An Introduction to Quantum Annealing

G. Rose, W. G. Macready
D-Wave Systems Inc., 100-4401 Still Creek Dr., Burnaby, BC V5C 6G9 Canada
(Dated: August 10, 2007)

This document describes how quantum annealing can be used as a heuristic approach to solving certain classes of hard optimization problems.

INTRODUCTION

D-Wave processors are designed to harness a fundamental principle of nature that operates in both quantum and classical regimes - the propensity for all physical systems to minimize their free energy [1].

Free energy minimization in a classical system is often referred to as annealing [2]. For example, in metallurgy, annealing a metal involves heating it and then cooling it. This type of thermal annealing allows a metal that is originally filled with defects (a metastable 'high energy' state) to become crystalline and defect-free (the minimum free energy state).

The simulation of this type of thermal annealing using classical computers is known as simulated annealing, which is a commonly used heuristic approach to solving certain classes of hard optimization problems [3]. Instead of having a fixed landscape through which to anneal (as in the metallurgical example), in simulated annealing a programmer defines what the energy landscape is. This energy landscape is crafted so that its global minimum is the answer to the problem to be solved, and low-lying local minima are good approximations.

Free energy minimization in a quantum system is referred to as quantum annealing. As in classical annealing, all quantum systems are driven to minimize their free energy. In non-programmable scenarios (the analog of the metal annealing example, where Nature sets the energy landscape), it has been demonstrated that quantum annealing can hasten the energy minimization process [4, 5].

D-Wave processors compute by piggybacking on quantum annealing. A quantum annealing processor can be operated as a universal quantum computer. In this regime of operation, the computational model is referred to as adiabatic quantum computation (AQC), which can be thought of as the long-time limit of quantum annealing.

D-Wave Hydra processors are analog embodiments of the optimization version of the two dimensional Ising model in a magnetic field problem. This problem is NP-hard [6]. The design philosophy is to use quantum annealing to hasten convergence of the energy of the system towards the ground state energy. If the system can reach its ground state, the configuration of variables returned is the exact solution of the problem. If it can reach a low-lying local minimum, the configuration of variables returned is an approximate solution.

A Hydra processor has the Hamiltonian

$$H_0(t) = \sum_{i=1}^{N} h_i Z_i + \sum_{i<j=2}^{N} J_{ij} Z_i Z_j + \sum_{i=1}^{N} \Delta_i(t) X_i \quad (1)$$

where $Z_i \equiv \hat{\sigma}_z^i$ and $X_i \equiv \hat{\sigma}_x^i$ are the Z and X Pauli matrices respectively for qubit i, h_i is the local bias on qubit i, $\Delta_i(t)$ is the tunneling matrix element for qubit i, and J_{ij} is the coupling strength between qubits i and j.

A problem instance is encoded in the h and J values, which are user-programmable. The transverse term is used to control the quantum annealing schedule as defined in the following section.

CLASSICAL, SIMULATED AND QUANTUM ANNEALING

Consider first a classical system characterized by a vector $p(x)$ giving the occupation probability of all states x. Classically at finite temperature, p is determined by minimizing the free energy functional

$$F_T[p] = E_p(V) - TS[p] = \sum_x E(x)p(x) + T\sum_x p(x)\ln p(x) \quad (2)$$

where $S[p]$ is the entropy of p and $E_p(V)$ is the expected value of the energy function $V(x)$. The free energy divides into two terms. The entropy contribution is convex, i.e. there is a single global minimum at the maximum entropy state where all probabilities are equal. The second contribution arises from the expected energy contribution and for $V(x)$ with local minima is not convex, and has local minima when viewed as a function of p. The idea of simulated annealing is to start at large T so the free energy has a single global minimum which is easily located. As we cool T, the energy term begins to contribute and new local minima can appear. But, if we are at the global minimum of $F_T[p]$ then cooling $T \to T - dT$ where dT is small enough means that we can follow the gradient dF/dp to locate the global minimum of F_{T-dT}. In this way we ratchet down to $T = 0$.

In the simulated annealing algorithm we don't actually determine p, but instead set up a Markov chain which samples from p at each temperature (assuming you let things equilibrate at each T).

The broader story is that at finite temperature there are two forces driving the evolution, a V minimizing term and a term which wants to spread out p (make p uniform). An exactly analogously thing is going on quantum mechanically.

In the QM case, instead of p we need to consider the density matrix ρ. The free energy is now

$$F_T[\rho] = Tr(\rho H) - TS[\rho] \tag{3}$$

where the entropy of ρ is $S[\rho] = -Tr(\rho \ln \rho)$. As above, ρ is determined by minimizing the free energy. The same story holds in this quantum case - the entropy functional is convex while the expected energy is not. In the QM case even at zero temperature (where the entropy term vanishes) there remains a term which is minimized by spreading the wave function out (like making p uniform classically). This is the 'kinetic energy' K where $H = K + V$. In our case the role of kinetic energy is played by the transverse field terms $\sim X_i$ while in continuous systems it is the negative Laplacian. In this zero temperature case we have

$$F_0[\rho] = Tr(\rho V) + Tr(\rho K) \tag{4}$$

The second term is convex and we can play the same game weighting the importance of the two terms so that we can define

$$F_0[\rho] = Tr(\rho V) + \Gamma\ Tr(\rho K) \tag{5}$$

and gradually anneal out ($\Gamma \rightarrow 0$) the convex term. Viewed this way QA is just a different mechanism uniformizing occupation probabilities across states x.

Hydra processors operate in a regime where we must consider the most general case, with free energy

$$F_T[\rho] = Tr(\rho V) + \Gamma\ Tr(\rho K) + T\ Tr(\rho \ln \rho) \tag{6}$$

We have the freedom to choose annealing paths within the whole (T, Γ) space and not just $(T, 0)$ paths (simulated annealing) or $(0, \Gamma)$ paths (quantum annealing).

PERFORMANCE CHARACTERIZATION OF SIMULATED AND QUANTUM ANNEALING

Generic analysis of annealing heuristics on NP-hard optimization problems is difficult for a number of reasons, including instance dependence and the existence of vastly differing time-scales characterizing short-time and asymptotic (long-time) behavior.

A meaningful performance measure is the residual energy as a function of time. For both quantum and classical annealing, the residual energy is believed to approach zero asymptotically. In other words, choosing an exponentially long (in the problem size) annealing schedule will, for both approaches, guarantee asymptotic convergence to the exact solution. The functional forms describing the long-time asymptotic behavior for a range of relevant instance types are believed to be

$$\epsilon_c(t) = A \log^{-2} \alpha t \tag{7}$$

for classical annealing and

$$\epsilon_q(t) = B \log^{-6} \beta t \tag{8}$$

for quantum annealing, where A and B are prefactors with units of energy and α and β are rates with units of s^{-1}. [5]

To make a relative performance characterization of classical and quantum annealing when both are in their long-time limits (where these asymptotic forms might be expected to be indicative of real performance), we equate the two residual energies and ask how long the quantum annealer has to run (t_q) in order to match the performance of a simulated annealer running for some time t_c. This gives

$$t_q = \frac{1}{\beta} 2^{\left(\frac{B}{A}\right)^{1/6} \log^{1/3} \alpha t_c} \tag{9}$$

In practice, a simulated annealing approach in the long-time limit is typically operated for 1-10 hours. To compare the time for a quantum annealer to achieve the same levels of accuracy for these times, we can estimate the quantities $\alpha \sim 10^9$, $\beta \sim 10^3$ and $A \sim B$. With these assumptions we obtain times of $\sim 10ms$ for the quantum annealer for 10 hours of simulated annealing time–a speed-up of more than six orders of magnitude.

It is important to note that for any given problem, heuristics superior to simulated annealing almost always exist. Therefore comparing the performance benefits of quantum vs. classical annealing does not fully answer the question of what the expected speed-up of quantum annealing over the best known classical approaches is. In order to perform this analysis, more specificity with the instance class involved and the specific heuristic being used to solve the problem are required.

In the short-time limit, good heuristics exist that will quickly find local minima–for example there are one-pass heuristics for the two-dimensional Ising model in a magnetic field that scale at worst quadratically with the number of variables. However based on the analysis above, after $\sim 10ms$ the quantum annealer will have already produced a very good answer. Assuming that even in the short-time limit a classical heuristic requires more than this time, the relative performance of the quantum vs. classical annealers will be even more dramatic.

Analyzing the performance of any heuristic approach (and quantum annealing is explicitly heuristic) has many pitfalls. Ultimately the actual performance of the quantum annealer must be extracted empirically by measuring the residual energy as a function of time on benchmark problems of increasing size and complexity.

DEFINITIONS

Adiabatic Quantum Computation

Adiabatic quantum computation (AQC) is an algorithm that yields an exact solution to a computation problem encoded in the ground state of a Hamiltonian H_P. The system is initialized in the ground state of an initial Hamiltonian H_B, whose ground state is easily accessible, and then is adiabatically evolved by slowly changing the Hamiltonian from H_B to H_P. An example evolution Hamiltonian is $H = (1-s)H_B + sH_P$, where s changes monotonically from 0 to 1 during the evolution. The performance of the computation is assessed by the ground state probability P_{0f} vs. the total evolution time t_f. The computation time is the value of t_f that yields $P_{0f} \sim 1$. In general there is no restriction on H_B and H_P except that $[H_B, H_P] \neq 0$. The system must stay in its ground state during the entire computation.

Quantum Annealing

Quantum annealing (QA) is an optimization algorithm that uses quantum mechanics as a source of disorder during the annealing process. In QA, the optimization problem is encoded in a Hamiltonian H_P. The algorithm starts by introducing strong quantum fluctuations by adding a disordering Hamiltonian H' that does not commute with H_P. An example case is $H = H_P + \Gamma H'$, where Γ changes from a large value to zero during the evolution. The disorder is slowly removed by removing H' (reducing Γ). If the process is slow enough, the system will settle in a local minimum close to the exact solution. The slower the evolution, the better the solution that will be achieved. The performance of the computation is assessed via the residual energy (distance from exact solution using the objective function) vs. evolution time. The computation time is the time required to generate a residual energy below some acceptable threshold value. In QA, H_P encodes an optimization problem and therefore H_P is diagonal in the subspace of the qubits that encode the solution, but the system does not necessarily stay in the ground state at all times.

---

[1] The free energy referred to is the Helmholtz free energy F=E-TS; see for example `http://en.wikipedia.org/wiki/Helmholtz_free_energy`.

[2] See for example `http://en.wikipedia.org/wiki/Annealing`.

[3] See for example `http://en.wikipedia.org/wiki/Simulated_annealing`.

[4] Brooke et al., *Science* **284**, 779 (1999).

[5] Santoro et al., *Science* **295**, 2427 (2002).

[6] For an introduction to complexity classes and the concept of NP-hardness, see for example `http://en.wikipedia.org/wiki/NP-hard`.