White- and Black-Box Computing and Measurements under Limited Resources: Cloud, High Performance, and Quantum Computing, and Two Case Studies – Robotic Boat and Hierarchical Covid Testing

Vladik Kreinovich, Martine Ceberio, and Olga Kosheleva

University of Texas at El Paso
El Paso, Texas 79968, USA
vladik@utep.edu, mceberio@utep.edu, olgak@utep.edu

1. Formulation of the Problem

- Major objective of science: to predict the future.
- Major objective of engineering: make the future better.
- For both problems, we need to measure different quantities and process measurement results.
- Some measurements and computations require a lot of resources, but our resources are limited.
- We need to develop ways to measure and compute under different resource limitations.
- In this talk, we overview major resource limitations and how to handle them.
- We will distinguish between white-box and black-box (proprietary or classified code) situations.
- We also (briefly) describe two case studies.

2. White-Box Computing: Three Types of Situations and Related Resource Limitations

- We distinguish regular-scale, large-scale (high performance), and small-scale (e.g., cell phone) computing.
- For *small-scale computing*, computational ability is not a problem.
- The main need for such small-scale devices comes from the fact that regular computers are not very portable.
- This portability is a problem: to perform computations, we need energy.
- For portable devices, energy is a problem:
  - we can only plug it in once in a while, and
  - in a small volume, we can only store a limited amount of energy.
- So, the main resource limitation is *energy*.

3. White-Box Computing (cont-d)

- This requires a serious change in algorithms – since most traditional algorithms minimize computation time.
- In particular, auxiliary procedures like garbage collection have to be performed only based on need.
- For *regular-size computing:*
  - computational ability is not a problem,
  - otherwise, we would have needed a high-performance computer.
- Energy is also not a problem: we just plug in.
- The main limited resource is a very mundane one: money.
- We can save money since the amount of needed computations changes with time.
- To cover sometimes excessive need, we can rent computation time; this is known as *cloud computing*.

4. Cloud Computing (cont-d)

- How much to rent and how many computers to buy?
- Suppose that in-house we spend c_0 per computation, out-of-house c_1, and $\rho(x)$ is pdf of computer needs.
- We need to select the computer power x_0 to buy that minimizes the total cost:

$$c_0 \cdot x_0 + \int_{x_0}^{\infty} c_1 \cdot (x - x_0) \cdot \rho(x)\, dx.$$

- The optimal x_0 satisfies $F(x_0) = 1 - \dfrac{c_0}{c_1}$ for cdf $F(x)$.
- Similar algorithms exist for more complex situations – e.g., when we know the values with uncertainty.

5. Large-Scale Computing: Resource Limitations

- A high-performance computer consists of usual processors, needs a lot of energy.
- When we design a high-performance computer, we maximize the overall number of computations per second.
- So we run all processors at maximum speed.
- To save energy, we use more processors – but run them at speed f that minimizes watt/operation $F(f)/f$.
- Processors not needed should be fully idle.
- How can we further speed up? Due to speed of light, a 30-cm laptop take 1 ns to cross: time of 4 operations.
- To speed up, we need to make computers smaller – this leads to the micro-size domain of quantum physics.

6. Quantum Computing: Resource Limitations

- Many efficient quantum algorithms exist.
- E.g., Grover's algorithm finds an element in an unsorted array of n elements in time $\sqrt{n}$.
- In quantum physics, instead of a bit, we have qubits – superpositions $c_0|0\rangle + c_1|1\rangle$.
- Grover's algorithm requires n qubits; qubits are now the main limiting resource; we have $s \ll n$ qubits.
- Solution: we divide the array into n/s subarrays, take time $\sqrt{s}$ for each, overall time $n/\sqrt{s} \ll n$.

7. Black-Box Computing: Resource Limitations

- Usually, commercial software provides a turn-key solution to the corresponding problem:
  - the software produces the result $y = f(x_1, \ldots, x_n)$ of processing the inputs $x_1, \ldots, x_n$,
  - but *not* the accuracy of this result.
- This is important: if oil deposit estimate is 200 ± 20, start drilling, but if 200 ± 300, maybe there is no oil?
- Usually, the measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ are small, so, we can keep only linear terms in

$$\Delta y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n):$$

$$\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i, \text{ where } c_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}.$$

8. Black-Box Computing (cont-d)

- Sometimes, we know the probability distribution of each measurement error Δx_i.
- Then, we can use Monte-Carlo simulations to find the distribution of Δy, or compute $\sigma = \sqrt{\sum\limits_{i=1}^{n} c_i^2 \cdot \sigma_i^2}$.
- Sometimes, we only know the upper bound Δ_i on each measurement error Δx_i: $|\Delta x_i| \le \Delta_i$.
- In this case, the upper bound Δ on Δy is $\sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.
- If we compute it directly, we need $n + 1$ calls to f:
  - one call to apply f to inputs, and
  - n calls to compute n partial derivatives.
- For large n and complex f, this too long.

9. Black-Box Computing (cont-d)

- *Alternative:* simulate Δx_i as Cauchy-distributed with parameter Δ_i; $\rho_i(\Delta x_i) \sim 1/(1+(x_i/\Delta_i)^2)$.
- Then, Δy is Cauchy-distributed with desired Δ.
- If we use N iterations, we get Δ with accuracy $1/\sqrt{N}$.
- So, to get accuracy 10%, we need $N = 100$ calls to f.
- For large $n \gg 100$, this is much faster than numerically computing n partial derivatives.
- Additional speed up is attainable if some inputs are irrelevant.
- Quantum Deutsch-Jozsa algorithm helps decide on this.
- E.g., for 1-bit input, deciding whether $f(0) = f(1)$ is done in 1 call to f.

10. Measurements under Limited Resources

- Measurement resources – time, energy, etc. – are also limited.
- We describe case studies corresponding to two possible types of situations:
  - when we can only measure individual quantities, and
  - when we can measure combinations of different quantities.
- First case study: a robotic boat floats with the small river and provides a detailed map.
- In some parts, the depths etc. do not change much. To save energy, we need to measure rarely.
- In other parts, the river changes fast, so we need frequent measurements.

11. Robotic Boat (cont-d)

- Idea: we fit measurements-so-far by, e.g., a polynomial; this predicts future depths and their accuracy.
- Least squares can do it for probabilistic uncertainty.
- Linear programming helps when we only know upper bounds Δ_i.
- We only resume measurements when the predicted inaccuracy exceeds a certain threshold.

12. 2nd Case Study: Covid-19 Testing

- The number of test kits is a limitation.
- Known idea: apply each test to a combined sample from several (s_1) people.
- Those from a positive group need to be tested further, but others are good.
- We then combine to-be-tested folks into groups of $s_2 < s_1$, test again, etc.; on stage $n+1$, we test individually.
- What is the optimal arrangement?
- Let p is an empirically known frequency of Covid in population.
- p is small, so for sufficiently small s_k, the probability that we have two positive folks in each group is small.
- So, after k stages, we have $N \cdot p \cdot s_k$ possibly-positives.

13. Covid-19 Testing (cont-d)

- The overall number of tests is
$$\frac{N}{s_1} + \frac{N \cdot p \cdot s_1}{s_2} + \ldots + \frac{N \cdot p \cdot s_{k-1}}{s_k} + \frac{N \cdot p \cdot s_k}{s_{k+1}} + \ldots$$
- Minimizing w.r.t. s_k leads to $-\frac{s_{k-1}}{s_k^2} + \frac{1}{s_{k+1}} = 0.$
- So $s_k/s_{k+1} = s_{k-1}/s_k$ and s_k is a geometric progression.
- At the end, we check individually, so $s_{n+1} = 1$, $s_n = q$, $s_{n-2} = q^2$, $\ldots$, $s_1 = q^n$, so $n = \ln(s_1)/\ln(q)$.
- The number of tests is
$$\frac{N}{s_1} + n \cdot N \cdot p \cdot q = \frac{N}{s_1} + \frac{\ln(s_1)}{\ln(q)} \cdot N \cdot p \cdot q.$$
- Minimizing with respect to q means minimizing $q/\ln(q)$, so $q = e$, and the number of tests is $\frac{N}{s_1} + N \cdot p \cdot e \cdot \ln(s_1)$.

14. Covid-19 Testing (cont-d)

- *Reminder:* the number of tests is
$$\frac{N}{s_1} + N \cdot p \cdot e \cdot \ln(s_1).$$
- Minimizing with respect to s_1 leads to $s_1 = \dfrac{1}{p \cdot e}$.
- So, we start with groups of this size, then, we take:
$$s_2 = \frac{s_1}{e}, \quad s_3 = \frac{s_1}{e^2}, \ldots$$
- Overall, we need $\sim N \cdot p \cdot \ln(p)$ tests.
- One can show that this is asymptotically optimal.

15. Acknowledgments

This work was supported in part by the National Science Foundation grants:

- 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science);
- HRD-1834620 and HRD-2034030 (CAHSI Includes).