Why Encubation?

Vladik Kreinovich,
Rohan Baingolkar,
Swapnil S. Chauhan, and
Ishtjot S. Kamboj

Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu
rubaingolkar@miners.utep.edu
sschauhan@miners.utep.edu
iskambo@miners.utep.edu

1. What Is Encubation

- It is known that:
  - some algorithms are feasible, and
  - some take too long to be practical.
- For example:
  - if the running time of an algorithm is 2^n, where $n = \text{len}(x)$ is the bit size of the input x,
  - then already for $n = 500$, the computation time exceeds the lifetime of the Universe.
- In computer science, it is usually assumed that an algorithm A is feasible if and only if A is *polynomial-time*.

2. What Is Encubation (cont-d)

- In other words, an algorithm is feasible if:
  - its number of computational steps $t_A(x)$ on any input x
  - is bounded by a polynomial $P(n)$ of the input length $n = \text{len}(x)$.
- An interesting *encubation* phenomenon is that:
  - once we succeed in finding a polynomial-time algorithm for solving a problem,
  - eventually it turns out to be possible to further decrease its computation time
  - until we either reach the cubic time $t_A(x) \approx n^3$ or reach some even faster time n^α for $\alpha < 3$.

3. How to Explain Encubation?

- According to modern physics, the Universe has $\approx 10^{90}$ particles.
- There are $\approx 10^{42}$ moments of time.
- The number of moments of time can be obtained if we divide:
  - the lifetime of the Universe ($T \approx 20$ billion years)
  - by the smallest possible time Δt.
- Δt is the time that light passes through the size-wise smallest possible stable particle – a proton.
- This means that overall:
  - even if each elementary particle is a processor that operates as fast as physically possible,
  - the largest possible number of computational steps that we can perform is $10^{90} \cdot 10^{42} = 10^{132}$.

4. How to Explain Encubation (cont-d)

- This is the largest possible number of computational steps $t(n)$.
- The largest possible input size comes if you input 1 bit per unit time.
- Thus, during the lifetime of the Universe, the largest possible length of the input is $n \approx 10^{42}$ bits.
- If an algorithm is feasible, then:
  - for the largest possible length n of the input
  - it should still perform the physically possible number of steps.
- For $t(n) \approx n^\alpha$ and $n \approx 10^{42}$ this means that

$$t(n) \approx n^\alpha \le 10^{132}.$$

- Thus, we get $\alpha \le \dfrac{132}{42} = \dfrac{22}{7} \approx 3.$

5. How to Explain Encubation (cont-d)

- For $t_A(n) = n^\alpha$, we got $\alpha \le \dfrac{22}{7} \approx 3$.
- This is exactly what we want to explain.
- *Comment.* Since $\dfrac{22}{7} \approx \pi$, maybe π and not 3 is the actual upper bound?

6. What About Human Computations?

- What if instead computability in a computer we consider computability in a human brain?
- Let us repeat similar computations for such human computing.
- A human life lasts for ≈ 80 years.
- Each year has ≈ 30 million second, so overall, we get $\approx 2.4 \cdot 10^9$ seconds.
- Brain processing is performed by neurons.
- Typical neurons involved in thinking and processing data have an operation time about 100 milliseconds.
- This is about 0.1 seconds.
- Thus, during the lifetime, we have $\approx 2.4 \cdot 10^{10}$ moments of time.

7. Human Computations (cont-d)

- There are about 10^{10} neuron in a brain.
- Thus, overall:
  - if all the neurons are active all the time,
  - we can perform $t(n) \approx (2.4 \cdot 10^{10}) \cdot 10^{10} \approx 10^{20}$ computational steps.
- Similarly to the physical case:
  - we can gauge the largest possible size
  - by assuming that enter 1 bit every single moment of time.
- Thus, the largest input size is $n \approx 10^{10}$.

8. Human Computations (cont-d)

- Similarly to the physical case, let us check for which α:
  - the number of computational steps $t(n)$ needed to process the largest possible input $n \approx 10^{10}$
  - does not exceed the largest possible number of computation al steps: $t(n) = n^\alpha \leq 10^{20}$.
- In this case, we conclude that $\alpha \leq 2$.
- So, only quadratic-time (and faster) algorithms are feasible in terms of human computations.
- This makes sense; for example:
  - sorting algorithms that describe how we sort by hand (such as insertion sort),
  - are indeed quadratic-time.