16

Quantum adiabatic algorithm

The authors are grateful to Dong An for reviewing this chapter.

Rough overview (in words)

The *quantum adiabatic algorithm* (QAA) [382], sometimes referred to as *adiabatic state preparation*, is a continuous-time procedure for (approximately) preparing an eigenstate (typically the ground state) of a particular Hamiltonian of interest on a quantum device. The QAA also forms the basis for a model of quantum computation called *adiabatic quantum computation* which acts as an alternative to the standard quantum circuit model.

The main idea of the QAA is to begin in an eigenstate of a simpler Hamiltonian that is easy to prepare, and then slowly change the Hamiltonian to be equal to the more complex Hamiltonian of interest. The adiabatic theorem (see [16] and references therein), a celebrated concept from physics, dictates that if the evolution is sufficiently slow, the system will evolve to (approximately) remain in the instantaneous eigenstate of the continuously varying Hamiltonian and thus finish in the desired state. The length of time required for the evolution to succeed depends on the spectral properties of the Hamiltonian path and in particular on the minimum spectral gap. The adiabatic algorithm can be simulated on a gate-based quantum computer with time-dependent Hamiltonian simulation.

Rough overview (in math)

Let $H(s)$, where s varies as $0 \leq s \leq 1$, denote a single-parameter path through the space of Hamiltonians, and let $|\phi_j(s)\rangle$ and $E_j(s)$ denote the eigenstates and eigenvalues of $H(s)$, indexed by j in increasing order. The goal of the QAA is to prepare a certain eigenstate $|\phi_j(1)\rangle$ of $H(1)$. Let $|\psi(t)\rangle$ denote the state of

https://doi.org/10.1017/9781009639651.019 Published online by Cambridge University Press

our system at time t and let T be the total evolution time. The procedure calls for beginning in the state $|\psi(0)\rangle = |\phi_j(0)\rangle$ and allowing $|\psi(t)\rangle$ to evolve by the Schrödinger equation according to the Hamiltonian $H(t/T)$, that is, $\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H(t/T)|\psi(t)\rangle$ from $t = 0$ to $t = T$. Thus, as T is made larger, the path from $H(0)$ to $H(1)$ is traversed increasingly slowly.

Dominant resource cost (gates/qubits)

The main resource for the continuous-time QAA is the total evolution time T. The adiabatic theorem suggests that if T is chosen sufficiently large, and as long as eigenvalue E_j is nondegenerate along the entire path, then $|\psi(T)\rangle \approx |\phi_j(1)\rangle$ will hold. The often-quoted heuristic condition [16] for success is that

$$T \gg \max_{0\le s\le 1} \frac{\left\|\frac{\mathrm{d}H}{\mathrm{d}s}\right\|}{\Delta(s)^2}\,, \tag{16.1}$$

where $\Delta(s)$ is the spectral gap, that is, $\min_{i\neq j}|E_i(s) - E_j(s)|$, and $\|\cdot\|$ denotes the spectral norm. Thus, the runtime needed for the QAA to have small error is primarily governed by the minimum size of the spectral gap along the adiabatic path. This aspect of the QAA is a common sticking point as it is often difficult to produce lower bounds on $\Delta(s)$ that would suffice for proving upper bounds on T. In practice, the value of T can be chosen heuristically, or by trial-and-error, but a more detailed understanding of $\Delta(s)$ would inform smarter choices of Hamiltonian path $H(s)$.

While Eq. (16.1) is nonrigorous and potentially loose in specific scenarios, the polynomial dependence of T on the inverse spectral gap is an essential feature of the QAA. For example, it was shown in [161] that in general, any rigorous bound on the QAA runtime must scale at least linearly in the inverse spectral gap.

The QAA is typically formulated as a continuous-time procedure, but a gate-based quantum computer can simulate the QAA by discretizing the path and approximately implementing the evolution from time t to $t + \delta t$ with product formulas or with more advanced techniques for time-dependent Hamiltonian simulation. This incurs error in addition to the adiabatic error of the continuous-time QAA. The number of gates needed to do this can be made proportional to T (up to logarithmic corrections), polynomial in the number of qubits needed to hold the state $|\psi(t)\rangle$, and logarithmic in the approximation error incurred (e.g., [615]).

Caveats

A technical caveat of the QAA is that rigorous formulations of sufficient conditions for success (e.g., [567, 376]) are more complex than Eq. (16.1) and

https://doi.org/10.1017/9781009639651.019 Published online by Cambridge University Press

likely looser than what is necessary in practice. Also, in most cases, the dependence of the runtime T on the final approximation error $\epsilon = \||\psi(T)\rangle - |\phi_j(1)\rangle\|$ goes as $T = \text{poly}(1/\epsilon)$, rather than $T = \text{polylog}(1/\epsilon)$. To circumvent this and achieve polylog$(1/\epsilon)$ dependence, one can choose more sophisticated Hamiltonian paths $H(s)$ for which all time derivatives vanish at $s = 0$ and $s = 1$ [411, 16].

A practical caveat of the QAA is that the spectral gap—the main determiner of the resource cost—is difficult to study theoretically. Numerically, it can often be computed only for small system sizes, and it is unclear whether extrapolations to larger system sizes would be accurate.

Furthermore, in many end-to-end applications the spectral gap is not only unknown, but also expected to be extremely small, implying that the QAA requires a large runtime (see §Example use cases, below). For the QAA to be efficient, the spectral gap must decay only like an inverse polynomial of the system size, but such scaling requires the problem instance to have a special structure. Such structure should not be assumed to exist without justification.

NISQ implementations

The QAA is closely related to the concept of *quantum annealing* [591], a term used especially in the context of near-term implementations on existing quantum hardware. In quantum annealing, the system is exposed to a time-dependent Hamiltonian, typically a transverse-field Ising model. The strength of the transverse field is slowly reduced, eventually to zero, where the Hamiltonian is equal to a classical Ising model encoding a hard combinatorial optimization problem. If implemented perfectly and sufficiently slowly, this would be a manifestation of the QAA, and one would obtain the solution to the problem. However, the typical setting of quantum annealing is to consider faster implementations, and to possibly allow for some amount of control noise and finite-temperature effects (rather than evolving under a closed system at zero temperature), which induce transitions from the ground state to excited states. The goal is relaxed from ending in the exact ground state of the final Hamiltonian to ending in a low-energy state that can be considered an *approximately* optimal solution to the problem. The success metric is often the quality of the solution produced rather than the runtime required to find the best solution. As such, it is a heuristic algorithm and must be compared with classical heuristic algorithms, where evidence of a scalable advantage is mixed. See, for example, [323] for a perspective on quantum annealing and the most promising related directions.

Separately, the QAA can be related to variational quantum algorithms, which are NISQ friendly. In particular, by applying product formulas to the

https://doi.org/10.1017/9781009639651.019 Published online by Cambridge University Press

QAA, one obtains alternating time evolutions by $H(0)$ and by $H(1)$; in the case that $H(0)$ is a transverse field and $H(1)$ is a classical cost function, this is precisely an instance of the quantum approximate optimization algorithm (QAOA) [384], a leading NISQ algorithm. In the limit of large depth, the QAOA can fully simulate the QAA to arbitrarily small precision. However, in a NISQ setting, the depth of the QAOA would need to be restricted, and the QAOA would not exactly follow the QAA.

Example use cases

- Combinatorial optimization: The QAA was first invented [382] as a way to solve hard classical combinatorial optimization problems on a quantum computer. An example is constraint satisfaction problems, where one is given a Hamiltonian $H(1)$ that is diagonal in the computational basis (i.e., "classical") and equal to the sum of various constraints on n bits. The ground state of $H(1)$ is the bit string that violates the fewest constraints. One typically chooses the initial Hamiltonian to be $H(0) = -\sum_{i=1}^{n} X_i$, where X_i denotes the Pauli-X operator on qubit i, whose ground state is an easy-to-prepare product state. The QAA is guaranteed to find the ground state of $H(1)$ if it is run with sufficiently large evolution time. However, in general, it is expected that the spectral gaps along the adiabatic path become exponentially small in n [638, 1071, 517, 23, 1032], indicating that the QAA requires exponentially long runtime.
- Quantum chemistry and condensed matter physics: A central problem of quantum chemistry and computational condensed matter physics is the problem of finding the ground state energy of a molecule, material, or lattice model. This can be solved efficiently with quantum phase estimation (QPE) so long as one can prepare a state that has substantial overlap with the ground state of the Hamiltonian. Adiabatic state preparation has been proposed as a method for producing such a state (see, e.g., [1056, 870, 1006, 644, 966, 1031, 1072]). This initial state preparation is often the bottleneck in the end-to-end quantum solution, as it can require exponential time for systems of interest (see, e.g., [670]).
- Quantum linear system solvers: The state-of-the-art quantum linear system solvers [313] leverage the QAA to produce a quantum state $|x\rangle$ corresponding to the solution of a linear system $Ax = b$ (see also [964, 31, 689, 572]). In particular, this method allows the runtime to scale linearly in the condition number of the matrix A.

https://doi.org/10.1017/9781009639651.019 Published online by Cambridge University Press

Further reading

- See [16] for a comprehensive 2018 review of the QAA and adiabatic quantum computation more generally.
- See [162] for a digital version of the QAA for a gate-based quantum computer, but distinct from a direct simulation of the QAA. The idea is to choose a sequence of s values $0 = s_0 < s_1 < s_2 < \cdots < s_T = 1$ and perform measurements of $H(s_t)$ for $t = 0, \ldots, T$ in sequence using QPE. As long as the difference between consecutive values of s is sufficiently small, the quantum Zeno effect guarantees that each measurement will project onto the correct eigenstate $|\phi_j(s_t)\rangle$ with high probability (see also [944, 673]). One can also take larger jumps, and amplify their success probability with fixed-point amplitude amplification. The resource cost has a similar dependence on the spectral gap as the continuous-time QAA: if the "path length" traced by the eigenstate $|\phi_j(s)\rangle$ is L, the minimum gap is Δ, and the target error is ϵ, then the gate cost of the algorithm is $O(L \log(L/\epsilon)/\Delta)$. The path length L can be upper bounded by $\|dH/ds\|/\Delta$, which roughly recovers Eq. (16.1).
- Along the lines of the previous bullet, [1012] gives an alternative way to effect adiabatic state preparation on a gate-based computer with polylog$(1/\epsilon)$ overall error dependence, via quasi-adiabatic continuation.

https://doi.org/10.1017/9781009639651.019 Published online by Cambridge University Press