thehindu.com

# How quantum algorithms solve problems that classical computers can't

*S. Srinivasan*

7–8 minutes

---

We often hear that quantum computers efficiently solve problems that are very difficult to solve with a classical computer. But even if the hardware is available to build a quantum computer, exploiting its quantum features requires us to write smart algorithms.

An algorithm is a sequence of logically connected mathematical steps that solve a problem. For example, an algorithm to add three numbers can have two steps: add the first two numbers in the first step and the result to the third number in the second step.

## Quantum v. classical algorithms

A more involved example of an algorithm is the search for the largest number in a finite list of numbers.

An algorithm can start by assuming that the first number on the list is the largest. Next, it can compare this number with the second number on the list. If the second number is larger than or equal to the first number, the second number is now deemed to be the largest. Otherwise, the first number remains the largest at this stage. The algorithm then moves to the third number on the list – and so on until it has finished comparing all the numbers on the list. The number that is the largest as of the final step will be the answer.

A *quantum algorithm* is also a series of steps, but its implementation requires *quantum gates*. Some problems may need fewer steps on the part of a quantum algorithm than the number of steps required by a classical algorithm. That is, the quantum algorithm can speed up the computation.

One factor that controls this speed-up is the possibility of superposition of the states of quantum bits, or qubits, that encode information. Whereas a classical computer uses semiconductor-based gadgets as bits to encode information, quantum computers use qubits. In both cases, the bit or the qubit can have two distinct states, 0 *or* 1; but qubits have the additional ability to be partly 0 *and* partly 1 at the same time.

**Shor's algorithm**

One of the earliest quantum algorithms is the factorisation algorithm developed by Peter Shor. It requires fewer steps to factorise a number than one that operates with classical principles.

Shor's algorithm identifies the factors of a given integer. For example, 2 is a factor of 20 (since 2 divides 20 without a remainder). Similarly, 4, 5, and 10 are also factors of 20. However, identifying all the factors requires a greater and greater number of steps if the number becomes larger.

The efficiency of an algorithm is related to the number of steps required as the size of the input increases. An algorithm is more efficient if it requires fewer steps (and thus less time). From this perspective, Shor's algorithm is far more efficient than any known classical algorithm for factorisation.

Technically, in Shor's algorithm, the number of steps increases as a polynomial in the size (more precisely, the logarithm of the size) of the input whereas it is a superpolynomial for the best classical algorithm known today.

To understand the difference, compare multiplying 10 with itself thrice (i.e. 10^3) and multiplying 3 ten times (i.e. 3^10). The former

is a polynomial in 10 whereas the latter is a superpolynomial in 10. A polynomial increase is always lower than a superpolynomial increase for a sufficiently large input size. Thus, classical factorisation algorithms are far less efficient compared to Shor's algorithm, which is a quantum algorithm.

Modern cryptography – which is used to secure user accounts on the internet, for example – depends on the fact that there are no efficient classical algorithms that can factorise large integers. This is the source of the claim that the availability of quantum computers (with an adequate number of qubits) will challenge the safety of classical cryptography.

**Grover's and Deutsch-Jozsa algorithms**

Another popular quantum algorithm is the quantum search algorithm developed by Lov Grover. It looks for a numerical pattern in a large list of numbers. A deterministic classical algorithm requires almost half the number of steps as there are patterns in the list. That is, to identify a pattern from a list of one-million patterns, the classical approach may need half a million steps. The quantum algorithm will require only a thousand steps, however. In fact, for every 100x increase in the list's size, Grover's algorithm will need only 10x more steps. This is the kind of speed-up this quantum algorithm achieves.

Yet another scheme that showcases the exponential speed-up is the Deutsch-Jozsa algorithm. Imagine a set containing two-digit numbers whose digits are either 0 or 1; let's call this Set A: 00, 01, 10, and 11. For each number from Set A, associate a number from another set, Set B, containing 0 and 1 as the only members.

Next, consider two categories of relation between the two sets. A relation is *constant* if all the members of the first set are associated with only 0 or only 1. A relation is *balanced* if two of the numbers from the first set are associated with 0 and the other two with 1.

Say the output is 0. A classical computer will require three steps at

most to determine if the mapping is constant or balanced. (Can you figure out what they are?)

But a quantum computer can figure it out with only one computation. This is thanks to superposition – the ability of the value of a qubit to be partly 0 and partly 1 at the same time.

As this author wrote previously, "If a qubit is in a superposition, then measuring the qubit will cause it to collapse to one of the two states [*either* 0 *or* 1]. However, we can only predict the probability that it will collapse to one state."

When the inputs are in superposition, the output will be as well, and in a way that corresponds to the states in the input superposition. The output will also have a sign – positive or negative – depending on whether the association is balanced or constant.

So the Deutsch-Jozsa algorithm can determine the mapping with one computation independent of the size of the input. We just need to make sure there are enough qubits available to represent the number of digits in the input. (Of course, this requirement would apply to bits as well).

**Wait for reliable devices**

Scientists already know of more quantum algorithms that can solve problems in optimisation, drug design, and pattern search, among other fields more efficiently.

When reliable, large-scale devices become available, quantum computing will help address many otherwise intractable problems as well. Research in quantum algorithms is highly interdisciplinary, involving computer science, mathematics, and physics. The field is also still evolving, and there are plenty of opportunities to make significant contributions.

*S. Srinivasan is a professor of physics at Krea University.*