Numeric Experiments on the Commercial Quantum Computer

Richard H. Warren

This paper describes the creativity that is needed to solve an optimization problem on an adiabatic quantum computer. Current features of adiabatic quantum computing are discussed. Research questions are posed. Quantum complexity is briefly addressed.

We will also discuss the mathematics at the forefront of this new era in computing, describe some of the initial work done at Lockheed Martin on an adiabatic quantum computer, and indicate the distinct mindset that is used when programming this type of machine. The paper begins with a description of the commercial, adiabatic quantum computer and concludes with a discussion of current limitations that point to future research.

Quantum computing is motivated by an expected speedup for solving large problems. In 2010 experimenters estimated the time for adiabatic quantum optimization would be about 4 to 6 orders of magnitude faster than for classical solvers of large problems [8]. New results show a variation of speedup. The largest reported is a speedup of about 3600 times for two types of problems on specific software packages [10]. The adiabatic quantum computing model is polynomially equivalent to the quantum circuit model [1], which is the standard quantum computation method. This means that Shor's factoring algorithm can be implemented on an adiabatic quantum machine, but to our knowledge no one has done this.

Richard H. Warren is retired after thirty years with Lockheed Martin as an applied mathematician. He learned his applied skills at Wright-Patterson Air Force Base, Ohio, as a new Ph.D. His email address is `rwarre01@villanova.edu`.

DOI: http://dx.doi.org/10.1090/noti1059

Background Information

Commercial quantum computers are constructed only by D-Wave Systems, a Canadian technology firm. Lockheed Martin Corporation purchased the first one in 2011 for a reported 10 million dollars. The machine was moved from Canada to the University of Southern California's School of Engineering [15]. A large percentage of the operating time is devoted to academic computing, with the rest reserved for Lockheed Martin applications.

This computer is an adiabatic quantum computer, not the popularized gate array type. Essentially, it does one thing: it solves discrete minimization problems extremely fast. In order to do this, it needs an objective function and constraints, both expressed in binary variables. This requires declaring the variables, their types, and their coefficients. There are no procedural instructions, such as "if X, then do Y".

In 2013 it was announced that a consortium of Google, NASA AMES, and the nonprofit Universities Space Research Association had purchased an adiabatic quantum computer from D-Wave [7].

Adiabatic Quantum Computing

In the commercial adiabatic quantum computer, the quantum bits (qubits) are loops of superconducting wire, the coupling between qubits is magnetic wiring, and the machine is supercooled. Reference [8] describes this superconducting adiabatic quantum processor. Fabrication limits the number of pairwise-coupled qubits, which in turn limits the number of variables for problems that are implemented on the computer.

Quantum annealing is a process where the qubits achieve an optimal state of low energy when supercooled. The Ising objective function for this

Photo courtesy University of Southern California.
Photographer: Steve Cohn.

On the left: Professor Daniel Lidar, scientific director of the Quantum Computing Center, with USC engineering dean Yannis Yortsos in front of an adiabatic quantum computer.

optimal state is

$$\min\left(\sum_{(i,j)} s_i J_{ij} s_j + \sum_i h_i s_i\right), \tag{1}$$

where i and j are qubits, s_i is the input state of qubit i (either 0 or 1), h_i is the energy bias for qubit i, and J_{ij} is the coupling energy between qubits i and j. Quantum annealing can be thought of as a path from an initial state to a final state according to the weights h_i and J_{ij} which minimizes energy. There are difficulties achieving the theoretical features of quantum annealing, but there are techniques to mitigate the obstacles [10].

Also, there are questions whether D-Wave machines are true quantum systems. Boixo and his colleagues report some evidence that they are [3], [4]. The discussion is ongoing [16], [17].

Next we will describe how to transform optimization problems into the form of the Ising function (1) and solve them on an adiabatic quantum computer. This means the optimization problem needs to be expressed in binary variables corresponding to s in the Ising function and is limited to linear and quadratic terms corresponding to $h_i s_i$ and $s_i J_{ij} s_j$ in the Ising function. This requires creativity to exploit a problem's structure, often in new ways, in order to adapt it to an adiabatic quantum computer. Thus, a unique mindset is used to program this machine.

Reference [9] indicates that many NP-complete problems have been transformed into quadratic binary problems that the Ising model (1) requires. In addition, the work at Lockheed Martin has transformed the traveling salesman problem [13], the job shop problem for one machine, and a logistics problem into the form of the Ising model (1). The first two problems were implemented on the Lockheed Martin adiabatic quantum machine. Numeric results showed that the probability of finding an optimal solution increases as the quantum annealing process is done more frequently. This correlates with the theory in [12].

Other problems that have been implemented on an adiabatic quantum computer include machine learning and anomaly detection [11], quadratic unconstrained binary optimization, weighted maximum 2-satisfiability, and the quadratic assignment problem [10]. Also, [10] cites finding Ramsey numbers, binary classification in image matching, and 3D protein folding on earlier D-Wave machines with fewer qubits.

Adapting an Optimization Problem to an Adiabatic Quantum Computer

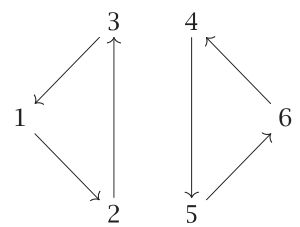
As in 0-1 integer programming, an initial step identifies binary variables for the problem, i.e., variables that are restricted to 0 or 1 and represent the problem. The goal is to use the binary variables (with real coefficients) to write an objective function that represents a solution to the optimization problem. It is important to describe the condition that the binary variables represent when they are 1 and when they are 0. Next we suggest describing the (real) coefficients of the variables and what they represent. Using the binary variables and their coefficients, we write an objective function that has the pattern of (1); i.e., the degree of each term is at most 2. Since $x^2 = x$ when x is binary, this reduction may be useful for obtaining a quadratic objective function.

For example, in the traveling salesman problem [13], the distance d_{ij} to proceed directly from city i to city j is known for all cities i and j. The objective is to find a route through all the cities that returns to the starting city and is minimum in length. Our binary variables are x_{ij} for the path directly from city i to city j. Our variable x_{ij} is 1 if the salesman proceeds directly from city i to city j; otherwise the value of x_{ij} is 0. The coefficients of the variables are the d_{ij}. Assuming the number of cities is n, our objective function is

$$\sum_{i,j=1}^{n} d_{ij} x_{ij} \tag{2}$$

such that $i \neq j$. We want the adiabatic quantum computer to assign 0, 1 to the variables x_{ij} so that the objective function (2) is a minimum over all n-cycles representing routes for the salesman.

The next step is to identify constraints that need to be satisfied to prevent minimum solutions that are not feasible. What are the boundaries for the optimization problem? What type of minimizations might occur that should be excluded? For example, in the traveling salesman problem, we need to ensure that the route for a solution enters each city once and departs from each city once. Also, we need to prevent subtours, which are cycles using fewer than all of the cities. In the case for 6 cities, this includes preventing a solution that has a 3-cycle through cities 1, 2, and 3 and another 3-cycle through cities 4, 5, and 6.



Usually the constraints are equations or inequalities with real coefficients and the previously identified variables. The degree of a constraint is restricted by (1) to be at most quadratic. For a 6-city traveling salesman problem, the constraint $x_{21} + x_{31} + \cdots + x_{61} = 1$ ensures that the salesman uses exactly one route into city 1. The inequality $x_{12} + x_{23} + x_{31} \leq 2$ will prevent a 3-cycle through cities 1, 2, and 3 in a solution for a 6-city problem.

The final step in the binary characterization is verification that the objective function and constraints represent the optimization function.

Forming the Hamiltonian

The Hamiltonian is a square, symmetric matrix with a row and column for each variable. The diagonal entries of the Hamiltonian are the values assigned to qubits. The off-diagonal entries in the Hamiltonian are the values assigned to the connections between qubits. The entries in the Hamiltonian are the coefficients of the terms from the sum of the objective function and the penalty functions (to be described in the next paragraphs).

Since equations and inequalities cannot be combined with a function, the usual technique is to reformulate the constraints as penalty functions. The constraints that are equations can be changed to penalty functions by reversing the algebraic sign of all terms on one side of the equation and deleting the equality sign. Most likely, the minimum for the result is not the same as the solution for the constraint equation. Thus we square the result, simplify it with the property $x^2 = x$ for binary variables, and delete the constant term.

In summary, we have shown a method to convert a constraint equation to a penalty function. The purpose is to add the objective function and the penalty functions to obtain an expression for an optimization problem that corresponds to (1). The coefficients in this sum are the entries in the Hamiltonian.

Exercise 1. Let x and z be binary variables. Find a penalty function for the constraint $z = \neg x$.[1]

Exercise 2. Let x, y, and z be binary variables. Find a penalty function for the constraint $z = x \vee y$.[2]

Next we describe a technique to convert a constraint inequality to a penalty function. We insert slack variables to change the inequality to an equation, as is done in the simplex algorithm for linear programming. We require the slack variables to be binary 0, 1. Then the above method can be used to change the equation to a penalty function. For example, consider the constraint $x + y + z \leq 2$ where x, y, and z are binary variables. We insert slack, binary variables s and t to obtain $x + y + z + s + t = 2$. Exercises 3 and 4 verify correctness.

Exercise 3. Let x, y, and z be binary variables such that $x + y + z \leq 2$. Show that there are binary values for s and t such that $x + y + z + s + t = 2$.

Exercise 4. Let x, y, z, s, and t be binary variables such that $x + y + z + s + t = 2$. Show that the variables x, y, and z satisfy $x + y + z \leq 2$.

If the weight of the objective function greatly exceeds the weight of the constraints, then the quantum solution is likely to be tilted toward a solution that satisfies the objective function but violates a constraint. On the other hand, if the weight favors the constraints, then the quantum solution is apt to find a suboptimal solution that satisfies the constraints. As in the simplex algorithm for linear programming, a lambda factor is used to balance the coefficients in the objective function and the coefficients in the penalty functions. See the discussion in [9, pp. 239–40] about the scalar P in Transformation 1. Lockheed Martin experiments on the quantum machine showed that the lambda factor has a narrow range that is dependent on the coefficients.

Complexity Considerations

After creating the objective function and constraints, the major steps to solve a minimization problem on an adiabatic quantum computer are:

- Form the Hamiltonian.
- Input the Hamiltonian and computing parameters to the machine.
- Wait while the Hamiltonian transitions to a final state on the machine.
- Read the final Hamiltonian on the machine.
- Interpret the final Hamiltonian to the variables.

[1] *A solution for Exercise 1:* $2xz - x - z$.

[2] *A solution for Exercise 2:* $xy + (x + y)(1 - 2z) + z$.

- Verify that the result is optimal.

A critical input parameter is the number of times to iterate a problem on the quantum machine. This is due to errors [18] and the analog nature of the machine when it treats input parameters [5].

The Hamiltonian is usually formed by a preprocessor. This is not considered part of quantum complexity [5]. Input and output steps are treated as instantaneous in [5]. A postprocessor may interpret the final Hamiltonian and prove the result is optimal. Thus, in both theory and practice, adiabatic quantum complexity is viewed as a function of the time to evolve the initial Hamiltonian to the final Hamiltonian. The (scaled) Schrödinger equation [5, equation (2)] contains a term τ representing the time for a quantum system to evolve from an initial state to a final state. The adiabatic theorem [2], [14] in quantum mechanics ensures that under certain conditions the evolution is close to the ground state of the final Hamiltonian, i.e., evolves to an optimal solution.

Cao and Elgart [5] cite the literature trail for theoretic bounds on τ for variations of the quantum database search algorithm of Grover. This work assumes conditions on a final Hamiltonian and asks what initial Hamiltonians and input parameters minimize τ and what the optimal value of τ is. Partial answers are cited. Cao and Elgart [5] extend the results.

Turning to the experimental side of quantum complexity, the current adiabatic quantum computer is too small to verify the theory. It has 512 qubits. If all qubits are operational, the largest complete graph that can be embedded in it has 33 vertices (qubits).[3] This limits the traveling salesman problem (TSP) to 6 cities on the current adiabatic quantum machine, since the TSP requires a complete graph and an n-city problem has $n(n-1)$ variables x_{ij} for the objective function (2).

In summary, Cao and Elgart [6] point out that quantitative characterization of the speedup of adiabatic quantum computing is largely unknown.

Quantum Hardware Effects on Computation

There is an accuracy difficulty, since for a single program loaded into the adiabatic quantum computer and iterated 100 times, about 1 percent of the time none of the 100 solutions are optimal. The theoretic work in [5] acknowledges the accuracy problem by indicating "the AQC algorithm is probabilistic in the sense that it gives a correct answer with the probability γ^2. The probability of failure can be decreased to the desired value (namely $O(1/N)$) by repeating the algorithm $(\ln N)/\gamma^2$ times. We set $\gamma = 1/5$ throughout this paper." Here AQC = adiabatic quantum computation, $N = 2^n$, and n is the dimension of the input space.

There is another hardware difficulty. The number of fully connected qubits limits the number of variables. This is significant, because the number of variables may grow rapidly as the size of the problem increases. Variables in the Ising model (1) equate to qubits that need to be connected, physically or logically, in order to represent their relationships. Current development methods are impractical for fabricating connectivity between qubits as the number of qubits increases [8]. This leads to the following open problems.

Find a decomposition algorithm that breaks a large optimization problem into small problems that an adiabatic quantum computer can solve. Find a reassembly algorithm that combines the solutions into a global optimal solution. Early work indicates that the algorithms may depend on properties of the Hamiltonian.

Summary and Research Areas

An adiabatic quantum machine provides a remarkable setting for computing some discrete optimization problems. The machine needs a quadratic objective function, binary variables, and quadratic penalty functions that are balanced with the objective function. Then quantum annealing can be expected to produce a near optimal solution extremely fast.

Theory and experimental verification are needed to refine the quantum annealing process so that an optimal solution is obtained in one iteration. A dual problem technique has been recommended so there is a simple, uniform method to verify that the result is optimal. Much work is needed to achieve a quantified method to express complexity for adiabatic quantum computing. Lastly, compilers are needed to simplify the pre- and postprocesses.

Acknowledgments

Daniel M. Davenport, Maxwell D. Henderson, and Joseph A. Angelozzi, who were my colleagues at Lockheed Martin, are thanked for discussions and ideas that improved the foundations for this report.

References

[1] D. Aharonov et al., Adiabatic quantum computation is equivalent to standard quantum computation, *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science* (2004), pp. 42–51.

[2] J. E. Avron and A. Elgart, The adiabatic theorem of quantum mechanics (1998), http://www.ma.utexas.edu/mp_arc/c/98/98-370.ps.gz

[3] S. Boixo, T. F. Rønnow et al., Quantum annealing with more than one hundred qubits (2013), arXiv:1304.4595v1.

[3] *Source is W. G. Macready,* D-Wave Systems.

[4] S. Boixo, T. Albash et al., Experimental signature of programmable quantum annealing (2013), *Nat. Commun.* 4.2067, DOI: 10.1038/ncomms3067.

[5] Z. Cao and A. Elgart, On efficiency of Hamiltonian-based quantum computation for low-ranked matrices, *J. Math. Phys.* **53** (2012), 032201.

[6] ———, *Adiabatic Quantum Computation: Enthusiast and Skeptic's Perspectives*, Virginia Tech. University, Blacksburg, VA, 2010, arXiv:1004.5409.

[7] N. Jones, Google and NASA snap up quantum computer, *Nature* (2013), DOI: 10.1038 nature.2013.12999, `http://www.nature.com/news/google-and-nasa-snap-up-quantum-computer-1.12999`.

[8] K. Karimi et al., Investigating the performance of an adiabatic quantum optimization processor, *Quantum Inf. Process.* **11** (2012), 77–88, DOI:10.1007/s11128-011-0235-0.

[9] G. A. Kochenberger et al., A unified modeling and solution framework for combinatorial optimization problems, *OR Spectrum* **26** (2004), 237–250.

[10] C. C. McGeoch and C. Wang, Experimental evaluation of an adiabatic quantum system for combinatorial optimization, *Proceedings of the ACM International Conference on Computing Frontiers* (2013), DOI:10.1145/2482767.2482797.

[11] K. L. Pudenz and D. A. Lidar, Quantum adiabatic machine learning, *Quantum Inf. Process.* **12** (2013), 2027–2070.

[12] A. T. Rezakhani et al., Accuracy versus run time in an adiabatic quantum search, *Phys. Rev. A* **82** (2010), 052305.

[13] R. H. Warren, Adapting the traveling salesman problem to an adiabatic quantum computer, *Quantum Inf. Process.* **12** (2013), 1781–1785, DOI:10.1007/s11128-012-0490-8.

[14] Wikipedia, Adiabatic theorem, available at `http://en.wikipedia.org/wiki/Adiabatic_theorem`.

[15] Operational Quantum Computing Center established at USC Viterbi School of Engineering, news release, October 29, 2011, `http://viterbi.usc.edu/news/news/2011/operational-quantum-computing334119.htm`.

[16] J. Smolin and G. Smith, *Classical signature of quantum annealing*, IBM T. J. Watson Research Center, Yorktown Heights, NY, 2013, arXiv:1305.4904v1.

[17] L. Wang et al., *Comment on: "Classical signature of quantum annealing"*, 2013, arXiv:1305.5837v1.

[18] K. L. Pudenz et al., *Error corrected quantum annealing with hundreds of qubits*, 2013, arXiv:1307.8190v1.