Phase transition and computational complexity in a stochastic prime number generator

L Lacasa\textsuperscript{1,3}, B Luque\textsuperscript{1} and O Miramontes\textsuperscript{2}

\textsuperscript{1} Departamento de Matemática Aplicada y Estadística, ETSI Aeronáuticos, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, Madrid 28040, Spain
\textsuperscript{2} Departamento de Sistemas Complejos, Instituto de Física, Universidad Nacional Autónoma de México, Mexico 01415 DF, Mexico
E-mail: lucas@dmae.upm.es

\textit{New Journal of Physics} 10 (2008) 023009 (16pp)
Received 13 August 2007
Published 8 February 2008
Online at http://www.njp.org/
doi:10.1088/1367-2630/10/2/023009

Abstract. We introduce a prime number generator in the form of a stochastic algorithm. The character of this algorithm gives rise to a continuous phase transition which distinguishes a phase where the algorithm is able to reduce the whole system of numbers into primes and a phase where the system reaches a frozen state with low prime density. In this paper, we firstly present a broader characterization of this phase transition, both in analytical and numerical terms. Critical exponents are calculated, and data collapse is provided. Further on, we redefine the model as a search problem, fitting it in the hallmark of computational complexity theory. We suggest that the system belongs to the class \textit{NP}. The computational cost is maximal around the threshold, as is common in many algorithmic phase transitions, revealing the presence of an easy-hard-easy pattern. We finally relate the nature of the phase transition to an average-case classification of the problem.
1. Introduction

Computer science and physics, although different disciplines in essence, have been closely linked since the birth of the former. More recently, computer science has met together with statistical physics in the so called combinatorial problems and their relation to phase transitions and computational complexity (see [1] for a compendium of recent works). For instance, Erdös and Renyi, in their pioneering work on graph theory [2], found the existence of zero-one laws in their study of cluster generation. These laws have a clear interpretation in terms of phase transitions, which appear extensively in many physical systems. The computer science community has recently detected this behavior in the context of algorithmic problems [3]–[11]. The so called threshold phenomenon [1] distinguishes zones in the phase space of an algorithm where the problem is, computationally speaking, either tractable or intractable. It is straightforward that these three phenomena can be understood as a unique concept, in such a way that building bridges between them is an appealing idea.

Related to the concept of a phase transition is the task of classifying combinatorial problems. The theory of computational complexity distinguishes problems which are tractable, that is to say, solvable in polynomial time by an efficient algorithm, from those which are not. The so-called *NP* (nondeterministic polynomial time) class gathers problems that can be solved in polynomial time by a non-deterministic Turing machine [4]. This class generally includes many hard or eventually intractable problems, although this classification is denoted *worst-case*, that is to say, a rather pessimistic one, since the situations that involve long computations may ultimately be rare. In the last few years, numerical evidence suggests the presence of the threshold phenomenon in *NP* problems. These phase transitions may in turn characterize the *average-case* complexity of the associated problems, as pointed out recently [6].

In this paper, we discuss a stochastic algorithm [12]–[14] that generates prime numbers by means of a stochastic integer decomposition. The paper is organized as follows: in section 2, we will describe the model, which stands as a stochastic prime number generator. In section 3, we will characterize the phase transition present in the system, following the steps depicted in...
2. The model

Suppose [12], that we have a pool of positive integers \{2, 3, \ldots, M\}, from which we randomly extract a certain number \(N\) of them (this will constitute the system under study). Note that the chosen numbers can be repeated, and that the integer 1 is not taken into account. Now, given two numbers \(n_i\) and \(n_j\) taken from the system of \(N\) numbers, the algorithm division rules are the following:

1. Rule 1: if \(n_i = n_j\) there is no division, and the numbers are not modified.
2. Rule 2: if the numbers are different (say \(n_i > n_j\)), a division will take place only if \(n_j\) is a divisor of \(n_i\), i.e. if \(n_i \mod n_j = 0\). The algorithm’s outcome is then schematized as

   \[ n_i \oplus n_j \mapsto n_k \oplus n_j, \quad (1) \]

   where \(n_k = (n_i/n_j)\).
3. Rule 3: if \(n_i > n_j\) but \(n_i \mod n_j \neq 0\), no division takes place.

The result of a division will be the extinction of \(n_i\) and the introduction of a smaller one, \(n_k\).

The algorithm goes as follows: after randomly extracting from the pool \{2, 3, \ldots, M\} a set of \(N\) numbers, we pick at random two numbers \(n_i\) and \(n_j\) from the set. We then apply the division rules. In order to have parallel updating, we will establish \(N\) repetitions of this process (\(N\) Monte Carlo steps) as a time step. Note that the algorithm rules tend to reduce numbers, hence this dynamic when iterated may generate prime numbers in the system. We say that the system has reached stationarity when no more divisions can be achieved, whether because every number has become a prime or because rule 2 cannot be satisfied in any case—a frozen state. The algorithm then stops.

3. Phase transition

3.1. Preliminary insight

As stated in the previous section, this algorithm clearly tends to generate primes as far as possible: when the algorithm stops, one may expect the system to have a large number of primes or at least have a frozen state of non-divisible pairs. A first indicator that can evaluate properly
Figure 1. Numerical simulation of the steady values of $r$ versus $N$, for a pool size $M = 2^{14}$. Each run is averaged over $2 \times 10^4$ realizations in order to avoid fluctuations. Note that the system exhibits a phase transition which distinguishes a phase where every element of the system becomes a prime in the steady state and a phase with low prime density.

This feature is the unit percentage or ratio of primes $r$, that a given system of $N$ numbers reaches at stationarity [14]. In figure 1, we present the results of Monte Carlo simulations calculating, as a function of $N$ and for a concrete pool size $M = 2^{14}$, the steady values of $r$. Every simulation is averaged over $2 \times 10^4$ realizations in order to avoid fluctuations. We can clearly distinguish two phases in figure 1, a first one where $r$ is small and a second one where the prime number concentration reaches unity. This is the portrait of a phase transition, where $N$ would stand as the control parameter and $r$ as the order parameter. In the phase with small $r$, the average steady state distribution of the $N$ elements is plotted in the left side of figure 2: the distribution is uniform (note that the vertical scale is zoomed in such a way that if we scale it between $[0, 1]$ we would see a horizontal line), which is related to a homogeneous state. In this regime, every number has the same probability of appearing in the steady state. On the other hand, the average steady state distribution of the $N$ numbers in the phase of high $r$ is plotted in the right side of figure 2: the distribution is now a power law, which is related to a biased, inhomogeneous state. In this regime, the probability of having—in the steady state—a composite number is practically null, and the probability of having the prime $x$ is in turn proportional to $1/x$. The breaking of this symmetry between steady distributions leads us to assume an order–disorder phase transition, the phase with small proportion of primes being the disordered phase and the ordered phase being the one where $r$ tends to one.

This is something quite intuitive if we take into account that there are typically $N/2$ multiples of two, $N/3$ multiples of three, and so on: the probability that the prime $x$ appears in a random composite number is on average $1/x$.\footnote{This is something quite intuitive if we take into account that there are typically $N/2$ multiples of two, $N/3$ multiples of three, and so on: the probability that the prime $x$ appears in a random composite number is on average $1/x$.}
Figure 2. The left figure stands for the steady state distribution (averaged over $2 \times 10^4$ realizations) of the $N$ elements, for $N = 10$ and $M = 10^4$ (phase with low $r$): this one is a uniform distribution $U(2, M)$ (note that the distribution is not normalized). The right figure stands for the same plot for $N = 110$ and $M = 10^4$ (phase where $r$ reaches unity): this one is a power law $P(x) \sim 1/x$.

Figure 3. Plot of $r$ versus $N$, for different pool sizes $M$ (each simulation is averaged over $2 \times 10^4$ realizations).

A second feature worth investigating is the size dependence of the transition. In figure 3, we plot $r$ versus $N$, for a set of different pool sizes $M$. Note that the qualitative behavior is size invariant, however the transition point increases with $M$. This size dependence will be considered in a later section.

As a third preliminary insight, we shall study the temporal evolution of the system. In figure 4 we plot, for a given pool size $M = 10^4$, the cumulated number of divisions that a system of $N$ numbers needs to make in order to reach stationarity. According to this, in figure 5 we plot, for the same ($N$, $M$), the evolving value $r(t)$. In the disordered phase, we can see that the system is rapidly frozen: the algorithm is not efficient in producing primes, and $r$ is asymptotically small. In the ordered phase the system needs more time to reach stationarity: this is due to the
Figure 4. Number of cumulated positive reactions (rule 2 is satisfied) as a function of the time steps, for three different configurations: (top) disordered phase $N \ll N_c$, (middle) critical phase $N \sim N_c$, (bottom) ordered phase $N \gg N_c$.

Figure 5. Ratio $r$ as a function of the time steps for the same configurations as for figure 4.

fact that the algorithm is producing many primes, as the evolving value of $r$ reaches unity. It is however in the neighborhood of the transition where the system takes longer times to reach the steady state: the system is producing many divisions, but not that many primes. This fact can be related to a critical slowing down phenomenon, and is studied later in the text.
It is worth noting in figures 1 and 3 that in the disordered phase the order parameter does not vanish, as it should. This is due to the fact that in a pool of $M$ numbers, following the prime number theorem, one finds on average $M/\log(M)$ primes [15]. Thus, there is always a residual contribution to the ratio $1/\log(M)$ not related to the system’s dynamics which only becomes relevant for small values of $N$, when the algorithm is not able to produce primes.

3.2. New order parameter

Let us now see how this phase transition can be understood as a dynamical process embedded in a network having integer numbers as the nodes. Consider two numbers of that network, say $a$ and $b$ ($a > b$). These numbers are connected ($a \rightarrow b$) if they are exactly divisible, that is to say, if $a/b = c$ with $c$ being an integer. The topology of similar networks has been studied in [16]–[18], concretely in [18] it is shown that this network exhibits scale-free topology [19]: the degree distribution is $P(k) \sim k^{-\lambda}$ with $\lambda = 2$. In our system, fixing $N$ is equivalent to selecting a random subset of nodes in this network. If $a$ and $b$ are selected they eventually can give $a/b = c$; in terms of the network this means that the path between nodes $a$ and $b$ is traveled thanks to the ‘catalytic’ presence of $c$. We may say that our network is indeed a catalytic one [20, 21] where there are no cycles as attractors but two different stationary phases: (i) for large values of $N$ all resulting paths sink into prime numbers, and (ii) if $N$ is small only a few paths are traveled and no primes are reached. Notice that in this network representation, primes are the only nodes that have input links but no output links (by definition, a prime number is only divisible by the unit and by itself, acting as an absorbing node of the dynamics). When the temporal evolution of this algorithm is explored for small values of $N$, we have observed in figures 4 and 5 that the steady state is reached very quickly. As a consequence, there are only few traveled paths over the network and since $N$ is small the probability of catalysis is small as well, hence the paths ending in prime nodes are not traveled. We say in this case that the system freezes in a disordered state. In contrast when $N$ is large enough, many divisions take place and the network is traveled at large. Under these circumstances, an arbitrary node may be catalyzed by a large $N - 1$ quantity of numbers, its probability of reaction being high. Thus, on average all numbers can follow network paths towards the prime nodes: we say that the system reaches an ordered state.

In light of the preceding arguments, it is meaningful to define a new order parameter $P$ as the probability that the system has for a given $(N, M)$ to reduce every number from $N$ into primes, that is to say, to reach an ordered state. In practice, $P$ is calculated in the following way: given $(N, M)$, for each realization we check, once stationarity has been reached, whether the whole set of elements are primes or not, and we subsequently count the fraction of runs in which all the remaining numbers are prime in the steady state.

In figure 6, we plot $P$ versus $N$ for different pool sizes $M$. The phase transition that the system exhibits has now a clear meaning: when $P = 0$, the probability that the system has to be able to reduce the whole system into primes is null (disordered state), and vice versa when $P \neq 0$.

In each case, $N_c(M)$, the critical value separating the phases $P = 0$ and $P \neq 0$, can now be defined. Observe in figure 6 that $N_c$ increases with the pool size $M$. In order to describe this size dependence, we need to find some analytical argument by means of which we can define a system’s characteristic size. As we will see in a few lines, this is not $M$ as one would expect at first.
3.3. Annealed approximation

The system under consideration shows highly complex dynamics: correlations take place between the $N$ numbers of the system at each time step in a nontrivial way. Finding an analytical solution to the former problem is thus completely out of the scope of this paper. However, an annealed approximation can still be performed. The main idea is to obviate these two-time correlations, assuming that at each time step, the $N$ elements are randomly generated. This way, we can calculate, given $N$ and $M$, the probability $q$ that at a single time step, no pair of numbers among $N$ are divisible. Thus, $1 - q$ will be the probability that there exists at least one reacting pair. Note that $1 - q$ will somehow play the role of the order parameter $P$, in this oversimplified system.

In a first step, we can calculate the probability $p(M)$ that two numbers randomly chosen from the pool $M$ are divisible:

$$p(M) = \frac{2}{(M - 1)^2} \sum_{x=2}^{\lfloor M/2 \rfloor} \left\lfloor \frac{M - x}{x} \right\rfloor \approx \frac{2 \log M}{M},$$

where the floor brackets stand for the integer part function. Obviously, $1 - p(M)$ is the probability that two numbers randomly chosen are not divisible in any case. Now, in a system composed of $N$ numbers, we can make $N(N - 1)/2$ distinct pairs. However, these pairs are not independent in the present case, so that the probability $q(N, M)$ is not simply $(1 - p(M))^N(N-1)/2$. Correlations between pairs must be somehow taken into account. At this point, we can make the following ansatz:

$$q(N, M) \approx \left(1 - \frac{2 \log M}{M}\right)^{N^{1/\alpha}}.$$  

Figure 6. Order parameter $P$ versus $N$, for the same pool sizes as figure 3 (averaged over $2 \times 10^4$ realizations). Note that $P$ is now a well-defined order parameter, as long as $P \in [0, 1]$. Again, $N_c$ depends on the pool size $M$.  

New Journal of Physics 10 (2008) 023009 (http://www.njp.org/)
where $\alpha$ characterizes the degree of independence of the pairs. The relation $1 - q(N, M)$ versus $N$ is plotted in figure 7 for different values of the pool size $M$. Note that for a given $M$, the behavior of $1 - q(N, M)$ is qualitatively similar to $P$, the order parameter in the real system.

For convenience, in this annealed approximation we will define a threshold $N_c$ as the one for which $q(N_c, M) = 0.5$. This value is the one for which half of the configurations reach an ordered state. This procedure is usual for instance in percolation processes, since the choice of the percolation threshold, related to the definition of a spanning cluster, is somewhat arbitrary in finite size systems [22]. Taking logarithms in equation (3) and expanding up to first order, we easily find a scaling relation between $N_c$ and $M$, that reads

$$N_c \sim \left[ \frac{M}{\log M} \right]^{\alpha}.$$  \hspace{1cm} (4)

This relation suggests that the system’s characteristic size is not $M$, as one would expect initially, but $M/\log(M)$. In figure 8, we plot, in log–log, the scaling between $N_c$ and the characteristic size $M/\log(M)$ that can be extracted from figure 7. The best fit provides a relation of the shape (4) where $\alpha = 0.48 \pm 0.01$ (note that the scaling is quite good, which gives consistency to the leading order approximations assumed in equation (4)).

3.4. Data collapse and critical exponents

The annealed approximation introduced in the preceding section suggests that the characteristic size of the system is not $M$ as one would expect but rather $M/\log(M)$. This is quite reasonable if we have in mind that the number of primes that a pool of $M$ integers has is on average $M/\log(M)$ [15]; the quantity of primes does not grow linearly with $M$. 

**Figure 7.** Numerical simulations calculating the probability $1 - q(N, M)$ (as explained in the text) versus $N$, for different values of pool size $M$, in the annealed approximation.
Figure 8. Scaling of $N_c$ versus the system’s characteristic size in the annealed approximation. The plot is log–log: the slope of the straight line provides the exponent $\alpha = 0.48$ of equation (4).

Figure 9. Scaling of the critical point $N_c$ versus the characteristic system size $M/\log(M)$ in the prime number generator, for pool size $M = \{2^{10} \ldots 2^{18}\}$. The plot is log–log: the slope of the curve provides an exponent $\alpha = 0.59$.

In order to test if this conjecture also applies to the prime number generator, in figure 9 we present (in log–log) the values of $N_c$ (obtained numerically from the values where $P(N, M)$ becomes non-null for the first time) as a function of $M/\log(M)$. We find the same scaling relation as for the annealed system (equation (4)), but with a different value for $\alpha = 0.59 \pm 0.05$ due to the obviation of correlations.
Let us apply generic techniques of finite size scaling in order to calculate the critical exponents of this phase transition. Reducing the control parameter as $n = \frac{N}{M \log(M)}$, the correlation exponent $\nu$ is defined as:

$$|n - n_c| \sim \left( \frac{M}{\log(M)} \right)^{-1/\nu},$$

(5)

where we find $n_c(\infty) = 0$ and $\nu = 1.69 \pm 0.05$. Note that the transition tends to zero in the thermodynamical limit because its value increases more slowly than the system’s size.

The critical exponent of the order parameter $\beta$ can be deduced from the calculation of the correlation exponent $\nu$ and from the finite size scaling relation

$$P(n_c) \sim \left( \frac{M}{\log(M)} \right)^{-\beta/\nu}.$$

(6)

The best fitting provides a value of $\beta = 3.4 \pm 0.2$.

In figure 10, we have collapsed all curves $P(N, M)$ from figure 6 according to the preceding relations. Note that the collapse is excellent, something which provides consistency to the full development.

4. Computational complexity

As pointed out in [7], phase transitions quite similar to the former one such as percolation processes for instance can be easily related to search problems. In the case under study we can redefine the system as a decision problem in the following terms: one could ask when is the clause every number of the system is prime when the algorithm reaches stationarity satisfied.
It is clear that through this focus, the prime number generator can be understood as a SAT-like problem [1], as long as there is an evident parallelism between the satisfiability of the preceding clause and our order parameter $P$. Thereby, in order to study the system from the focus of computational complexity theory, we must address the following questions: what is the algorithmic complexity of the system, and how is the observed phase transition related to the problem’s tractability?

4.1. Worst-case classification

The algorithm under study is related to both primality test and integer decomposition problems. Although primality was believed to belong to the so-called NP problems [23] (solvable in non-deterministic polynomial time), it has recently been shown to be in $P$ [24]: there exists at least an efficient deterministic algorithm that tests if a number is prime in polynomial time. The integer decomposition problem is in turn a harder problem, and to find an algorithm that would factorize numbers in polynomial time is an unsolved problem of computer science. Furthermore, exploring the computational complexity of the problem under consideration could eventually shed light on these aspects. For this task, let us determine how the search space grows when we increase $N$. In a given time step, the search space corresponds to the set of configurations that must be checked in order to solve the decision problem: this is nothing but the number of different pairs that can be formed using $N$ numbers. Applying basic combinatorics, the set of different configurations $G$ for $N$ elements and $N/2$ pairs is:

$$G(N) = \frac{N!}{2!^{N/2}(N/2)!} = (N - 1)!!.$$

(7)

We get that the search space increases with $N$ as $(N - 1)!!$. On the other hand, note that the decision problem is rapidly checked (in polynomial time) if we provide a candidate set of $N$ numbers to the algorithm. These two features lead us to assume that the problem under consideration belongs, in a worst-case classification [1], to the NP complexity class.

Note that this is not surprising: the preceding sections led us to the conclusion that the process is embedded in a (dynamical) scale-free non-planar network [25]. Now, it has been shown that non-planarity in this kind of problem usually leads to NP-completeness [26] (for instance, the Ising model in two-dimensions is, when the underlying network topology is non-planar, in $NP$).

4.2. Easy-hard-easy pattern

An ingredient which is quite universal in the algorithmic phase transitions is the so called easy-hard-easy pattern [1]: in both phases, the computational cost of the algorithm (the time that the algorithm requires to find a solution, that is, to reach stationarity) is relatively small. However, in the neighborhood of the transition, this computational time reaches a peaked maximum. In terms of search or decision problems, this fact has a clear interpretation: the problem is relatively easy to solve as long as the input is clearly in one phase or the other, but not in between. In the system under study, the algorithm is relatively fast in reaching an absorbing state of low concentration of primes for small $N$ because the probability of having positive divisions is small. On the other hand, the algorithm is also fast in reaching an absorbing state of high concentration of primes for high $N$, because the system has enough ‘catalytic candidates’ at each time step to be able to reduce them, the probability of having positive divisions is high. In the transition’s vicinity, the
Figure 11. Characteristic time $\tau$ as defined in the text versus $N$, for different pool sizes, from left to right: $M = 2^{10} - 2^{14}$. Every simulation is averaged over $2 \times 10^4$ realizations. Note that for each curve and within the finite size effects $\tau(N)$ reaches a maximum in the neighborhood of its transition point (this can be easily explored in figure 6).

system is critical. Divisions can be achieved, however, the system needs to make an exhaustive search of the configuration space in order to find these divisions: the algorithm requires in this region much more time to reach stationarity.

Note that this easy-hard-easy pattern is related, in second-order phase transitions, to the phenomenon of critical slowing down, where the relaxation time in the critical region diverges [1].

We have already seen in figures 4 and 5 that the system reaches the steady state in a different manner, depending on in which phase the process is located. More properly, when $N \ll N_c$ (disordered phase), the system rapidly freezes, practically without achieving any reaction. When $N \gg N_c$ (ordered phase), the system takes more time to reach the steady state, but it is in the regime $N \sim N_c$ where this time is maximal. In order to be able to properly compare these three regimes, let us define a characteristic time in the system $\tau$ as the number of average time steps that the algorithm needs to take in order to reach stationarity. Remember that we defined a time step $t$ as $N$ Monte Carlo steps ($N$ operations). Thus, normalizing it over the set of numbers considered, it is straightforward to define a characteristic time as:

$$\tau(N) = \frac{t}{N}. \quad (8)$$

Note that $\tau$ can be understood as a measure of the algorithm’s time complexity [4]. In figure 11, we plot $\tau$ versus $N$ for a set of different pools $M = 2^{10}, \ldots, 2^{14}$ (simulations are averaged over $2 \times 10^4$ realizations). Note that given a pool size $M$, $\tau$ reaches a maximum in the neighborhood of its transition point $N_c(M)$, as can be checked according to figure 6. As expected, the system exhibits an easy-hard-easy pattern, as long as the characteristic time $\tau$ required by the
algorithm to solve the problem has a clear maximum in the neighborhood of the phase transition. Moreover, the location of the maximum shifts with the system’s size according to the critical point scaling found in equation (4). On the other hand, this maximum also scales as:

$$\tau_{\text{max}}(M/\log(M)) \sim (M/\log(M))^\delta,$$

where the best fit provides $\delta = 0.13 \pm 0.1$. Note that in the thermodynamic limit, the characteristic time would diverge in the neighborhood of the transition. It is straightforward to relate this parameter to the relaxation time of a physical phase transition. According to these relations, we can collapse the curves $\tau(N, M)$ of figure 11 into a single universal one. In figure 12, this collapse is provided: the goodness of the collapse supports the validity of the scaling relations.

4.3. Average-case classification

While computational complexity theory has generally focused on worst-case classification, one may readily generalize this study to different complexity definitions. Average-case analysis is understood as the classification of algorithms according to which is the resource usage needed to solve them on average (for instance, the execution time needed to solve an algorithm often depends on the algorithm’s inputs and consequently one can define an average execution time). This classification seems more relevant for practical purposes than the worst-case one. As a matter of fact, although $NP$-complete problems are generally thought of as being computationally intractable, some are indeed easy on average (however some remain complete in the average case, indicating that they are still difficult on randomly generated instances).

The system under study has been interpreted in terms of a search problem, belonging to the $NP$ class in a worst-case classification. Now, an average-case behavior, which is likely to be
more useful in order to classify combinatorial problems, turns out to be tough to describe. In [6], Monasson et al showed that where \( NP \) problems exhibit phase transitions (related to dramatic changes in the computational hardness of the problem), the order of the phase transition is in turn related to the average-case complexity of the problem. More specifically, that second-order phase transitions are related to a polynomial growth of the resource requirements, instead of the exponential growth, associated with first-order phase transitions.

It has been shown that the system actually exhibits a second-order phase transition and an easy-hard-easy pattern. Following Monasson et al [6], while our prime generator is likely to belong to the \( NP \) class, its average-case complexity class is only polynomial. This means that as the pool size \( M \) grows, the execution time that the algorithm needs in order to solve the problem (to reach the steady state) increases polynomially on average (on average means in the ensemble of all possible initial configurations, i.e. considering many realizations with an initial random configuration). We may argue that one of the reasons for this hardness reduction is that the algorithm does not realize a direct search but on the contrary this search is stochastic: the search space is not exhaustively explored. Thereby, the average behavior of the system and thus the average decision problem can be easily solved by the algorithm, in detriment to the probable character of this solution.

5. Conclusions

In this paper, a (stochastic) algorithmic model which stands for a prime number generator has been studied. This model exhibits a phase transition which distinguishes a phase where the algorithm has the ability to reduce every element into a prime, and a phase where the system is rapidly frozen. Analytical and numerical evidence suggests that the transition is continuous. In a second part, the model has been reinterpreted as a search problem. As long as the model searches paths to reduce integers into primes, the combinatorial problem is related to primality tests and decomposition problems. It has been shown that this model belongs to the \( NP \) class in a worst-case classification; moreover, an easy-hard-easy pattern has been found, as is common in many algorithmic phase transitions. According to the fact that the transition is continuous, and based on previous works, it has been put forward that the average-case complexity may be only polynomial. This hardness reduction is in turn related to the fact that the algorithm only yields probable states.

Acknowledgments

We thank the Instituto de Fisica at UNAM for support and hospitality. The research was supported by UNAM-DGAPA grant IN-118306 (OM), grants FIS2006-26382-E and FIS2006-08607 (BL and LL) and the IF-UNAM Tomas Brody Lectureship 2007 awarded to BL.

References

[1] Percus A, Istrate G and Moore C (ed) 2006 Computational Complexity and Statistical Physics (Oxford: Oxford University Press)
[2] Erdos P and Renyi A 1959 Publ. Math. 6 290
[3] Cheeseman P, Kanefsky B and Taylor W M 1992 Computational complexity and phase transitions Workshop on Physics and Computation (PhysComp ’92) pp 63–8 ISBN 0-8186-3420-0

New Journal of Physics 10 (2008) 023009 (http://www.njp.org/)
[4] Mertens S 2000 Computational complexity for physicists Preprint cond-mat/0012185
[5] Kirkpatrick S and Selman B 1994 Science 264 5163
[6] Monasson R, Zecchina R, Kirkpatrick S, Selman B and Troyansky L 1999 Nature 400 133–7
[7] Hogg T, Huberman B A and Williams C P 1996 Artif. Intell. 81 1–15
[8] Istrate G 2000 Computational complexity and phase transitions Proc. 15th Ann. IEEE Conf. on Computational Complexity pp 104–15
[9] Mertens S 1998 Phys. Rev. Lett. 81 20
[10] Biroli G, Cocco S and Monasson R 2002 Physica A 306 381–94
[11] Mézard M, Parisi G and Virasoro M A 1987 Spin Glass Theory and Beyond (Singapore: World Scientific)
[12] Dittrich P, Banzhaf W, Rauhe H and Ziegler J 1998 Proc. Second German Workshop on Artificial Life pp 19–22
[13] Dittrich P 2001 Artif. Life 7 225–75
[14] Luque B, Lacasa L and Miramontes O 2007 Phys. Rev. E 76 010103
[15] Schroeder M R 1997 Number Theory in Science and Communications (Berlin: Springer)
[16] Corso G 2004 Phys. Rev. E 69 036106
[17] Achter J D 2004 Phys. Rev. E 70 058103
[18] Zhou T, Wang B H, Hui P M and Chan K P 2006 Physica A 367 613–8
[19] Albert R and Barabasi A L 2002 Rev. Mod. Phys. 74 47–97
[20] Kaufman S A 1993 The Origins of Order (Oxford: Oxford University Press)
[21] Jain S and Krishna S 1998 Phys. Rev. Lett. 81 25
[22] Gould H and Tobochnik J 1996 An Introduction to Computer Simulation Methods (Reading, MA: Addison-Wesley)
[23] Pratt V R 1975 SIAM J. Comput. 4 214–20
[24] Agrawal M, Kayal N and Saxena N 2004 Ann. Math. 160 2
[25] Bollobas B 2002 Modern Graph Theory (Berlin: Springer)
[26] Istrail S 2000 Proc. 32nd ACM Symp. on the Theory of Computing (New York: ACM Press) pp 87–96