Modelling the LLL algorithm by sandpiles

Manfred Madritsch[1] and Brigitte Vallée[1]

GREYC, CNRS and University of Caen, 14032 Caen Cedex (France)

Abstract The LLL algorithm aims at finding a "reduced" basis of a Euclidean lattice. The LLL algorithm plays a primary role in many areas of mathematics and computer science. However, its general behaviour is far from being well understood. There are already many experimental observations about the number of iterations or the geometry of the output, that pose challenging questions that remain unanswered and lead to natural conjectures which are yet to be proved. However, until now, there exist few experimental observations about the precise execution of the algorithm. Here, we provide experimental results which precisely describe an essential parameter of the execution, namely the "logarithm of the decreasing ratio". These experiments give arguments towards a "regularity" hypothesis (R). Then, we propose a simplified model for the LLL algorithm, based on the hypothesis (R), which leads us to discrete dynamical systems, namely sandpiles models. It is then possible to obtain a precise quantification of the main parameters of the LLL algorithm. These results fit the experimental results performed on general input bases, which indirectly substantiates the validity of such a regularity hypothesis and shows the usefulness of such a simplified model.

Introduction

Lenstra, Lenstra, and Lovász designed the LLL algorithm [10] in 1982 for solving integer programming problems and factoring polynomials. This algorithm belongs to the general framework of lattice basis reduction algorithms, and solves a general problem: Given a basis for a lattice, how to find a basis for the same lattice, which enjoys good euclidean properties? Nowadays, this algorithm has a wide area of applications and plays a central algorithmic role in many areas of mathematics and computer science, like cryptology, computer algebra, integer linear programming, and number theory. However, even if its overall structure is simple (see Figure 1), its general probabilistic behaviour is far from being well understood. A precise quantification of the main parameters which are characteristic of the algorithms –principally, the number of iterations, the geometry of reduced bases– is yet unknown. The works of Gama, Nguyen and Stehlé [6,11] provide interesting experiments, which indicate that the geometry of the output seems to be largely independent of the input distribution, whereas the number of iterations is highly dependent on it. The article of Daudé and Vallée [5] provides a precise description of the probabilistic behaviour of these parameters (number of iterations, geometry of the output), but only in the particular case in which the vectors of the input basis are independently chosen in the unit ball. This

input distribution does not arise naturally in applications. In summary, the first works [6,11] study general inputs, but do not provide proofs, whereas the second one [5] provides proofs, but for non realistic inputs. Furthermore, none of these studies is dedicated to the fine understanding of the internal structure of the algorithm.

The LLL algorithm is a multidimensional extension, in dimension n, of the Euclid algorithm (obtained for $n = 1$) or the Gauss algorithm (obtained for $n = 2$). In these small dimensions, the dynamics of the algorithms is now well understood and there exist precise results on the probabilistic behaviour of these algorithms [12,13,14] which are obtained by using the dynamical systems theory, as well as its related tools. However, even in these small dimensions, the dynamics is rather complex and it does not seem possible to directly describe the fine probabilistic properties of the internal structure of the LLL algorithm in an exact way.

This is why we introduce in this paper a simplified model of the LLL algorithm, which is based on a regularity hypothesis: Whereas the classical version deals with a decreasing factor which may vary during the algorithm, the simplified version assumes *this decreasing factor to be constant*. Of course, this appears to be a strong assumption, but we provide arguments towards this simplification. This assumption leads us to a classical model, the *sandpile model*, and this provides another argument for such a simplification.

Sandpile models are instances of dynamical systems which originate from observations in Nature [9]. They were first introduced by Bak, Tang and Wiesenfeld [3] for modelling sandpile formations, snow avalanches, river flows, etc.. By contrast, the sandpiles that arise in a natural way from the LLL algorithm are not of the same type as the usual instances, and the application of sandpiles to the LLL algorithm thus needs an extension of classical results.

Plan of the paper. Section 1 presents the LLL algorithm, describes a natural class of probabilistic models, and introduces the simplified models, based on the regularity assumption. Section 2 provides arguments for the regularity assumption. Then, Section 3 studies the main parameters of interest inside the simplified models —the number of iterations, the geometry of reduced bases, the independence between blocks—. Section 4 then returns to the actual LLL algorithm, within the probabilistic models of Section 1, and exhibits an excellent fitting between two classes of results : the proven results in the simplified model, and the experimental results that hold for the actual LLL algorithm. This explains why these "regularized" results can be viewed as a first step for a probabilistic analysis of the LLL algorithm.

1 The LLL algorithm and its simplified version.

1.1. Description of the algorithm. The LLL algorithm considers a Euclidean lattice $\mathcal{L}$ given by a system B of n linearly independent vectors in the ambient space $\mathbb{R}^p$ $(n \leq p)$. It aims at finding a reduced basis $\widehat{B}$ formed with vectors that are *almost orthogonal and short enough*. The algorithm operates with the

matrix $\mathcal{P}$ which expresses the system B as a function of the Gram–Schmidt orthogonalized system B^*; the generic coefficient of the matrix $\mathcal{P}$ is denoted by $m_{i,j}$. The lengths ℓ_i of the vectors of the basis $B^\star$ and the ratios r_i's between successive ℓ_i,

$$r_i := \frac{\ell_{i+1}}{\ell_i}, \qquad \text{with} \quad \ell_i := \|b_i^\star\|. \tag{1}$$

play a fundamental role in the algorithm. The algorithm aims at obtaining lower bounds on these ratios, by computing a *s–Siegel reduced* basis $\widehat{B}$ that fulfills, for any $i \in [1..n-1]$, the Siegel condition $\mathcal{S}_s(i)$,

$$|\widehat{m}_{i+1,i}| \le \frac{1}{2}, \quad \widehat{r}_i := \frac{\widehat{\ell}_{i+1}}{\widehat{\ell}_i} \ge \frac{1}{s}, \qquad \text{with} \quad s > s_0 = \frac{2}{\sqrt{3}}. \tag{2}$$

In the classical LLL algorithm, a stronger condition, the Lovasz condition $\mathcal{L}_t(i)$,

$$|\widehat{m}_{i+1,i}| \le \frac{1}{2}, \quad \widehat{\ell}_{i+1}^2 + \widehat{m}_{i+1,i}^2\, \widehat{\ell}_i^2 \ge \frac{1}{t^2}\widehat{\ell}_i^2 \qquad (\text{with} \quad t > 1), \tag{3}$$

must be fulfilled for all $i \in [1..n-1]$. When s and t are related by the equality $1/t^2 = (1/4) + 1/s^2$, Condition $\mathcal{L}_t(i)$ implies Condition $\mathcal{S}_s(i)$

The version of the LLL algorithm studied here directly operates with the Siegel conditions (2). However, the behaviours of the two algorithms are similar, as it is shown in [2], and they perform the same two main types of operations:

(i) *Translation* (i,j) *(for* $j < i$*).*[1] The vector b_i is translated with respect to the vector b_j by : $\quad b_i := b_i - \lfloor m_{i,j} \rceil b_j, \quad$ with $\lfloor x \rceil :=$ the integer closest to x.
This translation does not change ℓ_i, and entails the inequality $|m_{i,j}| \le (1/2)$.

(ii) *Exchange* $(i, i+1)$. When the condition $\mathcal{S}_s(i)$ is not satisfied, there is an exchange between b_i and b_{i+1}, which modifies the lengths ℓ_i, ℓ_{i+1}. The new value $\check{\ell}_i$ is multiplied by a factor ρ and satisfies

$$\check{\ell}_i^2 := \ell_{i+1}^2 + m_{i+1,i}^2\, \ell_i^2, \qquad \text{so that} \quad \check{\ell}_i = \rho\, \ell_i \qquad \text{with} \quad \rho^2 = \frac{\ell_{i+1}^2}{\ell_i^2} + m_{i+1,i}^2, \tag{4}$$

while the determinant invariance implies the relation $\check{\ell}_i\, \check{\ell}_{i+1} = \ell_i\, \ell_{i+1}$, hence the equality $\check{\ell}_{i+1} = (1/\rho)\, \ell_{i+1}$. This entails that ρ defined in (4) satisfies

$$\rho \le \rho_0(s) \qquad \text{with} \quad \rho_0^2(s) = \frac{1}{s^2} + \frac{1}{4} < 1; \tag{5}$$

The "decreasing factor" ρ plays a crucial rôle in the following.
Figure 1 describes the standard strategy for the LLL algorithm, where the index i is incremented or decremented at each step. However, there exist other strategies which perform other choices for the next position of reduction, which can be any index i for which Condition $\mathcal{S}_s(i)$ does not hold (See Section 2). Each execution conducted by a given strategy leads to a random walk. See Figure 9 for some instances of random walks in the standard strategy.

[1] In the usual LLL algorithm, all the translations $(i+1, j)$ are performed at each step when the condition $\mathcal{S}_s(i)$ is satisfied. These translations do not change the length ℓ_{i+1}, but are useful to keep the length of b_{i+1} small. Here, we look at the trace of the algorithm only on the ℓ_i, and the translations $(i+1, j)$, with $j < i$, are not performed.

RLLL (ρ, s)
with $s > 2/\sqrt{3}$, $\rho \le \rho_0(s) < 1$

Input. A sequence $(\ell_1, \ell_2, \dots \ell_n)$
Output. A sequence $(\widehat{\ell}_1, \widehat{\ell}_2, \dots \widehat{\ell}_n)$
with $\widehat{\ell}_{i+1} \ge (1/s)\widehat{\ell}_i$.

$i := 1$;
`While` $i < n$ `do`
If $\ell_{i+1} \ge (1/s)\ell_i$, **then** $i := i+1$
else $\ell_i := \rho\, \ell_i$;
$\ell_{i+1} := (1/\rho)\, \ell_{i+1}$;
$i := \max(i-1, 1)$;

ARLLL (α) with $\alpha > \alpha_0(s)$.

Input. A sequence $(q_1, q_2, \dots q_n)$
Output. A sequence $(\widehat{q}_1, \widehat{q}_2, \dots \widehat{q}_n)$
with $\widehat{q}_i - \widehat{q}_{i+1} \le 1$.

$i := 1$;
`While` $i < n$ `do`
If $\widehat{q}_i - \widehat{q}_{i+1} \le 1$, **then** $i := i+1$
else $q_i := q_i - \alpha$;
$q_{i+1} := q_{i+1} + \alpha$;
$i := \max(i-1, 1)$;

Figure 1. Two versions of the LLL algorithm. On the left, the classical version, which depends on parameters s, ρ, with $\rho_0(s)$ defined in (5). On the right, the additive version, which depends on the parameter $\alpha := -\log_s \rho$, with $\alpha_0 := -\log_s \rho_0(s)$.

1.2. What is known about the analysis of the LLL algorithm? The main parameters of interest are the number of iterations and the quality of the output basis. These parameters depend a priori on the strategy. There are classical bounds, which are valid for any strategy, and involve the potential $D(B)$ and the determinant $\det B$ defined as

$$D(B) = \prod_{i=1}^{n} \ell_i^i, \qquad \det(B) = \prod_{i=1}^{n} \ell_i.$$

Number of iterations. This is the number of steps K where the test in step 2 is negative. There is a classical upper bound for K which involves the potential values, the initial one $D(B)$ and the final one $D(\widehat{B})$, together with the constant $\rho_0(s)$ defined in (5). We observe that K is *can be exactly expressed* with the potential values and the mean $\overline{\alpha}$ of the values $\alpha := -\log_s \rho$ used at each iteration

$$K(B) = \frac{1}{\overline{\alpha}(B)} \log_s \frac{D(B)}{D(\widehat{B})}, \qquad \text{so that} \quad K(B) \le \frac{1}{\alpha_0} \log_s \frac{D(B)}{D(\widehat{B})}, \tag{6}$$

where $\alpha_0 := -\log_s \rho_0(s)$ is the minimal value of α.

Quality of the output. The first vector $\widehat{b}_1$ of a s-Siegel reduced basis $\widehat{B}$ is short enough; there is an upper bound for the ratio $\gamma(B)$ between its length and the n-th root of the determinant,

$$\gamma(B) := \frac{||\widehat{b}_1||}{\det(B)^{1/n}} \le s^{(n-1)/2}. \tag{7}$$

The two main bounds previously described in (6) and (7) are worst–case bounds, and we are interested here in the "average" behaviour of the algorithm: What are the mean values of the number K of steps? of the output parameter γ?

1.3. Our probabilistic model. We first define a probabilistic model for input bases, which describe realistic instances, of variable difficulty. We directly choose a distribution on the actual input instance, which is formed with the coefficients

$m_{i,j}$ of the matrix $\mathcal{P}$, together with the ratios r_i. As Ajtai in [1], we consider lattice bases of full–rank (i.e, $n = p$) whose matrix B is triangular: in this case, the matrix $\mathcal{P}$ and the ratios r_i are easy to compute as a function of $b_i := (b_{i,j})$,

$$r_i = \frac{b_{i+1,i+1}}{b_{i,i}}, \qquad m_{i,j} = \frac{b_{i,j}}{b_{j,j}}.$$

Furthermore, it is clear that the main parameters are the ratios r_i, whereas the coefficients $m_{i,j}$ only play an auxilliary rôle. As Ajtai suggests it, we choose them (for $j < i$) independently and uniformly distributed in the interval $[-1/2, +1/2]$. Since Ajtai is interested by worst-case bounds, he chooses very difficult instances where the input ratios r_i are *fixed* and very small, of the form $r_i \sim 2^{-(a+1)(2n-i)^a}$ with $a > 0$. We design a model which produces instances with variable difficulty, and, each ratio r_i is here a random variable which follows a power law :

$$\forall i \in [1..n-1],\ \exists \theta_i > 0 \text{ for which } \quad \mathbb{P}\left[r_i \leq x\right] = x^{1/\theta_i} \qquad \text{for } x \in [0,1]. \tag{8}$$

The difficulty of the instance increases when the parameters θ_i become large. This distribution arises in a natural way in various frameworks, in the two dimensional case [13] or when the initial basis is uniformly chosen in the unit ball. See [14] for a discussion about this probabilistic model.

1.4. An additive version. First, we adopt an additive point of view, and thus consider the logarithms of the main variables

$$q_i := \log_s \ell_i, \qquad c_i := -\log_s r_i = q_i - q_{i+1} \qquad \alpha := -\log_s \rho \tag{9}$$

Then, the Siegel condition becomes $q_i \leq q_{i+1} + 1$ or $c_i \leq 1$, and the exchange in the LLL algorithm is rewritten as (see Figure 1. right)

$$\textbf{If} \quad q_i > q_{i+1} + 1, \quad \textbf{then} \quad [\check{q_i} = q_i - \alpha, \quad \check{q_{i+1}} = q_{i+1} + \alpha]. \tag{10}$$

In our probabilistic model, each c_i follows an exponential law of the form

$$\mathbb{P}[c_i \geq y] = s^{-y/\theta_i} \qquad \text{for} \quad y \in [0, +\infty[\qquad \text{with} \quad \mathbb{E}[c_i] = \frac{\theta_i}{\log s}. \tag{11}$$

This model is then called the `Exp-Ajtai`(θ) model. Remark that, if we restrict ourselves to non-reduced bases, we deal with the `Mod-Exp-Ajtai`(θ) distribution,

$$\mathbb{P}[c_i \geq y + 1] = s^{-y/\theta_i} \qquad \text{for} \quad y \in [0, +\infty[, \qquad \text{with} \quad \mathbb{E}[c_i] = 1 + \frac{\theta_i}{\log s}. \tag{12}$$

1.5. The regularized version of the LLL algorithm. The main difficulty of the analysis of the LLL algorithm is due to the fact that the decreasing factor ρ defined in (4) can vary throughout the interval $[0, \rho_0(s)]$. For simplifying the behaviour of the LLL algorithm, we assume that the following Regularity Hypothesis holds (R):

(R). *The decreasing factor ρ (and thus its logarithm $\alpha := -\log_s \rho$) are constant.*

Then, Equation (10) defines a sandpile model which is studied in Section 3.

There are now three main questions:

– Is Hypothesis (R) reasonable? This is discussed in Section 2.

– What are the main features of the regularized versions of the LLL algorithm, namely sandpiles? This is the aim of Section 3.

– What consequences can be deduced on the probabilistic behaviour of the LLL algorithm? This is done in Section 5 that transfers results of Section 4 to the framework described in Section 2, with the arguments discussed in Section 4.

2 Is the LLL algorithm regular?

2.1. General bounds for α. Since the evolution of the coefficients $m_{i+1,i}$ seems less "directed" by the algorithm, we may suppose them to be uniformly distributed inside the $[-1/2, +1/2]$ interval, and independent of the Siegel ratios. The average of the square m^2 is then equal to $1/12$, and if we assume m^2 to be constant and equal to $1/12$, then the value of α satisfies (with s near $s_0 = 2/\sqrt{3}$),

$$-\frac{1}{2}\log_{s_0}\left(\frac{3}{4}+\frac{1}{12}\right) \leq \alpha := -\frac{1}{2}\log_{s_0}\left(r^2+\frac{1}{12}\right) \leq -\frac{1}{2}\log_{s_0}\left(\frac{1}{12}\right).$$

Then $\alpha \in [0.5, 8.5]$ most of the time. This fits with our experiments.

2.2. General study of parameter α. We must make precise the regularity assumption. Of course, we cannot assume that there is a universal value for $\alpha := -\log_s \rho$, and we describe the possible influence of four variables on the parameter α, when the dimension n becomes large:

(a) The input distribution of `Exp-Ajtai` type is described by $\Theta = (\theta_1, \ldots, \theta_{n-1})$.

(b) The position $i \in [1..n(B)-1]$ is the index where the reduction occurs.

(c) The discrete time $j \in [1..K(B)]$ is the index when the reduction occurs,

(d) The strategy defines the choice of the position i at the j–th iteration, inside the set $\mathcal{N}(j)$ which gathers the indices for which Condition $\mathcal{S}(i)$ is not satisfied. We consider three main strategies Σ : – The standard strategy Σ_s chooses $i := \mathrm{Min}\mathcal{N}(j)$ – The random strategy Σ_r chooses i at random in $\mathcal{N}(j)$ – The greedy strategy Σ_g chooses the index $i \in \mathcal{N}(j)$ for which the ratio r_i is minimum.

The study of α decomposes into two parts. First, we study the variations of α during one execution, due to the position i or the time j. Second, we consider the variable $\overline{\alpha}$, defined as the mean value of α during one execution, and study the influence of the input distribution, the strategy, and the dimension on $\overline{\alpha}$.

We consider a set $\mathcal{B}$ of input bases, and we determine a maximal value M of α for this set of inputs. In order to deal with fixed intervals for positions, times, and values, we choose three integers X, Y, Z, and we divide

– the interval $[1..n]$ of positions into X equal intervals of type I_x with $x \in [1..X]$,

– the interval $[1..K]$ of times into Y equal intervals of type J_y with $y \in [1..Y]$, – the interval $[0, M]$ of values into Z equal intervals. of type L_z with $z \in [1..Z]$

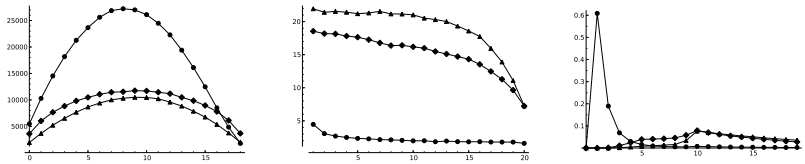
Then the parameters $\alpha_{\langle x \rangle}, \alpha^{\langle y \rangle}$ are respectively defined as the restriction of α for $i \in I_x$, (resp. for $j \in J_y$).

2.3. Distribution of the variable α. Here, the parameter Θ of the input distribution and the strategy $\Sigma \in \{\Sigma_s, \Sigma_r, \Sigma_g\}$ are fixed, . and we consider a set $\mathcal{N}$ of dimensions. We first consider the global variable α, study its distribution,

(1) The distribution of the parameter α for $n = 5\,(\bullet); n = 10\,(\blacksquare); n = 15\,(\blacktriangle); n = 20\,(\blacklozenge)$

(2) The two functions $y \mapsto \overline{\alpha}^{\langle y \rangle}$(left, with $Y = 20$) and $x \mapsto \overline{\alpha}_{\langle x \rangle}$ (right, with $X = 5$), for $n = 5\,(\bullet); n = 10\,(\blacksquare); n = 15\,(\blacktriangle); n = 20\,(\blacklozenge)$

(3) On the left, the distribution of $\alpha^{\langle y \rangle}$ with $Y = 20$ and $y = 2\,(\bullet); 5\,(\blacksquare); 10\,(\blacktriangle); 15\,(\blacklozenge); 20\,(\blacktriangledown)$
On the right, the distribution of $\alpha_{\langle x \rangle}$ with $X = 5$ and $x = 1\,(\bullet); 2\,(\blacksquare); 3\,(\blacktriangle); 4\,(\blacklozenge); 5\,(\blacktriangledown)$



(4) The curves are associated to $\bullet$ for Σ_s (standard), $\blacktriangle$ for Σ_g (greedy), and $\blacklozenge$ for Σ_r (random). On the right, the functions $x \mapsto A_{\langle x \rangle}$. In the middle, the functions $y \mapsto \overline{\alpha}^{\langle y \rangle}$. On the left, the distribution of α.

Figure 2. Experiments about the Regularity Hypothesis: Study of the global parameter α. Influence of position and time. Influence of the strategy for $n = 20$.

and its mean, for each $n \in \mathcal{N}$ [See Figure 3(1)]. We observe that the distribution of α gets more and more concentrated when the dimension grows, around a value which appears to tend to 2.5.

2.4. Variations of α during an execution. Figure 3(2) describes the two functions $x \mapsto \overline{\alpha}_{\langle x \rangle}$ and $y \mapsto \overline{\alpha}^{\langle y \rangle}$, for each dimension $n \in \mathcal{N}$. Figure 3(3) provides (for $n = 20$) a description of the distribution of parameters $\alpha_{\langle x \rangle}, \alpha^{\langle y \rangle}$ for various values of (x, y). We observe that the variations of the functions $y \mapsto \overline{\alpha}^{\langle y \rangle}$ and $x \mapsto \overline{\alpha}_{\langle x \rangle}$ are small, and become smaller when the dimension n increases. The distributions of $\alpha_{\langle x \rangle}$ and $\alpha^{\langle y \rangle}$ are also concentrated, at least for y's not too small and for central values of x.

2.5. Influence of the strategy. Here, for $n = 20$, we describe the influence of the trategy on the functions $\quad x \mapsto A_{\langle x \rangle}, \quad y \mapsto \overline{\alpha}^{\langle y \rangle}, \quad z \mapsto \mathbb{P}[\alpha \in L_z]$.
The experimental results, reported in Figure 4, show the important influence of the strategy on the parameter α. They are of independent interest, since, to the best of our knowledge, the strategy is not often studied. There are two groups: On the one hand, the standard strategy[2] is the least efficient: it performs a larger number of steps, and deals with a parameter α whose value is concentrated below $\alpha = 5$. On the other hand, the other two ones, (random and greedy) are much more efficient, with a much smaller number of steps; they deal with values of α which vary in the whole interval $[5, 20]$ and decrease with the discrete time. These two strategies (random and greedy) must be used if we wish more efficient algorithms. If we wish simulate with sandpiles the LLL algorithm under these two strategies, we have to consider different values of α, for instance, at the beginning, in the middle and at the end of the execution.

2.6. Influence of the input distribution. We study the influence of the parameter Θ of the `Exp-Ajtai` distribution on $\overline{\alpha}$. We first recall what happens in two dimensions, where the LLL algorithm coincides with the Gauss algorithm. The paper [13] studies this algorithm when the input $c := -\log_s r$ follows an exponential law with mean θ and proves that the number of steps K of the Gauss algorithm follows a geometric law of ratio $\lambda(1+1/\theta)$, where $\lambda(s)$ is the dominant eigenvalue of the transfer operator associated to the Gauss algorithm.

$$\text{The relations} \qquad -\log_s \mathbb{P}[K \geq k] \sim -\log_s \mathbb{P}\left[c \geq k\overline{\alpha}\right] \sim \frac{\mathbb{E}_\theta[\overline{\alpha}]}{\theta}\, k$$

entail that the mean $\mathbb{E}_\theta[\overline{\alpha}]$ depends on θ as $\quad \mathbb{E}_\theta[\overline{\alpha}] \sim -\theta \log_s \lambda\left(1 + \frac{1}{\theta}\right)$.

Then, properties of the pressure[3] imply that the function $\mathbb{E}_\theta[\overline{\alpha}]$ satisfies

$$\mathbb{E}_\theta[\overline{\alpha}] \sim \frac{|\lambda'(1)|}{\log s} \quad \text{for } \theta \to \infty, \qquad \text{and} \quad \mathbb{E}_\theta[\overline{\alpha}] \sim \frac{2}{\log s} \log(1+\sqrt{2}) \quad \text{for } \theta \to 0,$$

[2] We have not reported the results relative to the anti-standard strategy which chooses $i := \mathrm{Max}\,\mathcal{N}(j)$, but they are of the same type as the standard one.
[3] In dynamical theory setting, the pressure is the logarithm of the dominant eigenvalue.

where $|\lambda'(1)| \sim 3.41$ equals the entropy of the Euclid centered algorithm. This entails that, in two dimensions, the mean value $\mathbb{E}[\overline{\alpha}]$ varies in the interval $[14, 23]$. Led by the dynamical point of view, we set a conjecture which extends the previous two–dimensional property to any dimensions.

Entropy Conjecture. *Consider the probabilistic* `Exp-Ajtai`(θ) *model in n dimensions. Then, for $\theta \to \infty$, the mean of the variable $\overline{\alpha}$ tends to the entropy $\mathcal{E}_n$ of the dynamical system underlying the LLL algorithm.*

$$\lim_{\theta \to \infty} \mathbb{E}_{(\theta,n)}[\overline{\alpha}] = \frac{\mathcal{E}_n}{\log s}$$

3 Study of the sandpile model.

There are three main questions about the RLLL algorithm :

($Q1$) Does the RLLL algorithm depend on the strategy?

($Q2$) How does the behaviour of the RLLL algorithm depend on the value of parameter α? What about the number of iterations? the output configuration? Are there lower bounds on average in relations (6) and (7)?

($Q3$) Does there exist a characterisation for two blocks to be independent in the RLLL algorithm? We can run the execution of the LLL algorithm in parallel, on the block B_- formed with the first vectors and on the block B_+ formed on the last vectors The two blocks B_- and B_+ are said to be independent if the total basis formed by concatening the two reduced bases $\widehat{B}_-$ and $\widehat{B}_+$ is reduced.

Here, we answer these three main questions. As we already said previously, the additive version of the regularized algorithm (see Figure 1.right) deals with the sandpile model. Even if this model is very well known, the modelling of the RLLL algorithm gives rise to non classical instances of sandpile models.

3.1. Main objects for sandpiles. Here, H, h denote strictly positive real numbers.

A sandpile model $\mathbf{Q}_n(\mathbf{q}, H, h)$ describes all the possible evolutions of the configuration $\mathbf{q} = (q_1, \ldots, q_n)$ under the action of functions f_i

$$f_i(\mathbf{q}) = \begin{cases} q_j - h & \text{if } j = i \text{ and } q_i - q_{i+1} > H, \\ q_j + h & \text{if } j = i+1 \text{ and } q_i - q_{i+1} > H, \\ q_j & \text{else.} \end{cases}$$

We associate to $\mathbf{q} := (q_1, \ldots, q_n)$ the configuration $\mathbf{c} := \Delta(\mathbf{q})$ formed with the differences between the components, $c_i = q_i - q_{i+1} \quad \text{for} \quad i \in [1..n-1]$.

The strategy graph, denoted by $\mathcal{G}(\mathbf{q}, H, h)$, is a directed graph whose vertices are all the configurations that are reachable from $\mathbf{q}$; there is an edge from $\mathbf{u}$ to $\mathbf{v}$ (with $\mathbf{u} \neq \mathbf{v}$) if there exists an index $i \in [1..n-1]$ for which $\mathbf{v} = f_i(\mathbf{u})$.

The energy E and the total mass M of the configuration $\mathbf{q}$ are defined by

$$E(\mathbf{q}) = \sum_{i=1}^{n} i \cdot q_i, \quad \text{and} \quad M(\mathbf{q}) = \sum_{i=1}^{n} q_i, \tag{13}$$

and satisfy $\quad M(f_i(\mathbf{q})) = M(\mathbf{q}), \qquad E(f_i(\mathbf{q})) = E(\mathbf{q}) + h.$

3.2. Various kinds of sandpiles. The usual sandpiles are basic and decreasing:

Definition 1. *(i) A sandpile* $\mathbf{q}$ *is basic if the configuration* $\Delta(\mathbf{q})$ *is integral and parameters* (H, h) *equal* $(1, 1)$
(ii) A sandpile is (H, h)*–integral if the components* c_i *of* $\mathbf{c} := \Delta(\mathbf{q})$ *belong to the same discrete line* $H + \mathbb{Z}h$
(iii) A basic sandpile $\mathbf{q}$ *is decreasing if the components of* $\mathbf{c} := \Delta(\mathbf{q})$ *are positive* $(c_i \geq 0)$*. It is strictly decreasing if* $\mathbf{c}$ *is strictly positive. It is increasing if all the components of* $\mathbf{c}$ *are negative.*

The sandpiles used in the RLLL algorithm are not basic. However, the following result shows that any general sandpile is isomorphic to a basic sandpile.

Proposition 1. *The mapping* $\psi : \mathbf{q} \mapsto \mathbf{q}'$ *defined by*

$$c'_i := 1 - \left\lfloor \frac{H - c_i}{h} \right\rfloor, \qquad q'_n = 0 \tag{14}$$

transforms a general sandpile into a basic sandpile. Moreover, the two graphs $\mathcal{G}(\mathbf{q}, H, h)$ *and* $\mathcal{G}(\psi(\mathbf{q}), 1, 1)$ *are isomorphic.*

A general sandpile $\mathbf{q}$ is decreasing (resp. strictly decreasing, increasing) if $\psi(\mathbf{q})$ is decreasing (resp. strictly decreasing, increasing). A general sandpile decomposes into strictly decreasing configurations, separated by increasing configurations.

Definition 2. *Two adjacent strictly decreasing sandpiles* $\mathbf{q}^-, \mathbf{q}^+$ *are independent if the configuration obtained by concatening the two final configurations* $\widehat{\mathbf{q}}^-$ *and* $\widehat{\mathbf{q}}^+$ *is a final configuration.*

3.3. Study of a general sandpile. Here, we obtain (easy) extensions of results of Goles and Kiwi in [8] to a general sandpile. These authors only consider particular basic decreasing sandpiles.

Theorem 1. *The following holds for any sandpile* $\mathcal{Q}(\mathbf{q}, H, h)$*:*
(i) The graph $\mathcal{G}(\mathbf{q}, H, h)$ *is finite, with a unique final configuration* $\widehat{\mathbf{q}}$*. The length of a path* $\mathbf{q} \to \widehat{\mathbf{q}}$ *is the same for any path. This is the number of steps* $T(\mathbf{q})$*,*

$$T(\mathbf{q}) = \frac{1}{h}\left[E(\widehat{\mathbf{q}}) - E(\mathbf{q})\right] = \frac{1}{2h}\sum_{i=1}^{n-1} i(n-i)\,(c_i - \widehat{c}_i)$$

(ii) If $\mathcal{Q}_n(\mathbf{q}, H, h)$ *is decreasing, then the components of the output configuration* $\widehat{\mathbf{c}}$ *satisfy* $H - 2h < \widehat{c}_i \leq H$*, and the number of iterations satisfy*

$$0 \leq T(\mathbf{q}) - \frac{1}{2h}\sum_{i=1}^{n-1} i(n-i)\,(c_i - H) \leq 2A(n) \qquad \text{with} \quad A(n) := n\frac{n^2-1}{12}$$

(iii) If $\mathcal{Q}_n(\mathbf{q}, H; h)$ *is strictly decreasing, then there exists* $j \in [1..n-1]$ *for which*

$$\forall i \neq j,\ H - h < \widehat{c}_i \leq H, \qquad \text{and} \quad H - 2h < \widehat{c}_j \leq H - h,$$

and the number of steps $T(\mathbf{q})$ *satisfies*

$$0 \leq T(\mathbf{q}) - \left[A(n) + \frac{1}{2h} \sum_{i=1}^{n-1} i(n-i)\,(c_i - H) \right] \leq \frac{1}{8} n^2$$

(*iv*) *For a general sandpile* $\mathcal{Q}_n(\mathbf{q}, H, h)$*, the output configuration satisfies*

$$H - 2h < \widehat{c}_i \leq H \quad \text{if} \quad c_i > H - h, \qquad \widehat{c}_i \geq c_i \quad \text{if} \quad c_i \leq H - h$$

and the number of steps $T(\mathbf{q})$ *satisfies*

$$\frac{1}{2h} \sum_{i=1}^{n-1} i(n-i)(c_i - H + h) \leq T(\mathbf{q}) \leq \frac{1}{2h} \sum_{i=1}^{n-1} i(n-i) \max(c_i - H + h, 0)$$

(*v*) *A sufficient condition for two adjacent sandpiles* $\mathcal{Q}_p(\mathbf{q}_-, H, h), \mathcal{Q}_n(\mathbf{q}_+, H, h)$ *to satisfy the independence condition of Definition 2 is*

$$\frac{1}{p} M(\mathbf{q}_-) - \frac{1}{n} M(\mathbf{q}_+) \leq \left(\frac{n+p}{2} \right) (H - h) - h$$

and for a sandpile (H, h)*–integral:* $\quad \dfrac{1}{p} M(\mathbf{q}_-) - \dfrac{1}{n} M(\mathbf{q}_+) \leq \left(\dfrac{n+p}{2} \right) H - 2h.$

4 Returning to lattices.

We now return to the LLL algorithm, with the framework of Section 1, and apply the results of Section 3 to the so–called ρ–regular executions of the LLL algorithm, for which the decreasing factor is constant and equal to ρ. We recall that, in this case, the execution of the algorithm in dimension n can be viewed as a sandpile model $\mathbf{Q}_n(\mathbf{q}, 1, \alpha)$ associated to a parameter $\alpha := -\log_s \rho$, and an initial configuration $\mathbf{q}$ related to the lengths ℓ_i of the orthogonalised basis $B^\star$ of the input basis B via the equalities $q_i := \log_s \ell_i$. The main objects associated to the basis B, namely the potential $D(B)$ or the determinant $\det(B)$ are then related to the energy $E(\mathbf{q})$ or the total mass $M(\mathbf{q})$,

$$E(\mathbf{q}) = \log_s D(B), \qquad M(\mathbf{q}) = \log_s \det(B).$$

We are interested in two kinds of input bases:

(*i*) We first study totally non-reduced bases, for which Condition $\mathcal{S}_s(i)$ is never satisfied on the input. In this case, the sandpile is strictly decreasing. [Sections 4.1 and 4.2]

(*ii*) We then study a general input basis, which is a sequence of blocks, some of them are totally non-reduced, and other ones are totally reduced [Section 4.3]

We compare here the results that are proven for regular executions of the LLL algorithm, (by an easy transfer of results of Section 3) and the experimental results that are performed on general executions of the algorithm. We will see that there is a good fitting between these two kinds of results. This good fitting has two main consequences:

– This provides an indirect validation of the property : "The executions of the LLL algorithm are very often regular enough".

– This shows that long experiments on the LLL algorithm can be simulated by fast computations in the sand pile model (with a good choice of parameter α).

As in Section 3, we study the final configurations, the number of steps, and the independence of blocks.

4.1. Final configurations. When the initial basis is totally non reduced, the sandpile is strictly decreasing. Then, with Theorem 1 (*ii*), each output Siegel ratio $\widehat{r}_i$ and the first vector of the output basis satisfy

$$\rho s \le \frac{1}{\widehat{r}_i} = \frac{\widehat{\ell}_i}{\widehat{\ell}_{i+1}} \le s, \qquad \rho(s\cdot\rho)^{(n-1)/2} \le \gamma(\widehat{B}) = \frac{||\widehat{b}_1||}{(\det L)^{1/n}} \le s^{(n-1)/2}. \quad (15)$$

Then, we have proven:

Theorem 2. *Consider a totally non reduced basis B on which the execution of the LLL algorithm is ρ–regular. Then, the output parameter $\gamma(\widehat{B})$ defined in (7) satisfies*

$$\frac{2}{n-1}\log_s \gamma(\widehat{B}) \in [1-\alpha, 1], \qquad \text{with } \alpha := -\log_s \rho.$$

This is compatible with experiments done on general executions by Nguyen and Stehlé [11], which show that there is a mean value $\beta \sim 1.04$, such that, for most of the output bases $\widehat{B}$, the ratio $\gamma(\widehat{B})$ is close to $\beta^{(n-1)/2}$. The relation $\beta \sim s\sqrt{\rho}$ is then plausible, so that the "usual" ρ would be close to 0.81.

4.2. Number of iterations. Suppose that the (totally non reduced) input basis follows the `Mod-Exp-Ajtai`(θ) distribution. Then, the configuration $\mathbf{c}'$ associated to $\mathbf{c}$ via Theorem 1 follows a geometric law,

$$\mathbb{P}[c'_i \ge 1+k] = \rho^{k/\theta}, \qquad \mathbb{E}[c'_i - 1] = \frac{\rho^{1/\theta}}{1-\rho^{1/\theta}}$$

Then, Theorem 1 (*iii*) entails:

Theorem 3. *Consider an input basis B, which follows the* `Mod-Exp-Ajtai` *distribution of parameter θ. If the execution of the LLL algorithm in dimension n is ρ–regular on the basis B, the number of iterations satisfies*

$$K_n(\rho,\theta) \sim \frac{n^3}{12\alpha}\left(\frac{\rho^{1/\theta}}{1-\rho^{1/\theta}}\right) \qquad (n\to\infty).$$

If the Entropy Conjecture of Section 3.6 is true, then

$$\lim_{\theta\to\infty} K_n(\theta) \sim \left(\frac{\theta \log s}{12}\right)\frac{n^3}{\mathcal{E}_n^2} \qquad \text{where } \mathcal{E}_n \text{ is the entropy of the LLL algorithm.}$$

This results fits with the experiments done for general executions of the LLL algorithm by Nguyen and Stehlé [11]. In particular, for the choice of Ajtai, namely $\theta = n^a$, the experiments show a number of iterations of order $\Theta(n^{3+a})$.

4.3. An instance of the independence property. The question of the independence between blocks is important. We now describe such an instance of this phenomenon in the framework of Coppersmith's method. In the paper [4], Boneh and Durfee present a method for breaking the RSA cryptosystem based on Coppersmith's method. Coppersmith's method uses the LLL algorithm in order

to find a small root of a polynomial modulo an integer E. For the cryptanalysis of RSA, one deals with the public exponent E. We let $L := \log_s E$.

The initial configuration is formed with $m+1$ blocks, indexed from $k=0$ to m. The k-th block has length $k+1$, is $(1,\alpha)$-integral, and the components c_i of the configuration $\mathbf{c}$ are equal to $L/2$. However, the total configuration is not totally decreasing, but the (second) sufficient condition of Theorem 1 (v) is always true. Then, Theorem 1(v) entails:

Theorem 4. *Suppose that the execution of the LLL algorithm is ρ-regular on the Coppersmith lattice described in [4]. Then, the blocks of the lattice are always independent, and the reduction can be done in parallel on each block. The number of iterations K_p performed in this parallel strategy is then*

$$K_p = \frac{m^3}{12\alpha}\left(\frac{L}{2}-1\right) \qquad \textit{to be compared to} \quad K_s = \sum_{i=1}^{m} K_i \sim \frac{m^4}{48\alpha}\left(\frac{L}{2}-1\right),$$

which is the number of steps in the sequential strategy.

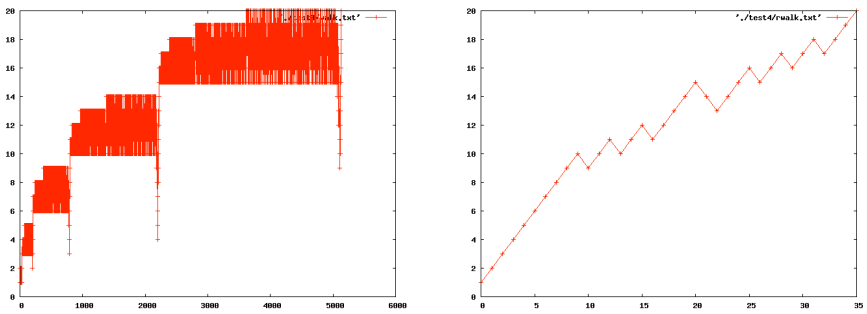


Figure 3. On the left, the random walk of the actual LLL algorithm on a Coppersmith lattice of dimension 21 (related to $m = 5$). On the right, the random walk of the execution of the LLL algorithm on the basis formed by the concatenation of the reduced blocks.

Of course, the execution of the LLL algorithm on the Boneh-Durfee latticecannot be totally regular : the first vector of the reduced lattice basis would be the first vector of the initial basis, and the method would fail! However, it is possible to compare (see [7]) the result of Theorem 4 to an execution of the actual LLL algorithm on a Boneh–Durfee lattice (see Figure 3 left). We first see that, on each block, the number of iterations is quite large (the blocks are totally non reduced) and this fits with the order $\Theta(k^3)$ which is proven for a ρ–regular execution. We also remark that the blocks are not independent but almost independent: the basis obtained by concatening the reduced bases of each block is not totally reduced, but few reduction steps are needed for reducing it, as Figure 3 (right) shows it. Such a strategy, whose first step can be performed in a parallel way, is very efficient in this case

Conclusion. This paper presents a simplified model of the LLL algorithm, under a "regularity" hypothesis which assumes that the decreasing factor ρ is constant. Of course, this hypothesis does not exactly hold in the reality, and we have provided experimental results about its validity. We have also explained why this simplified model is very useful for understanding the LLL algorithm in an intuitive way, and for testing (at least qualitative) conjectures on the algorithm. The excellent fitting of this model on a class of Coppersmith lattices is also striking. In fact, the sandpile model represents a good compromise between simplicity and adequation to the reality.

Acknowlegments. This research was supported by the LAREDA Project (LAttice REDuction Algorithms: Dynamics, Probability, Experiments) of the ANR (French National Research Agency). The authors thank Ali Akhavi, Julien Clément, Mariya Georgieva, Fabien Laguillaumie, Loïck Lhote, Damien Stehlé, Antonio Vera, and the whole group LAREDA for interesting discussions on the subject.

References

1. Ajtai, M. : Optimal lower bounds for the Korkine-Zolotareff parameters of a lattice and for Schnorr's algorithm for the shortest vector problem. Theory of Computing 4(1): 21-51 (2008)
2. Akhavi, A. : Random lattices, threshold phenomena and efficient reduction algorithms. Theoret. Comput. Sci. **287**(2) (2002) 359–385
3. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: An explanation of the 1/f noise. Phys. Rev. Lett. **59**(4) (Jul 1987) 381–384
4. Boneh, D. and Durfee, G. : Cryptanalysis of RSA with private key d less than $N \leq 0.292$, IEEE Trans. Inform. Theory **46** (2000), no. 4, 1339–1349.
5. Daudé, H. and Vallée, B : An upper bound on the average number of iterations of the LLL algorithm. Theoretical Computer Science 123, 1 (1994), 95–115.
6. Gama, N. and Nguyen, P. : Predicting Lattice Reduction, Proceedings of Eurocrypt 2008, LNCS 4965, 31-51
7. Georgieva, M. : Étude expérimentale de l'algorithme LLL sur certaines bases de Coppersmith, Master Thesis, University of Caen (2009).
8. Goles, E., Kiwi, M.A. : Games on line graphs and sandpiles. Theoret. Comput. Sci. **115**(2) (1993) 321–349
9. Jensen, H.J. : Self-organized criticality. Volume 10 of Cambridge Lecture Notes in Physics. Cambridge University Press, Cambridge (1998) Emergent complex behavior in physical and biological systems.
10. Lenstra, A.K., Lenstra, Jr., H.W., Lovász, L. : Factoring polynomials with rational coefficients. Math. Ann. **261**(4) (1982) 515–534
11. Nguyen, P. and Stehlé, D. : LLL on the average, Proceedings of the 7th Algorithmic Number Theory Symposium (ANTS VII), Springer LNCS vol. 4076, (2006), 238–256
12. Vallée, B. : Euclidean Dynamics, Discrete and Continuous Dynamical Systems, 15 (1) May 2006, pp 281-352.
13. Vallée, B., Vera, A. : Probabilistic analyses of lattice reduction algorithms. Chapter 3 of the book "The LLL Algorithm", collection Information Security and Cryptography Series, Springer (2009)
14. Vera, A.: Analyses de l'algorithme de Gauss. Applications à l'analyse de l'algorithme LLL, PhD Thesis, Universiy of Caen (2009)