EATCS Bulletin Feb. 95 (no. 55) pp. 136-138

The Structural Complexity Column

by

Juris HARTMANIS

Cornell University, Department of Computer Science
Ithaca, NY 14853, U.S.A.
jh@cs.cornell.edu

On the Weight of Computations

One of the Grand Challenges to computer science is to understand what is and is not feasibly computable. Recursive function theory clarified what is and is not effectively computable and in the process extended our understanding of Goedel incompleteness results about the limits of the power of formal mathematical methods. Since computing is universal and encompasses the power of mathematics, the understanding of the limits of the feasibly computable could give a deeper understanding of the limits of rational intellectual processes and insights into the power and limits of scientific theories.

The search for what is and is not feasibly computable has two distinct aspects. The first is to determine (estimate) how much and what kind of computing power will be available in the foreseeable future. The other problem is to determine what kind of problems can be solved with these available computing resources. The first problem is a technological assessment of the existing and potential computing technologies to estimate what kind and how much computing work our machines will be able to render. The other problem leads us to the central questions of complexity theory: what is the intrinsic complexity of important classes of problems we wish to solve. Clearly, the P=NP=PSPACE? problems are among the best known in this area.

In all these considerations, the exponential function seems to give a crude upper limit for the feasibly computable. May it be time, memory or weight requirements, if they grow exponentially, then the computations are not feasible. We do not know what computations require exponential amounts of resources, nor do all instances of problems in exponential complexity classes require exponential resources. Even if the exact solutions require exponential resources, good approximations to the solution may not. But if indeed the problem requires exponential resources, then it is clearly not feasibly solvable already for moderate size instances of the problem.

New Computing Technologies

Recently there have been some very interesting results about new modes of -computing with hints that for some computations there may be an escape from the exponential curse. The two most interesting technologies are quantum computing and molecular (DNA) computing.

We will leave quantum computing for a later time and reflect on Adleman's exciting paper, "Molecular Computation of Solutions to Combinatorial Problems" [1]. Adleman's paper appeared on November 11, 1994, and has received a lot of national publicity since then, including *a New York Times* article on December 13, 1995, about Adleman's scientific career by Gina Kolata.

The paper describes how the Hamiltonian path problem for a seven node graph was encoded in DNA sequences and the Hamiltonian path was extracted as a single DNA string using standard lab techniques after seven days of lab work. This is indeed a very impressive achievement and may stimulate through exploration of the potential of molecular computing. As pointed out in this article, the numbers of "operations" performed in biological computing can be very high and the energy requirements are surprisingly small. At the same time, as we will observe later, even these computations can not escape the exponential curse; if the computations are indeed exponential, then their weight is prohibitive.

The following nondeterministic algorithm was used to solve the directed Hamiltonian path problem:

```
Step 1:  Generate random paths through the graph.

Step 2:  Keep only paths that begin with in-node and end with out-node.

Step 3:  If the graph has n vertices, then keep only those paths that enter exactly n
         vertices.

Step 4:  Keep only those paths that enter all of the vertices of the graph at least once.

Step 5:  If any paths remain, say "Yes'; otherwise, say "No".
```

If this algorithm is used then in step 1 one has to expect to generate almost all possible paths through the graph to find a path satisfying the remaining conditions. Though the paths are encoded molecularly (with DNA strands) they have weight and the question arises: How heavy will these computations get? In short, we have to consider a new computational complexity measure: the weight of the computation to assess its feasibility.

Consider a graph with 200 nodes and assume that to extract a Hamiltonian path we need to generate an exponential number of paths of length *n*. A lower bound for the encoding of the edges and the paths in the graph in DNA sequences is $\log_4 200$ bases per edge and a low estimate of the weight per base is 10^{-25}kg. Thus the biologically encoded set of paths will weigh more than

$$2^{200} \bullet \log_4 200 \bullet 10^{-25}\text{ kg} \geq (2^4)^{50} \bullet 3 \bullet 10^{-25}\text{ kg} \geq 3 \bullet 10^{25}\text{ kg}$$

which is more than the weight of the Earth.

Adleman's molecular solution of the Hamiltonian path problem is indeed a magnificent achievement and may initiate a more intensive exploration of molecular computing and computing in biological systems. At the same time, the exponential function grows too fast and the atoms are a bit too heavy to hope that molecular computing can break the exponential barrier, this time the weight barrier.

This leaves us with the difficult task of understanding what computations can be performed below the exponential computational resource bounds imposed by nature.

References

Adleman, Leonard M, "Molecular Computation of Solutions of Combinatorial Problems." *Science*, Vol. 266, 11 November, 1994, pages 1021-1024.

* Supported in part by National Science Foundation Grant *CCR-91-23730.*