LLL AND STOCHASTIC SANDPILE MODELS

JINTAI DING, SEUNGKI KIM, TSUYOSHI TAKAGI, YUNTAO WANG

ABSTRACT. The aim of the present paper is to suggest that statistical physics provides the correct language to understand the practical behavior of the LLL algorithm, most of which are left unexplained to this day. To this end, we propose sandpile models that imitate LLL with compelling accuracy, and prove for these models some of the most desired statements regarding LLL. We also formulate a few conjectures that formally capture our heuristics and would serve as milestones for further development of the theory.

1. INTRODUCTION

1.1. **The mysteries of LLL.** The LLL algorithm ([22]) is one of the most celebrated algorithmic inventions of the twentieth century, with countless applications to pure and computational number theory, computational science, and cryptography. It is also the most fundamental of lattice reduction algorithms, in that nearly all known reduction algorithms are generalizations of LLL in some sense, and they also utilize LLL as their subroutine. (We refer the reader to [28] for a thorough survey on LLL and these related topics.) Thus it is rather curious that much of the observed behavior of LLL in practice is left totally unexplained, not even in a heuristic, speculative sense, even to this day.

The most well-known among the mysteries of LLL is the gap between its worst-case root Hermite factor(RHF) and the observed average-case, as documented in Nguyen and Stehlé ([27]). It is a theorem from the original LLL paper ([22]) that the shortest vector of an LLL-reduced basis, with its determinant normalized to 1, has length at most $(4/3)^{\frac{n-1}{4}} \approx 1.075^n$, whereas in practice one almost always observes $\approx 1.02^n$, regardless of the way in which the input is sampled. This is a strange phenomenon in the light of the work of Kim and Venkatesh ([20]), which, roughly speaking, proves that, for almost every lattice, nearly all of its LLL bases have RHF close to the worst bound. This leads to the suspicion that the LLL algorithm must be operating in a complex manner that belies the simplicity of its code.

There are also many other phenomena regarding LLL that are unaccounted for. One is the geometric series assumption(GSA), originally proposed by Schnorr ([33]), and its partial failure at the boundaries, both of which are observed in other blockwise reduction algorithms as well e.g. BKZ ([34]). There are also questions raised regarding the time complexity of LLL. Nguyen and Stehlé ([27]) suggest that, in some situations, the average time complexity is lower than the worst-case, and in others, the worst-case is attained. The complexity of the optimal LLL algorithm — i.e. the parameter δ equals 1 — is not proven to be polynomial-time, although observations suggest that it is (see Akhavi ([1]) and references therein).

1.2. **This paper.** Our main idea is that statistical physics may provide a correct language and concepts to study the practical behavior of the LLL algorithm. As we demonstrate throughout this paper, for each LLL phenomenon, there is a corresponding sandpile

phenomenon, which can be captured and then studied with the already well-established methods of physics. Indeed, from the physical perspective, there is no reason not to regard the LLL algorithm as a proper member of the family of *stochastic sandpile models*. Since the identification of the correct language often does much good in mathematics, this may open up a path to a systematic understanding of the various unexplained phenomena of LLL, via the mathematics and physics of the sandpile models.

Some similarities of LLL to sandpiles have been noticed previously and utilized to some extent. To the best of our knowledge, this is first pointed out in Madritsch and Vallée ([29]), and also in Vallée ([37]), albeit briefly. Some aspects of this analogy have also been applied to BKZ as well — see [16] and [4] for instance.

One of the new contributions made by the present paper is the introduction of stochastic sandpile models that are both impressively close to LLL (precisely speaking, its Siegel variant) and mathematically accessible. We propose two models of LLL, which we name *LLL-SP* and *SSP* respectively. LLL-SP (Algorithm 2) is a sandpile model that exhibits nearly identical quantitative behavior to that of LLL in many aspects, suggesting that the two algorithms operate under the same principles. This claim can be formulated in a precise language, in terms of the mixing property of the μ variables; see Conjecture 2 below. SSP (Algorithm 4) is a simple stochastic variant of the abelian sandpile model (ASM; Algorithm 3) that serves as a useful toy model for LLL that is mathematically far more tractable than LLL-SP, and still imitates the important aspects of the output statistics of LLL.

We also demonstrate that it is possible to prove some of the most desired statements for LLL on these models. On SSP, we can establish an upper bound on the average-case RHF that is significantly smaller than the worst-case (Theorem 3; the proof is deferred to [21]). This seems much harder to achieve on LLL-SP, but here we are still able to provide a probabilistic lower bound on the time complexity (Theorem 5), whose order matches the well-known upper bound, and to resolve the optimal LLL problem (Theorem 6). These results support our idea that the sandpile interpretation may be the correct approach to the real-world behavior of LLL.

In addition, it is worth noting that this physical perspective on LLL provides convincing heuristics on a number of LLL behavior, which are scattered throughout this paper. For instance, it explains why the output statistics of LLL appears independent of the input distribution, and has the shape as described by the GSA and its failure at the boundaries. Given that there has been not even a vague heuristic accounting for most practical behavior of LLL, this provides yet another good reason to pursue this line of development. Moreover, we try to capture the essence of these heuristics with formal language, with Conjecture 2 below and the "parallelepiped argument" in the proof of Theorem 3 (also see Conjecture 4). These provide some concrete paths for further research.

1.3. **Cryptographic considerations.** LLL is of fundamental importance to lattice-based cryptography, and for this reason alone it deserves to be understood well. Our understanding of LLL may affect our understanding of all other reduction algorithms, particularly BKZ, the current standard method for challenging lattice-based systems.

Specifically, there are certain questions regarding LLL — and any reduction algorithm in general — that may pose a threat to lattice-based cryptography as a whole. For instance, it could be that a yet undiscovered small trick may improve the RHF of LLL without meaningfully increasing the time complexity, say, to 1.002. Absurd as this may sound at first, recall that we already "improved" LLL from (RHF) ≈ 1.075 to ≈ 1.02

merely by implementing it, and that currently we lack the device to form even a vague argument against such catastrophic possibility.

It seems that the practitioners of lattice-based cryptography are well aware of such uncertainties as to the scope and limit of reduction algorithms, and are reacting accordingly. According to Tables 5–10 of [2], most lattice-based submissions to the recent NIST call for proposals ([24]) claim about half or less as may bits of security as estimated with the state-of-art techniques. On the other hand, one observes no such reservation in submissions from multivariate cryptography, for example. By developing a framework for a systematic study of reduction algorithms, we hope to be able to relieve some of these concerns in the area of lattice-based cryptography.

1.4. **Assumptions and notations.** Throughout this paper, instead of the original LLL reduction from [22], we work with its Siegel variant, a slight simplification of LLL. The Siegel reduction shares with LLL all its idiosyncrasies, and a bit easier to handle technically; hence a reasonable starting point for our research.

n always means the dimension of the relevant Euclidean space. Our lattices in $\mathbb{R}^n$ always have full rank.

A basis $\mathcal{B}$, besides its usual definition, is an *ordered* set, and we refer to its i-th element as $\mathbf{b}_i$. Denote by $\mathbf{b}_i^*$ the component of $\mathbf{b}_i$ orthogonal to all vectors preceding it, i.e. $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. Also, for $i > j$, define $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$. Thus the following equality holds in general:

$$\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*.$$

We will write for shorthand $\alpha_i := \|\mathbf{b}_i^*\|/\|\mathbf{b}_{i+1}^*\|$, and $Q_i = (\alpha_i^{-2} + \mu_{i+1,i}^2)^{-1/2}$. When discussing lattices, $r_i := \log \alpha_i$, and when discussing sandpiles, r_i refers to the "amount of sand" at vertex i.

2. Modeling LLL by a sandpile

2.1. **The LLL algorithm.** We briefly review the LLL algorithm; for details, we recommend [22], in which it is first introduced, and also [17] and [28]. A pseudocode for the LLL algorithm is provided in Algorithm 1. Be reminded that, whenever we mention LLL, we are really referring to its Siegel variant.

Algorithm 1 The LLL algorithm (Siegel variant)

---

0. Input: a basis $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ of $\mathbb{R}^n$, a parameter $\delta < 0.75$
1. while true, do:
2.   Size-reduce $\mathcal{B}$.
3.   (Lovász test) choose the lowest $k \in \{1, \ldots, n-1\}$ such that $\delta\|\mathbf{b}_k^*\|^2 > \|\mathbf{b}_{k+1}^*\|^2$
4.   if there is no such k, break
5.   swap $\mathbf{b}_k$ and $\mathbf{b}_{k+1}$ in $\mathcal{B}$
6. Output $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$, a δ-reduced LLL basis.

Proposition 1. *After carrying out Step 5 in Algorithm 1, the following changes occur:*

(i) $\alpha_{k-1}^{new} = Q_k \alpha_{k-1}$
(ii) $\alpha_k^{new} = Q_k^{-2} \alpha_k$
(iii) $\alpha_{k+1}^{new} = Q_k \alpha_{k+1}$

FIGURE 1. An illustration of a (legal) toppling T_i.

(iv) $\mu^{new}_{k,k-1} = \mu_{k+1,k-1}$
(v) $\mu^{new}_{k+1,k} = Q^2_k \mu_{k+1,k}$
(vi) $\mu^{new}_{k+2,k+1} = \mu_{k+2,k} - \mu_{k+2,k+1}\mu_{k+1,k}$
(vii) $\mu^{new}_{k,l} = \mu_{k+1,l}, \mu^{new}_{k+1,l} = \mu_{k,l}$ for $1 \le l \le k-1$
(viii) $\mu^{new}_{l,k} = \mu_{l,k+1} - \mu_{l,k+1}\mu_{k+1,k}\mu^{new}_{k+1,k} + \mu_{l,k}\mu^{new}_{k+1,k}$ for $l \ge k+2$
(ix) $\mu^{new}_{l,k+1} = \mu_{l,k} - \mu_{l,k+1}\mu_{k+1,k}$ for $l \ge k+2$

and there are no other changes. The superscript "new" refers to the corresponding variable after the swap.

Proof. Straightforward calculations (see e,g, [22]).   □

2.2. **Sandpile basics.** We also briefly review the basics of the sandpile models. For references, see Dhar ([10], [11]) or Perkinson ([30]).

A sandpile model is defined on a finite graph $\mathcal{G}$, with one distinguished vertex called the *sink*. In the present paper, we only concern ourselves with the cycle graph, say A_n, consisting of vertices $\{v_1, \ldots, v_n\}$ and one unoriented edge for each adjacent pair v_i and v_{i+1}. We also consider v_1 and v_n as adjacent. We designate v_n as the sink.

A *configuration* is a function $r : \{v_1, \ldots, v_n\} \to \mathbb{R}$. Just as reduction algorithms work with bases, sandpile models work with configurations. We write for short $r_i = r(v_i)$. One may think of r_i as the amount or *height* of the pile of sand placed on v_i.

Just as LLL computes a reduced basis by repeatedly swapping neighboring basis vectors, sandpiles compute a *stable configuration* by repeated *toppling*. Let $T, I \in \mathbb{R}_{>0}$. A configuration is *stable* if $r_i \le T$ for all $i \ne n$. A *toppling operator* T_i $(i \ne n)$ replaces r_i by $r_i - 2I$, and r_{i-1} by $r_{i-1} + I$ and r_{i+1} by $r_{i+1} + I$. An illustration is provided in Figure 1. Applying T_i when $r_i > T$ is called a *legal toppling*. By repeatedly applying legal topplings, all excess "sand" will eventually be thrown away to the sink, and the process will terminate.

In our paper, T — *threshold* — will always be a fixed constant, but I — *increment* — could be a function of the current configuration, or a random variable, or both. In the former case, we say that the model is *nonabelian* — otherwise *abelian*. In the second case, we say that the model is *stochastic*. The (non-stochastic) abelian sandpile theory is quite well-developed, with rich connections to other fields of mathematics — see e.g. [23]. Other sandpile models are far less understood, especially the nonabelian ones.

2.3. **The LLL sandpile model.** Motivated by Proposition 1, especially the formulas (i) – (iii), we propose the following Algorithm 2, which we call the *LLL sandpile model*, or LLL-SP for short.

Algorithm 2 The LLL sandpile model (LLL-SP)

---

0. Input: $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$, $\mu_{2,1}, \ldots, \mu_{n,n-1} \in [-0.5, 0.5]$, a parameter $\delta < 0.75$
1. Rewrite $r_i := \log \alpha_i$, $\mu_i := \mu_{i+1,i}$ $T := -0.5 \log \delta$
2. while true, do:
3.   choose the lowest $k \in \{1, \ldots, n-1\}$ such that $r_k > T$
4.   if there is no such k, break
5.   subtract $2 \log Q_k$ from r_k
6.   add $\log Q_k$ to r_{k-1} (if $k - 1 \geq 1$) and r_{k+1} (if $k + 1 \leq n - 1$)
7.   (re-)sample $\mu_{k-1}, \mu_k, \mu_{k+1}$ uniformly from $[-0.5, 0.5]$
8. Output: real numbers $r_1, \ldots, r_{n-1} \leq T$

The only difference between LLL (Algorithm 1) and LLL-SP (Algorithm 2) lies in the way in which the μ's are replaced after each swap or topple. Our experimental results below demonstrate that this change hardly causes any difference in their behavior. A theoretical perspective is discussed at the end of this section.

The decision to sample μ_i's uniformly is largely provisional, though some post hoc justification is provided in Figure 4. One could refine the model by updating μ_i's with the formulas in Proposition 1, and then re-sampling $\mu_{i+2,i}$'s uniformly.

2.4. **Numerical comparisons.** Figure 2 shows the average shape of the output bases and configurations by LLL and LLL-SP. We also ran the same experiments where the method of choosing k is tweaked, e.g. choose randomly among eligible k's. We omitted them here due to limited space, but the same lessons are obtained anyway.

For each dimension $n = 80, 100, 120$, we ran each algorithm 5,000 times with the same set of input bases of determinant $\approx 2^{10n}$, generated using the standard method suggested in Section 3 of [27]. A point (i, y) in each plot indicates that the average of $r_i := \log \alpha_i$'s over those 5,000 outputs equal y. We used fpLLL ([12]) for the LLL algorithm.

One easily observes that the algorithms yield nearly indistinguishable outputs. In particular, since RHF can be computed directly from the r_i's by the formula

$$(1) \qquad \qquad \text{RHF} = \exp \left(\frac{1}{n^2} \sum_{i=1}^{n-1} (n - i) r_i \right),$$

we expect both algorithms to yield about the same RHF. Indeed, Figure 3 shows that the RHF distribution of LLL and LLL-SP are in excellent agreement.

The resemblance of the two algorithms runs deeper than on the level of output statistics. See Figure 4, which depicts the plot of points $(i, Q_{k(i)}^{-2})$ and $\mu_{k(i)+1,k(i)} = \mu_{k(i)}$ as we ran LLL and LLL-SP on dimension 80, where $k(i)$ is k chosen at i-th iteration.[1] The two plots are again indistinguishable, yet another evidence that LLL and LLL-SP possess nearly identical dynamics.

---

[1] We have the same results in higher dimensions, but they are too cumbersome to present here.

FIGURE 2. Average output of LLL (orange square) and LLL-SP (blue circle) in dimensions 80, 100, and 120.

FIGURE 3. RHF distributions of LLL and LLL-SP in dimension 120. We obtain the same results in other dimensions.

FIGURE 4. Plot of $Q_{k(i)}^{-2}$(top) and $\mu_{k(i)+1,k(i)} = \mu_{k(i)}$(bottom) during a typical run of LLL(left) and LLL-SP(right) in dimension 80.

2.5. **Discussion.** The only difference between LLL and LLL-SP has to do with the way they update the $\mu_k(=\mu_{k+1,k})$'s. For LLL-SP, the μ_k-variables are i.i.d. and independent of the r_k-variables. For LLL, μ_k is determined by a formula involving its previous value and r_k. However, it seems plausible that the μ_k's in LLL, as a stochastic process, is *mixing*, which roughly means that they are close to being i.i.d, in the sense that a small perturbation in μ_k causes the next value μ_k^{new} to become near unpredictable. Numerically, this is robustly supported by the graphs at the bottom of Figure 4. Theoretically, our intuition comes from the fact that the formula $\mu_k^{new} = \mu_k/(\mu_k^2 + \alpha_k^{-2})$ (mod 1) is an approximation of the Gauss map $x \mapsto \{1/x\}$, which is well-known to have excellent mixing properties (see e.g. Rokhlin ([31]) and the references in Bradley ([7]) for more recent works).

The above discussion can be summarized and formulated in the form of a mathematical conjecture, which can then be considered a rigorous version of the statement "LLL is essentially a sandpile model." Below is our provisional formulation of such a conjecture.

Conjecture 2. *Choose a distribution $\mathcal{D}$ on the set of bases in $\mathbb{R}^n$, to be used to sample inputs for LLL. Define $k(i)$, as earlier, to be the index of the pile toppled at i-th iteration. Then $k(i)$ is a random variable depending on the input distribution, and so is $\mu_{k(i)}$. Then, if $\mathcal{D}$ is "generic," then*

  (i) *$(|\mu_{k(i)}|)_{i=1,2,\dots}$ is strongly mixing as a stochastic process. (Roughly speaking, this means $|\mu_k(N)|$ is nearly independent of $|\mu_k(M)|$ for which $N - M$ is large; see the text [6] for a precise definition.)*
 (ii) *each $|\mu_{k(i)}|$ is contained in a compact subset S of the set of all probability density functions on $[0, 0.5]$ with respect to the L^∞-norm. S is independent of the dimension, the input distribution, or any other variable.*

Ideally, Conjecture 2 is to be designed so that what is provable for LLL-SP would also be provable for LLL by an analogous argument (e.g. the theorems in Section 4), while retaining the flexibility as to what the correct distribution of μ_k might be. It is to be updated accordingly as our understanding of LLL and LLL-SP progresses. Then at some point Conjecture 2 may come within reach of mathematics; or maybe it already is.

As for the expression "generic" in the statement of Conjecture 2, it is hard to pin down its precise meaning at this point, as is sometimes the case in mathematics. But there are two criteria that a generic $\mathcal{D}$ must fulfill, which we suspect are also sufficient conditions. First, most of the samples from $\mathcal{D}$ must not ruin the mixing property of μ_k^{new}; a counterexample is a basis of a lattice with a huge discrepancy between two successive minima $\lambda_k \ll \lambda_{k+1}$, which would cause $\mu_{k(i)}$ to be abnormally small whenever $k(i) = k$. The second criterion concerns the shape of supp $\mathcal{D}$, which we illustrate in the next section.

3. Abelian sandpile analogue of LLL

The drawback of LLL-SP as a model of LLL is that, being nonabelian, it is difficult to study. The existing literature on nonabelian sandpile models is rather thin, and to the best of our knowledge, there is no theoretical treatment on this subject. We hope that our work here provides some motivation to pursue nonabelian models in depth.

In this section, we introduce a certain abelian stochastic sandpile model that we named SSP, which is in a sense an abelianized version of LLL-SP. A priori, SSP appears completely unrelated to LLL. Surprisingly, though, its average output shape turns out to be extremely close to that of LLL. Moreover, SSP admits a mathematical theory that is analogous to that of ASM developed by Dhar ([9], see also [11]), which is developed in a separate paper

by the second-named author ([21]). This allows one to prove some pleasant statements such as Theorem 3. Therefore, SSP is a useful toy model that could yield insights into some of the most prominent features of the output statistics of LLL. It may yield some hints as to how to start analyzing the non-abelian LLL-SP as well.

3.1. **Background on ASM.** To facilitate reader's understanding, we briefly describe the abelian sandpile model (ASM), the most basic of sandpile models, and parts of its theory that is relevant to us. Its pseudocode is provided in Algorithm 3. See Dhar ([9]), where the theory is originally developed, or Perkinson ([30]) for an articulate and readable exposition.

Algorithm 3 Abelian sandpile model (ASM)

---

0. Input: $r_1, \ldots, r_{n-1} \in \mathbb{Z}$, parameters $T, I \in \mathbb{Z}$, $0 < I \leq T/2$
1. while true, do:
2.   choose a $k \in \{1, \ldots, n-1\}$ such that $r_k > T$
3.   if there is no such k, break
4.   subtract $2I$ from r_k
5.   augment I to r_{k-1} and r_{k+1}
6. Output: integers $r_1, \ldots, r_{n-1} \leq T$

The important ASM concepts for us are that of the *recurrent configurations* and the *steady state*. Let M be the set of all stable (non-negative) configurations of ASM. Given two configurations $r, s \in M$, we have the operation

$$r \oplus s = (\text{stabilization of } r + s),$$

which is the outcome of ASM with input being the configuration $r+s$ defined by $(r+s)_i = r_i + s_i$ for each i. Unlike LLL, the output of ASM is independent of the choice of toppling order — hence the term "abelian" — and thus $\oplus$ is well-defined. This operation makes M into a commutative monoid.

Define $g \in M$ to be the configuration with $g_1 = 1$ and $g_2 = \ldots = g_{n-1} = 0$. We call $r \in M$ *recurrent* if

$$\underbrace{g \oplus \ldots \oplus g}_{m \text{ times}} = r \text{ for infinitely many } m.$$

One can actually take any g for which at least one g_i is coprime to the g.c.d. of T and I (this condition is nothing but only to avoid concentration on a select few congruence classes). Equivalently, with LLL in mind, we can also define that r is recurrent if there exist infinitely many input configurations such that their stabilization results in r. It is a theorem that the set R of the recurrent configurations of ASM forms a group under $\oplus$.

One may ask, given an $r \in R$, what is the proportion of $m \in \mathbb{Z}_{>0}$ that satisfies $g \oplus \ldots \oplus g \, (m \text{ times}) = r$? It turns out that the answer is $1/|R|$ for any $r \in R$, that is, each element of R has the same chance of appearing. This distribution, say ρ, on R is called the *steady state* of the system. And the phrase *average output shape* that we have been using in the empirical sense obtains a formal definition as $\sum_{r \in R} \rho(r) r$. The steady state is unique in the following sense: choose an $r \in R$ according to ρ, and take any configuration s; then the stabilization of $r + s$, which is a random variable, has distribution ρ.

Equivalently, this means that, in terms of our second definition of recurrent, if we sample the input configuration for ASM from a "generic" distribution e.g. uniformly from a large rectangular set of $\mathbb{Z}_{>0}^{n-1} \cong$ (the space of nonnegative configurations), we would

obtain each $r \in R$ with about probability $\rho(r)$. To elaborate, in the case of ASM, the set of inputs whose output becomes a given $r \in R$ is contained in a coset $r + \Lambda$ of a sublattice $\Lambda \subseteq \mathbb{Z}^{n-1}$. So the word "generic" here means "containing about the same number of each coset representatives of Λ." This is essentially the second condition for genericity that we mentioned in the discussion after Conjecture 2.

3.2. **Introduction to SSP.** A pseudocode for SSP is provided in Algorithm 4. This is exactly the same as ASM, except for Step 4, which determines the amount of sand to be toppled at random. The decision to sample from the uniform distribution is an arbitrary one; we could have chosen something else, and much of the discussion below still apply.

Algorithm 4 Stochastic sandpile (SSP)

---

0. Input: $r_1, \ldots, r_{n-1} \in \mathbb{Z}$, parameters $T, I \in \mathbb{Z}$, $0 < I \le T/2$
1. while true, do:
2.   choose a $k \in \{1, \ldots, n-1\}$ such that $r_k > T$
3.   if there is no such k, break
4.   sample γ uniformly from $\{1, \ldots, I\}$
5.   subtract 2γ from r_k
6.   augment γ to r_{k-1} and r_{k+1}
7. Output: integers $r_1, \ldots, r_{n-1} \le T$

The average output shape of this stochastic sandpile model (SSP) is shown in Figure 5. Figure 5 not only shares all the major characteristics of Figure 2, but they are also quantitatively alike. The values r_i's are nearly identical in the middle, which gradually decreases as i approaches the boundary, starting at around $i = 15$ and $n-15$. Furthermore, in both figures, the differences between the threshold and the middle values, and the differences between the middle and the boundary value, are equal; in the case of SSP, it is $\approx I/4$, and for LLL, it equals about 0.08. Outside the paradigm we are developing here, this should come as quite surprising. Algorithms 1 and 4 are simply so different that there is no reason to expect that anything similar would come out of them.

This finding suggests that sandpiles may shed light on the geometric series assumption(GSA) and its partial failure at the boundary, yet another important unexplained phenomenon of LLL-like reduction algorithms. Indeed, "GSA" is a quite general phenomenon that is fairly well-understood in statistical physics, via the finite-size scaling theory (see e.g. [15]). We will explore this connection in a forthcoming paper.

3.3. **Mathematical properties of SSP.** A mathematical theory of SSP closely analogous to that of ASM has been recently developed ([21]), largely motivated by the experimental result above. Every aspect of the above-mentioned ASM theory has its appropriate counterpart in the SSP theory, except that instead of configurations one works with a distribution on the set of configurations, due to its stochastic nature. Most importantly, SSP, like ASM, possesses the unique steady state. Figure 5 is a reflection of that steady state of SSP, from which one can compute its average "RHF" via (1).

As some readers might have noticed in the earlier discussion, the notion of the steady state alone clarifies numerous aspects of the practical behavior of LLL. The number "1.02" obtains a neat mathematical meaning as a certain invariant of the steady state. Moreover, that the steady state is unique — i.e. it is the only attractor in the associated dynamics — explains why the number 1.02 seems independent of how the input bases are sampled for LLL, an observation in [26] that is also declared as a conjecture there. Therefore, most

FIGURE 5. Average output of SSP, $n = 100$, $I = 200$ and $T = 400$.

of the frequently asked questions regarding the average output of LLL — why 1.02, why GSA, why independent of input sampling — comes down to studying the quantitative properties of the steady state.

There are two difficulties. First, we do not yet know how to prove that LLL indeed possesses a steady state, even upon fixing the order of toppling. Second, studying the steady state is nontrivial, even for ASM or SSM, where it reduces to tricky problems in combinatorics. Still, we can prove some of the statements for SSP that one wishes of LLL. For example, it is possible to rigorously bound the average RHF of SSP from above:

Theorem 3. *The worst-case* $\log{(\mathrm{RHF})}$ *of SSP is* $T/2 + o_n(1)$. *The average* $\log{(\mathrm{RHF})}$ *of SSP is bounded from above by* $T/2 - I/2e^2 + o_n(1)$.

We note that empirically one observes $\log{(\mathrm{RHF})} \approx T/2 - I/8$ on average.

Sketch (and discussion) of proof. This is essentially Proposition 8 of [21]. We present the idea of the proof for completeness; it takes only a moderate amount of labor to turn it into a formal argument.

Take an unstable configuration r. If r is sufficiently far away from the origin in the configuration space, we must topple on each and every vertex at least once — in fact, arbitrarily many times — on the way of stabilizing r. So consider $T_1 T_2 \ldots T_{n-1} r$, where T_i is the toppling operator on vertex i. Since the outcome of SSP is a random variable, $T_1 T_2 \ldots T_{n-1} r$ should be thought of as a probability distribution on the configuration space. As such, it is a distribution that is supported on a parallelepiped-shaped cluster, as illustrated in the top of Figure 6 in case $n = 3$ and $I = 4$; the upper-right vertex in the parallelogram is $r - (1, 1, \ldots, 1)$, which is r toppled on every vertex by the minimum possible amount.

Applying T_i to this parallelepiped-shaped distribution amounts to "pushing" the parallelepiped in the direction of i, resulting in another parallelepiped-shaped distribution. The middle graph in Figure 6 illustrates this process, by indicating with x marks the outcome of applying T_1 to the original distribution (assuming that the horizontal axis represents r_1). Repeating, we eventually reach the situation as in the bottom of Figure 6, where none of the T_i would preserve the shape of the parallelepiped, since $(T, T, \ldots, T)$ is already a stable configuration and thus T_i leaves it there. From this point on, it is rather

FIGURE 6. The parallelepiped argument.

tricky to describe the action of the T_i's, which is the source of the difficulty of studying the steady state of SSP.

However, we claim that, for any r sufficiently far enough from the origin, the distribution on the parallelepiped obtained by the time the upper-right corner reaches $(T, \ldots, T)$ is arbitrarily close to a certain limiting distribution $\wp$. To see this, consider the action of T_i on the distribution on the parallelepiped, while forgetting the information about where that parallelepiped is located in the configuration space. Then one notices that each T_i

acts as a linear operator on the space of such distributions. Simultaneously diagonalizing all T_i's — possible because they pairwise commute — one finds that 1 is the single largest eigenvalue of multiplicity one, whose corresponding eigenvector is $\wp$. Upon repeated applications of T_i's, the components corresponding to the lesser eigenvalues converge to zero, proving the claim.

Observe that, when the upper-right corner equals $(T, \ldots, T)$ (again see bottom of Figure 6), fully stabilizing $\wp$ yields the steady state. $\wp$ itself is easily computed by hand, but the steady state is not. Still, we can prove, using the information about $\wp$, that the maximum density of the steady state occurs at $(T, \ldots, T)$, and that its value equals $\approx (I/2)^{-(n-1)}$. Note that this is enough to deduce a nontrivial upper bound on the average RHF: estimate the number $N(\alpha)$ of stable configurations whose $\log(\text{RHF})$ are greater than α, and take α such that $N(\alpha) \cdot (I/2)^{-(n-1)}$ vanishes as $n \to \infty$. It turns out we can choose $\alpha = T/2 - I/2e^2$.

$\square$

There are a couple of difficulties in directly applying the above argument to LLL (even assuming Conjecture 2) or LLL-SP. For instance, because the increment depends on the r_i's for those systems, the effect of T_i is not as neat as illustrated in Figure 6. It would push the side of the parallelepiped with "uneven force," skewing the shape of the parallelepiped. Specifically, for LLL or LLL-SP, the higher r_i is, the greater is the toppling induced by T_i, so the upper side is pushed further than the lower side. Technically speaking, since the T_i's depend on the r_i-coordinate it is impossible to separate the parallelepiped from the coordinate space as we have done above, and since the T_i's do not commute we cannot use linear algebra to find the attractor of the dynamics. These are the obstacles to proving the existence of the steady state of LLL.

To prove that the "average" RHF is bounded strictly away from the worst-case, it is sufficient to show a much weaker statement that the maximum density of the output distribution is not too large. This seems feasible yet quite vexing; we state it as a conjecture below for future reference. As in the SSP case, we expect that the maximum density is attained on the upper-right corner, and the T_i's perturb it at most marginally, once they are iterated sufficiently many times.

Conjecture 4. *For a generic distribution $\mathcal{D}$ on the set of bases of $\mathbb{R}^n$, the probability density function of the corresponding output distribution $\mathcal{D}^\circ$ of LLL (or LLL-SP) is bounded from above by a constant C that depends only on n.*

4. Regarding time complexity

Although expanding the SSP theory, and Theorem 3 in particular, to LLL-SP seems challenging for the time being, we are able to prove some attractive statements for LLL-SP with respect to its complexity, which we present below. We also consider their extensions to LLL assuming the truth of Conjecture 2.

4.1. **A lower bound.** The theorem below gives a probabilistic *lower* bound on the complexity of LLL-SP, which agrees up to constant factor with the well-known upper bound. There are two ingredients in the proof: (i) measuring the progress of the LLL algorithm by the quantity *energy*, a well-known idea from the original LLL paper ([22]) (ii) bounding the performance of LLL-SP by a related SSP.

Theorem 5. *Consider LLL-SP, and an input configuration r whose* log-energy $E = E(r)$, *defined by*

$$E(r) = \sum_{j=1}^{n-1} \sum_{i=j}^{n-1} (n-i) r_i,$$

is sufficiently large — in fact, $E > 10H$ works, with H defined as in (2). Then the probability that LLL-SP is not terminated in $E/4$ steps is at least $1 - CE^{-1/2}$ for an absolute constant $C > 0$.

Observe that the familiar upper bound $O(n^2 \log \max_i \|\mathbf{b}_i\|)$ on the number of required steps is equivalent to $O(E)$, with the implicit constant depending on δ.

Proof. If the algorithm is terminated, then E must have become less than

$$\sum_{i=1}^{n} (n-i+1)(n-i) T/2,$$

(where $T := -\log \delta^{1/2} > 0$) which equals,

(2) $$H := \frac{T}{6}(n^3 - n).$$

Taking converse, we see that if E is greater than (2), then LLL-SP has not yet terminated. At k-th toppling, E decreases by at most $\log \mu_{k(i)}^{-2}$, where $k(i)$ is the index of the vertex in which i-th toppling occured. If toppled N times, the decrease in E is bounded by at most $F_N := \sum_{i=1}^{N} \log \mu_{k(i)}^{-2}$. In sum,

(3) $$\mathrm{Prob}(E - F_N > H)$$

gives the lower bound on the probability that LLL-SP is not terminated after N swaps. Hence, it suffices to show that (3) is bounded from below by $1 - CE^{-1/2}$ when $N = E/2$.

The central limit theorem is applicable on F_N, since $\mu_{i(k)}$ are i.i.d. More precisely, we apply the Berry-Esseen theorem, which asserts the following. Suppose we have i.i.d. random variables $X_1, X_2, \ldots$, so that $m = \mathbb{E}(X_1)$, $\sigma = (\mathbb{E}(X_1^2) - \mathbb{E}(X_1)^2)^{1/2}$, and $\rho = \mathbb{E}(X_1^3)$ are all finite. Furthermore, let $Y_N = \sum_{i=1}^{N} X_i$, and let $G_N(x)$ be the cumulative distribution function of Y_N, and $\Phi_N(x)$ be the cumulative distribution function of the normal distribution $N(Nm, N\sigma^2)$. Then for all x and N,

$$|G_N(x) - \Phi_N(x)| = O(N^{-1/2}),$$

where the implied constant depends on m, σ, ρ only.

We let $X_i = \log \mu_{k(i)}^{-2}$ so that $F_N = G_N$, and apply the Berry-Esseen. It is easy to compute and check that m, σ, ρ are all finite e.g. $m = 2(1 + \log 2) \approx 3.386$ and $\sigma = 2$. Then, for a random variable $\mathcal{N}_N \sim N(Nm, N\sigma^2)$, (3) is bounded by

$$\mathrm{Prob}(E - \mathcal{N}_N > H)$$

plus an error of $O(N^{-1/2})$.

Now choose $N = E/4$, so that $\mathcal{N}_N \sim N((1 + \log 2)E/2, E)$. Using Chebyshev's inequality we can prove

$$\mathrm{Prob}(\mathcal{N}_N \geq 0.9E) \leq O(E^{-1}),$$

where the implied constant is absolute. Thus if E is large enough so that $E - H > 0.9E$, we have that (3) is at least $1 - CE^{-1/2}$ for some $C > 0$, as desired. With marginally more effort, it is possible to determine an explicit value for C.

$\square$

Remark. 1. We can use the same idea to obtain a lower bound on the average RHF of LLL-SP, but it turns out to be slightly less than 1, which happens to be useless in the context we are in.

2. There exists a central limit theorem for a strong mixing process ([6]), and also a central limit theorem for a sequence of independent but non-identical sequence of random variables (e.g. the Lyapunov CLT). Conjecture 2 states that the $|\mu_{k(i)}|$ of LLL is strong mixing (weaker than independent) and non-identical (though contained in a compact set). We do not know whether there exists a central limit theorem that applies in this context, though we suspect that there should be.

4.2. **The optimal LLL problem.** The optimal LLL problem (see e.g. [1]) asks whether LLL with the optimal parameter $\delta = 3/4$ terminates in polynomial time. The following theorem, while crude, shows that this is true for LLL-SP with arbitrarily high probability.

Theorem 6. *For any $\eta > 0$ small, LLL-SP with $\delta = 3/4$ terminates after $O_\eta(E)$ steps with probability $1 - \eta$.*

Proof. Write μ for the random variable uniformly distributed in $[0, 1/2]$. In case $\delta < 3/4$, the complexity bound of LLL is established with the observation that, with each swap, the energy E decreases by at least $c := \log(\delta + 1/4)^{-1} > 0$, and thus the algorithm must terminate within E/c steps. Similarly, in case $\delta = 3/4$, we try to show that the minimum change of energy $\log(\delta + \mu^2)^{-1}$ is strictly bounded away from zero almost all the time.

(If I was the increment for a given toppling operation, it is easy to show that the energy decreases by $2I$ after such a step.)

Choose a small $\varepsilon > 0$, and let $p = \text{Prob}(\mu \leq 1/2(1 - \varepsilon)) = 1 - \varepsilon$. Let $d = \log(3/4 + p^2/4)^{-1}$, which is the minimum possible change in energy provided $\mu \leq 1/2(1 - \varepsilon)$. Now take $10E/d$ samples $\mu_1, \mu_2, \ldots$ of μ (there is nothing special about the constant 10 here). If at least E/d of those samples are less than $1/2(1-\varepsilon)$, LLL-SP would terminate. Proving that this probability is arbitrarily close to 1 is now a simple exercise with the binomial distribution.

$\square$

Observe that the above proof carries over to the case of LLL assuming Conjecture 2; the compactness condition on the $\mu_{k(i)}$ distributions allows control on the probability that they are all simultaneously bounded away from $1/2(1 - \varepsilon)$.

REFERENCES

[1] A. Akhavi. Worst-case complexity of the optimal LLL algorithm. *LATIN 2000: Theoretical Informatics*, 355-366.

[2] M. Albrecht, B. Curtis, A. Deo, Alex Davidson, Rachel Player, E. Postlethwaite, Fernando Virdia, Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! Available at https://github.com/estimate-all-the-lwe-ntru-schemes.

[3] P. Bak, C. Tang, and K. Wieselfeld. Self-organized criticality: an explanation of $1/f$ noise. *Phys. Rev. Lett.*, 59:381-384, 1987.

[4] S. Bai, D. Stehlé, W. Wen. Measuring, simulating and exploiting the head concavity phenomenon in BKZ. *Advances in cryptology — ASIACRYPT 2018*. Part I, 369-404, Lecture Notes in Comput. Sci., 11272, Springer, Cham, 2018.

[5] D. Bernstein and T. Lange. Post-quantum cryptography. *Nature*, 549(7671):188-194, 2017.

[6] P. Billingsley. Probability and Measure, 3rd ed. John Wiley & Sons, 1995.

[7] R. C. Bradley. Basic properties of strong mixing conditions. A survey and some open questions. *Probability Surveys*, Vol. 2, 107-144, 2005.

[8] J. Cardy (ed). Finite-size scaling. Elsevier Science Publishers B.V., 1988.

[9] D. Dhar. Self-organized critical state of sandpile automaton models. *Phys. Rev. Lett.*, 64(14):1613-1616, 1990.

[10] D. Dhar. The abelian sandpile and related models. *Physica A*, 263(1999) vol. 4, 4-25.

[11] D. Dhar. Theoretical studies of self-organized criticality. *Physica A*, 369(2006) 29-70.

[12] The fpLLL team. fpLLL, a lattice reduction library. Available at https://github.com/fplll/fplll.

[13] N. Gama, P. Nguyen. Predicting lattice reduction. *Advances in cryptology — EUROCRYPT 2008*, 31-51, *Lecture Notes in Comput. Sci.*, 4965, Springer, Berlin, 2008.

[14] C. Gentry. A fully homomorphic encryption scheme. PhD thesis, Stanford University, 2009.

[15] P. Grassberger, D. Dhar, and P. K. Mohanty. Oslo model, hyperuniformity, and the quenched Edwards-Wilkinson model. *Physical review E* 94, 042314 (2016).

[16] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. *Advances in cryptology—CRYPTO 2011*, 447-464, Lecture Notes in Comput. Sci., 6841, Springer, Heidelberg, 2011.

[17] A. Joux and J. Stern. Lattice reduction: a toolbox for the cryptanalyst. *J. Cryptology* (1998) 11: 161. https://doi.org/10.1007/s001459900042

[18] S. Kim. On the shape of a high-dimensional random lattice. PhD thesis, Stanford University, 2015.

[19] J. Kim, M. Kim, and S. Kim. LLL via the Lenstra graph. Preprint. Available at https://sites.google.com/view/seungki/

[20] S. Kim and A. Venkatesh. The behavior of random reduced bases. *Int. Math. Res. Not.*, rnx074.

[21] S. Kim and Y. Wang. A stochastic variant of the abelian sandpile model. Preprint. Available at https://sites.google.com/view/seungki/

[22] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515-534, 1982.

[23] L. Levine. What is . . . a sandpile? *Notices Amer. Math. Soc.* 57 (2010), no. 8, 976-979.

[24] National Institute of Standards and Technology. Post-quantum cryptography, call for proposals. Available at https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization/Call-for-Proposals.

[25] A. Neumaier and D. Stehlé. Faster LLL-type reduction of lattice bases. *Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation*, 373-380, ACM, New York, 2016.

[26] P. Nguyen and D. Stehlé. Floating-point LLL revisited. *Advances in cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Comput. Sci.*, 215-233. Springer, Berlin, 2005.

[27] P. Nguyen and D. Stehlé. LLL on the average. *Algorithmic number theory*, volume 4076 of *Lecture Notes in Comput. Sci.*, pages 238 - 256. Springer, Berlin, 2006.

[28] P. Nguyen and B. Vallee (eds). The LLL Algorithm: Survey and Applications. Springer, 2010.

[29] M. Madritsch and B. Vallé. Modelling the LLL algorithm by sandpiles. *LATIN 2010: theoretical informatics*, 267-281, *Lecture Notes in Comput. Sci.*, 6034, Springer, Berlin, 2010.

[30] D. Perkinson. Notes for AIMS Cameroon: Abelian Sandpile Model. Available at http://people.reed.edu/ davidp/cameroon

[31] V. Rohlin. Exact endomorphisms of a Lebesgue space. *Izvest. Akad. Nauk* 25 (1961), 499-530.

[32] T. Sadhu and D. Dhar. Steady state of stochastic sandpile models. *J. Stat. Phys.* (2009), 134: 427-441.

[33] P. Schnorr. Lattice reduction by random sampling and birthday methods. *STACS 2003*, 145-156, *Lecture Notes in Comput. Sci.*, 2607, Springer, Berlin, 2003.

[34] P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming* 66 (1994), no. 2, Ser. A, 181-199.

[35] C.L. Siegel. A mean value theorem in geometry of numbers. Ann. of Math. 46(2) (1945), 340-347.

[36] Damien Stehlé. Floating-point LLL: theoretical and practical aspects. In Phong Nguyen and Brigitte Vallée (eds), *The LLL Algorithm, Survey and Applications*, Informaton Security and Cryptography, chapter 5, pages 179-213. Springer-Verlag, Berlin, Heidelberg, 2010.

[37] B. Vallée. Genealogy of lattice reduction: algorithmic description and dynamical analyses. Preprint.

[38] Y. Yu and L. Ducas. Second Order Statistical Behavior of LLL and BKZ. *Selected Areas in Cryptography*, SAC 2017, pp.3-22