

New biological device not faster than regular computer

Jonas Einarsson^{a,1}

Nicolau et al. (1) describe a proof-of-concept biological computing device that performs parallel computations by letting protein filaments simultaneously explore all branches of a problem-specific structured network. This innovative device represents a new direction in alternative means of computing, and its construction is an impressive technical achievement. However, I find it necessary to comment on the authors' statements on solving large combinatorial problems. In particular, the new device does not circumvent the superpolynomial time requirement to solve the nondeterministic-polynomial-time (NP) complete subset sum problem (SSP), and it does not move forward the limit of the size of combinatorial problems that can be solved.

The SSP has a known solution that runs in $O(NT)$ time, where N is the number of integers in the set and T is the sum of the magnitudes of all integers in the set (2, 3). This so-called dynamic programming algorithm solves the problem for subsets of increasing size using the known result from previous steps. Pisinger (3) gives an improved algorithm that is $O(NW)$, where W is the largest (in magnitude) integer in the set. NP-complete problems that have such polynomial solutions are called weakly NP-complete.

The SSP on the N first primes can thereby be solved for $N = 100$ in less than a second on a regular 2015 laptop computer, and it does not need exponential amounts of any resource, contrary to the statements of Nicolau et al. (1). However, for sets of exponentially

increasing integers, for example $\{2^1, \dots, 2^N\}$, the total sum $T \sim 2^N$ and, consequently, the seemingly polynomial $O(NT)$ algorithm require exponential time in N . It is customary (2) to consider the complexity of the algorithm in the size of the problem as measured by the number of bits $B \propto \log_2 T$ required to encode the problem set. In this sense, the above algorithm is of exponential complexity in the size B .

As explained by Nicolau et al. (1) in their *SI Appendix*, the required physical size and computing time of their device grows as $O(T)$. The computing time for their device thus scales similarly to the computing time of the conventional algorithm on a regular computer.

The authors write that "we are trading the need of time for the need of molecular mass" (1), but, in fact, no such trade is necessary. Both their device and a regular computer can solve the SSP with slowly growing integers in polynomial time. In addition, their device requires an exponential amount of computing agents.

For general instances of the SSP, both their device and the conventional algorithms require exponential computing time in the size B of the problem. Therefore, the new device does not move forward the limit of the size of combinatorial problems that can be solved.

Acknowledgments

I am grateful to Erik Werner for helpful discussions.

¹ Nicolau DV, Jr, et al. (2016) Parallel computation with molecular-motor-propelled agents in nanofabricated networks. *Proc Natl Acad Sci USA* 113(10):2591–2596.

² Cormen TH (2009) *Introduction to Algorithms* (MIT Press, Cambridge, MA).

³ Pisinger D (1999) Linear time algorithms for knapsack problems with bounded weights. *J Algorithms* 33(1):1–14.

^aDepartment of Physics, University of Gothenburg, 412 96 Gothenburg, Sweden

Author contributions: J.E. wrote the paper.

The author declares no conflict of interest.

¹Email: me@jonaseinarsson.se.