BALANCED INTEGER SOLUTIONS OF LINEAR EQUATIONS

KONSTANTINOS A. DRAZIOTIS

Abstract. We use lattice based methods in order to get an integer solution of the linear equation $a_1x_1+\cdots+a_nx_n = a_0$, which satisfies the bound constraints $|x_j| \leq X_j$. Further we study the corresponding homogeneous linear equation under constraints and finally we apply our method to Knapsack problem.

1. Introduction-Statement of results

Let $a_j \in \mathbb{Z} - \{0\},\ (0 \leq j \leq n)$. We consider the linear equation

$$f(x_1, ..., x_n) = \sum_{j=1}^{n} a_j x_j = a_0. \tag{1.1}$$

We are interested in the integer solutions of (1.1) under the constraints $|x_j| \leq X_j$, for some $X_j \in \mathbb{Z}_{>0}$. This is an NP-complete problem, but without the bound constraints is solved in polynomial time. This problem has some important applications in discrete optimization, in designing integrated circuits [1] and is also applied in Merkle-Hellman and the Chor-Rivest knapsack cryptography systems [10, 12]. Further we shall apply our method to the knapsack problem of density 1 and dimension ≤ 40.

In [1] the authors found a method for the solutions of equation (1.1) under the bound constraints $0 < x_j < X_j$. Their method contain two parts, one deterministic (application of LLL) having polynomial time complexity and the other heuristic. Further, in [15] the method of Rosser, starts with the matrix $M = (I_n, \mathbf{a}^T),\ \mathbf{a} = (a_1, ..., a_n)$ and a new matrix M' is obtained (using linear integer transformations) with $M' = (U, \mathbf{d}^T),\ \mathbf{d} = (0, 0, ..., \gcd(a_1, ..., a_n))$ and $U \in SL_n(\mathbb{Z})$. Then the general solution can be expressed in terms of the row vectors of M'. Many recent methods are based on the original idea of Rosser.

Also there is another approach to this problem which is based to the Closest Vector Problem (CVP). Let $\mathbf{y} = (y_1, y_2, ..., y_n)$ (the target vector) be a solution of (1.1), not necessarily small (for instance one can use Euclidean algorithm). Let L be the lattice generated by the solutions of the homogeneous equation $\sum_{j=1}^{n} a_j x_j = 0$. We solve the CVP instance $CVP(L, \mathbf{y})$ and we get the solution, say $\mathbf{t} = (t_1, t_2, .., t_n)$ (cvp vector). Then $\mathbf{x} = \{\text{target vector}\} - \{\text{cvp vector}\} = \mathbf{y} - \mathbf{t}$ is an integer solution of (1.1) and has small absolute value $||\mathbf{x}||$. For this approach for instance see [13]. Implementations of CVP can be found in fplll [14] and in Magma [3]. Here we use fplll. We shall provide some examples which compare the CVP approach with our algorithm and we shall conclude (based on experiments) that the better strategy is to combine the two methods (for $n \leq 60$). For large values of n, say $n \geq 80$, the

2000 *Mathematics Subject Classification.* Primary 11D04; Secondary 11Y50.
Key words and phrases. Linear Diophantine Equation, Lattice, LLL, CVP.

CVP-solver of fplll (and Magma) is (relatively) slow. So in this case our strategy is to apply our algorithm which is fast for large values of n. For instance, we made some tests for $n = 100$ and our algorithm terminated (with a small solution) in less than 5 seconds in all the examples while fplll needed at least three hours of running time in a 3Ghz Dual Core Pc (without getting any solution). Further, the algorithm given in [1] is also fast for large values of n (since it uses LLL) but in general it gives longest vectors than ours (see example (ii) in section 6).

Before we state our basic result we must set up some notation. Let $\{\mathbf{e}_1, ..., \mathbf{e}_{n+2}\}$ be the standard basis of $\mathbf{R}^{n+2}$, that is $\mathbf{e}_j = (..., \delta_{ij}, ...)$, where δ_{ij} is the delta of Kronecker and is located in the jth entry of the vector $\mathbf{e}_j$. We define the vectors

$$\mathbf{b}_j = \frac{1}{X_j}\mathbf{e}_j + a_j\mathbf{e}_{n+2} \quad (1 \leq j \leq n).$$

Let L be the lattice of $\mathbf{R}^{n+2}$ spanned by the vectors $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$. In matrix form L is spanned by the rows of the $n \times (n+2)$ matrix

$$B = [\mathbf{b}_1, .., \mathbf{b}_n] = \begin{bmatrix} \frac{1}{X_1} & 0 & \cdots & 0 & 0 & a_1 \\ 0 & \frac{1}{X_2} & \cdots & 0 & 0 & a_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \frac{1}{X_n} & 0 & a_n \end{bmatrix}.$$

(For reasons that later will become clear, we added the column with zeros). Let $\{\mathbf{b}'_1, ..., \mathbf{b}'_n\}$ be its LLL-reduced basis. We set $\mathbf{b}'_j = (b'_{j1}, ..., b'_{j,n+2})$. We consider the Gram-Schmidt orthogonalization process

$$\mathbf{b}_1^{'*} = \mathbf{b}'_1, \ \mathbf{b}_i^{'*} = \mathbf{b}'_i - \sum_{j=1}^{i-1} \mu_{ij}\mathbf{b}_j^{'*}, \tag{1.2}$$

where

$$\mu_{ij} = \frac{\mathbf{b}_i^{'} \cdot \mathbf{b}_j^{'*}}{B_j^2}, \quad B_j = ||\mathbf{b}_j^{'*}||.$$

Also if $\lfloor \cdot \rfloor$ is the floor function, we define $\lceil x \rfloor = \lfloor x + 0.5 \rfloor$ with $x \in \mathbf{R}$. That is the closest integer to x.

We shall prove the following Theorem.

Theorem 1.1. *Let* $\gcd(a_1, ..., a_n) = 1$. *If the following two assumptions hold*

A$_1$. $(a_n X_n)^2 + (a_j X_j)^2 < \dfrac{1}{2^{n+1}}(X_n X_j)^2, \ j = 1, 2, ..., n-1, \ X_j \in \mathbb{Z}_{>0}$,

A$_2$. $\left\lceil \dfrac{a_0}{B_n^2} \right\rfloor = a_0$,

then there is an integer solution $(x_1, ..., x_n)$ *of equation (1.1), such that*

$$|x_j| < c(n) \prod_{i=1}^{n} X_i, \quad j = 1, 2, ..., n, \ c(n) = \sqrt{3}(1.25)^{(n-1)/2}. \tag{1.3}$$

Further, we obtain this solution in polynomial time.

Assumption A$_1$ guarantees that $LLL-$ reduction will generate (some) solutions of the homogeneous linear equation (thus balanced multipliers for the $\gcd(a_1, ..., a_n)$).

Further, this assumption is crucial for the computation of integer solutions of equation (1.1). Assumption A_2 can be rewritten

$$a_0 \leq \frac{a_0}{B_n^2} + \frac{1}{2} < a_0 + 1,$$

thus,

$$B_n^2\big(a_0 - \frac{1}{2}\big) \leq a_0 < B_n^2\big(a_0 + \frac{1}{2}\big). \tag{1.4}$$

This assumption guarantees that our procedure will end up with a solution to equation (1.1). The type of lattice we used here it looks like the one used by Coppersmith in [4], which are used in order to attack the RSA cryptosystem. Although the bound is seemingly theoretical, in practice we get the desire solution (if there is any) $|x_j| \leq X_j$, without the strong assumption A_1. As far as I know, the methods given in the bibliography for the solution of the problem (1.1) under $|x_j| \leq X_j$ provide us with a small solution but there is not any theoretical result that guarantees that the method will work. Our method provides a theoretical result, in the sense that if assumptions A_1, A_2 are fulfilled, then the method shall work and also experiments show that the method will end up with solutions satisfying $|x_j| < X_j$, instead of inequality (1.3).

If $x_j \in \mathbb{N}$ and $a_0 = 0$ (i.e. the homogeneous version of our problem) then the problem of deciding if there is any integer solution is NP-complete [11]. We study a variant of this problem [see Lemma 4.1 and Proposition 4.2]. In fact we prove that if there is a Shortest Vector Problem (SVP) oracle then we can find an integer solution of $\sum_{j=1}^{n} a_j x_j = 0$, with $|x_j| < \sqrt{2}\max_{1\leq i\leq n}\{|a_i|\}$ $(1 \leq j \leq n)$. Finally, If $a_j > 0$ and we restrict the solutions $x_j \in \{0, 1\}$, then we have the $0-1$ Knapsack or subset sum problem. The decisional version is NP-complete [7]. This has many applications in public key cryptography. We shall apply our method in this problem

We give a brief outline of the paper. In the next section we give some preliminaries propositions about LLL and in section 3 some basic auxiliary results which we shall use for the proof of our Theorem in section 5. In section 4 we study the homogeneous linear equation and in section 6 we provide some examples. In the last section we use our approach to knapsack problem and give some examples.

2. Preliminaries on LLL

Our method uses LLL reduction algorithm, but with a different lattice than the one used in [1]. Lattices have many applications in cryptanalysis, for example [4, **?**]. Here we shall not provide analytically the theory of lattices and LLL algorithm. For instance, the reader can study [8, 17].

Definition 1. *A subset $L \subset \mathbf{R}^n$ is called a lattice if there exists linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ of $\mathbf{R}^n$ such that*

$$L = \Big\{ \sum_{j=1}^{k} \alpha_j \mathbf{b}_j : \alpha_j \in \mathbb{Z}, 1 \leq j \leq k \Big\} := L(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k).$$

The vectors $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ are called a lattice basis of L.

All the bases have the same number of elements, this common number is called *dimension* or *rank* of the lattice.

Lemma 2.1. *Let $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_k$ be an LLL−reduced basis of the lattice $L \subset \mathbf{R}^n$ and $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t$ linearly independent vectors in L. Then for all $j \leq t$ we have*

$$||\mathbf{b}_j||^2 \leq 2^{n-1} \max\{||\mathbf{x}_1||^2, ..., ||\mathbf{x}_t||^2\}.$$

Proof. [17, Theorem 7.10]. □

3. AUXILIARY RESULTS

Let L be the lattice of $\mathbf{R}^{n+2}$ spanned by the vectors $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ and let $\{\mathbf{b}'_1, ..., \mathbf{b}'_n\}$ be its LLL-reduced basis. We set

$$\mathbf{b}'_j = (b'_{j1}, ..., b'_{jn}, b'_{j,n+1}, b'_{j,n+2}).$$

Further, from Gramm-Schimdt orthogonalization process we get the vectors,

$$\mathbf{b}_i^{'*} = (\hat{b}^*_{i1}, ..., \hat{b}^*_{in}, \hat{b}^*_{i,n+1}, \hat{b}^*_{i,n+2}),\ (1 \leq i \leq n),$$

where $\hat{b}^*_{ij} = \dfrac{b^*_{ij}}{X_j}$ and $\hat{A} = [\hat{b}^*_{ij}]_{1\leq i,j\leq n}$. We shall prove the following Lemma.

Lemma 3.1.

$$\prod_{j=1}^{n} X_j = \frac{1}{|\det \hat{A}|}.$$

Proof. Let B, B' be the matrices $[\mathbf{b}_1, ..., \mathbf{b}_n]^T$, $[\mathbf{b}'_1, ..., \mathbf{b}'_n]^T$, respectively (we work row-wise). Then there is a matrix $U = [\lambda_{ij}]_{1\leq i,j\leq n} \in SL_n(\mathbb{Z})$ such that $B' = UB$. Indeed, we consider the $n \times n$ matrix $U_1 = \operatorname{diag}\big(\frac{1}{X_1}, ..., \frac{1}{X_n}\big)$. We apply LLL to the rows of B, thus we get B'. The linear changes and swaps of rows made by LLL in B also applied to U_1, since $B = [U_1, \mathbf{a}^T]$ where, $\mathbf{a} = (a_1, ..., a_n)$. Then we get a new $n \times n$ matrix U_2 which is equals to $\big[\frac{\lambda_{ij}}{X_j}\big]_{1\leq i,j,\leq n}$. So

$$|\det U_1| = |\det U_2| = \frac{1}{\prod_{j=1}^n X_j}.$$

Also,

$$|\det U_2| = \frac{1}{\prod_{j=1}^n X_j} |\det U|,$$

thus $|\det U| = 1$. Now we shall prove that

$$|\det[b^*_{ij}]| = |\det[\lambda_{ij}]| = 1. \tag{3.1}$$

Indeed, from $B' = UB$ we get

$$\mathbf{b}'_i = \sum_{j=1}^{n} \lambda_{ij} \mathbf{b}_j = \big(b'_{i1}, ..., b'_{in}, *, *\big),$$

where λ_{ij}, b'_{ij} are related by $\lambda_{ij} = X_j b'_{ij}$. From relations (1.2) we get

$$\hat{b}^*_{1j} = b'_{1j},\ \hat{b}^*_{2j} = b'_{2j} - \mu_{21} b'_{1j}, ...(1 \leq j \leq n)$$

thus, multiplying by X_j we get

$$b^*_{1j} = \lambda_{1j},\ b^*_{2j} = \lambda_{2j} - \mu_{21}\lambda_{1j}, ...(1 \leq j \leq n),$$

so (3.1) is proved. We conclude that

$$|\det \hat{A}| = \frac{1}{\prod_{j=1}^n X_j} |\det[b^*_{ij}]| = \frac{1}{\prod_{j=1}^n X_j}.$$

□

Also we get the following Proposition.

Proposition 3.2. *Assume that*

$$(a_nX_n)^2 + (a_jX_j)^2 < \frac{1}{2^{n+1}}(X_nX_j)^2, \tag{3.2}$$

for all j such that $1 \le j \le n-1$. Then

$$b'_{j,n+1} = b'_{j,n+2} = 0 \text{ for } 1 \le j \le n-1.$$

Further,

$$b'_{n,n+1} = 0,\ |b'_{n,n+2}| = 1.$$

Proof. We set $\mathbf{r}_j = a_n\mathbf{b}_j - a_j\mathbf{b}_n$. These are independent in L, so from Lemma 2.1 we get

$$||\mathbf{b}'_j||^2 \le 2^{n+1} \max_{1\le j\le n-1}\{||\mathbf{r}_j||^2\} \le 2^{n+1} \max_{1\le j\le n-1}\Big\{\frac{a_n^2}{X_j^2} + \frac{a_j^2}{X_n^2}\Big\}.$$

Since

$$(a_nX_n)^2 + (a_jX_j)^2 < \frac{1}{2^{n+1}}(X_nX_j)^2$$

we get

$$2^{n+1}\Big(\frac{a_n^2}{X_j^2} + \frac{a_j^2}{X_n^2}\Big) < 1 \ (1 \le j \le n-1).$$

Thus $||\mathbf{b}'_j||^2 < 1$. Now we assume that $b'_{j,r} \neq 0$ for $r \in \{n+1, n+2\}$ and $1 \le j \le n-1$. Since $b'_{j,n}, b'_{j,n+1} \in \mathbb{Z}$ we get $||\mathbf{b}'_j||^2 \ge 1$, contradicting to the previous inequality. The first part of the Proposition follows.

Since $LLL-$ algorithm makes only linear transformation of the form

$$\mathbf{b}_j \leftrightarrow \mathbf{b}_i \text{ or } \mathbf{b}_j \leftarrow \mathbf{b}_j - r\mathbf{b}_i,\ r \in \mathbb{Z},\ j > i$$

and the $(n+1)$th column consists only from zeros, we get $b'_{n,n+1} = 0$. Further, the gcd of the last column remain the same in every step of the $LLL-$process. Since we assumed that $\gcd(a_1, ..., a_n) = 1$ we get

$$1 = \gcd(a_1, ..., a_n) = \gcd(b'_{1,n+2}, b'_{2,n+2}, ..., b'_{n,n+2}) = \gcd(0, 0, ..., 0, b'_{n,n+2}) = |b'_{n,n+2}|.$$

□

Remark 3.3. From assumption (3.2) we get

$$|a_n| < \frac{1}{2^{(n+1)/2}}X_j,\ (1 \le j \le n-1) \text{ and } |a_j| < \frac{1}{2^{(n+1)/2}}X_n.$$

We shall show that, under the assumption (3.2), we have $\hat{b}^*_{i,n+1} = \hat{b}^*_{i,n+2} = 0$ for $1 \le i \le n-1$ and $\hat{b}^*_{n,n+1} = 0,\ \hat{b}^*_{n,n+2} = \pm 1$. Indeed, it is easy for $\mathbf{b}_1^{'*}$ since it is equal to $\mathbf{b}'_1 = (..., 0, 0)$. Also for

$$\mathbf{b}_2^{'*} = \mathbf{b}'_2 - \mu_{21}\mathbf{b}_1^{'*} = (*, *, ..., 0, 0) - \mu_{21}(*, *, ..., 0, 0) = (*, *, ..., 0, 0).$$

Inductively we can show that $\mathbf{b}_i^{'*} = (*, *, ..., 0, 0)$, for $1 \le i \le n-1$. For

$$\mathbf{b}_n^{'*} = (*, *, ..., 0, \pm 1) - \mu_{n2}(*, *, ..., 0, 0) - \cdots - \mu_{n,n-1}(*, *, ..., 0, 0) = (*, *, ..., 0, \pm 1).$$

So we proved the following.

Corollary 3.4. *Under the assumption (3.2) we get*

$$\mathbf{b}_i^{'*} = (\hat{b}_{i1}^*, ..., \hat{b}_{in}^*, 0, 0) \ (1 \leq i \leq n-1) \ \textit{and} \ \hat{b}_{n,n+1}^* = 0, \ |\hat{b}_{n,n+2}^*| = 1.$$

4. THE HOMOGENEOUS CASE

We get the following Lemma concerning the integer solutions of the corresponding homogeneous linear equation.

Lemma 4.1. *Under the assumption A_1, we can find an integer solution $(x_1, ..., x_n)$ of the homogeneous linear equation $\sum_{j=1}^n a_j x_j = 0$, with $|x_j| < X_j$, in polynomial time.*

Proof. Every vector of the $LLL-$reduced basis of the lattice L is of the form

$$\Big(\frac{\lambda_1}{X_1}, ..., \frac{\lambda_n}{X_n}, \beta, \sum_{j=1}^n \lambda_j a_j - \beta a_0\Big),$$

with $\lambda_j, \beta \in \mathbb{Z}$ (note also that the two last coordinates are integers). From the previous Proposition we get $\beta = 0$ and $\sum_{j=1}^n \lambda_j a_j = 0$ for the first $n-1$ vectors of the $LLL-$reduced basis. Thus, the $n-$tuples

$$\big\{\lambda_{ij} : 1 \leq j \leq n\big\}$$

for $i = 1, 2, ..., n-1$, are solutions of the equation $a_1 x_1 + \cdots + a_n x_n = 0$. Where λ_{ij} are as in Lemma 3.1. Moreover, since $||\mathbf{b}_i'||^2 < 1$ for each $i = 1, 2, ..., n-1$, we conclude that $|\lambda_{ij}| < X_j$ for each $j \in \{1, 2, ..., n-1\}$. □

In the previous Lemma we used LLL reduction. It is known that LLL runs in polynomial time and it provides vectors that are exponentially longer than the shortest vectors. Assume now that we have a SVP oracle. That is a probabilistic algorithm which computes with high probability a shortest vector of a lattice in polynomial time. LLL algorithm behave as a SVP oracle for small dimensions (≤ 40) and further $BKZ-20$ reduction algorithm for dimensions ≤ 50 (see [6], Figure 1) (remark that we do not have any proof that BKZ runs in polynomial time, but in practice for dimensions ≤ 50 is fast) . Assuming the existence of a SVP oracle, let $\mathbf{b}$, a shortest vector of our lattice, which has the form

$$\Big(\frac{\lambda_1}{X_1}, ..., \frac{\lambda_n}{X_n}, \beta, \sum_{j=1}^n \lambda_j a_j - \beta a_0\Big), \ \lambda_j \in \mathbb{Z}.$$

Assume that there is some j_0 with $|a_{j_0}| \neq |a_n|$ and set

$$X = X_1 = \cdots = X_n = \sqrt{2} \max_{1 \leq j \leq n} |a_j|.$$

Since $||\mathbf{b}|| \leq ||\mathbf{r}_j||, \ (1 \leq j \leq n-1)$ we get

$$||\mathbf{b}||^2 \leq ||\mathbf{r}_{j_0}||^2 = \frac{a_n^2 + a_{j_0}^2}{X^2} < 1.$$

Thus necessarily, $\beta = 0$, since if not $||\mathbf{b}|| \geq 1$. So $\beta = 0$ and

$$\Big|\frac{\lambda_j}{X}\Big| < 1, \ \sum_{j=1}^n \lambda_j a_j = 0.$$

So we proved the following.

Proposition 4.2. *Assume that there is a SVP oracle and at least one* $|a_j| \neq |a_n|$. *Then we can compute in polynomial time an integer solution* $(x_1, ..., x_n)$ *of the homogeneous equation* $\sum_{j=1}^{n} a_j x_j = 0$, *with* $|x_j| < \sqrt{2} \max\limits_{1 \leq i \leq n} |a_i|$.

5. PROOF OF THE THEOREM

We set $\mathbf{b}_{n+1} = \mathbf{e}_{n+1} - a_0 \mathbf{e}_{n+2}$. As usual $L = L(\mathbf{b}_1, ..., \mathbf{b}_n)$ and $L' = L'(\mathbf{b}'_1, ..., \mathbf{b}'_n)$ be the $LLL-$reduced basis of L. We consider the lattice $\hat{L}$ generated by the set $\{\mathbf{b}'_1, ..., \mathbf{b}'_n, \mathbf{b}_{n+1}\}$. We apply size reduction to $\hat{L}$ that is

$$row(n+1) \leftarrow row(n+1) - \lceil \mu_{n+1,n} \rfloor row(n). \tag{5.1}$$

Note that at this stage $\mu_{n+1,j} = 0$, for $1 \leq j \leq n-1$. Also, from Proposition 3.2 we get

$$\mu_{n+1,n} = \frac{1}{B_n^2}(\mathbf{b}_{n+1} \cdot \mathbf{b}'_n) = \frac{1}{B_n^2}\big((0, 0, ..., 1, -a_0) \cdot (*, *, ..., 0, \pm 1)\big) = \frac{\pm a_0}{B_n^2}.$$

From assumption A_2, we get $\lceil \mu_{n+1,n} \rfloor = a_0$ if $b'_{n,n+2} = -1$ and $\lceil \mu_{n+1,n} \rfloor = -a_0$ if $b'_{n,n+2} = 1$. That is always

$$b'_{n+1,n+2} = -a_0 - \lceil \mu_{n+1,n} \rfloor \cdot b'_{n,n+2} = 0.$$

We continue with

$$row(n+1) \leftarrow row(n+1) - \lceil \mu_{n+1,j} \rfloor row(j) \text{ for } j = 1, 2, ..., n-1.$$

Let $\mathbf{b}'_{n+1}$ the size reduced (last) row. Then $\mathbf{b}'_{n+1}$ has the form $(\hat{x}_1, ..., \hat{x}_n, 1, 0)$. The new basis $\{\mathbf{b}'_1, ..., \mathbf{b}'_n, \mathbf{b}'_{n+1}\}$ has the property $\mu_{n+1,j} < 1/2$ for $1 \leq j \leq n$. Since

$$\mu_{n+1,j} = \frac{\mathbf{b}'_{n+1} \cdot \mathbf{b}'^{*}_{j}}{B_j^2}, \quad B_j = ||\mathbf{b}'^{*}_{j}||$$

we get (using Corollary (3.4)) the system,

$$\hat{x}_1 \hat{b}^*_{j1} + \cdots + \hat{x}_n \hat{b}^*_{jn} = \varepsilon_j, \ 1 \leq j \leq n \tag{5.2}$$

where

$$\hat{b}^*_{ij} = \frac{b^*_{ij}}{X_j} \text{ and } |\varepsilon_j| = |\mu_{n+1,j}| \ B_j^2 < \frac{B_j^2}{2}.$$

Since $B_j^2 = ||\mathbf{b}'^{*}_{j}||^2 < ||\mathbf{b}'_j||^2$, we get

$$|\varepsilon_j| < \frac{||\mathbf{b}'_j||^2}{2} < \frac{1}{2} \ \ (1 \leq j \leq n-1).$$

For the case $j = n$ we get

$$|\varepsilon_n| = |\mu_{n+1,n}| \ B_n^2 < \frac{1}{2} B_n^2 < \frac{1}{2} 2 = 1. \tag{5.3}$$

We used that $B_n^2 < 2$. Indeed, from the first part of inequality (1.4) we get

$$B_n^2 < \frac{2a_0}{2a_0 - 1}.$$

Since $a_0 \in \mathbb{Z} - \{0\}$, we get

$$0 < \frac{2a_0}{2a_0 - 1} < 2,$$

thus $B_n^2 < 2$. Let $\hat{A}$ be the matrix of the system, that is $\hat{A} = [\hat{b}_{ij}^*]_{1\leq i,j\leq n}$. If we substitute the jth column of $\hat{A}$ with the column vector $(\varepsilon_1, ..., \varepsilon_n)^T$ we get the matrix $\hat{A}_j$. Then

$$|\hat{x}_j| = \Big|\frac{\det \hat{A}_j}{\det \hat{A}}\Big|.$$

From Lemma (3.1), the system (5.2) has determinant $|\det \hat{A}| = \dfrac{1}{\prod_{i=1}^n X_i}$. Thus

$$|\hat{x}_j| = |\det \hat{A}_j| \prod_{i=1}^n X_i. \tag{5.4}$$

We apply Hadamard inequality to $\det \hat{A}_j$. We shall get

$$|\det \hat{A}_j|^2 < \prod_{i=1}^n ||row[\hat{A}_j]_i||^2. \tag{5.5}$$

The ith row of the matrix $\hat{A}_j$ is

$$row[\hat{A}_j]_i = \Big(\frac{b_{i1}^*}{X_1}, ..., \varepsilon_i, ..., \frac{b_{in}^*}{X_n}\Big),$$

where ε_i is in the $j-$entry. The square of its length for $j \neq n$ is

$$||row[\hat{A}_j]_i||^2 < B_i^2 + \varepsilon_i^2 < 1 + \varepsilon_i^2 < 1 + \frac{1}{4} = \frac{5}{4} = 1.25.$$

For the case $j = n$ we get

$$||row[\hat{A}_j]_n||^2 = \Big(\frac{b_{n1}^*}{X_1}\Big)^2 + \cdots + \Big(\frac{b_{n,n-1}^*}{X_{n-1}}\Big)^2 + \varepsilon_n^2 < B_n^2 + 1 < 3.$$

Thus, from inequality (5.5) we get

$$|\det \hat{A}_j| < \sqrt{3}(1.25)^{(n-1)/2} = c(n).$$

So from relation (5.4) we get

$$|\hat{x}_j| < c(n) \prod_{i=1}^n X_i, \quad j = 1, 2, ..., n.$$

Since all the previous computations can be done in polynomial time (LLL and size reduction), the Theorem follows.

Remark 5.1. In the case where $a_0 < 0$, instead of relation (5.3) we have the better inequality $|\varepsilon_n| < 1/2$, since $B_n^2 < 1$. Thus $c(n) = (1.25)^{n/2}$.

6. EXAMPLES

The first example is given only to explicitly show how we apply our method.
(**i**). Let

$$84 \cdot 10^5 x_1 + 4 \cdot 10^6 x_2 + 15688 x_3 + 6720 x_4 + 15 x_5 = 371065262.$$

This is example 1 of [1] and they get the solution $\mathbf{x} = (36, 17, 39, 8, -22)$, with $||\mathbf{x}|| \simeq 60.44$. Assumption A_1 is fulfilled if

$$\max_{1\leq j\leq 4} |a_j| < \frac{1}{8} X_5, \ |a_5| < \frac{1}{8} \max_{1\leq j\leq 5} |X_j|.$$

So it is enough to choose

$$X = X_1 = \cdots = X_5 = 8 \cdot 84 \cdot 10^5 + 1.$$

We consider the matrix

$$M = \begin{bmatrix} \frac{1}{67200001} & 0 & 0 & 0 & 0 & 0 & 8400000 \\ 0 & \frac{1}{67200001} & 0 & 0 & 0 & 0 & 4000000 \\ 0 & 0 & \frac{1}{67200001} & 0 & 0 & 0 & 15688 \\ 0 & 0 & 0 & \frac{1}{67200001} & 0 & 0 & 6720 \\ 0 & 0 & 0 & 0 & \frac{1}{67200001} & 0 & 15 \end{bmatrix}$$

Applying LLL to the rows of M we get

$$M_{LLL} = \begin{bmatrix} -\frac{10}{67200001} & \frac{21}{67200001} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{15}{67200001} & -\frac{35}{67200001} & -\frac{8}{67200001} & 0 & 0 \\ \frac{1}{67200001} & -\frac{2}{67200001} & -\frac{25}{67200001} & -\frac{1}{67200001} & -\frac{72}{67200001} & 0 & 0 \\ \frac{5}{67200001} & -\frac{10}{67200001} & -\frac{95}{67200001} & -\frac{76}{67200001} & \frac{72}{67200001} & 0 & 0 \\ \frac{2}{67200001} & -\frac{4}{67200001} & -\frac{42}{67200001} & -\frac{21}{67200001} & \frac{1}{67200001} & 0 & -1 \end{bmatrix}$$

Then applying size reduction to the lattice $\hat{L}$ generated by the set $\{\mathbf{b}'_1, ..., \mathbf{b}'_5, \mathbf{b}_6\}$, where $\mathbf{b}_6 = (0, 0, 0, 0, 0, 1, -a_0)$, we get

$$\hat{M}_{LLL} = \begin{bmatrix} -\frac{10}{67200001} & \frac{21}{67200001} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{15}{67200001} & -\frac{35}{67200001} & -\frac{8}{67200001} & 0 & 0 \\ \frac{1}{67200001} & -\frac{2}{67200001} & -\frac{25}{67200001} & -\frac{1}{67200001} & -\frac{72}{67200001} & 0 & 0 \\ \frac{5}{67200001} & -\frac{10}{67200001} & -\frac{95}{67200001} & -\frac{76}{67200001} & \frac{72}{67200001} & 0 & 0 \\ \frac{2}{67200001} & -\frac{4}{67200001} & -\frac{42}{67200001} & -\frac{21}{67200001} & \frac{1}{67200001} & 0 & -1 \\ \frac{36}{67200001} & \frac{17}{67200001} & \frac{39}{67200001} & \frac{8}{67200001} & -\frac{2}{6109091} & 1 & 0 \end{bmatrix}$$

We take the 6th row and multiply each entry with X, then we shall get the vector $\mathbf{x}$. If we want the coordinates of the solution vector to satisfy

$$|x_1| < 30, |x_j| < 50 \ (2 \leq j \leq 5),$$

then taking $X_1 = 30, X_2 = X_3 = X_4 = X_5 = 50$ we get the following solution $\mathbf{x} = (26, 38, 39, 8, -22)$. This solution has absolute value 64.72. Notice that is larger (with respect to euclidean length) than the previous solution, but it has the advantage that satisfies our constraints.

(**ii**). We consider now $n = 50$.

$\{a_j\}_j = [872934629013064, 362643350651979, 231593889792433, 1084529488472651, 152647947850799, 739407904067188, 1078361055147110, 522287723336618, 1048278073142822, 71464720981315, 1026144865997912, 401128969656441, 1104125375426692, 223040948030783, 259134135114376, 477165086702863, 693696459173357, 956101007737750, 1076391779531258, 887808907972169, 154289043341408, 1123813906929138, 100640784930380, 1028038257417354, 126747913149526, 345001039716371, 173180910604612, 376756743710801, 462057825850822, 105084485099476, 193285152829384, 663950233902816, 1005024177016821, 350981819196027, 1049577315489835, 455051495653072, 1014366278972062, 905067265314795, 972603957926899, 1110054606397627, 768533772552959, 798515502008744, 705587377794293, 64248048456242, 771519628719865, 190006526706907, 481482852515889, 916067763534188, 768875611228651, 666640039086558]$
$a_0 = 17297404087862459$

Using our algorithm with $X_j = 3$ $(1 \leq j \leq n)$ we get (using Sage [16])

$\mathbf{x}_1 = [1, 0, 1, 2, 0, 0, 1, -1, 0, 0, 1, 2, -1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 2, 2, 0, 0, -1, 1, 1, -1, -1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0]$

with euclidean length 6.324. It took less than 5 seconds in order to find it. This solution satisfies our bound $|x_j| \leq X_j = 3$. Using the algorithm of [1] we get

$$\mathbf{x}_2 = [0,0,0,1,1,2,-1,3,0,1,2,1,2,2,0,0,0,0,1,-1,0,-1,0,2,1,0,0,1,2,\\ 0,-1,0,1,0,0,0,1,0,1,1,0,0,0,0,1,0,0,1,1,0]$$

which has euclidean length 7.141. Now using CVP method (as implemented in fplll) we manage to get the vector

$$\mathbf{x}_3 = [0,0,0,1,0,1,1,0,1,0,2,0,1,0,0,0,1,1,1,0,0,0,0,1,\\ 0,0,0,1,0,0,0,0,1,-1,1,0,0,1,0,0,0,2,1,0,0,0,1,0,0,1]$$

which has norm 5. It took almost 4 minutes in order to compute it. We worked as follows. Let L be the lattice generated by the homogeneous equation $\sum_{j=1}^{n} a_j x_j = 0$. Then we choose as target vector, a vector $\mathbf{t}$ which is a solution of $\sum_{j=1}^{n} a_j x_j = a_0$. If $\mathbf{c}$ is the output of the CVP instance $CVP(L, \mathbf{t})$, then a small solution is given by $\mathbf{x} = \mathbf{t} - \mathbf{c}$. We noticed that, if the target vector has large length then fplll provide us with a large solution $\mathbf{x}$ (but smaller than the length of target vector). That is the CVP solvers are "sensitive" to the choice of the target vector. In order to get the previous solution $\mathbf{x}_3$ we used as target vector, the vector $\mathbf{x}_1$ from the application of our method. So it seems that a nice strategy in order to get a small solution is to combine the CVP solvers and our algorithm (which shall give us the target vector). In case we have large n, say $n \geq 80$ then the CVP solver of fplll(and Magma)[1] is slow. Thus, we conclude (at least experimentally) that for large dimensions is better to use our algorithm and for smaller say $n < 80$ a combination of CVP and our method.

7. An application to Knapsack problem

The subset sum or knapsack problem is the following. Given a list of n positive integers $\{a_1, ..., a_n\}$ and an integer s such that $\max\{a_i\}_i \leq s \leq \sum_{i=1}^{n} a_i$ find a binary vector $\mathbf{x} = (x_i)_i$ such that $\sum_{i=1}^{n} x_i a_i = s$. The decisional version is known to be NP-complete [7]. The variant, multiple knapsack problem is used in many loading and scheduling problems in operational research.

We use the following lattice

$$B = [\mathbf{b}_1, .., \mathbf{b}_{n+1}] = \begin{bmatrix} \frac{N_1}{X_1} & 0 & \cdots & 0 & 0 & N_1 a_1 \\ 0 & \frac{N_1}{X_2} & \cdots & 0 & 0 & N_1 a_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \frac{N_1}{X_n} & 0 & N_1 a_n \\ 0 & 0 & \cdots & 0 & N_2 & -N_1 s \end{bmatrix},$$

for some positive integers N_1, N_2. When we apply $LLL-$ algorithm to the previous lattice we get vectors of the form

$$\Big(\frac{\lambda_1 N_1}{X_1}, ..., \frac{\lambda_n N_1}{X_n}, \beta N_2, N_1\big(\sum_{j=1}^{n} \lambda_j a_j - \beta s\big)\Big).$$

This is because LLL algorithm uses transformations of the form

$$\mathbf{b}_j \leftrightarrow \mathbf{b}_i \text{ or } \mathbf{b}_j \leftarrow \mathbf{b}_j - r\mathbf{b}_i,\ r \in \mathbb{Z},\ j > i.$$

Since we expect small vectors from LLL, we probable get vectors of the form

$$\Big(\frac{\lambda_1 N_1}{X_1}, ..., \frac{\lambda_n N_1}{X_n}, N_2, 0\Big)$$

[1]For a detailed account how these solvers work see [9].

with $\lambda_j \in \{0,1\}$ (in fact for small dimensions $n \leq 40$ is very probable, at least experimentally, to get vectors of the previous form). Then a solution of the knapsack is $(\lambda_1, ..., \lambda_n)$. We shall fix N_1, N_2 and we randomly choose $X_1, ..., X_n \in \{1, 2, ..., k\}$ with k say ≤ 10.

After making many experiments with dimension $n \leq 40$ we concluded that whenever the algorithm of Coster et al. [5] is working is faster than ours. But there are cases where the algorithm of [5] is not working as we will see below. For large values of n say $n > 40$ and density very close to 1 our algorithm is slow. All the examples below have density very close to 1.

Example 1 ($n = 20$). Let

$$\mathbf{a} = [231578, 90066, 426782, 989541, 428396, 861588, 366246, 430412, 329226, 299869,$$

$$179689, 288142, 916676, 447222, 1040519, 271141, 652751, 132316, 548527, 907547]$$

and $s = 6507929$. Using the algorithm of Coster et al. with $N = 15$ we did not manage to get a solution. Our algorithm with $N_1 = 90, N_2 = 80$ and using random denominators from the set $\{1, 2, 3, 4, 5\}$, in the $38th$ round we got the solution $\mathbf{x} = (0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0)$.

Example 2 ($n = 30$). We set

$$\mathbf{a} = [257957069, 211449890, 453588748, 460393904, 806269638, 965676997, 722998227,$$

$$557173347, 544414881, 605777707, 308224438, 609694552, 614806334, 86201849,$$

$$3033849, 54567875, 749134183, 136657534, 339166263, 622170807, 339856371, 565613209,$$

$$66643022, 732672773, 874884984, 522967114, 168924289, 405266804, 946333809, 879669424]$$

and $s = 6835888107$. Again using Coster et al, with $N = 10$ we did not manage to get any solution. Using our algorithm with $N_1 = 250, N_2 = 230$ and using random denominators from the set $\{1, 2, 3\}$, we got

$$\mathbf{x} = (1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0)$$

in the 439th round. We got this results in some minutes.

Example 3 ($n = 35$). We set

$$[25757712619, 1703301249, 29787913497, 12224812308, 10842851796,$$

$$12515371588, 32028450775, 34098294238, 3343156310, 27995252025, 8010200960,$$

$$15769634246, 23243451953, 18423819032, 4905368619, 18951710032, 18461896729,$$

$$31018788743, 33944716414, 30577978749, 19433865371, 21833994553,$$

$$16822791334, 9873829642, 32574703247, 16993191260, 34144724289, 6412642125,$$

$$15206763392, 17781019093, 29173151234, 25267831499, 32387438669, 18801581598, 19492385639].$$

and $s = 206027365036$. Again using Coster et al, with various values of $N \in \{5, 10, 15\}$ we did not get any solution. Using our algorithm with $N_1 = 25000, N_2 = 20000$ and using random denominators from the set $\{1, 2, 3, 4\}$, we got

$$\mathbf{x} = (1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0)$$

after 163.2 min.

References

[1] Aardal, Karen; Hurkens, Cor.; Lenstra, Arjen K.; Solving a linear Diophantine equation with lower and upper bounds on the variables. Integer programming and combinatorial optimization p.229-242, Lecture Notes in Comput. Sci., **1412**, Springer, Berlin, 1998.

[2] Beihoffer, Dale; Hendry, Jemimah; Nijenhuis, Albert; Wagon, Stan; Faster algorithms for Frobenius numbers. Electron. J. Combin. **12** (2005), Research Paper 27, 38 pp.

[3] Bosma, Wieb, Cannon, John and Playoust, Catherine; The Magma algebra system. I. The user language, J. Symbolic Comput.Vol. **24** (1997).

[4] Coppersmith, D.: Finding small solutions to small degree polynomials. Lecture Notes in Computer Science **2146** (2001) p.20-31.

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

[6] Gama N. and Nguyen P.Q., Predicting Lattice reduction, Eurocrypt 2008, LNCS **4965**, p.31-51 (2008)

[7] Garey, M.R.;Johnson, D.S.; Computers and intractability : A guide to the theory of NP-completeness. W.H.Freeman and Company, NY (1979).

[8] Galbraith, S.; Mathematics of Public Key Cryptography, Version 1.1, December 1, 2011, http://www.math.auckland.ac.nz/ sgal018/crypto-book/crypto-book.html.

[9] Hanrot G., Pujol X. and D.Stehle. Algorithms for the Shortest and Closest Lattice Vector Problems. IWCC'11.

[10] Lenstra, H.W.; On the Chor-Rivest knapsack cryptosystem; Journal of Cryptology, Volume **3**, Number 3 (1991), p.149-155.

[11] Lueker, G.S; Two NP-complete problems in nonnegative integer programming, report **178** CSL, Princeton University (1975).

[12] Merkle, R., Hellman, M.; Hiding information and Signatures in trapdoor cryptosystem, IEEE Trans.Inf.Theory **IT-24**(1978), p.525-530.

[13] Nguyen, P.Q., Stern, J., The two faces of Lattices in Cryptography, Cryptography and lattices. 1st international conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001. Revised papers. Berlin: Springer. Lect. Notes Comput. Sci. **2146**, 146-180 (2001).

[14] Pujol X., Stehle D., fplll mathematic software (version 4.0), http://xpujol.net/fplll.

[15] Rosser, J. B.; A note on the linear Diophantine Equation, American Maths. Monthly **48**(1941).

[16] Stein, W.A. et al. Sage Mathematics Software (Version 4.5.1), The Sage Development Team, 2012, http://www.sagemath.org.

[17] Vasilenko, O.N.; Number-Theoretic Algorithms in Cryptography, Translations of Mathematical Monographs (AMS), Volume **232** ,Translated by Alex Martsinkovsky.

Email address: drazioti@csd.auth.gr

Aristotle University of Thessaloniki, Department of Informatics, 54124 Thessaloniki, Greece