1-2020

Physical Randomness Can Help in Computations

Olga Kosheleva
The University of Texas at El Paso, olgak@utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:

Technical Report: UTEP-CS-20-02

Physical Randomness Can Help in Computations

Olga Kosheleva and Vladik Kreinovich
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
olgak@utep.edu, vladik@utep.edu

Abstract

Can we use some so-far-unused physical phenomena to compute something that usual computers cannot? Researchers have been proposing many schemes that may lead to such computations. These schemes use different physical phenomena ranging from quantum-related to gravity-related to using hypothetical time machines. In this paper, we show that, in principle, there is no need to look into state-of-the-art physics to develop such a scheme: computability beyond the usual computations naturally appears if we consider such a basic notion as randomness.

1 Introduction

While traditional computers have achieved great results, there are still many important problems for which computations on current computers are too slow – not to mention that there are many problem for which it has been proven that they cannot be algorithmically solved on modern computers; see, e.g., [5, 6, 11]. To overcome this problem, many researchers and engineers are working on making computers faster – and to achieve that goal, they are figuring out if we can utilize additional physical processes. Several such schemes have been proposed, using different physical phenomena ranging from quantum effects to black holes to even hypothetical phenomena such as causal anomalies (time machines); see, e.g., [1, 3, 4, 5, 6, 7, 10].

In this paper, we show that there is yet another potential way to new computational abilities: namely, the use of physical randomness.

2 Randomness Is Important

In classical (pre-quantum) physics, randomness was mainly subjective. All the way to the beginning of the 20 century, all known laws of physics were deterministic. A good example is Newton's laws. Once we know the initial positions and velocities of all the planets, we can predict – many years ahead –

where the planets will be at any future moment. This is not just a theoretical possibility: astronomers did predict the planets' positions with very high accuracy. This accuracy was indeed very high. For example, one of the motivations for developing General Relativity was that the predicted position of Mercury differed from the predicted one, with the difference growing by 43 arcseconds per 100 years.

In that period, randomness was not in the physical theory itself, randomness was used – starting with Gauss – to describe the fact that we do not know the exact positions, the exact velocities, and in general, the exact values of other physical characteristics. In other words, in those days, randomness was mainly subjective, reflecting our lack of knowledge.

The appearance of objective randomness. At the beginning of the 19 century, it became clear that, in addition to subjective randomness – that describes our ignorance – there is also *objective* randomness, randomness which is essential for the corresponding physical process itself. The first such phenomenon was *radioactivity*, when atoms emit high-energy particles. A radioactive material such as radium consists of numerous absolutely identical atoms. If the corresponding physics was deterministic, they would all emit the corresponding particles at exactly the same time – but this is not what we observe. What we observe is that at any given moment of time:

- some atoms remain stable, while
- other atoms emit the particles.

There is no regularity in which atoms are stable are which are not; in this sense, the corresponding process is truly random.

According to modern physics, randomness is ubiquitous. Studies of radioactivity and related phenomena led to the development of new physics – quantum physics, according to which all real-life processes are probabilistic; see, e.g., [2, 12]. Thus means, in particular, that, in general:

- it is not possible to predict future events,
- it is only possible to predict the *probabilities* of different future events.

When we repeat the same experiment with random results again and again, we get a random sequence.

Randomness is also important to conciliate reversibility of physical laws with irreversibility of many physical processes. Later, it turned out that randomness is important already in the non-quantum approximation to the true physics: namely, it is important so that we will be able to reconcile physical equations with observations.

Why do we need to reconcile them? Because all fundamental physical equations, starting with Newton's laws, are reversible: if we keep all the particles where they are and reverse all the velocities, the system will reverse its trajectory and eventually reach the original state. This is easy to show in the mathematical level, this is easy to illustrate on the example of a simple mechanical system.

However, in real life, many processes are irreversible. If we break a cup, there is no way to make the pieces come together into the original unbroken shape. This seeming contradiction was known already to Boltzmann, the father of statistical physics. The modern explanation for this seeming paradox is that, in addition to *equations*, we also need to take into account *initial conditions*. There may be some restrictions on these initial conditions, but within these restrictions, we do not expect any additional regularities. In other words, the initial conditions should be *random*.

This additional assumptions breaks the original symmetry between past and future: when initial conditions are random, the resulting future state is not random at all: e.g., if we start with a random distribution of matter, we end up with the current hierarchical structure of galaxies, stars, etc.

3 How Can We Describe Randomness in Precise Terms?

Need to go beyond the traditional probability theory. At first glance, it may appear that the question raiseds in the title of this section have been answered long time ago. Starting with Gauss, we have used probabilities to describe randomness.

This is true, probability theory has indeed been very successful in describing physical phenomena. However, as noticed already by Kolmogorov — the father of modern probability theory – this theory deal with ensembles, with mass phenomena. This is great, but physicists also deal with individual phenomena. For example:

- If we measure the value of some quantity at consequent moments of time and get 010101..., clearly we have a regular process – namely, a periodic process.
- On the other hand, if we flip a coin several times – or perform some quantum experiment with random outcomes – we get a sequence with no regularities, a sequence that, from the physical (and commonsense) viewpoint is *random*.

In formalizing the difference between the two cases, traditional probability theory is of no help: according to this theory, there is no different between the sequence 0101... and any other sequence – all these sequences have the same probability 2^{-n}, where n is the number of bits.

To describe this difference, Kolmogorov and other researchers came up with a special notions of *Kolmogorov complexity* and *algorithmic randomness*; see, e.g., [9]. Let us describe these notions.

Kolmorogov complexity and algorithmic randomness. What is the difference between a regular sequence like 0101... and a truly random sequence?

- A regular sequence like 01010..., no matter how long, can be generate by a very short and simple program: it is sufficient to run a loop repeatedly producing 01.
- On the other hand, the vert fact that a binary sequence is random means that it has not regularities, no such short program is possible, and the only way to general a random sequence 0110... of length n is to write something like `print`(0110..). This program requires that we reproduce the whole sequence, so its length (= number of bits) is close to n.

In other words:

- a regular sequence x of length $\mathrm{len}(x) = n$ can be generated by a short program, a program whose length is much shorter than n;
- on other hand, the only to generate a random binary sequence x of length n is to have a program whose length is close to n.

To formalize this idea, researchers came with the following notion of *Kolmogorov complexity* $K(x)$ of a given string x. We fix some universal programming language, and we define $K(x)$ as the length of the shortest program p that generates x:

$$K(x) \overset{\text{def}}{=} \min\{\mathrm{len}(p) : p \text{ generates } x\}.$$

According to the above arguments:

- for a regular sequence, $K(x) \ll \mathrm{len}(x)$;
- on the other hand, for a truly random sequence x, we have $K(x) \approx \mathrm{len}(x)$.

This leads to the following formal definition [8, 9]:

Definition 1. *Let C be an integer. We say that a binary sequence x is C-*random *if $K(x) \geq \mathrm{len}(x) - C$.*

4 Back to Physics

Back to physics. So, whenever we want to describe that some physical sequence is random, we formalize it as saying that this sequence is C-random for some appropriate value C (and, of course, for an appropriate selection of a programming language).

One more thing about physics. Physicists usually believe that:

- if some phenomenon is never happening in the world,
- then there must be a reason for this never-happening.

This sounds natural. Interestingly, we can reverse this statement and formulate a logically equivalent statement which may sound not as natural – that:

- if some phenomenon does not contradict any physical laws,
- this means that eventually, we will observe this phenomenon.

From this viewpoint, if the only restriction on a binary sequence (obtained from observing a physically random phenomenon) is that this sequence should be random (in the sense to Definition 1), this means that *every sequence which is random according to this definition will eventually occur as a result of some observation.*

5 How Does This Affect Computations

If a sequence is not random, we will eventually find it out. Let us first show that if a binary sequence x is *not* C-random, then we will eventually find it out. Indeed, if the sequence x is not C-random, this means that its Kolmogorov complexity $K(x)$ is smaller than $\text{len}(x) - C$. By definition of the Kolmogorov complexity, this means that there exists a program p of length

$$\text{len}(p) < \text{len}(x) - C$$

that generates the sequence x.

There are finitely many sequences of any given length. Thus, there are finitely many programs p of length $\text{len}(p) < \text{len}(x) - C$. So, all we have to do is to start all these programs. If a sequence x is not truly random, one of these programs will eventually generate x – and thus, we will know that this sequence x is not C-random.

There is a physical way to check whether a given binary sequence is C-random. Let us show that this enables us to decide, for each binary sequence x, whether this sequence is C-random or not. To do that, we simultaneously start two processes:

- in the first process (as described in the previous subsection), we run all possible programs p of length $\text{len}(p) < \text{len}(x) - C$ and wait until one of these programs generates the given sequence x;
- in the second process, we perform all possible measurements of random phenomena and wait until in one of these measurements, we get exactly the given sequence x.

As a result:

- If a sequence if not C-random, then one of the programs from the first process will generate x and thus, we will know that the given sequence x is not C-random.
- On other hand, if the sequence x is C-random, then, according to the arguments from the previous section, it will eventually appear in some observations – so we will know that this sequence x is C-random.

So, by using physical world, we will always be able to tell whether a given sequence is C-random or not.

This may sound natural and simple, but without the use of physical world, we cannot check C-randomness. Interestingly, the above simple scheme enable us to solve the problem which is not algorithmically solvable on a normal computer! Indeed, here is a simple proof.

Proposition 1. *No algorithm is possible that, given a binary sequence x, checks whether this sequence is C-random.*

Proof. This proof uses a known auxiliary result that, for each $C > 0$ and for each n, there exists a sequence x of length n for which $K(x) \geq \text{len}(x) - C$.

Indeed, by definition of Kolmogorov complexity, each binary sequence which is *not* C-random, i.e., for which $K(x) < \text{len}(x) - C = n - C$, can be generated by a program of length $< n - C$ – i.e., equivalently, of length $\leq n - C - 1$. There are:

- $2^1 = 2$ binary strings of length 1,
- $2^2 = 4$ strong of length 2, etc., and
- 2^{n-C-1} strings of length $n - C - 1$.

Thus, overall, there are

$$2^1 + 2^2 + 2^3 + \ldots + 2^{n-C-1} = 2^{n-C} - 2 < 2^{n-C}$$

possible binary strings of length $\leq n - C - 1$. Thus, there are less than 2^{n-C} programs of length $\leq n - C - 1$.

- Each program generates only one output, so all these programs can generate $\leq 2^{n-C}$ different strings.
- On the other hand, there are $2^n > 2^{n-C}$ strings of length n.

Thus, some strings of length n are C-random.

On the set of all binary sequences of length n there is a natural order – corresponding to the numerical values of these sequences interpreted as binary numbers, so that:

- 00...0 is 0,
- 00...01 is 1, etc., all the way to
- 11...1 which is $2^n - 1$.

By using this order, among several possible C-random sequences of length n, we can select one sequence r_n which corresponds to the binary sequence with the smallest possible number.

Now, we are ready to prove the proposition by contradiction. Let us assume that there exists a program P (of some length $\text{len}(P)$) that:

- given a binary sequence x,
- checks whether $K(x) \geq \text{len}(x) - C$.

Then, we can compute r_n as follows:

- We enumerate all the numbers 0, 1, 2, $\ldots$, and for each number, we use the program P to check whether this number is C-random.
- Once we reached the C-random number, we stop and output this number – this will be the desired number r_n.

The resulting program R for computing r_n consists of:

- the program P and
- a few lines describing adding 1.

So, the length of the program R does not depend on n.

Hence, for large enough n, this length is $< n - C$ – but it contradicts to the definition of r_n as one of the C-random sequences – i.e., by definition, sequences which cannot be generated by any sequence whose length is smaller than $n - C$.

This contradiction shows that our assumption was wrong, and thus, indeed, no algorithm is possible that:

- given a binary sequence x,
- checks whether this sequence is C-random.

The proposition is proven.

A word of caution. In principle, the above scheme allows us, by using the physical world, to solve the general problem that cannot be algorithmically solved only by existing computers. But is it practical? Not really – or, in more optimistic way – not yet.

Indeed, the above procedure requires that, for each sequence x of length n, we perform experiments until we encounter this sequence (or until we find a short program generating this sequence). There are 2^n possible sequences of length n. Most of them – as one can easily prove – are C-random. Thus to get a given C-random sequence, we need between 1 and $\approx 2^n$ measurements. So, to get the given sequence, on average, we will need to have $\approx 0.5 \cdot 2^n$ measurements. Thus, this procedure requires exponential time – and is, therefore, not feasible; see [6, 11].

A word of hope. The above procedure is not practical yet. However, the fact that such a procedure exists in the first place makes us hope that a feasible (or at least more feasible) version of this procedure will be eventually found.

Acknowledgments

This work was supported in part by the US National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] S. J. Aaronson, "NP-complete problems and physical reality", *ACM SIGACT News*, 2005, Vol. 36, No. 1, pp. 30–52.

[2] R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.

[3] M. Koshelev and V. Kreinovich, "Towards Computers of Generation Omega – Non-Equilibrium Thermodynamics, Granularity, and Acausal Processes: A Brief Survey", *Proceedings of the International Conference on Intelligent Systems and Semiotics ISAS'97*, National Institute of Standards and Technology Publ., Gaithersburg, MD, 1997, pp. 383–388.

[4] O. Kosheleva and V. Kreinovich, "What can physics give to constructive mathematics," *Mathematical Logic and Mathematical Linguistics*, Kalinin, 1981, pp. 117–128 (in Russian).

[5] O. Kosheleva and V. Kreinovich, "Space-Time Assumptions Behind NP-Hardness of Propositional Satisfiability", *Mathematical Structures and Modelling*, 2014, Vol. 29, pp. 13–30.

[6] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[7] V. Kreinovich, "Designing, Understanding, and Analyzing Unconventional Computation: The Important Role of Logic and Constructive Mathematics", *Applied Mathematical Sciences*, 2012, Vol. 6, No. 13, pp. 629–644.

[8] L. A. Levin, "Randomness conservation inequalities: information and independence in mathematical theories", *Information and Control*, 1984, Vol. 61, pp. 15–37.

[9] M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, New York, 2008.

[10] D. Morgenstein and V. Kreinovich, "Which algorithms are feasible and which are not depends onthe geometry of space-time", *Geombinatorics*, 1995, Vol. 4, No. 3, pp. 80–97.

[11] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994.

[12] K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2017.