

Running quantum software on a classical computer

August 3 2021

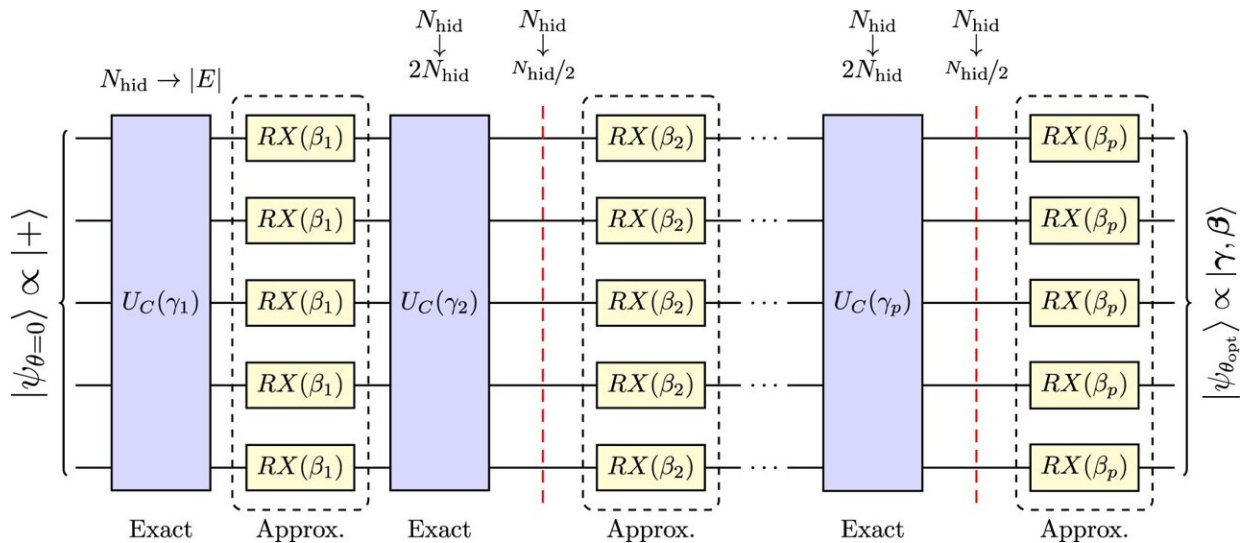


Fig. 1: The QAOA quantum circuit. A schematic representation of the QAOA circuit and our approach to simulating it. The input state is trivially initialized to $|+\rangle|+\rangle$. Next, at each p , the exchange of exactly (UC) and approximately ($RX(\beta) = e^{-i\beta X}$) applicable gates is labeled (see Sec. Methods). As noted in the main text, each (exact) application of the UC gate leads to an increase in the number of hidden units by $|E|$ (the number of edges in the graph). In order to keep that number constant, we "compress" the model (see Sec. Methods), indicated by red dashed lines after each UC gate. The compression is repeated at each layer after the first, halving the number of hidden units each time, immediately after doubling it with UC gates. After the final layer, the RBM is parametrized by θ_{opt} , approximating the final QAOA target state $|\gamma, \beta\rangle|\gamma, \beta\rangle$. Credit: DOI: 10.1038/s41534-021-00440-z

Two physicists, from EPFL and Columbia University, have introduced an approach for simulating the quantum approximate optimization algorithm using a traditional computer. Instead of running the algorithm on advanced quantum processors, the new approach uses a classical machine-learning algorithm that closely mimics the behavior of near-term quantum computers.

In a paper published in *Nature Quantum Information*, EPFL professor Giuseppe Carleo and Matija Medvidović, a [graduate student](#) at Columbia University and at the Flatiron Institute in New York, have found a way to execute a complex quantum computing [algorithm](#) on traditional computers instead of quantum ones.

The specific "quantum software" they are considering is known as Quantum Approximate Optimization Algorithm (QAOA) and is used to solve classical optimization problems in mathematics; it's essentially a way of picking the best solution to a problem out of a set of possible solutions. "There is a lot of interest in understanding what problems can be solved efficiently by a quantum computer, and QAOA is one of the more prominent candidates," says Carleo.

Ultimately, QAOA is meant to help us on the way to the famed "quantum speedup," the predicted boost in processing speed that we can achieve with quantum computers instead of conventional ones. Understandably, QAOA has a number of proponents, including Google, who have their sights set on quantum technologies and computing in the near future: in 2019 they created Sycamore, a 53-qubit [quantum processor](#), and used it to run a task it estimated it would take a state-of-the-art classical supercomputer around 10,000 years to complete. Sycamore ran the same task in 200 seconds.

"But the barrier of "quantum speedup" is all but rigid and it is being continuously reshaped by new research, also thanks to the progress in the

development of more efficient classical algorithms," says Carleo.

In their study, Carleo and Medvidović address a key open question in the field: can algorithms running on current and near-term quantum computers offer a significant advantage over classical algorithms for tasks of practical interest? "If we are to answer that question, we first need to understand the limits of classical computing in simulating quantum systems," says Carleo. This is especially important since the current generation of quantum processors operate in a regime where they make errors when running quantum "software," and can therefore only run algorithms of limited complexity.

Using conventional computers, the two researchers developed a method that can approximately simulate the behavior of a special class of algorithms known as variational quantum algorithms, which are ways of working out the lowest energy state, or "ground state" of a quantum system. QAOA is one important example of such family of quantum algorithms, that researchers believe are among the most promising candidates for "quantum advantage" in near-term quantum computers.

The approach is based on the idea that modern machine-learning tools, e.g. the ones used in learning complex games like Go, can also be used to learn and emulate the inner workings of a quantum computer. The key tool for these simulations are Neural Network Quantum States, an artificial neural network that Carleo developed in 2016 with Matthias Troyer, and that was now used for the first time to simulate QAOA. The results are considered the province of quantum computing, and set a new benchmark for the future development of quantum hardware.

"Our work shows that the QAOA you can run on current and near-term quantum computers can be simulated, with good accuracy, on a classical [computer](#) too," says Carleo. "However, this does not mean that alluseful quantum algorithms that can be run on near-term quantum processors

can be emulated classically. In fact, we hope that our approach will serve as a guide to devise new quantum algorithms that are both useful and hard to simulate for classical computers."

More information: Matija Medvidović et al, Classical variational simulation of the Quantum Approximate Optimization Algorithm, *npj Quantum Information* (2021). [DOI: 10.1038/s41534-021-00440-z](https://doi.org/10.1038/s41534-021-00440-z)

Provided by Ecole Polytechnique Federale de Lausanne

Citation: Running quantum software on a classical computer (2021, August 3) retrieved 20 April 2024 from <https://phys.org/news/2021-08-quantum-software-classical.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.