DRAFT — NOT FOR DISTRIBUTION

A Note on 'Memcomputing NP-complete problems...' and (Strong) Church's Thesis

Ken Steiglitz
Computer Science Department
Princeton University

August 25, 2015

1 Background

Traversa at al. [1] have recently described what amounts to a classical analog computer, with the claim that it solves an NP-complete problem (SUBSET SUM) in polynomial time and with polynomial resources. The purpose of this note is to point out that essentially the same idea, solving PARTITION with frequency generators and multipliers, was proposed in 1986 [2], and to review the reasons adduced at that time to conclude that such a machine is doomed to failure. As far as I know, nothing has changed since then to change that view.

As mentioned, the idea of using classical analog machines to solve NP-complete problems fast goes back at least to 1986 [2, 3]. In [2] I proposed solving PARTITION [4, 5] using oscillators and multipliers to produce a signal that indicates whether or not a problem instance is YES or NO. The signal is a product of sinusoids, just as given in both [1] and [2]. The question comes down to whether the integral of the signal is zero or not, which is the same as asking whether it has a nonzero zero-frequency component. As pointed out in [2], there's a catch: The decision problem requires distinguishing between 0 and 2^{-n}, where n is a measure of the problem size. Thus, there is reason to believe that noise, which is inevitable in a real machine, kills the idea. Traversa at al. [1] arrive at the same obstacle, but argue that the difficulty can be circumvented by error-correcting coding, but the details are not spelled out. I'd like to repeat the argument in [2] that any such scheme cannot succeed.

2 Strong Church's Thesis and the Meta-argument

Just as we can argue that perpetual motion machines cannot be built because of the Second Law of Thermodynamics, we can argue from general principles that classical[1] analog machines cannot solve the NP-complete problems fast—at least not if (1) $P \neq NP$, and (2) Strong Church's Thesis holds. The latter is, as proposed in [2],

Definition 1 Strong Church's Thesis (SCT) *An (classical) analog device can be simulated by a Turing Machine using resources (including time) polynomial in the resources of the machine.*

Feynman [6] stated essentially the same principle in 1982, and the idea was also referred to as "Physical Church's Thesis". The supporting argument is simply that any finite chunk of classical

[1] If we are allowed to use quantum machines all bets are off.

space-time can be described by mechanisms that are well understood, and can be simulated efficiently using standard numerical methods. See [3] for more discussion, and a proof of a special case.

The rest of the argument is extremely simple: If we can build a classical machine that solves an NP-complete problem in polynomial time, simulate it on a Turing Machine. This Turing machine then shows that P=NP. Put another way, if Memcomputers can solve NP-complete problems in polynomial time, then, assuming P≠NP, SCT is false. It is not evident to me what classical hardware there is in a Memcomputer that cannot be simulated efficiently on an ordinary digital computer.

References

[1] F.L. Traversa, C. Ramella, F. Bonani, and M. Di Ventra. Memcomputing NP-complete problems in polynomial time usng polynomial resources and collective states. Science Advances, 1(6, e1500031), July 3, 2015. http://advances.sciencemag.org/content/1/6/e1500031.full.

[2] K. Steiglitz. Two non-standard paradigms for computation: Analog machines and cellular automata. In J.K. Skwirzynski, editor, Proceedings of the NATO Advanced Study Institute on Performance Limits in Communication Theory and Practice, II Ciocco, Castelvecchio Pascoli, Tuscany, Italy, July 7-19, 1986. NATO Advanced Study Institutes Series E, No. 142, pages 173–192, Dordrecht, The Netherlands, 1988. Kluwer Academic Publishers. http://link.springer.com/chapter/10.1007%2F978-94-009-2794-0_11#page-1 and http://www.cs.princeton.edu/~ken/TwoNonStandard88.pdf.

[3] A. Vergis, K. Steiglitz, and B.D. Dickinson. The complexity of analog computation. Mathematics and Computers in Simulation, 28:91–113, 1986. http://www.cs.princeton.edu/~ken/MCS86.pdf.

[4] M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, CA, 1979.

[5] D. Plaisted. Some polynomial and integer divisibility problems are np-hard. pages 264–267, 1976.

[6] R.P. Feynman. Simulating physics with computers. Int. J. Theor. Phys., 21(6/7):467–488, 1982.