

# Unconventional Computing: Do We Dream Too Much?

Gheorghe Păun<sup>(✉)</sup>

Institute of Mathematics of the Romanian Academy, PO Box 1-764,  
014700 București, Romania  
gpaun@us.es

**Abstract.** After briefly mentioning the motivation and the “dreams” of unconventional computing (with an eye on natural computing, especially on bio-inspired computing), we ask ourselves whether these “dreams” are realistic, and end with a couple of related research issues from the membrane computing area.

## 1 Motivation of Unconventional Computing

The motivation of unconventional computing, in particular, of natural computing (here we understand it mainly in the sense of bio-inspired computing, although in many cases – see, e.g., [22] – also quantum computing and cellular automata are included), comes from at least three directions: (i) the limits of current (“Turing-von Neumann”) computers, (ii) the need for new modeling and simulating tools for sciences like biology, ecology, even physics, (iii) the intrinsic human curiosity, the need to know, to predict, to build mathematical models.

The computers are the most influential invention of the last century, but they have both theoretical limits and practical limits (the two categories do not completely overlap – but this can be the subject of a separate discussion). Among the theoretical ones, two are fundamental: (1) current computers (although they are not Turing machines in the strict, mathematical meaning of the term) cannot compute “beyond the Turing barrier”, cannot compute what is Turing uncomputable, and (2) for current computers, problems of a complexity higher than the polynomial one are intractable, they cannot be solved in a feasible time. In general, problems which are **NP**-complete are (considered) intractable – although most non-trivial practical problems are of this type. (Cryptography is crucially dependent on this assumption.)

Breaking “the Turing barrier” was a constant concern of computability, and the term *hypercomputability* was coined to name research in this direction. The bibliography is large, we mention here only [6, 24].

Much larger is the literature related to complexity classes **P** and **NP** and, especially in the unconventional computing area, the bibliography of attempts to “break the **NP** barrier”. Symmetrical to the hypercomputation concept, the term *fypercomputation* was proposed in [18] as a name for the research aiming to find polynomial time solutions to computationally hard problems, typically, **NP**-complete problems (the initial “f” is taken from “fast”).

Hypercomputability is the primary goal of unconventional computing and this is a very important issue from a practical point of view, [5], although it is believed, see, e.g., [6], that “computing the uncomputable” could have even more important consequences than proving that  $\mathbf{P} = \mathbf{NP}$ .

## 2 Dreams of Unconventional Computing

The term “unconventional” is not precisely defined, we use it here in the vague sense of “non-classic” in theory (not belonging to the “standard” automata theory, based on Turing machines and their variants, restrictions and generalizations) and “not electronic” in the form of implementation, having in mind especially bio-inspired computing (DNA computing, membrane computing, evolutionary and neural computing, the long list of algorithms abstracted from bio-processes, such as immune, ant colony, bee colony, swarm, cuckoo, strawberry algorithms), water flowing algorithms and cultural computing, as well as quantum computing and analog computation, optical computing and many others.

As said before, solving problems considered of an exponential complexity (we assume  $\mathbf{P} \neq \mathbf{NP}$ ) in polynomial time is the first dream of all these directions of research (even if the solution is not provably optimal, but “good enough”, whatever this means – e.g., close to optimal with a known probability). In evolutionary-like computing the strategy is based on an intriguing slogan: when you do not know where to, go randomly! Actually, the random walk through the space of candidate solutions is controlled in a way which imitates the Darwinian evolution, or the way the ants and the bees look for food, and so on and so forth. Combined with the impressive brute force of the modern computers, this strategy, which is only a metaphoric imitation of biological processes, proves to work surprisingly well in surprisingly many situations.

Although usually not explicitly stated, also hypercomputability is a dream of unconventional computing, starting, for instance, from the (debatable, of course) observation that “the brain is not a Turing machine” and, similarly, from other “computations” taking place in nature which seem to be of a non-Turing type. There are many basic ideas of hypercomputing, [24], which lead to devices more powerful than Turing machines (typically, able to answer the halting problem). Some of these ideas were extended also to natural computing models. An example is the acceleration, see [3] (the first step takes one time unit, but the machine learns, so the next step takes only half of a time unit, and so on, each step taking half of the time needed for the previous step; note the important distinction between the internal clock and the external one: infinitely many internal steps are performed in two external time units).

There also are other goals/dreams of unconventional computing, which can themselves be called “unconventional” with respect to the classical computer science. We only list some of them: energy efficiency, also related to the reversibility issue; adaptability, learnability, evolvability; self-healing, robustness with respect to hardware errors.

We somewhat moved from theory to engineering, so let us go further towards practice (the three dimensions, theory, computer engineering, and applications,

are intimately related, of course). The internet brought into the stage old-new concepts, such as unsynchronized/asynchronous computation, amorphous computing, cloud computing. Biology asks for models of cells (this is considered the main challenge of the bioinformatics, after the completion of the Genome Project), medicine asks for nano-robots able to scan the body and deliver medicines in the necessary places, repair genes, kill viruses (a project of such a nano-robot, built from DNA molecules, was presented in [2]; note that repairing genes, identifying and breaking in parts viruses are string editing operations, hence they pertain to DNA computing).

Then, on top of all these, there is a “meta-dream”, actually, a forecast: both physics and biology will gain a lot from bringing among their central paradigms the information and the computability, new ages of these sciences are foreseen, based on these paradigms. For physics, this seems to be enhanced by the progresses in quantum computing (see, e.g., [10,27]). For biology, the issue is more urgent, taking into account that biology is not yet a mathematized science, the biologists need tools and techniques for modeling and simulating processes at all levels, from cells to eco-systems. A huge quantity of empirical data was gathered, but the tools to process these data were not correspondingly developed. In [14] the term *infobiotics* was proposed for this informational-computational biology, in [19] we propose the term *infobiology* (symmetrical to the currently used term *bioinformatics*).

Significant for our discussion, a lot of words were spent around the *systems biology* syntagma, a planned research area aiming to transform biology and medicine into a “precise engineering” (see, e.g., [12]).

### 3 Difficulties and Limits

And now, we come to the question in the title: are these goals and dreams realistic, or we dream too much? Of course, we do not have to underestimate the progresses in science and technology, in particular, in bioengineering, there are many funny examples of this kind in the history. However, in many cases, after a successful experiment (e.g., in bio-computing), the progresses were disappointing, not confirming the initial enthusiasm. The typical example is that of DNA computing, with Adleman’s experiment, [1], opening a new research area, which, after twenty years has not confirmed yet the big hopes of the beginning.

There are many details to be discussed here. The scientists are, in general, moderately enthusiastic, but they are “forced” to be so in order to “sell” their results, to get projects, hence money. Then, mass-media is always ready to inflate the facts, to predict “scientific revolutions” (see, as an example, the media echoes of the “doctor in a cell” from [2] and of similar – science fiction at this moment – ideas). As another illustration we can mention the noise around systems biology, in many respects not too much more than an reincarnation of system theory applied to biology, [25], not very successful in its first stages, in sixties, because of various reasons, connected to both computer science and biology development at that time.

All these pertain to the politics and the sociology of science. Here we are more interested in the difficulties and, especially, the limits coming from science and technology. There are many of them.

On paper we work with idealized objects, e.g., DNA molecules; in a test tube or in a cell they behave “slightly” differently. There are three different “worlds”, *in vivo*, *in vitro*, *in info*, and the notions, ideas, models cannot be directly transferred among them. The models of a cell are made of symbols, they are reductionistic, abstract, they always work as prescribed, which is not the case in reality. The goal of life is life, not computing. We, the computer scientists see computations everywhere. The biochemistry is in a large extent nondeterministic, probabilistic, context-dependent, the controls are complex and not always known or easy to find. Nature is redundant, it has “enough time”, it affords to try and, if the result is not acceptable, to discard it. All these are far from what we need and what we can do in computer science. That is why the modeling and the simulation of a cell, the minimal entity which is unanimously considered alive, is so difficult. That is why it is difficult to ask the biomolecules or the cells to compute.

Going upwards from the cell, the things become still more difficult, first because the systems and processes we have to deal with are much more complex. Small is beautiful, big is necessary (but difficult to handle). At the level of “big” there appear such phenomena as emergence, synergy, system effect, which are often non-predictable (this can be related to precise mathematical results: Rice theorem tells us that no non-trivial question about a model of the cell which is known to be Turing equivalent – and most such models in membrane computing are so – can be algorithmically answered).

How to model such “things” like life and intelligence providing that we do not even have good (mathematical) definitions of them?!... This raises the question whether such definitions are possible, in general, or, more realistic, possible in the framework of the today mathematics. It was said in several places (see the references of [17], from where several ideas are recalled here) that it is perhaps necessary to wait for a new mathematics in order to face such tasks, like modeling life and intelligence.

But, also in terms of what we have now we can find intrinsic limits of unconventional computing.

Three are my favorite results of this kind. They are mentioned in the order of the time of their publication.

The first one is Gandy’s approach to what is (to use Hilbert’s term) *mechanically computable*, see [9]. In the attempt to free the Turing-Church thesis of any anthropic meaning, Gandy considered a general algebraic definition of a computing device, and then proved that any computing machine fulfilling the (four) conditions in this definition can be simulated by a Turing machine. The generality of Gandy’s definition and his theorem can be seen as a proof that hypercomputing is a difficult task – and this fits with Martin Davis opinion [7, 8] that “hypercomputing is a myth”, moreover, that the computing machineries more powerful than the Turing machine which were reported so far are based

on dishonest tricks (the power is introduced in the definition, in disguise, in the form of infinite processes, real numbers, other infinite ingredients, hence there is no surprise that the device is powerful).

Next, one must recall Conrad theorems, [4]. Basically, they say that the three desired characteristics of computing devices, universality (hence programmability), efficiency and learnability/evolvability, are contradictory, there cannot exist a computing device simultaneously having all these three properties. These theorems are of a kind which is famous in mathematics, *impossibility theorems*, proving that when we demand simultaneously certain properties, it happens that there is no object having all these properties. Gödel theorems and Arrow theorem (in social choice) are classic examples.

As the third limiting theorem in natural computing/optimization we can consider the (in) famous *no free lunch* theorem of Wolpert and Macready, [26], saying, in short, that in average all methods of approximate optimization are equally good over all optimization problems. “Equally good” can also be read “equally bad”, which can explain the plethora of approximate optimization methods – each new one finds a niche where it is better than others, and so on....

Returning to applications in biology, one often makes lists of properties which, for instance, the mathematical models should possess: adequacy and relevance, programmability and scalability, efficiency, understandability, emergent behavior. The question which arises is obvious: can all these properties be simultaneously reached, or also in the biological modeling area there exist impossibility theorems, like Conrad theorems? I bet for the existence of such impossibility results.

## 4 Research Topics

The last lines of the previous section already formulated a research topic. There are many others – of course, we are interested only in those which are related to the discussion here, concerning the goals/dreams of unconventional computing.

Two are the basic directions of future research in this respect: hypercomputing and fypercomputing. We consider them only in terms of membrane computing, the area we know better, [21]. The only ideas considered so far in the direction of hypercomputing were acceleration [3] and evolutionary lineages of P systems [23]. Many further ideas (well, “tricks”, in terms of Martin Davis) can be found in the literature [6, 24] – which ones can be extended also to membrane computing? Which of them can also have a biological motivation? In particular, what about spiking neural P systems [11] able of hypercomputation? No result of this type is known, not even the acceleration used in [3] was extended to SN P systems. This issue is especially relevant, taking into account the fact that the brain is considered non-Turing.

Fypercomputing is a basic research topic in membrane computing, encouraged by the fact that there are many biological processes which can be used in order to produce an exponential space in linear time, so that we can trade-off time for space, thus essentially speeding-up the computation. Membrane division, membrane creation (plus the possibility of producing exponentially many

objects in linear time), membrane separation, neuron division and neuron budding, string replication are such processes. An almost systematic study of the effect of these operations on various classes of P systems was carried out – but still efforts are needed in this respect to complete the map.

A special case is that of numerical P systems from [20], proved in [13] to be efficient for a variant used in robot control, those with “enzymatic control”, [16]. What about numerical P systems without the enzymatic control? In the above mentioned papers, these systems are used for computing functions and no research was reported where numerical P systems are used for solving decision problems. This remains as a research topic. To this aim, is it necessary to introduce further ingredients, such as membrane division, or these systems are intrinsically efficient?

An interesting and natural issue is to try to transform the ideas which lead to hypercomputation to tools for obtaining hypercomputation. As an example, acceleration is such a tool, in two time units (external time), any computation ends (although internally one performs an infinite number of steps). The first task is to define complexity classes for such devices, then to compare them with each other and with standard complexity classes.

Natural (bio-inspired) computing raises certain complexity problems which were not considered in the classical complexity theory, [15]. We only mention three of them, in the form already discussed – but not completely settled – in the membrane computing framework: (1) allowing non-uniform solutions (called semi-uniform when the algorithms are produced in polynomial time, starting from instances, not from the size of the problem), comparing uniform and semi-uniform complexity classes; (2) using pre-computed resources, an arbitrarily large initial workspace, without containing “too much” information, activated by introducing a problem in a well delimited portion of it; (3) allowing non-determinism, but taking care that the device is *confluent*, either converges to a unique configuration, from where the computation continues deterministically (strong confluence), or all computations halt and provide the same result (weak, logical confluence).

Of course, a major problem concerns the implementations. Most chapters of natural computing aim at finding ways to better use the existing computers – see as a typical example the case of evolutionary computing. DNA computing came with a new promise, of using molecules as a support for computation. This goes close to analog computing, where the device is the big novelty, not the possible model behind it. At this moment, no commercial unconventional computer is known – maybe the D-wave computers are a counterexample, but the extent in which they can be considered quantum computers is still debatable.

But, as we said before, we have to be careful, not to underestimate the progress, in particular, that of unconventional computing!...

## References

1. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. *Science* **226**, 1021–1024 (1994)
2. Benenson, Y., Shapiro, E., Gill, B., Ben-Dor, U., Adar, R.: Molecular computer. A ‘Smart Drug’ in a test tube. In: Ferretti, C., Mauri, G., Zandron, C. (eds.) *Proceedings of Tenth DNA Computing Conference*, vol. 49, Milano, 2004, (abstract of invited talk), Univ. of Milano-Bicocca (2004)
3. Calude, C., Păun, G.: Bio-steps beyond turing. *BioSystems* **77**, 175–194 (2004)
4. Conrad, M.: The price of programmability. In: Herken, R. (ed.) *The Universal Turing Machine: A Half-Century Survey*, pp. 285–307. Kammerer and Unverzagt, Hamburg (1988)
5. Cook, S.: The importance of the P versus NP question. *J. ACM* **50**(1), 27–29 (2003)
6. Copeland, B.J.: Hypercomputation. *Mind. Mach.* **12**(4), 461–502 (2002)
7. Davis, M.: The myth of hypercomputation. In: Teuscher, C. (ed.) *Alan Turing: The Life and Legacy of a Great Thinker*, pp. 195–212. Springer, Berlin (2004)
8. Davis, M.: Why there is no such discipline as hypercomputation. *Appl. Math. Comput.* **178**, 4–7 (2006)
9. Gandy, R.: Church’s thesis and principles for mechanisms. In: Barwise, J., et al. (eds.) *The Kleene Symposium*, pp. 123–148. North-Holland, Amsterdam (1980)
10. Gruska, J.: *Quantum Computing*. McGraw-Hill, Maidenhead (1999)
11. Ionescu, M., Păun, G., Yokomori, T.: Spiking neural P systems. *Fundamenta Informaticae* **71**, 279–308 (2006)
12. Kitano, H.: Systems biology. a brief overview. *Science* **295**, 1662–1664 (2002)
13. Leporati, A., Mauri, G., Porreca, A.E., Zandron, C.: Enzymatic numerical P systems using elementary arithmetic operations. In: Alhazov, A., Cojocaru, S., Gheorghie, M., Rogozhin, Y., Rozenberg, G., Salomaa, A. (eds.) *CMC 2013. LNCS*, vol. 8340, pp. 249–264. Springer, Heidelberg (2014)
14. Manca, V.: *Infobiotics: Information in Biotic Systems*. Springer, Berlin (2013)
15. Papadimitriou, C.H.: *Computational Complexity*. Addison-Wesley, Reading (1994)
16. Pavel, A.B., Vasile, C.I., Dumitrache, I.: Robot localization implemented with enzymatic numerical p systems. In: Prescott, T.J., Lepora, N.F., Mura, A., Verschure, P.F.M.J. (eds.) *Living Machines 2012. LNCS*, vol. 7375, pp. 204–215. Springer, Heidelberg (2012)
17. Păun, G.: From cells to (silicon) computers, and back. In: Cooper, B.S., Lowe, B., Sorbi, A. (eds.) *New Computational Paradigms. Changing Conceptions of what is Computable*, pp. 343–371. Springer, New York (2008)
18. Păun, G.: Towards “fypercomputations” (in membrane computing). Essays dedicated to jürgen dassow on the occasion of his 65th birthday. In: Bordihn, H., Kutrib, M., Truthe, B. (eds.) *Languages Alive. LNCS*, vol. 7300, pp. 207–220. Springer, Heidelberg (2012)
19. Păun, G.: *Looking for Computers in the Biological Cell. After Twenty Years. The Publishing House of the Romanian Academy, Bucharest (2014)*. (in Romanian)
20. Păun, G., Păun, R.: Membrane computing and economics: numerical P systems. *Fundamenta Informaticae* **73**, 213–227 (2006)
21. Păun, G., Rozenberg, G., Salomaa, A. (eds.): *The Oxford Handbook of Membrane Computing*. Oxford University Press, New York (2010)
22. Rozenberg, G., Bäck, T., Kok, J.N. (eds.): *Handbook of Natural Computing*. Springer, Berlin (2012)

23. Sosík, P., Valík, O.: On evolutionary lineages of membrane systems. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2005. LNCS, vol. 3850, pp. 67–78. Springer, Heidelberg (2006)
24. Syropoulos, A.: Hypercomputation: Computing Beyond the Church-Turing Barrier. Springer, Berlin (2008)
25. Wolkenhauer, O.: Systems biology: the reincarnation of systems theory applied in biology? *Brief. Bioinform.* **2**(3), 258–270 (2001)
26. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67 (1997)
27. Zenil, H. (ed.): *A Computable Universe. Understanding and Exploring Nature as Computation.* World Scientific, Singapore (2013)
28. The P Systems Website. <http://ppage.psystems.eu>