(n-k))/4}(\sqrt{\|a\|^2+1})^{1-k/n} \\ &\le 2^{(k(n-k)+1)/4}/\|a\|^{k/n} \\ &= \|a\| f(a,k), \end{aligned} \tag{4.5.42}$$

with the second inequality coming the lower bound on $\|a\|$. This shows (1) and the rest of the proof follows verbatim the proof of Theorem 4. □

Theorem 5 also has a successive variant, which is

Theorem 8. *Suppose* $\|a\| \ge 2^{(n/2+1)n}$. *Let* V *be a matrix whose columns are an LLL-reduced basis of* $\mathbb{N}(a)$, b *an integral column vector with* $ab = 1$, $k \le n-1$ *an integer, and* P_k *the (integral) submatrix of* $(V,b)^{-1}$ *consisting of the next-to-last* k *rows.*

Furthermore, let $a(k)$ *be the projection of* a *onto the subspace spanned by the rows of* P_k, $r = a - a(k)$ *and*

$$\lambda_k := \|a(k)\|/\det(P_k P_k^T)^{1/2}.$$

Then $r \ne 0$ *and*

(1) $(\det(P_k P_k^T))^{1/2} \parallel r \parallel \leq \parallel a \parallel g(a,k)$;

(2) $|\sin(a, a(k))| \leq \parallel r \parallel / \lambda \leq 2g(a,k)$.

Proof sketch We will use the notation of Theorem 5. We need to replace (4.3.23) with

$$\begin{aligned} \det L_{n-1} &= \det L_{n-1}^{\perp} = \parallel a \parallel, \\ \det L_{n-1-k} &= \det L_{n-1-k}^{\perp} = (\det(P_k P_k^T))^{1/2} \parallel r \parallel . \end{aligned} \tag{4.5.43}$$

Theorem 2 implies

$$\det L_{n-1-k} \leq 2^{k(n-1-k)/4} (\det L_{n-1})^{1-k/(n-1)}. \tag{4.5.44}$$

Plugging the expressions for $\det L_{n-1}$ and $\det L_{n-1-k}$ from (4.5.43) into (4.5.44) gives

$$\begin{aligned} (\det(P_k P_k^T))^{1/2} \parallel r \parallel &\leq 2^{k(n-1-k)/4} \parallel a \parallel^{1-k/(n-1)} \\ &= g(a,k) \parallel a \parallel, \end{aligned} \tag{4.5.45}$$

proving (1). The rest of the proof is an almost verbatim copy of the corresponding proof in Theorem 5. □

4.6 Discussion

Computing a near parallel vector can be done in other ways as well. The relevance of Theorems 4 and 5 is not just finding near parallel vectors: it is finding a near parallel p, which corresponds to a unit vector in the rangespace and nullspace reformulations, thus leading to the analysis of Theorem 3.

Finding an integral vector, which is near parallel to an other integral or rational one has other applications as well. In [24] Huyer and Neumaier studied several notions of near parallelness, presented numerical algorithms, and applications to verifying the feasibility of a linear system of inequalities.

Theorems 4 and 5 approximate a by a single vector, last row of U^{-1}. In Chapter 6, we will show that branching on multiple rows in succession (i.e., on last row of U^{-1}, ..., first row of U^{-1}) is also beneficial in solving the majority of the randomly generated knapsack problems.

In the next chapter, we show that for a low density subset sum problem, there is a polynomial time computable certificate of infeasibility for almost all β integer right hand sides. This implies that for

almost all right hand sides, the integer width along the last variable in the rangespace reformulation of a low density subset sum problem is zero.

CHAPTER 5

Branching Proofs of Infeasibility in Low Density Subset Sum Problems

5.1 Introduction

In this chapter, we prove that the subset sum problem

$$\begin{aligned} ax &= \beta \\ x &\in \{0,1\}^n \end{aligned} \qquad (SUB)$$

has a polynomial time computable certificate of infeasibility for all a with density at most $1/(2n)$ and for almost all integer right hand sides β. The certificate is branching on a hyperplane.

The proof has two ingredients. We first prove that a "short" vector that is near parallel to a is a suitable branching direction, regardless of the density. Then we show that for a low density a such a short and near parallel vector can be computed using diophantine approximation, via a methodology introduced by Frank and Tardos in [15]. We also show that the last row of U^{-1}, the inverse of the transformation matrix, in the rangespace reformulation can also be used to prove the same result, which implies that the infeasibility of almost all low density subset sum problems can be proved by branching on the last variable after the problem is reformulated using the rangespace reformulation.

5.2 Literature Review

The subset sum problem (SUB) is one of the original NP-complete problems introduced by Karp [29]. A particular reason for its importance is its applicability in cryptography. With a being a public

key and x the message, one can transmit $\beta = ax$ instead of x. An eavesdropper would need to find x from the intercepted β and the public a, i.e., solve (SUB), while a legitimate receiver can use a suitable private key to decode the message. In cryptography applications, instances with low density are of interest, with the density of $a \in \mathbb{Z}^n$ defined as

$$d(a) = \frac{n}{\log_2 \| a \|_\infty}. \tag{5.2.1}$$

A line of research started in the seminal paper of Lagarias and Odlyzko [33], focused on solving such instances. In [33] the authors proved that the solution to (SUB) can be found for all but at most a fraction of $1/2^n$ of all a vectors with $d(a) < c/n$ and assuming that the solution exists. Here c is a constant approximately equal to 4.8. Frieze in [16] gave a simplified algorithm to prove their result.

From now on we will say that a statement is true for almost all elements of a set S, if it is true for at least a fraction of $1 - 1/2^n$ of them, with the value of n always clear from the context.

Furst and Kannan in [17] pursued an approach that looked at both feasible and infeasible instances. In [17] they showed that for some constant $c > 0$, if $M \geq 2^{cn \log n}$, then for almost all $a \in \{ 1, \ldots, M \}^n$ and all β the problem (SUB) has a polynomial size proof of feasibility or infeasibility. Their second result shows that for some constant $d > 0$, if $M \geq 2^{dn^2}$, then for almost all $a \in \{ 1, \ldots, M \}^n$ and all β the problem (SUB) can be *solved* in polynomial time.

All the above proofs construct a candidate solution to (SUB) as a "short" vector in a certain lattice. Finding a vector whose length is off by a factor of at most $2^{(n-1)/2}$ from the shortest one is done utilizing the LLL basis reduction method.

Assuming the availability of a *lattice oracle*, which finds the shortest vector in a lattice, Lagarias and Odlyzko in [33] show a similar result under the weaker assumption $d(a) < 0.6463$. The current best result on finding the solution of almost all solvable subset sum problems using a lattice oracle is by Coster et al. [12]: they require only $d(a) < 0.9408$. It is an open question to prove the infeasibility of almost all subset sum problems with density upper bounded by a constant, without assuming the availibility of an oracle. For more references, we refer to [12] and [42].

5.3 Main Results

In this section we look at the structure of low density subset sum problems from a complementary, or dual viewpoint. With P a polyhedron and v an integral vector, it is clear that P has no integral point if vx is nonintegral for all $x \in P$. We will examine such proofs of infeasibility of (SUB). Let

$$G(a,v) \quad = \quad \{\, \beta \in \mathbb{Z} \,|\, vx \notin \mathbb{Z} \text{ for all } x \text{ with } ax = \beta,\ 0 \leq x \leq e \,\}, \tag{5.3.2}$$

where e denotes a column vector of all ones. We will say that for the right hand sides β in $G(a,v)$ the infeasibility of (SUB) is proven by branching on vx. The reason for this terminology is that letting $P = \{\, x \,|\, ax = \beta,\ 0 \leq x \leq e \,\}$, β is in $G(a,v)$ iff both the maximum and the minimum of vx over P are between two consecutive integers.

We shall write $\mathbb{Z}^n_+$, and $\mathbb{Z}^n_{++}$ for the set of nonnegative and positive integral n-vectors, respectively. We will throughout assume $n \geq 10$, and that the components of a are relatively prime. We only consider nontrivial right hand sides of (SUB), i.e., right hand sides from $\{\, 0, 1, \ldots, \|\, a \,\|_1 \,\}$.

Our first main result is:

Theorem 9. *Suppose $d(a) \leq 1/(2n)$. Then we can compute in polynomial time an integral vector v, such that for almost all right hand sides the infeasibility of (SUB) is proven by branching on vx.*

Also, $G(a,v)$ can be covered by the disjoint union of at most 2^{2n^2} intervals, each of length at least 2^n.

□

Note that Theorem 9 further narrows the range of hard instances from the work of Furst and Kannan in [17].

There are at most 2^n right hand sides for which (SUB) is feasible, so most right hand sides lead to an infeasible instance, when $d(a)$ is small. However, in principle, it may be difficult to *prove* the infeasibility of many infeasible instances. Fortunately, this is not the case, as shown by the following corollary.

Corollary 2. *Let a and v be as in Theorem 9. Then for almost all right hand sides for which (SUB) is infeasible, its infeasibility is proven by branching on vx.*

□

There is an interesting duality and parallel between the results on low density subset sum in [12, 17, 33] and Theorem 9. The proofs in [12, 17, 33] work by constructing a candidate solution, while ours works by branching, i.e., by a dual method. At the same time, they all rely on basis reduction. In our proof we find v by a method of Frank and Tardos in [15], which uses the simultaneous diophantine approximation method of Lenstra, Lenstra and Lovász [35], which in turn, also uses basis reduction.

Theorem 9 will follow from combining Theorems 10 and 11 below. Theorem 10 proves that a "large" fraction of righ hand sides in (SUB) have their infeasibility proven by branching on vx, if v is relatively short and near parallel to a. Theorem 11 will show that such a v can be found using diophantine approximation, when $d(a) \leq 1/(2n)$.

Theorem 10. *Let $v \in \mathbb{Z}^n_+$, $\lambda \in \mathbb{R}$, $r \in \mathbb{R}^n$ with $\lambda \geq 1$, $\| r \|_1 / \lambda < 1$, and*

$$a = \lambda v + r.$$

Then the infeasibility of all but at most a fraction of

$$\frac{2(\| r \|_1 + 1)}{\lambda} \tag{5.3.3}$$

right hand sides is proven by branching on vx.

In addition, $G(a, v)$ can be covered by the disjoint union of at most $\| v \|_1$ intervals, each of length at least $\lambda - \| r \|_1$.

Theorem 11. *Suppose $d(a) \leq 1/(2n)$. Then we can compute in polynomial time $v \in \mathbb{Z}^n_+$, $\lambda \in \mathbb{Q}$, $r \in \mathbb{Q}^n$ with $a = \lambda v + r$, and*

(1) $\| v \|_1 \leq 2^{2n^2}$;

(2) $\| r \|_1 / \lambda \leq 1/2^{n+2}$;

(3) $\lambda \geq 2^{n+2}$.

□

□

5.4 Proofs

Proof of Theorem 10 Let us fix a and v. Since a and v are nonnegative, and e is a column vector of all ones, it holds that

$$\| a \|_1 = ae \text{ and } \| v \|_1 = ve,$$

and we will use the latter notation for brevity.

Recall that for a row-vector w and an integer ℓ we write

$$\begin{aligned} \max(w, \ell) &= \max \{ wx \mid vx \leq \ell,\ 0 \leq x \leq e \}, \\ \min(w, \ell) &= \min \{ wx \mid vx \geq \ell,\ 0 \leq x \leq e \}. \end{aligned} \tag{5.4.4}$$

The dependence on v and on the sense of the constraint (i.e., $\leq$ or $\geq$) is not shown by this notation; however, we always use $vx \leq \ell$ with "max" and $vx \geq \ell$ with "min", and v is fixed.

Claim 3. *We have*

$$\min(a, k) \leq \max(a, k) \text{ for } k \in \{0, \ldots, ve\}, \tag{5.4.5}$$

$$\max(a, k) - \min(a, k) \leq \| r \|_1 \text{ for } k \in \{0, \ldots, ve\}, \text{ and} \tag{5.4.6}$$

$$\min(a, k+1) - \max(a, k) \geq -\| r \|_1 + \lambda > 0 \text{ for } k \in \{0, \ldots, ve - 1\}. \tag{5.4.7}$$

Proof The feasible sets of the optimization problems defining $\min(a, k)$ and $\max(a, k)$ contain $\{ x \mid vx = k,\ 0 \leq x \leq e \}$, so (5.4.5) follows.

The decomposition of a shows that for all ℓ_1 and ℓ_2 integers for which the expressions below are defined,

$$\begin{aligned} \max(a, \ell_1) &\leq \max(r, \ell_1) + \lambda \ell_1, \text{ and} \\ \min(a, \ell_2) &\geq \min(r, \ell_2) + \lambda \ell_2, \end{aligned} \tag{5.4.8}$$

hold. Therefore

$$\begin{aligned} \min(a, \ell_2) - \max(a, \ell_1) &\geq \min(r, \ell_2) - \max(r, \ell_1) + \lambda(\ell_2 - \ell_1) \\ &\geq -\| r \|_1 + \lambda(\ell_2 - \ell_1). \end{aligned} \tag{5.4.9}$$

follows, and (5.4.9) with $\ell_2 = \ell_1 = k$ implies (5.4.6), and with $\ell_2 = k+1$, $\ell_1 = k$ yields (5.4.7). Hence

$$\min(a,0) \le \max(a,0) < \min(a,1) \le \max(a,1) < \cdots < \min(a,ve) \le \max(a,ve). \tag{5.4.10}$$

We will call the intervals

$$[\min(a,0), \max(a,0)], \ldots, [\min(a,ve), \max(a,ve)]$$

bad, and the intervals

$$G_0 := (\max(a,0), \min(a,1)), \ldots, G_{ve-1} := (\max(a,ve-1), \min(a,ve))$$

good.

The nonnegativity of v and of a imply $\min(a,0) = 0$ and $\max(a,ve) = ae$, so the bad and good intervals partition $[0, ae]$: the pattern is bad, good, ..., good, bad. Some of the bad intervals may have zero length, but by (5.4.7) none of the good ones do.

Next we show that the good intervals contain exactly the right hand sides for which the infeasibility of (SUB) is proven by branching on vx.

Claim 4.

$$G(a,v) = \cup_{i=0}^{ve-1} G_i \cap \mathbb{Z}. \tag{5.4.11}$$

Proof By definition $\beta \in G(a,v)$ iff for some ℓ integer with $0 \le \ell < ve-1$ and for all x with $0 \le x \le e$, $ax = \beta$

$$\ell < vx < \ell + 1 \tag{5.4.12}$$

holds. We show that for this ℓ

$$\max(a,\ell) < \beta \text{ and} \tag{5.4.13}$$

$$\min(a,\ell+1) > \beta. \tag{5.4.14}$$

First, assume to the contrary that (5.4.13) is false, i.e., there exists x_1 with

$$ax_1 \geq \beta,\ vx_1 \leq \ell,\ 0 \leq x_1 \leq e. \tag{5.4.15}$$

Since $\ell \geq 0,$ denoting by x_2 the all-zero vector, it holds that

$$ax_2 \leq \beta,\ vx_2 \leq \ell,\ 0 \leq x_2 \leq e. \tag{5.4.16}$$

Looking at (5.4.15) and (5.4.16) it is clear that a convex combination of x_1 and $x_2,$ say $\bar{x}$ satisfies

$$a\bar{x} = \beta,\ v\bar{x} \leq \ell,\ 0 \leq \bar{x} \leq e, \tag{5.4.17}$$

which contradicts (5.4.13). Showing (5.4.14) is analogous.

End of proof of Claim 4

To summarize, Claim 4 implies that $G(a,v)$ is covered by the disjoint union of ve intervals. By (5.4.7) their length is lower bounded by $\lambda - \|r\|_1$.

Let us denote by b the number of integers in bad intervals and by g the number of integers in good intervals, i.e., $g = |G(a,v)|$. Using (5.4.6) and (5.4.7), and the fact that there are ve good intervals and $ve+1$ bad ones, we get

$$\begin{aligned} g &\geq ve(\lambda - \|r\|_1 - 1), \\ b &\leq (ve+1)(\|r\|_1 + 1), \end{aligned} \tag{5.4.18}$$

so

$$\frac{g}{b} \geq \frac{ve}{ve+1}\frac{\lambda - (\|r\|_1 + 1)}{\|r\|_1 + 1} \tag{5.4.19}$$

$$\geq \frac{1}{2}\frac{\lambda - (\|r\|_1 + 1)}{\|r\|_1 + 1} \tag{5.4.20}$$

$$\geq \frac{\lambda}{2(\|r\|_1 + 1)} - 1, \tag{5.4.21}$$

and from here

$$\frac{b}{g+b} = \frac{1}{1+g/b} \tag{5.4.22}$$

$$\leq \frac{2(\|r\|_1+1)}{\lambda}. \tag{5.4.23}$$

follows. □

Proof of Theorem 11

We will use a methodology due to Frank and Tardos introduced in [15]. Here the authors employ simultaneous diophantine approximation to decompose a vector with large norm into the weighted sum of smaller norm vectors. We will only need one vector that approximates a and the parameters will be somewhat differently chosen in the diophantine approximation.

We will rely on the following result of Lenstra, Lenstra and Lovász from [35]:

Theorem 12. *Given a positive integer N and $\alpha \in \mathbb{Q}^n$, we can compute in polynomial time $v \in \mathbb{Z}^n$, $q \in \mathbb{Z}_{++}$ such that*

$$\|q\alpha - v\|_\infty \leq \frac{1}{N} \text{ and} \tag{5.4.24}$$

$$q \leq 2^{n(n+1)/4}N^n. \tag{5.4.25}$$

□

We will use Theorem 12 with

$$\alpha = \frac{a}{\|a\|_\infty},$$

then set

$$\lambda = \frac{\|a\|_\infty}{q},\ r = a - \lambda v.$$

We have the following estimates with ensuing explanation:

$$\|v\|_1 \leq n\|v\|_\infty \leq nq \leq n2^{n(n+1)/4}N^n, \tag{5.4.26}$$

$$\frac{\|r\|_1}{\lambda} \leq \frac{n\|r\|_\infty}{\lambda} \leq \frac{n}{N}, \tag{5.4.27}$$

$$\lambda \geq \frac{\|a\|_\infty}{2^{n(n+1)/4}N^n} \geq \frac{2^{2n^2-n(n+1)/4}}{N^n}. \tag{5.4.28}$$

Here (5.4.26) follows from using (5.4.24), since $\|q\alpha\|_\infty = q$ and v is integral. The second inequality in (5.4.27) is actually equivalent to (5.4.24); and (5.4.28) comes from the definition of λ and (5.4.25).

Hence (1), (2), and (3) in Theorem 11 are satisfied when

$$n2^{n(n+1)/4}N^n \leq 2^{2n^2}, \tag{5.4.29}$$

$$\frac{n}{N} \leq \frac{1}{2^{n+2}}, \tag{5.4.30}$$

$$\frac{2^{2n^2-n(n+1)/4}}{N^n} \geq 2^{n+2}. \tag{5.4.31}$$

But (5.4.29) through (5.4.31) are equivalent to

$$n2^{n+2} \leq N \leq 2^{2n-(n+1)/4-1-2/n}, \tag{5.4.32}$$

and such an integer N exists, when $n \geq 10$. □

Proof of Corollary 2 Let $I(a)$ be the set of right hand sides for which (SUB) is infeasible. Theorem 9 states

$$\frac{|G(a,v)|}{\|a\|_1 + 1} \geq 1 - \frac{1}{2^n}. \tag{5.4.33}$$

Since $I(a) \subseteq \{0, \ldots, \|a\|_1\}$, Theorem 9 implies

$$\frac{|G(a,v)|}{I(a)} \geq 1 - \frac{1}{2^n}; \tag{5.4.34}$$

and since $G(a,v) \subseteq I(a)$, (5.4.34) means the desired conclusion. □

5.5 Discussion

Looking at the decomposition in Theorem 4, it is easy to see that, branching on p, last row of the inverse of the transformation matrix in the rangespace reformulation, proves the infeasibility of almost all subset sum problems when $\|a\|$ is large enough in the same way. I will briefly mention the results here without going into the details.

Let (*SUB-R*) denote the rangespace reformulation of (SUB).

Theorem 13. *Suppose* $a \in \mathbb{Z}^n$, $\|a\| \geq 2^{1.5n^2}$, *and let* p *be the last row of* U^{-1} *in the rangespace reformulation. Then*

(1) $\mathrm{iwidth}(p, (SUB)) \leq 1$ *for all* $\beta \in \{1, \ldots, \sum a_i\}$.

(2) $\mathrm{iwidth}(p, (SUB)) = 0$ *for almost all* $\beta \in \{1, \ldots, \sum a_i\}$.

Theorem 14. *Suppose* $a \in \mathbb{Z}^n$, $\| a \| \geq 2^{1.5n^2}$. *Then*

(1) $\mathrm{iwidth}(e_n, (\textit{SUB-R}) \leq 1$ *for all* $\beta \in \{1, \ldots, \sum a_i\}$.

(2) $\mathrm{iwidth}(e_n, (\textit{SUB-R}) = 0$ *for almost all* $\beta \in \{1, \ldots, \sum a_i\}$.

CHAPTER 6

Basis Reduction and the Complexity of Branch and Bound

The classical branch and bound algorithm for the integer feasibility problem

$$\text{Find } x \in Q \cap \mathbb{Z}^n, \text{ with } Q = \left\{ x \mid \begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} x \leq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \right\} \tag{6.0.1}$$

has exponential worst case complexity. We prove that it is surprisingly efficient on reformulations of (6.0.1), in which the columns of the constraint matrix are "short", and "near orthogonal", i.e., a reduced basis of the generated lattice.

The analysis builds on Furst and Kannan's work on the subset sum problem and also uses an upper bound on the size of the branch and bound tree based on Lenstra's analysis of his integer programming algorithm.

We show that when the entries of A are from $\{1, \ldots, M\}$ for a large enough M, branch and bound solves almost all reformulated instances at the root node, and explore practical aspects of this result. We compute numerical values of M which guarantee that 90 and 99 percent of the reformulated problems solve at the root: these turn out to be surprisingly small when the problem size is moderate.

A computational study also confirms that the reformulations of random integer programs become easier, as the coefficients grow.

6.1 Introduction and Main Results

The Integer Programming (IP) feasibility problem asks whether a polyhedron Q contains an integral point. Branch and bound (B&B) is a classical solution method. We will briefly introduce ordinary B&B. It starts with Q as the sole subproblem (node) (level $j = 0$). In a general step, one chooses a variable

x_i, and for each subproblem Q' at level j, the new subproblems $Q' \cap \{x | x_i = \gamma\}$ are created, where γ ranges over all possible integer values of x_i. Now all the subproblems are at the $(j+1)$st level of the B&B tree. We repeat this until all subproblems are shown to be empty or we find an integral point in one of them.

B&B enhanced by cutting planes is the workhorse method for integer programming implemented in most commercial software. However, instances in [3, 4, 8, 21, 25, 31] show that it is theoretically inefficient: it can take an exponential number of subproblems to prove the infeasibility of simple knapsack problems. Chvátal in [8] proves that this is true for the majority of knapsack problems in a certain natural family. While B&B is inefficient in the worst case, Cornuéjols et al. in [19] developed useful computational tools to give an early estimate on the size of the B&B tree in practice.

Since IP feasibility is NP-complete, one can ask for polynomiality of a solution method only in fixed dimension. All algorithms that achieve such complexity rely on advanced techniques. The algorithms of Lenstra [36] and Kannan [27] (see Section 2.3.1) first round the polyhedron (i.e., apply a transformation to make it have a spherical appearance), then use basis reduction to reduce the problem to a provably small number of smaller dimensional subproblems. On the subproblems the algorithms are applied recursively, e.g., rounding is done again. Generalized basis reduction, proposed by Lovász and Scarf in [39] avoids rounding, but needs to solve a sequence of linear programs to create the subproblems. In fixed dimension one can even *count* the number of feasible solutions in polynomial time: see the papers of Barvinok [6], and Dyer and Kannan [13]. We refer to [10, 37] for successful implementations of these theoretically efficient methods and to Haus et al. [22] for a finite augmentation type algorithm using basis reduction.

As explained in Section 2.3.2, there is a simpler way to use basis reduction in integer programming: preprocessing (6.0.1) to create an instance with short and near orthogonal columns in the constraint matrix, then simply feeding the resulting instance to an IP solver. We describe two such methods that were proposed recently. We assume that A is an integral matrix with m rows and n columns, and the w_i and ℓ_i are integral vectors.

The rangespace reformulation of (6.0.1) is

$$\text{Find } y \in Q_R \cap \mathbb{Z}^n, \text{ with } Q_R = \left\{ y \mid \begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} Uy \leq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \right\}, \tag{6.1.2}$$

where U is a unimodular matrix computed to make the columns of the constraint matrix a reduced basis of the generated lattice.

The nullspace reformulation is applicable, when $w_1 = \ell_1$. It is

$$\text{Find } y \in Q_N \cap \mathbb{Z}^{n-m}, \text{ with } Q_N = \{\, y \,|\, \ell_2 - x_0 \leq By \leq w_2 - x_0 \} \,, \tag{6.1.3}$$

where $x_0 \in \mathbb{Z}^n$ satisfies $Ax_0 = \ell_1$ and the columns of B are a reduced basis of the null lattice of A, $\mathbb{N}(A) = \{\, x \in \mathbb{Z}^n \,|\, Ax = 0 \,\}$.

We analyze the use of LLL, KZ and RKZ reduced bases in the reformulations (for more details, see Sections 2.2.1, 2.2.2 and 2.2.5). When Q_R is computed using RKZ reduction, we call it the RKZ-rangespace reformulation of Q; similarly we talk about an RKZ-nullspace, LLL-rangespace, LLL-nullspace, KZ-rangespace and KZ-nullspace reformulation.

Example 6. The polyhedron

$$\begin{array}{rcccl} 121 & \leq & 20x_1 + 18x_2 + 37x_3 & \leq & 125 \\ 0 & \leq & x_1, x_2, x_3 & \leq & 7 \end{array} \tag{6.1.4}$$

is shown on the first picture of Figure 6.1. It defines an infeasible and relatively difficult integer feasibility problem for B&B, as branching on either x_1, x_2 or x_3 yields at least 4 subproblems; and infeasibility can be proved only in the third level of the B&B tree. It is interesting to see how the various algorithms described above would work on the instance 6.1.4. Lenstra's and Kannan's algorithms would first transform this polyhedron to make it more spherical; generalized basis reduction would solve a sequence of linear programs to find the direction $x_1 + x_2 + 2x_3$ along which the polyhedron is thin.

The LLL-rangespace reformulation is

$$\begin{array}{rcccl} 121 & \leq & -x_1 - 2x_2 + 6x_3 & \leq & 125 \\ 0 & \leq & -x_1 - x_2 - 7x_3 & \leq & 7 \\ 0 & \leq & -x_1 + x_2 + 4x_3 & \leq & 7 \\ 0 & \leq & x_1 + 2x_3 & \leq & 7 \end{array} \tag{6.1.5}$$

shown on the second picture of Figure 6.1: now branching on y_3 proves integer infeasibility.

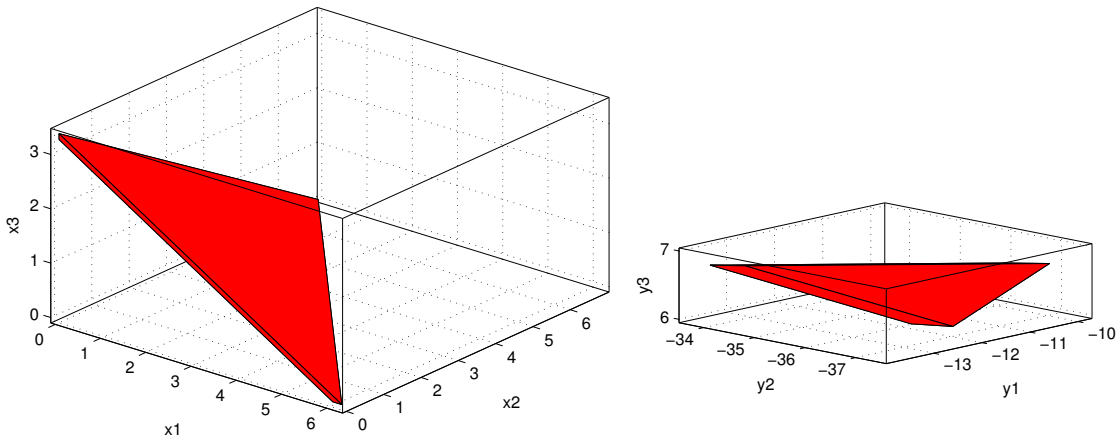

Figure 6.1: LP Relaxations of the Problem in Example 6 and its LLL-Rangespace Reformulation

Branching on $x_1 + x_2 + 2x_3$ (which is the last row of the inverse of the transformation matrix) in the original problem is equivalent to branching on y_3 in the reformulated problem.

The reformulation methods are very successful in practice in solving several classes of hard integer programs. Notably, the original formulations of the marketshare problems of Cornuéjols and Dawande in [11] are notoriously difficult for commercial solvers, while the nullspace reformulations are much easier to solve as shown by Aardal et. al. in [1].

However, they seem difficult to analyze in general. The only analysis that exists so far is for knapsack problems with a constraint vector of the form $a = \lambda p + r,$ with p and r integral vectors, and λ an integer, large compared to $\| p \|$ and $\| r \|$. Aardal and Lenstra in [3, 4] proved a lower bound on the norm of the last vector in the nullspace reformulation, and argued that branching on such a long vector creates a small number of B&B nodes. Krishnamoorthy and Pataki in [31] pointed out a gap in this proof, and showed that branching on the constraint px in Q (which creates a small number of subproblems, as λ is large) is equivalent to branching on the last variable in Q_R and Q_N.

A result one may hope for is proving polynomiality of B&B on the reformulations of (6.0.1) when the dimension is fixed. While this seems difficult, we give a different and perhaps even more surprising complexity analysis. It is in the spirit of Furst and Kannan's work in [17] on subset sum problems and builds on their results to bound the fraction of integral matrices for which the shortest vector of two corresponding lattices is short. We also use an upper bound on the size of the B&B tree, which is based on Lenstra's analysis of his integer programming algorithm in [36]. We introduce necessary notation

and state our results, then give a detailed comparison with [17].

Backward B&B is B&B branching on the variables in reverse order starting with the one of highest index. For a positive integer M we denote by $G_{m,n}(M)$ the set of matrices with m rows and n columns, and the entries drawn from $\{1, \ldots, M\}$. Remember that for an m by n integral matrix A with full row rank, we write $\gcd(A)$ for the greatest common divisor of the m by m subdeterminants of A. If B&B generates at most one node at each level of the B&B tree, we say that it solves an integer feasibility problem at the root node.

The main results of the paper follow.

Theorem 15. *Let* $0 < \epsilon < 1$.

(1) If

$$M > \frac{(2\gamma_n \parallel (w_1; w_2) - (\ell_1; \ell_2) \parallel +1)^{1+n/m}}{\epsilon^{1/m}}, \tag{6.1.6}$$

then for all but at most a fraction ϵ *of* $A \in G_{m,n}(M)$ *backward B&B solves the RKZ-rangespace reformulation of* Q *at the root node.*

(2) If

$$M > \frac{(2\gamma_{n-m} \parallel w_2 - \ell_2 \parallel +1)^{n/m}}{\epsilon^{1/m}}, \tag{6.1.7}$$

then for all but at most a fraction ϵ *of* $A \in G_{m,n}(M)$ *backward B&B solves the RKZ-nullspace reformulation of* Q *at the root node.*

□

Here $\gamma_i = max\{C_1, \ldots, C_i\}$, where C_i is the Hermite's constant. It is known that $\gamma_i \leq 1 + i/4$.

The proofs also show that when M obeys the above bounds, then Q has at most one element for all but at most a fraction of ϵ of $A \in G_{m,n}(M)$.

When a statement is true for all, but at most a fraction of $1/2^n$ of the elements of a set S, we say that it is true for *almost all* elements. So far, all polynomial time algorithms solving almost all subset sum instances required an M which is exponential in n, see for instance [16, 17, 33]. We note that when n/m is fixed and the problems are binary, the magnitude of M required for the RKZ-rangespace and RKZ-nullspace reformulations to solve almost all instances is a polynomial in n. To see this, we let $\epsilon = 1/2^n$ and observe that the lower bound on M is a polynomial in n when n/m is fixed.

Theorem 16. *Let* $0 < \epsilon < 1$.

(1) If

$$M > \frac{(2^{(n+1)/2} \parallel (w_1; w_2) - (\ell_1; \ell_2) \parallel +1)^{1+n/m}}{\epsilon^{1/m}}, \tag{6.1.8}$$

then for all but at most a fraction ϵ *of* $A \in G_{m,n}(M)$ *backward B&B solves the LLL-rangespace reformulation of* Q *at the root node.*

(2) If

$$M > \frac{(2^{(n-m+1)/2} \parallel (w_2 - \ell_2) \parallel +1)^{n/m}}{\epsilon^{1/m}}, \tag{6.1.9}$$

then for all but at most a fraction ϵ *of* $A \in G_{m,n}(M)$ *backward B&B solves the LLL-nullspace reformulation of* Q *at the root node.*

□

Theorem 17. *Let* $0 < \epsilon < 1$.

(1) If

$$M > \frac{(2n^{(1+\log n)/2} \parallel (w_1; w_2) - (\ell_1; \ell_2) \parallel +1)^{1+n/m}}{\epsilon^{1/m}}, \tag{6.1.10}$$

then for all but at most a fraction ϵ *of* $A \in G_{m,n}(M)$ *backward B&B solves the KZ-rangespace reformulation of* Q *at the root node.*

(2) If

$$M > \frac{(2(n-m)^{(1+\log(n-m))/2} \parallel (w_2 - \ell_2) \parallel +1)^{n/m}}{\epsilon^{1/m}}, \tag{6.1.11}$$

then for all but at most a fraction ϵ *of* $A \in G_{m,n}(M)$ *backward B&B solves the KZ-nullspace reformulation of* Q *at the root node.*

□

Furst and Kannan, based on Lagarias' and Odlyzko's [33] and Frieze's [16] work show that the subset sum problem is solvable in polynomial time for almost all weight vectors in $\{1, \ldots, M\}^n$ and all right hand sides, when M is sufficiently large and a reduced basis of the orthogonal lattice of the weight vector is available. The lower bound on M is $2^{cn \log n}$, when the basis is RKZ reduced, and 2^{dn^2}, when it is LLL reduced. Here c and d are positive constants.

Our Theorems 15, 16 and 17 generalize the solvability results from subset sum problems to bounded integer programs; also, we prove them via branch and bound, an algorithm considered inefficient from the theoretical point of view.

A practitioner of integer programming may ask for the value of Theorems 15, 16 and and 17. Proposition 3 and Theorems 18 and 19, and a computational study put these results into a more practical perspective. Proposition 3 shows that when m and n are not too large, already fairly small values of M guarantee that the RKZ nullspace reformulation (which has the smallest bound on M) of the majority of binary integer programs get solved at the root node.

Proposition 3. *Suppose that m, n are chosen according to Table 6.1, and M is as shown in the third column.*

n	m	M for 90 %	M for 99 %
20	10	100	125
30	10	3491	4394
30	20	31	35
40	20	229	257
40	30	21	23
50	20	1846	2071
50	30	93	100
50	40	18	19
60	30	410	443
60	40	59	62
60	50	16	17
70	30	1880	2030
70	40	193	205
70	50	45	47
70	60	15	15

Table 6.1: Values of M to make sure that the RKZ-nullspace reformulation of 90 ($\epsilon = 0.1$) or 99 ($\epsilon = 0.01$) % of the instances of type (6.1.12) solve at the root node

Then for at least 90% of $A \in G_{m,n}(M)$, and all b right hand sides, backward B&B solves the RKZ-nullspace reformulation of

$$\begin{aligned} Ax &= b \\ x &\in \{0,1\}^n \end{aligned} \tag{6.1.12}$$

at the root node. The same is true for 99% of $A \in G_{m,n}(M)$, if M is as shown in the fourth column.

□

Note that 2^{n-m} is the best upper bound one can give on the number of nodes when B&B is run on the original formulation (6.1.12); also, randomly generated IPs with for example $n - m = 30$ are nontrivial even for commercial solvers.

Theorems 18 and 19 gives another indication why the reformulations are relatively easy. One can observe that $\det(AA^T)$ is astronomically large even for moderate values of M, if $A \in G_{m,n}(M)$ is a random matrix. While we cannot give a tight upper bound on the size of the B&B tree in terms of this determinant, we are able to bound the width of the reformulations along the last unit vector for any A (i.e., not just almost all).

Theorem 18. *If Q_R and Q_N are computed using RKZ reduction, then*

$$\text{width}(e_n, Q_R) \leq \frac{\sqrt{n}\, \| (w_1; w_2) - (\ell_1; \ell_2) \|}{\det(AA^T + I)^{1/(2n)}}. \tag{6.1.13}$$

Also, if A has independent rows, then

$$\text{width}(e_{n-m}, Q_N) \leq \frac{\gcd(A)\sqrt{n-m}\, \| w_2 - \ell_2 \|}{\det(AA^T)^{1/(2n)}}. \tag{6.1.14}$$

□

Theorem 19. *If Q_R and Q_N are computed using RKZ reduction, then*

$$\text{width}(e_n, Q_R) \leq \frac{2^{(n-1)/4}\, \| (w_1; w_2) - (\ell_1; \ell_2) \|}{\det(AA^T + I)^{1/(2n)}}. \tag{6.1.15}$$

Also, if A has independent rows, then

$$\text{width}(e_{n-m}, Q_N) \leq \frac{\gcd(A) 2^{(n-m-1)/4}\, \| w_2 - \ell_2 \|}{\det(AA^T)^{1/(2n)}}. \tag{6.1.16}$$

□

These two theorems generalize the width results we have in Chapter 4.

6.2 Computational Study

According to Theorems 15, 16 and 17, random integer programs with coefficients drawn from $\{1, \ldots, M\}$ should get easier, as M grows. Our computational study confirms this somewhat counterintuitive hypothesis on the family of marketshare problems of Cornuéjols and Dawande in [11]. The original formulations are notoriously difficult for commercial solvers, while the nullspace reformulations are much easier to solve as shown by Aardal et al. in [1].

We generated twelve 4-by-30, and twelve 5-by-40 matrices with entries drawn from $\{1, \ldots, M\}$ with $M = 100, 1000$ and 10000 (this is 72 matrices overall), set $b = \lfloor Ae/2 \rfloor$, where e is the vector of all ones and constructed the instances of type (6.1.12). Tables 6.2 and 6.3 show the average number of nodes created to solve the twelve instances generated from each class.

The detailed tables can be found at the end of the chapter. Tables 6.4, 6.5 and 6.6 show the number of nodes that the commercial IP solver CPLEX 9.0 took to solve the original (non-reformulated), the rangespace reformulation and the nullspace reformulation of 4-by-30 marketshare problems.

Tables 6.7, 6.8 and 6.9 show the number of nodes that the commercial IP solver CPLEX 9.0 took to solve the rangespace reformulation and the nullspace reformulation of 5-by-40 marketshare problems. None of the original 5-by-40 instances we generated was solved in under an hour by CPLEX 9.0. We used a Sun Ultrasparc desktop computer running the Solaris 10 operating system with processor speed 410 MHz.

Since RKZ reformulation is not implemented in any software that we know of, we used the KZ reduction routine from the NTL library [48].

In Section 6.3 we introduce further necessary notation, and give the proof of the main results.

M	Original	Rangespace	Nullspace
100	$1,050,406.25$	1503.58	545.75
1000	$1,136,736.17$	235.08	81.92
10000	$1,235,433.42$	61.08	20.33

Table 6.2: Average number of B&B nodes to solve 4-by-30 marketshare problems

M	Rangespace	Nullspace
100	$86,858.08$	$17,531.92$
1000	$5,850.75$	$1,254.42$
10000	858.33	200.83

Table 6.3: Average number of B&B nodes to solve 5-by-40 marketshare problems

6.3 Further Notation and Proofs

Remember that the Euclidean norm of a shortest nonzero vector in L is denoted by $\lambda_1(L)$, and C_j is Hermite's constant.

We define

$$\gamma_i = \max\{C_1, \ldots, C_i\}. \tag{6.3.17}$$

A matrix A defines two lattices that we are interested in:

$$L_R(A) = \mathbb{L}(A; I),\ L_N(A) = \{x \in \mathbb{Z}^n | Ax = 0\}, \tag{6.3.18}$$

where we recall that $(A; I)$ is the matrix obtained by stacking A on top of I. Here $L_N(A)$ is the same as the null lattice of A.

If $b_1, \ldots, b_r$ are an RKZ reduced basis of the lattice L with Gram-Schmidt orthogonalization $b_1^*, \ldots, b_r^*$, then recall that

$$\| b_i^* \| \geq \lambda_1(L)/C_i. \tag{6.3.19}$$

holds. If they are an LLL-reduced basis, then

$$\| b_i^* \| \geq \lambda_1(L)/2^{(i-1)/2}. \tag{6.3.20}$$

If they are a KZ-reduced basis, then

$$\| b_i^* \| \geq \lambda_1(L)/i^{(1+\log i)/2}. \tag{6.3.21}$$

Lemma 3 is based on the ideas of Lenstra in [36] used in the analysis of his integer programming algorithm.

Lemma 3. *Let P be a polyhedron*

$$P = \{y \in \mathbb{R}^r \,|\, \ell \leq By \leq w\}\,, \tag{6.3.22}$$

and $b_1^*, \ldots, b_r^*$ the Gram-Schmidt orthogonalization of the columns of B. When backward B&B is applied to P, the number of nodes on the level of y_i is at most

$$\left(\left\lfloor \frac{\|\, w-\ell \,\|}{\|\, b_i^* \,\|} \right\rfloor + 1\right) \ldots \left(\left\lfloor \frac{\|\, w-\ell \,\|}{\|\, b_r^* \,\|} \right\rfloor + 1\right). \tag{6.3.23}$$

Proof First we show

$$\mathrm{width}(e_r, P) \leq \|\, w-\ell \,\| \;/\; \|\, b_r^* \,\|\,. \tag{6.3.24}$$

Let $x_{r,1}$ and $x_{r,2}$ denote the maximum and the minimum of x_r over P. Writing $\bar{B}$ for the matrix composed of the first $r-1$ columns of B, and b_r for the last column, it holds that there is $x_1, x_2 \in \mathbb{R}^{r-1}$ such that $\bar{B}x_1 + b_r x_{r,1}$ and $\bar{B}x_2 + b_r x_{r,2}$ are in P. So

$$\begin{aligned}\|\, w-\ell \,\| &\geq \|\, (\bar{B}x_1 + b_r x_{r,1}) - (\bar{B}x_2 + b_r x_{r,2}) \,\| = \|\, \bar{B}(x_1 - x_2) + b_r(x_{r,1} - x_{r,2}) \,\| \\ &\geq \|\, b_r^* \,\|\; |x_{r,1} - x_{r,2}| = \|\, b_r^* \,\|\; \mathrm{width}(e_r, P)\end{aligned}$$

holds, and so does (6.3.24).

After branching on $e_r, \ldots, e_{i+1}$, each subproblem is defined by a matrix formed of the first i columns of B, and bound vectors ℓ^i and w^i, which are translates of ℓ and w by the same vector. Hence the above proof implies that the width along e_i in each of these subproblems is at most

$$\|\, w-\ell \,\| \;/\; \|\, b_i^* \,\|, \tag{6.3.25}$$

and this completes the proof. □

Our Lemma 4 uses ideas from Furst and Kannan's Lemma 1 in [17], with inequality (6.3.27) also being a direct generalization.

Lemma 4. *For a positive integer k, let ϵ_R and ϵ_N be the fraction of $A \in G_{m,n}(M)$ with $\lambda_1(L_R(A)) \leq$*

k, *and* $\lambda_1(L_N(A)) \leq k$, *respectively. Then*

$$\epsilon_R \leq \frac{(2k+1)^{n+m}}{M^m}, \tag{6.3.26}$$

and

$$\epsilon_N \leq \frac{(2k+1)^n}{M^m}. \tag{6.3.27}$$

Proof We first prove (6.3.27). For $v,$ a fixed nonzero vector in $\mathbb{Z}^n,$ consider the equation

$$Av = 0. \tag{6.3.28}$$

There are at most $M^{m(n-1)}$ matrices in $G_{m,n}(M)$ that satisfy (6.3.28): if the components of $n-1$ columns of A are fixed, then the components of the column corresponding to a nonzero entry of v are determined from (6.3.28). The number of vectors v in $\mathbb{Z}^n$ with $\| v \| \leq k$ is at most $(2k+1)^n$, and the number of matrices in $G_{m,n}(M)$ is M^{mn}. Therefore

$$\epsilon_N \leq \frac{(2k+1)^n M^{m(n-1)}}{M^{mn}} = \frac{(2k+1)^n}{M^m}.$$

For (6.3.26), note that $(v_1; v_2) \in \mathbb{Z}^{m+n}$ is a nonzero vector in $L_R(A)$, iff $v_2 \neq 0,$ and

$$Av_2 = v_1. \tag{6.3.29}$$

An argument like the one in the proof of (6.3.27) shows that for fixed $(v_1; v_2) \in \mathbb{Z}^{m+n}$ with $v_2 \neq 0,$ there are at most $M^{m(n-1)}$ matrices in $G_{m,n}(M)$ that satisfy (6.3.29). The number of vectors in $\mathbb{Z}^{n+m}$ with norm at most k is at most $(2k+1)^{n+m}$, so

$$\epsilon_R \leq \frac{(2k+1)^{n+m} M^{m(n-1)}}{M^{mn}} = \frac{(2k+1)^{n+m}}{M^m}.$$

□

Proof of Theorems 15,16 and 17 Let $b_1^*, \ldots, b_n^*$ be the Gram-Schmidt orthogonalization of the columns of $(A; I)U$. Lemma 3 implies that the number of nodes generated by backward B&B applied

to Q_R is at most one, if

$$\| b_i^* \| > \| (w_1; w_2) - (\ell_1; \ell_2) \| \tag{6.3.30}$$

for $i = 1, \ldots, n$. Since the columns of $(A; I)U$ form an RKZ reduced basis of $L_R(A)$, (6.3.19) implies

$$\| b_i^* \| \geq \lambda_1(L_R(A))/C_i, \tag{6.3.31}$$

so (6.3.30) holds, when

$$\lambda_1(L_R(A)) > C_i \, \| (w_1; w_2) - (\ell_1; \ell_2) \| \tag{6.3.32}$$

does for $i = 1, \ldots, n$, which is implied by

$$\lambda_1(L_R(A)) > \gamma_n \, \| (w_1; w_2) - (\ell_1; \ell_2) \| \, . \tag{6.3.33}$$

By Lemma 4 (6.3.33) is true for all, but at most a fraction of ϵ_R of $A \in G_{m,n}(M)$ if

$$M > \frac{(\lfloor 2\gamma_n \, \| (w_1; w_2) - (\ell_1; \ell_2) \| + 1 \rfloor)^{(m+n)/m}}{\epsilon_R^{1/m}}. \tag{6.3.34}$$

□

The proof of part (2) of Theorem 15 is along the same lines: now $b_1^*, \ldots, b_{n-m}^*$ is the Gram-Schmidt orthogonalization of the columns of B, which is an RKZ reduced basis of $L_N(A)$. Lemma 3, and the reducedness of B implies that the number of nodes generated by backward B&B applied to Q_N is at most one, if

$$\lambda_1(L_N(A)) > \gamma_{n-m} \, \| w_2 - \ell_2 \|, \tag{6.3.35}$$

and by Lemma 4 (6.3.35) is true for all, but at most a fraction of ϵ_N of $A \in G_{m,n}(M)$ if

$$M > \frac{(\lfloor 2\gamma_{n-m} \, \| w_2 - \ell_2 \| + 1 \rfloor)^{n/m}}{\epsilon_N^{1/m}}. \tag{6.3.36}$$

□

The proof of Theorem 16 is an almost verbatim copy, now using the estimate (6.3.20) to lower bound $\| b_i^* \|$. The proof of Theorem 17 uses the estimate (6.3.21) to lower bound $\| b_i^* \|$.

□

Proof of Proposition 3 Let $N(n,k)$ denote the number of integral points in the n-dimensional ball of radius k. In the previous proofs we used $(2k+1)^n$ as an upper bound for $N(n,k)$. The proof of Part (2) of Theorem 15 actually implies that when

$$M > \frac{(N(n, \lceil \gamma_{n-m} \parallel w_2 - \ell_2 \parallel \rceil)^{1/m}}{\epsilon_N^{1/m}}, \tag{6.3.37}$$

then for all, but at most a fraction of ϵ_N of $A \in G_{m,n}(M)$ backward B&B solves the nullspace reformulation of (6.1.12) at the root node.

We use Blichfeldt's upper bound [7]:

$$C_i \leq \frac{2}{\pi}\Gamma\left(\frac{i+4}{2}\right)^{2/i}, \tag{6.3.38}$$

to bound γ_{n-m} in (6.3.37), dynamic programming to exactly find the values of $N(n,k)$, and the values $\epsilon_N = 0.1,$ and $\epsilon_N = 0.01$ to obtain Table 6.1.

We note that in general $N(n,k)$ is hard to compute, or find good upper bounds for; however for small values of n and k a simple dynamic programming algorithm finds the exact value quickly.

□

Proof of Theorems 18, and 19 If $b_1^*, \ldots, b_r^*$ is an RKZ reduced basis of the lattice L, then by [32]

$$\parallel b_r^* \parallel \geq \frac{(\det L)^{1/r}}{\sqrt{r}}; \tag{6.3.39}$$

if it is an LLL reduced basis, then multiplying the inequalities

$$\parallel b_i^* \parallel \leq 2^{(r-i)/2} \parallel b_r^* \parallel \quad (i = 1, \ldots, r), \tag{6.3.40}$$

and using $\parallel b_1^* \parallel \ldots \parallel b_r^* \parallel = \det L$ gives

$$\parallel b_r^* \parallel \geq \frac{(\det L)^{1/r}}{2^{(r-1)/4}}. \tag{6.3.41}$$

Using (6.3.39) and (6.3.41) with (6.3.24) and

$$\det L_R(A) = \det(AA^T + I)^{1/2}, \det L_N(A) = \det(AA^T)^{1/2}/\gcd(A)$$

completes the proof, where the last equation follows from Proposition 1. □

6.4 Detailed Computational Results

$m = 4 \quad M = 100$		Original		Rangespace		Nullspace	
Instance	Feasible	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes
1	No	283.97	1,054,683	2.22	1,285	1.24	691
2	No	309.38	1,133,723	3.10	1,750	1.19	598
3	No	143.86	454,841	2.75	1,578	1.18	650
4	No	260.55	853,396	2.25	1,307	1.09	591
5	No	461.50	1,545,838	2.90	1,536	1.06	573
6	No	253.66	915,263	3.00	1,514	0.99	481
7	No	250.95	961,809	3.76	1,987	1.14	644
8	No	332.91	1,177,425	2.63	1,461	1.12	597
9	No	270.02	1,023,709	2.97	1,636	1.23	709
10	Yes	139.12	493,628	1.40	728	0.01	0
11	No	325.41	1,273,732	2.62	1,462	0.87	461
12	No	439.65	1,716,828	3.52	1,799	1.01	554
Averages		289.25	1,050,406.25	2.76	1503.58	1.01	545.75

Table 6.4: Results for the randomly generated 4 by 30 marketshare instances when $M = 100$

$m = 4 \quad M = 1000$		Original		Rangespace		Nullspace	
Instance	Feasible	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes
1	No	403.49	1,555,724	0.29	116	0.26	98
2	No	214.16	776,562	0.74	315	0.21	83
3	No	380.93	1,530,221	0.58	232	0.24	84
4	No	216.43	795,414	0.56	227	0.27	100
5	No	359.52	1,197,797	0.52	235	0.13	40
6	No	406.98	1,539,789	0.52	188	0.13	54
7	No	309.96	1,083,217	0.77	309	0.29	88
8	No	290.95	1,125,457	0.52	221	0.28	96
9	No	222.30	824,831	0.65	261	0.26	110
10	No	322.83	1,226,286	0.52	202	0.21	69
11	No	303.99	1,050,540	0.75	302	0.17	63
12	No	286.12	934,996	0.57	213	0.28	98
Averages		309.81	1,136,736.17	0.58	235.08	0.23	81.92

Table 6.5: Results for the randomly generated 4 by 30 marketshare instances when $M = 1000$

$m = 4 \quad M = 10000$		Original		Rangespace		Nullspace	
Instance	Feasible	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes
1	No	374.97	1,317,740	0.20	52	0.09	24
2	No	335.90	1,268,528	0.17	58	0.08	18
3	No	336.27	1,212,268	0.21	52	0.09	16
4	No	459.64	1,578,143	0.15	44	0.10	22
5	No	316.60	1,227,520	0.26	84	0.10	24
6	No	329.39	1,294,314	0.10	22	0.08	16
7	No	338.52	1,314,576	0.24	68	0.09	21
8	No	288.91	1,038,989	0.22	64	0.09	20
9	No	385.84	1,421,441	0.23	71	0.07	18
10	No	231.09	861,344	0.19	56	0.09	16
11	No	418.04	1,409,049	0.26	78	0.08	15
12	No	270.07	881,289	0.27	84	0.13	34
Averages		340.44	1,235,433.42	0.21	61.08	0.09	20.33

Table 6.6: Results for the randomly generated 4 by 30 marketshare instances when $M = 10000$

$m = 5 \quad M = 100$		Rangespace		Nullspace	
Instance	Feasible	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes
1	No	343.57	104,536	66.32	22,952
2	No	253.49	80,733	51.08	16,821
3	No	472.31	135,423	60.13	18,730
4	Yes	110.37	36,150	29.92	9,220
5	No	236.34	73,788	60.44	20,503
6	No	301.88	95,048	54.32	19,855
7	No	267.08	77,978	40.96	13,209
8	No	247.08	80,369	62.57	20,752
9	No	308.71	86,990	91.42	28,610
10	No	458.25	134,083	54.40	17,758
11	Yes	242.19	63,263	15.93	4,849
12	No	253.56	73,936	50.28	17,124
Averages		291.24	86,858.08	53.15	17,531.92

Table 6.7: Results for the randomly generated 5 by 40 marketshare instances when $M = 100$

$m = 5 \quad M = 1000$		Rangespace		Nullspace	
Instance	Feasible	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes
1	No	22.67	4,993	4.09	1,177
2	No	21.84	5,138	3.66	982
3	No	29.99	6,947	3.24	879
4	No	33.03	7,360	3.83	991
5	No	20.74	4,715	4.29	1,115
6	No	34.15	7,794	5.67	1,536
7	No	28.42	6,455	4.75	1,384
8	No	22.41	5,183	3.26	914
9	No	23.58	5,399	5.96	1,637
10	No	21.99	4,830	4.41	1,186
11	No	27.26	6,443	6.69	1,577
12	No	21.61	4,952	5.58	1,675
Averages		25.64	5,850.75	4.62	1,254.42

Table 6.8: Results for the randomly generated 5 by 40 marketshare instances when $M = 1000$

$m = 5 \quad M = 10000$		Rangespace		Nullspace	
Instance	Feasible	Time (in sec)	B&B Nodes	Time (in sec)	B&B Nodes
1	No	3.81	868	1.14	283
2	No	4.61	1000	0.84	174
3	No	2.84	649	1.29	310
4	No	4.96	1052	0.58	126
5	No	2.59	581	1.01	228
6	No	2.46	578	0.63	142
7	No	5.14	1058	0.77	194
8	No	2.32	466	1.00	226
9	No	1.93	467	0.91	196
10	No	7.06	1380	0.75	170
11	No	5.45	1158	0.86	191
12	No	5.08	1043	0.67	170
Averages		4.02	858.33	0.87	200.83

Table 6.9: Results for the randomly generated 5 by 40 marketshare instances when $M = 10000$

CHAPTER 7

On the Hardness of Subset Sum Problems by Ordinary Branch and Bound

7.1 Introduction and Main Result

Vasek Chvátal in [8] identified a class of instances of the zero-one knapsack problem which are difficult to solve by a class of algorithms that was called "recursive" (see [8] for the details) which use the combined powers of branch and bound, dynamic programming and rudimentary divisibility arguments.

Specifically, it was shown that the time required to solve the zero-one knapsack problem

$$\begin{array}{lll} max & ax & \\ st & ax & \leq \beta \\ & x \in \{0,1\}^n & \end{array} \tag{7.1.1}$$

where each a_j is chosen uniformly and independently at random from the integers between 1 and $10^{n/2}$, and $\beta = \lfloor \sum_{j=1}^{n} a_j/2 \rfloor$ is bounded from below by $2^{n/10}$ for the majority of the problems when n is large enough.

The problem in (7.1.1) is the optimization version of the subset sum problem which for a set of given positive integers $a_1, a_2, \ldots, a_n$ and a positive integer β tries to find a subset of the indices $I \subset \{1, 2, 3, \ldots, n\}$ such that the sum $\sum_{i \in I} a_i$ is closest to, but not exceeding, β. The feasibility version of the subset sum problem looks for a subset of the indices $I \subset \{1, 2, 3, \ldots, n\}$ such that the sum $\sum_{i \in I} a_i$ is equal to β. If there is such an index set, then the problem is feasible, otherwise it is infeasible. Recall

the feasibility version of the subset sum problem

$$\begin{aligned} ax &= \beta \\ x &\in \{0,1\}^n. \end{aligned} \qquad (SUB)$$

In this chapter, we show that an overwhelming majority of the subset sum instances of (SUB) are hard (i.e., requiring exponential amount of time in the size of the input) for ordinary B&B. We show that if the right-hand-side β is chosen to be $\lfloor r \sum_{j=1}^{n} a_j \rfloor$ for a constant r such that $0 < r < 1$, and each a_j is chosen uniformly and independently at random from the set $\{1, 2, 3, \ldots, M\}$ where $M := \lfloor 10^{n/2} \rfloor$, then the time to solve almost all of the instances of (SUB) using ordinary B&B is bounded from below by $2^{n^{1-\epsilon}}$ (where ϵ is a constant satisfying $0 < \epsilon < 1$) when n is large enough.

First, we state our theorem, and then prove it using some lemmas.

Theorem 20. *Fix r, ϵ such that $0 < r < 1$ and $0 < \epsilon < 1$. Let $b = \lfloor r \sum_{j=1}^{n} a_j \rfloor$ and each a_j be chosen uniformly and independently at random from the set $\{1, 2, 3, \ldots, M\}$ where $M := \lfloor 10^{n/2} \rfloor$. Then the probability that the instance of (SUB) generated requires the creation of at least $2^{n^{1-\epsilon}}$ B&B nodes (when we branch on the individual variables in any order) in the process of solving (SUB) goes to one as n goes to infinity.*

The way the theorem is proven is similar to the proofs of Theorem 1 and Theorem 2 in [8]. We start with fixing a constant k such that $0 < k < \epsilon < 1$. We show that almost all of the coefficients a_j satisfy the following two properties when n is large enough:

P1 $\sum_{i \in I} a_i \leq \frac{1}{n^k} \sum_{j=1}^{n} a_j$ whenever $|I| \leq n^{1-\epsilon}$,

P2 There is no set I such that $\sum_{i \in I} a_i = \lfloor r \sum_{j=1}^{n} a_j \rfloor$.

Lemma 5. *The probability that the coefficients a_j satisfy P1 and P2 goes to one as n goes to infinity.*

Proof of Lemma 5 It was shown in [8] that P2 is satisfied by the coefficients with probability going to one as n goes to infinity. In [8], r was chosen to be $1/2$, but the proof works well for any r such that $0 < r < 1$.

Now, we shall show that P1 is satisfied almost surely. If P1 is violated, then there exists an index set I such that $|I| \leq n^{1-\epsilon}$ and

$$\sum_{i \in I} a_i > \frac{1}{n^k} \sum_{j=1}^{n} a_j.$$

Since each $a_i \leq M$, we obtain

$$\sum_{j=1}^{n} a_j < (Mn^{1-\epsilon})n^k = Mn^{1+k-\epsilon}. \tag{7.1.2}$$

To find an explicit upper bound for the probability that P1 is violated, we use the following identity

$$\sum_{\substack{i \geq (p+t)n \\ i \text{ integer}}} \binom{n}{i} p^i (1-p)^{n-i} < e^{-2t^2 n} \tag{7.1.3}$$

which is valid for $0 < p < 1$ and $t \geq 0$. (7.1.2) implies that at least $(n - 2n^{1+k-\epsilon})$ of the coefficients a_i must be $\leq M/2$, for otherwise $\sum_{j=1}^{n} a_j \geq (2n^{1+k-\epsilon})M/2 = Mn^{1+k-\epsilon}$. Using (7.1.3) with $p = \lfloor M/2 \rfloor / M$ and $t = 1/2 - 2n^{k-\epsilon}$, we get

$$\begin{aligned} \sum_{\substack{i \geq (1/2+t)n \\ i \text{ integer}}} \binom{n}{i} p^i (1-p)^{n-i} &= \sum_{\substack{i \geq n - 2n^{1+k-\epsilon} \\ i \text{ integer}}} \binom{n}{i} p^i (1-p)^{n-i} \\ \leq \sum_{\substack{i \geq (p+t)n \\ i \text{ integer}}} \binom{n}{i} p^i (1-p)^{n-i} &< e^{-2t^2 n} = e^{-2n(1/2 - 2n^{k-\epsilon})^2} \end{aligned}$$

which goes to zero as n goes to infinity. □

Proof of Theorem 20

Lemma 6. *For positive coefficients a_j satisfying P1 and P2, if $b \in \left[\frac{1}{n^k}\sum_{j=1}^{n} a_j, \left(1 - \frac{1}{n^k}\right)\sum_{j=1}^{n} a_j\right]$ and if (SUB) is infeasible, then the ordinary B&B creates at least $2^{n^{1-\epsilon}}$ B&B nodes.*

Proof of Lemma 2 We shall show that none of the nodes in the B&B tree is pruned by infeasibility unless more than $n^{1-\epsilon}$ of the variables are fixed.

Assume that at most $n^{1-\epsilon}$ of the variables are fixed to 0 or 1. Let I be the set of indices of the fixed variables and $\overline{I}$ be the set of indices of the unfixed variables. Since coefficients a_j satisfy P1, we have $\sum_{i \in \overline{I}} a_i > \left(1 - \frac{1}{n^k}\right)\sum_{j=1}^{n} a_j$. By assigning fractional values to x_i $i \in \overline{I}$, we get a feasible solution to the LP relaxation of (SUB). □

Note that when n is large enough, $\lfloor r \sum_{j=1}^{n} a_j \rfloor$ is guaranteed to lie in the above interval completing

the proof of Theorem 20.

7.2 Summary of the Solvability of Subset Sum Problems by Branch and Bound

This result shows that an overwhelming majority of the subset sum problems (all but a vanishing proportion of the problems as n increases) are hard for ordinary B&B. On the other hand, our results from Chapter 5 show that by using a generalized B&B method which branches on constraints, almost all subset sum problems can be solved at the root node in polynomial time. The following is a summary of the results on the solvability of the subset sum problems using B&B. We fix r such that $0 < r < 1$. We assume that the coefficients of (SUB) are chosen from $\{1, \ldots, M\}$ for a large M, and let $\beta = \lfloor r \sum_{j=1}^{n} a_j \rfloor$.

(1) An overwhelming majority of the subset sum problems created as above are hard for ordinary B&B (branching on variables).

(2) Almost all subset sum problems (all but at most a proportion of $1/2^n$ of the problems as n increases) created as above are easy (at most one B&B node is created) for generalized B&B (branching on constraints).

(3) Almost all subset sum problems are easy for ordinary B&B if the problem is reformulated using the rangespace or the nullspace reformulation.

CHAPTER 8

Summary and Future Research

We considered the three fundamental inequalities of Lenstra, Lenstra and Lovász, which express the "shortness" and "near orthogonality" of an LLL reduced basis. We proved a common generalization: even though the inequalities were proven 27 years ago, this is the first unifying inequality that we are aware of.

For a knapsack problem, we showed that branching on a "near parallel" integral vector to the constraint vector creates a small number of branch and bound nodes which becomes 1 when the Euclidean norm of the constraint vector is sufficiently large.

We showed that for a low density subset sum problem, the infeasibility of "almost all" integer right hand sides can be proven by branching on a "near parallel" vector which can be found using "Diophantine approximation" or "rangespace reformulation".

We considered the classical branch and bound algorithm for integer programming, which is known to have exponential worst case complexity. We proved that it is surprisingly efficient on reformulated integer programs; precisely when the entries of the constraint matrix are from $\{1, \ldots, M\}$ for a large enough M, branch and bound solves almost all reformulated instances at the root node, and explored practical aspects of this result.

We showed that even though "almost all" low density subset sum problems are solvable in polynomial time using (generalized) branch and bound, a "majority" of the low density subset sum problems are "hard" for ordinary branch and bound.

Several future research directions can be followed based on the results of this dissertation.

(1) Complexity of the Reformulation Methods

Even though the reformulation methods are very efficient on the majority of the instances, their

complexities are not yet fully understood. It is an open question if one can solve the reformulated integer programming problem in polynomial time for a fixed number of variables.

It would also be interesting to design a class of integer programs on which the performance of the reformulations is provably bad.

(2) Classes of Problems on which the Reformulations Work

Some classes of integer problems, such as marketshare problems, are turned into easy-to-solve instances after they are reformulated. But there are certain classes of problems for which the reformulations do not seem to work well. It would be beneficial to run a thorough computational study on different problem classes and determine which ones benefit most from the reformulations. Another important question is: is there a certain criterion based on which one can decide whether or not a problem will be made easy for branch and bound after the reformulation?

(3) Successive Approximation

In Section 4.5, we approximate the constraint vector of a knapsack problem by a sequence of integral vectors. Using the successive approximation, for a low density subset sum problem, is it possible to prove the infeasibility of a higher fraction of the right hand sides at the root node by branch and bound?

Bibliography

[1] Karen Aardal, Robert E. Bixby, Cor A. J. Hurkens, Arjen K. Lenstra, and Job W. Smeltink. Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances. *INFORMS Journal on Computing*, 12(3):192–202, 2000.

[2] Karen Aardal, Cor A. J. Hurkens, and Arjen K. Lenstra. Solving a system of linear Diophantine equations with lower and upper bounds on the variables. *Mathematics of Operations Research*, 25(3):427–442, 2000.

[3] Karen Aardal and Arjen K. Lenstra. Hard equality constrained integer knapsacks. *Mathematics of Operations Research*, 29(3):724–738, 2004.

[4] Karen Aardal and Arjen K. Lenstra. Erratum to: Hard equality constrained integer knapsacks. *Mathematics of Operations Research*, 31(4):846, 2006.

[5] László Babai. On Lovász lattice reduction, and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.

[6] Alexander I. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19(4):769–779, 1994.

[7] Hans Frederik Blichfeldt. A new principle in the geometry of numbers, with some applications. *Transactions of the American Mathematical Society*, 15(3):227–235, 1914.

[8] Vašek Chvátal. Hard knapsack problems. *Operations Research*, 28(6):1402–1411, 1980.

[9] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM.

[10] William Cook, Thomas Rutherford, Herbert E. Scarf, and David F. Shallcross. An implementation of the generalized basis reduction algorithm for integer programming. *ORSA Journal on Computing*, 5(2):206–212, 1993.

[11] Gérard Cornuéjols and Milind Dawande. A class of hard small 0–1 programs. In *6th Conference on Integer Programming and Combinatorial Optimization*, volume 1412 of *Lecture notes in Computer Science*, pages 284–293. Springer-Verlag, 1998.

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

[13] Martin Dyer and Ravi Kannan. On Barvinok's algorithm for counting lattice points in fixed dimension. *Mathematics of Operations Research*, 22(3):545–549, 1997.

[14] Friedrich Eisenbrand and Sören Laue. A linear algorithm for integer programming in the plane. *Mathematical Programming*, 102(2):249–259, 2005.

[15] András Frank and Éva Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

[16] Alan Frieze. On the Lagarias-Odlyzko algorithm for the subset sum problem. *SIAM Journal on Computing*, 15:536–540, 1986.

[17] Merrick Furst and Ravi Kannan. Succinct certificates for almost all subset sum problems. *SIAM Journal on Computing*, 18:550 – 558, 1989.

[18] Liyan Gao and Yin Zhang. Computational experience with lenstra's algorithm. *Technical Report, Department of Computational and Applied Mathematics, Rice University*, 2002.

[19] Miroslav Karamanov Gérard Cornuéjols and Yanjun Li. Early estimates of the size of branch-and-bound trees. *INFORMS Journal on Computing*, 18(1):86–96.

[20] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition edition, 1993.

[21] Zonghao Gu, George L. Nemhauser, and Martin W. P. Savelsbergh. Lifted cover inequalities for 0–1 integer programs: Complexity. *INFORMS J. on Computing*, 11:117–123, 1998.

[22] Utz-Uwe Haus, Matthias Köppe, and Robert Weismantel. A primal all-integer algorithm based on irreducible solutions. *Mathematical Programming B*, 96:205–246, 2003.

[23] C Hermite. Sur l'introduction des variables continues dans la th'eorie des nombres. *J. Reine Angew. Math*, (41):191–216, 1851.

[24] Walfred Huyer and Arnold Neumaier. Integral approximation of rays and verification of feasibility. *Reliable Computing*, 10:195–207, 2004.

[25] Robert G. Jeroslow. Trivial integer programs unsolvable by branch-and-bound . *Mathematical Programming*, 6:105–109, 1974.

[26] Ravi Kannan. Algorithmic geometry of numbers. *Annual Review of Computer Science*, 2:231–267, 1987.

[27] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

[28] Ravi Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal of Computing*, 8(4):499–507, 1979.

[29] Richard Karp. Reducibility among combinatorial problems. In J.W. Thatcher R.E. Miller, editor, *Complexity of Computer Computations*. Plenum Press, 1972.

[30] A. Korkine and G. Zolotarev. Sur les formes quadratiques. *Mathematische Annalen*, 6:366–389, 1873.

[31] Bala Krishnamoorthy and Gábor Pataki. Column basis reduction and higher level decomposable knapsack problems. *Discrete Optimization*, 6(3):242–270, 2009.

[32] Jeffrey C. Lagarias, Hendrik W. Lenstra, and Claus P. Schnorr. Korkine-Zolotarev bases and successive minina of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.

[33] Jeffrey C. Lagarias and Andrew M. Odlyzko. Solving low-density subset sum problems. *Journal of ACM*, 32:229–246, 1985.

[34] A. H. Land and Alison G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[35] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[36] Hendrik W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[37] Jesus A. De Loera, Raymond Hemmecke, Jeremy Tauzer, and Ruriko Yoshida. Effective lattice point counting in rational convex polytopes. *Journal of Symbolic Computation*, 38(4):1273–1302, 2004.

[38] László Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*. Number 50 in SIAM CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1986.

[39] László Lovász and Herbert E. Scarf. The generalized basis reduction algorithm. *Mathematics of Operations Research*, 17:751–764, 1992.

[40] Jacques Martinet. *Perfect Lattices in Euclidean Spaces*. Springer-Verlag, Berlin, 2003.

[41] Sanjay Mehrotra and Zhifeng Li. On generalized branching methods for mixed integer programming. *Research Report, Department of Industrial Engineering, Northwestern University*, 2004.

[42] D. Micciancio. *Complexity of lattice problems: a cryptographic perspective*. Kluwer Academic Publishers, 2002.

[43] Phong Nguyen and Jacques Stern. Merkle-Hellman revisited: A cryptanalysis of the qu-vanstone cryptosystem based on group factorizations. In *Advances in Cryptology CRYPTO '97*, volume 1294 of *Lecture notes in Computer Science*, pages 198–212. Springer-Verlag, 1997.

[44] Claus P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–225, 1987.

[45] Claus P. Schnorr. A more efficient algorithm for lattice reduction. *Journal of Algorithms*, 9(1):47–62, 1988.

[46] Claus P. Schnorr. Block reduced lattice bases and successive minima. *Combinatorics, Probability, and Computing*, 3:507–533, 1994.

[47] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, United Kingdom, 1986.

[48] Victor Shoup. NTL: A Number Theory Library, 1990. http://www.shoup.net.