SANDIA REPORT

SAND2013-10048
Unlimited Release
Printed November 2013

EVALUATING NEAR-TERM ADIABATIC QUANTUM COMPUTING

Ojas Parekh, John Aidun, Irene Dubicka, Andrew Landahl, Luke Schulenburger, Chris Tigges, and Jeremy Wendt

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-10048
Unlimited Release
November 2013

EVALUATING NEAR-TERM ADIABATIC QUANTUM COMPUTING

Ojas Parekh, John Aidun, Irene Dubicka, Andrew Landahl, Luke Schulenburger, Chris Tigges, and Jeremy Wendt

Discrete Math & Complex Systems, Advanced Device Technologies, Mission R&D, HEDP Theory, Photonic Microsystems Technology, Embedded Systems Engineering
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1322

ABSTRACT

This report summarizes the first year's effort on the Enceladus project, under which Sandia was asked to evaluate the potential advantages of adiabatic quantum computing for analyzing large data sets in the near future, 5-to-10 years from now. We were not specifically evaluating the machine being sold by D-Wave Systems, Inc; we were asked to anticipate what future adiabatic quantum computers might be able to achieve.

While realizing that the greatest potential anticipated from quantum computation is still far into the future, a special purpose quantum computing capability, Adiabatic Quantum Optimization (AQO), is under active development and is maturing relatively rapidly; indeed, D-Wave Systems Inc. already offers an AQO device based on superconducting flux qubits. The AQO architecture solves a particular class of problem, namely unconstrained quadratic Boolean optimization. Problems in this class include many interesting and important instances. Because of this, further investigation is warranted into the range of applicability of this class of problem for addressing challenges of analyzing big data sets and the effectiveness of AQO devices to perform specific analyses on big data. Further, it is of interest to also consider the potential effectiveness of anticipated special purpose adiabatic quantum computers (AQCs), in general, for accelerating the analysis of big data sets. The objective of the present investigation is an evaluation of the potential of AQC to benefit analysis of big data problems in the next five to ten years, with our main focus being on AQO because of its relative maturity. We are not specifically assessing the efficacy of the D-Wave computing systems, though we do hope to perform some experimental calculations on that device in the sequel to this project, at least to provide some data to compare with our theoretical estimates.

Our study hinges on the problems we seek to solve as much as it does on AQC. A significant part of our endeavor lies in understanding the types of problems amenable to efficient AQ algorithms. This is constrained by the fact that not many AQ algorithms exist, and an algorithmic breakthrough is unlikely within the project performance period. Additionally, AQC is currently in its infancy and may not reach a large scale for decades. There is currently a single commercial, special-purpose AQO device available, and the broader research community is, as yet, undecided whether this machine is in fact truly a quantum device.

Given these circumstances, we adopted the following guiding principles for this study:

- Devise ways of leveraging AQC assuming only modest scalability
- Evaluate extensions of existing AQ algorithms rather than aiming for a breakthrough
- Let the novelty and strengths of AQC guide our selection of potential applications
- Abstract existing AQ algorithms into general and reusable capabilities
- Do not lose sight of the practitioner's point of view

Our resulting approach to assessing the potential value of AQC on analyzing big data sets is multi-pronged. First, we selected a small set of powerful data analysis AQ algorithms that can serve as a toolkit for data analytics. Our toolkit, in some sense, pinpoints the current strengths of AQC. But the assessment of potential impact of AQC on big data problems can only be made in comparison with projected capabilities of classical computing for data analytics in the time frame under consideration. For this reason, we also identified the classical algorithmic tools that are currently providing the greatest impact in solving our types of large-scale problems. Thus, analogously to our AQC toolkit, we will also catalog classical state-of-the-art algorithmic workhorses routinely used to solve big data problems of interest to us. Next, we aim to bridge the divide by using our AQ tools to build replacements for some of the classical components. Thus our plan allows us to avoid the bottleneck of focusing on a single problem while still providing a potential path to AQ algorithms for problems of interest.

Because many real-world phenomena contained in big data sets can be easily and usefully represented as node-edge graphs and because identifying communities within the resulting very large graphs is a crucial analysis step, we selected *community detection* algorithms for our case study. Community detection naturally maps to the Max Cut problem, which is NP-hard but amenable to AQO implementation. Because community detection is a critical problem, many sub-optimal algorithms exist for finding acceptable-quality partitions of the graph into community. Several classical-computer algorithms have been created. Moreover, optimized implementations of many of these algorithms are freely available. Many of the core problems can be readily cast as Max Cut and QUBO instances for comparison with AQC.

We seek to estimate the performance of each chosen algorithm for problem instances of increasing size. We address two immediate questions: (i) what is an appropriate measure of performance; (ii) how do we select a family of instances in order to derive a meaningful and valid comparison. As is discussed in the report, we chose to focus on relative performance, as represented by growth curves indicating scalability over problem instances of increasing size, rather than absolute performance. Our goal in selecting problem instances is to capture some inherent notion of complexity that is independent of instance size. Random graphs are usually not representative of the real world and, often, are not even hard to analyze. On the other hand,

real data sets are not readily available in a range of instance sizes that all have similar characteristics in common, *i.e.*, being a family of instances. The report describes how we are attempting to cope with these challenges of generating graph instances, which remains a research endeavor.

Additionally near-term trends in classical computing capabilities were assessed through interviews with subject matter experts (SMEs) at Sandia National Laboratories. We focused on Sandia SMEs for convenience in the limited time frame of the project without having to be overly concerned about missing an important trend. But we also focused on Sandia SMEs who have current internally funded research projects (under Sandia's LDRD program) because this is information that would not otherwise be readily available to the present audience. We have produced a separate report containing a compilation of the SME interviews.

Results of our work to date are all concerned with development of the methodology, summarized above, for performing the assessment of interest. The AQC toolkit includes a machine learning algorithm, Quantum Boosting, which is the analog of the classical Adaptive Boosting algorithm. Combinatorial optimization is represented in the toolkit by Quantum Unconstrained Binary Optimization (QUBO). QUBO is a quadratic unconstrained Boolean optimization problem solvable by an AQO. Max Cut is easily expressed as an instance of QUBO. The last entry in our toolkit is the Quantum Page Rank algorithm. This algorithm can be extended to a general algorithm for sampling from an optimal eigenvector of a Hermitian matrix and has application in graph analysis. The classical community detection algorithms include Clauset-Newmann-Moore (CNM), Walktrap, Louvain, and weighted CNM (wCNM).

We investigated whether Multiplicative Attribute Graph model (MAG; used by MagFit and MagGen) offers a capability for generating graphs that faithfully represent real-world graphs. MagFit searches for attributes that fit an existing input graph. These attributes can then be used to generate similar graphs to the input graph. However, after experimenting with this method, we realized that the expected average degree (number of edges divided by n) scales with n. That is, the density of edges with respect to nodes grows more rapidly than we had expected. This has initiated a more detailed study of average degree in families of real-world graphs. We may be able to use alternative approaches to MagGen, such as BTER and random-walk-based sampling, to generate the families of graphs needed to carry out our benchmarking procedure; however, in the near term we will adopt a temporary expedient so that we can quickly have a family of graphs ready should we get an early opportunity to run experiments on a D-Wave AQO machine. We will generate a graph from a dataset that grows with time, such as the Border Gateway Protocol (BGP), which is available internet operations data. A family of graphs with similar character will be generated by using successively longer time intervals of this dataset. This is not a fully satisfactory method for generating a family of graphs, but it allows us to start benchmarking experiments.

In order to better understand the function of adiabatic quantum computing we are developing a two-pronged simulation capability. The first is a purely classical simulation of the time-dependent behavior of the annealing of a corresponding classical spin system. The second simulation capability we are developing is a quantum Monte Carlo algorithm to understand the Hamiltonians at any given point along the annealing.

We reviewed and critiqued recent evaluations and investigations reported in the research literature, in particular the work of Boixo and coworkers, and that of McGeoch and Wang. Both of these investigations report significant results for the D-Wave AQO device and so were of central interest to assessing the near-term potential for AQO on data analytics. Reviewing these articles helped us design a more effective methodology for assessing the potential of AQCs. These reviews and critiques are included in this report.

CONTENTS

FIGURES

1. INTRODUCTION

1.1 The inevitability—and advantages—of quantum computing

As computing technology is driven to operate ever more quickly, it is necessarily also driven to operate with ever smaller devices. This follows from special relativity—computer components must be brought closer together to exchange information at a faster pace. Eventually this drives computer components to a scale at which the "classical physics" of Newton and Maxwell can no longer be counted upon to give a faithful description of a computer's behavior. Instead, *quantum* physics is required, which has fundamental ramifications for both the limits and capabilities of computing.

For many years, quantum physics was viewed as an obstacle to computing at exceedingly small scales. After all, quantum physics asserts that one cannot completely know conjugate dynamical variables—such as the position and momentum of a computer component—at the same time. It also predicts that bits of information in a "quantum computer" are suddenly allowed to be in "superpositions" of the values 0 and 1 at the same time. Even worse, the outcome of measuring such "quantum bits" or "qubits" might lead to uncertain outcomes. Harnessing a device with such weird properties to serve as a functional computer seemed hopeless.

Over the course of the mid-1980s to the mid-1990s, this pessimistic view softened with a series of discoveries that gave a clearer picture of what quantum physics does and does not allow for computing, with the view essentially vanishing with the discovery of a fast quantum algorithm for factoring integers into primes made by Peter Shor in 1994 [Shor, 1994a]. It turns out that quantum physics isn't a barrier to computing—indeed it is quite the opposite—it is an *enabler* that can accelerate some computations far beyond what common sense suggests. By using a new set of rules (quantum physics), it appears one can solve some computational problems using fundamentally fewer steps than if one used the old set of rules (classical physics). This is qualitatively very different from the more familiar speedups that miniaturization usually affords, such as granting faster clock speeds. It is more like the speedup one attains when one transitions from using Roman numerals to using Arabic numerals when doing complex mathematics. To be clear, there is no mathematical proof that quantum computers are more powerful than classical computers. However, as researchers continue to develop quantum algorithms that are faster than their best-known "classical algorithm" counterparts (with over 50 known quantum speedups to date [Jordan, 2013a]), the evidence for the power of quantum computing has mounted to a level that cannot be ignored.

1.2 The quantum circuit architecture

To be useful, a quantum computer requires a well-defined *architecture* that specifies methods for input, output, and processing. The most widely studied quantum computing architecture is the quantum circuit architecture. The way a quantum circuit works is as follows. First one prepares a set of qubits in some known state. Then one applies a sequence of "elementary" unitary transformations, each of which is called a "quantum gate," to the state, where each gate acts on some small number of qubits. (Typically only one- and two-qubit gates are considered.) Finally one performs a sequence of measurements, each of which also acts on some small number of

qubits. (Typically only one-qubit measurements are considered.) The size and structure of a quantum circuit depends on the specific quantum algorithm of interest. Variations on the quantum circuit architecture revolve around how exactly the input, gates, and output are adaptively chosen as a function of the algorithm instance.

In the "standard" quantum circuit architecture, the state preparation sequence is always the same—each qubit begins prepared in the state $|0\rangle$ (where $|0\rangle$ and $|1\rangle$ denote the preferred *computational basis states* of the qubit—"preferred" because we assume the native quantum technology can readily prepare and measure qubits in this basis). The measurement sequence is always the same too—each qubit is measured in the computational basis. The one thing that changes from program to program is the gate sequence.

Two alternative quantum circuit architectures are worth mentioning. The first is the architecture variously described as the "cluster-state," "measurement-based," or "one-way" architecture, in which the (entangled) state preparations and (nonexistent) gate sequence are problem-independent but the measurement sequence depends (adaptively) on the problem. The second is the "programmable" quantum circuit architecture, in which the input is prepared to hold both the data and the program in a problem-dependent way, but the gate and measurement sequences are problem-independent. These variants are relevant for certain types of adiabatic quantum computing, described later.

An important feature of the quantum circuit architecture is that it is *universal*. This means that given any dynamics described by quantum mechanics, a quantum circuit can create an efficient "digital" approximation of the dynamics to any desired precision by using enough qubits, gates, and measurements. In other words, the quantum circuit architecture is as powerful as quantum mechanics will allow.

1.3 The adiabatic quantum architecture

While, in theory, adiabatic quantum computers (AQC) could be made to be universal quantum machines, this will require technological breakthroughs that enable a richer set of interactions among the qubits than just Ising two-body (anti-)parallel spin interactions. Currently, a special purpose adiabatic architecture, adiabatic quantum optimization (AQO), is under very active development, with the D-Wave company actually selling a device based on superconducting flux qubits. As AQO is the technology most mature and likely the most – or, only – available version of AQC for the next five to ten years and it shows significant potential to speed up some very interesting big data problems, we begin this section summarizing it. Indeed, as the objective of our investigation is an evaluation of the potential of AQC to benefit analysis of big data problems in the next five to ten years, our main focus has been on AQO because of its relative maturity. This section ends with a brief discussion of the challenges of developing and greater potential capabilities of a universal AQC.

1.3.1 The adiabatic quantum optimization architecture

Proposed in 2000 by Farhi *et al.* [Farhi, 2000a] as a quantum *algorithm* for "solving" the (NP-hard) satisfiability problem, the AQO algorithm lends itself to a novel full-fledged *architecture* for "solving" a large class of combinatorial optimization problems. The word "solving" is in quotes here because, while the AQO algorithm does indeed give a solution to these problems when run for a long enough time, it is an open research question as to just how long the algorithm needs to run as a function of the problem's instance size to give a correct solution with high probability. For many mathematicians, the perspective is that, if the required time scales exponentially with the instance size, the algorithm should not be said to be "solving" the problem. Classical computers can already "solve" the same class of problems in exponential time!

It is worth noting that it has become fashionable these days to refer to AQO as "quantum annealing." This is driven in part because of concerns by the research community that the D-Wave device, for example, is not really utilizing "adiabatic quantum physics" to solve combinatorial optimization problems. However, such hardware may be using some kind of "quantum effects" to do so in a manner reminiscent of a classical algorithm known as "simulated annealing."

Unlike the quantum circuit architecture, the AQO architecture is not a universal architecture. It is only able to solve unconstrained Boolean optimization problems, namely those of the form

$$\min f(x) \text{ subject to } x \in \{0,1\}^n.$$

Even though this does not represent universal quantum computation, solving this class of problems is highly useful; a great many optimization problems can be cast in this form, including problems in which billions of dollars have been invested. For example, AQO algorithms have been proposed for machine learning [Neven, 2008a; Pudenz, 2011a], PageRank [Garnerone, 2012a], satisfiability [Farhi, 2001a] and graph problems [Gaitan, 2012a; Gaitan, 2013a]. Quite generally, unconstrained Boolean optimization problems are known to be "NP-hard," which informally means that they are at least as hard as the hardest problems in the complexity class NP. Problems in NP are widely believed to unable to be solved in polynomial time (even with quantum computers!), and there is currently a $1M prize for anyone who can prove one way or the other whether problems in NP are tractable or not (meaning they can be solved in a time that grows polynomially with the input size). That said, even if these problems cannot be solved in polynomial time by an AQO machine, it may be the case that for finite instance sizes an AQO machine may be able to solve such problems more quickly than any existing or near-term classical machine is able to.

The AQO architecture implements algorithms via a three-step process. First, one prepares n qubits in the state $|+\rangle$ (*i.e.*, the state $|0\rangle + |1\rangle$). This is the lowest-energy configuration, or "ground state," of the Hamiltonian $H_I = \sum_i \sigma_x^{(i)}$. This initial Hamiltonian H_I is turned on and then interpolated to a final "problem" Hamiltonian $H_I = \sum_x f(x)|x\rangle\langle x|$, where $|x\rangle$ is a shorthand for the n-qubit computational basis state whose binary expansion is x. The ground state of the final Hamiltonian is a solution to the associated unconstrained Boolean optimization problem for

f. If the interpolation is performed slowly enough, then the state will have transformed "adiabatically" to the ground state of the final Hamiltonian. Finally, the state of each qubit is measured in the computational basis, returning the output of the algorithm. Like the quantum circuit architecture, the preparation and readout steps are problem-independent. Only the dynamics in the middle (described by the interpolating Hamiltonian) depends on the problem.

A natural question is, "How slowly does the interpolation have to be run for the output of the AQO algorithm to be correct with high probability?" The *adiabatic theorem* (or, more correctly, the *adiabatic approximation*) in quantum mechanics guarantees that as long as the total time for the interpolation, T, is chosen to be large enough, the final measurement will yield a value for x that minimizes f with high probability. The "rule of thumb" that physicists used for years for how long "long enough" needs to be [Messiah, 1961a] is that

$$T \gg \frac{\left|\langle E_1|\dot{H}|E_0\rangle\right|_{max}}{g_{min}^2}, \tag{1}$$

where $|E_0\rangle$ and $|E_1\rangle$ denote the instantaneous ground and first-excited states of the system, whose corresponding energies are E_0 and E_1 respectively, g denotes the difference in energy $|E_1 - E_0|$, also called the "(energy) gap," $\dot{H}$ denotes the derivative $\frac{d}{ds}H$, where $s := t/T$ is the interpolation parameter, and the maximization and minimization are taken over all $s \in [0,1]$. In this and subsequent formulas, the units can be corrected by inserting (powers of) Planck's constant $\hbar$ where necessary; in other words, here and henceforth we will work with units in which $\hbar = 1$.

For most (and perhaps all) AQO problems, it is fairly straightforward to choose an interpolation path without too many wiggles so that the numerator on the right hand side of Eq. (1) does not become too large. The challenge then becomes finding a path such that the minimum gap in the denominator on the right hand side of Eq. (1) does not become too small. For example, if the minimum gap became exponentially small as n increased, then the AQO problem would need to be run for an exponentially long time to ensure that it yields the answer with high probability.

In order to get a better handle on just how long the AQO problem must be run to succeed with high probability, tighter bounds than Eq. (1) have been developed in recent years. Indeed, it is the prospect of adiabatic quantum computing that has motivated mathematical physicists and computer scientists to sharpen this rather loose bound. One of the tightest bounds to date, although somewhat baroque, is the following [Jansen, 2007a]:

$$\||\psi(T)\rangle - |E_0(1)\rangle\| \leq \frac{1}{T}\left[\frac{\|\dot{H}\|}{g^2}(0) + \frac{\|\dot{H}\|}{g^2}(1) + \int_0^1 ds\left(\frac{7\|\dot{H}\|^2}{g^3} + \frac{\|\ddot{H}\|}{g^2}\right)\right]. \tag{2}$$

It is a common misperception to believe that the AQO architecture represents "analog" computing. Those who do not consider the architecture carefully may mistakenly dismiss it as riddled with all of the problems that analog classical computing has, most notably as having a

complete lack of robustness. In fact, quite the opposite is true—the AQO architecture is predicted to have exceptional implementation robustness. The AQO architecture is definitely digital—the input and output are completely digital. Just because time is modeled as a continuous variable during the interpolation, it does not mean that the state space of the computer is described by a continuous variable. In fact, even seemingly discrete-time architectures, such as the quantum circuit architecture, rely on continuous-time evolution when implemented in nature. Our best understanding of time in the universe is that it is a continuous, not discrete, variable, and all experiments to date have confirmed this. If time is, in fact, discrete in some manner, it must be so at a scale inaccessible to current experimentation.

As presented, the AQO architecture may require a highly nonlocal Hamiltonian. In many, perhaps all, physical systems, interactions are fundamentally local, meaning that they act on just two objects at a time. For example, if the qubits were represented by spin-1/2 particles, then a physically meaningful Hamiltonian would necessarily be restricted to acting on, at most, two spins at a time. With this restriction in mind, it is natural to consider the AQO architecture restricted to *quadratic unconstrained binary optimization* (QUBO) problems, defined as

$$\min f(x) = \sum a_i x_i + \sum b_{ij} x_i x_j \text{ subject to } x \in \{0,1\}^n.$$

By a linear transformation of variables from 0 and 1 to −1 and 1, an equivalent representation of the QUBO problem is the following:

$$\min f(x) = \sum h_i s_i + \sum J_{ij} s_i s_j \text{ subject to } s \in \{-1,1\}^n.$$

The final Hamiltonian in an AQO architecture for solving this problem is the Ising Hamiltonian,

$$H_{Ising} = \sum h_i \sigma_i^z + \sum J_{ij} \sigma_i^z \sigma_j^z,$$

which arises in many quantum information processing systems including semiconductors, superconductors, and laser-trapped atoms & ions.

In summary, adiabatic quantum optimization as realized by plausible near-term quantum technology will likely be narrowed to target solving QUBO problems. While seemingly narrow, such problems are still NP-hard and have billions of dollars invested in attempts to tackle them with classical devices. Whether such technology will be able to outperform existing or near-term classical methods for solving these problems is an open question; we address certain aspects of this question in this report, focusing on application to "big data" problems.

1.3.2 The universal adiabatic quantum computing architecture

While most of this report is focused on exploring how adiabatic quantum computers might fare when attempting to solve QUBO problems of practical interest, it is widely believed that, at sufficiently large scales, even quantum computers, no matter what architecture they use, will be unable to solve QUBO problems efficiently. What could be a real game changer is a *universal*

adiabatic quantum computer—a device that efficiently implements all quantum algorithms that have been or will be discovered. In particular, such a machine could factor numbers, simulate quantum systems, and discriminate knots in a time that is exponentially faster than the best-known classical algorithms we have. Unlike the scenario for AQO, there are strong theoretical proofs that universal adiabatic quantum computers can tackle these problems efficiently. *In terms of "tech surprise," a universal adiabatic quantum computer is a device to watch for.*

The basic idea behind universal adiabatic quantum computing architectures proposed to date is to have them directly simulate a quantum circuit by taking a description of that circuit as input. Adapting an idea of Feynman's [Feynman, 1986a], Kitaev [Kitaev, 2002a], and then others, [Aharanov, 2004a; Mizel, 2007a] proposed constructing a final Hamiltonian of an adiabatic evolution something like the following to simulate a quantum circuit defined by the gate sequence $U_T \ldots U_1$.

$$H = \sum_{t=1}^{T} U_t \sigma_t^+ \sigma_{t-1}^- + U_t^\dagger \sigma_t^- \sigma_{t-1}^+.$$

Numerous improvements to this Hamiltonian have occurred over the years, and it is now known that it suffices to have a final Hamiltonian of the following form, with coefficients that can be both positive and negative, to effect universal adiabatic quantum computation [Biamonte, 2007a].

$$H = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z + \sum_{ij} K_{ij} \sigma_i^x \sigma_j^x,$$

No technology has demonstrated a sufficiently tunable Hamiltonian of this form, but proposals exist for several technologies, including superconductors and semiconductors.

2. APPROACH

2.1 Near-term AQC

Our overarching project goal is to assess the "near-term" impact of AQC on solving problems across several domains at large scale. By near-term we mean five to ten years; however, our interest is not necessarily limited to this window, especially if there is a potential for high impact in a longer term. We seek to pinpoint strengths and weaknesses of AQC. This is especially important because if AQC is not viable in the near term as an algorithmic tool, we will want to understand the nature of the barriers preventing it from being advantageous.

Our study hinges on the problems we seek to solve as much as it does on AQC. It may very well be the case that there are a handful of interesting problems on which existing AQ algorithms shine. If this is the case, how might we discover such problems? Thus a significant part of our endeavor lies in understanding the types of problems amenable to efficient AQ algorithms. This becomes even more critical when we consider that an AQ algorithmic breakthrough is unlikely within the project performance period. Mathematical breakthroughs are extremely difficult to predict: proving Fermat's Last Theorem had remained open for 350 years before being solved in the 1990's. Who would have predicted in 1970 that this problem would be solved in his lifetime?

Scalability is another significant hurdle. Even if we conjecture that AQC scales on par with Moore's Law, AQC is currently in its infancy and may not reach a large enough scale to topple classical computation for decades. Other challenges also stem from the infancy of AQC. There is currently a single commercial special-purpose AQC available, and it is unclear whether this machine is in fact a truly quantum device. It is certainly manufactured with this intent; however, it is difficult to prove whether this machine is leveraging quantum computation. Another factor is that there are currently several viable technologies for implementing qubits, and it seems too early for any type of standardization. Classical computation has matured so that algorithms are expressed in machine-independent high-level languages such as C. However, this is far from the case with AQC. The physical layout of and the types of qubit interactions offered by an AQC device define its capabilities. In order to solve a problem using an AQC, one must of course devise a high-level AQ algorithm, but in addition one must "compile" this algorithm to match the hardware configuration of the AQC on which it will be executed. The resource requirements of the compiled version may dramatically outweigh those of the high-level algorithm.

Motivated by the considerations outlined above, we adopt the following guiding principles for this study:

- Devise ways of leveraging AQC assuming only modest scalability
- Evaluate extensions of existing AQ algorithms rather than aiming for a breakthrough
- Let the novelty and strengths of AQC guide our selection of potential applications
- Abstract existing AQ algorithms into general and reusable capabilities
- Do not lose sight of the practitioner's point of view

2.2 Building an AQC toolkit

Our ultimate goal is of course to understand and address the reach of AQC in solving large-scale problems. However, we feel that a top-down approach of pinning specific problems may be counterproductive in the short to medium term. Breakthrough quantum algorithms are rare and difficult to design; further constraining ourselves to a specific problem would likely amplify this difficulty. Instead we propose adopting a more bottom-up approach where we catalog a general collection of algorithmic AQ capabilities abstracted from existing AQ algorithms. The idea is to examine the novel and powerful components of existing AQ algorithms, and from these construct a toolbox of more generic capabilities currently afforded by AQC. This is akin to the way in which the Quantum Fourier Transform is a tool that is cleverly leveraged by a variety of (circuit-based) quantum algorithms.

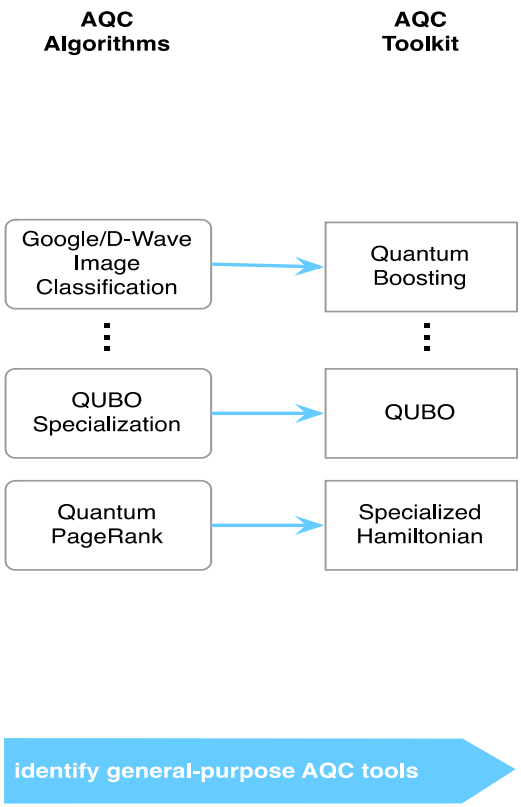

Figure 1: Deriving general-purpose AQC tools from existing AQC algorithms

Our toolkit, in some sense, pinpoints the current strengths of AQC; yet, this is only half our goal. We must also estimate whether our tools, or perhaps improved versions of them, might be used to solve large-scale problems in our domain within the next five to ten years. To this end we also seek to understand which classical algorithmic tools are currently providing the greatest impact in solving our types of large-scale problems. Thus, analogously to our AQC toolkit, we will also catalog classical state-of-the-art algorithmic workhorses routinely used to solve big data problems of interest to us. For example, dimensionality reduction techniques are a common critically used classical component for clustering, classification, and machine learning algorithms.

2.3 Bridging the gap to large-scale problems

With these two collections in hand, one outlining the algorithmic strengths of AQC and the other outlining the algorithmic needs of large-scale applications, our goal will be to bridge the divide by trying to use our AQ tools to build replacements for some of the classical components. Thus our plan allows us to avoid the bottleneck of focusing on a single problem while still providing a potential path to AQ algorithms for problems of interest.

Although the above approach may produce new and effective AQ algorithms, a natural concern is the scalability of these algorithms. Our current thoughts are that a near-term AQC breakthrough in big data will not directly leverage quantum effects on a large scale. We feel that AQC is better positioned to solve relatively small but very hard "kernel" problem instances. A hybrid algorithm, which uses classical techniques to decompose large-scale instances and parcels off smaller kernel instances to an AQC, is more likely to have near-term impact. The recent AQC machine learning algorithm developed by Google and D-Wave for image classification may be viewed as a hybrid algorithm [Neven, 2008a].

2.4 Hybrid algorithms

A natural and tempting response to the realization of quantum computation is simply framing it as a drop-in replacement for classical computation. The danger in this perspective is that such drop-in quantum replacements may take a long time to mature, which discounts the potential impact of near-term quantum computation. We have had considerable time to understand classical systems, as well as their limitations, both from a theoretical and practical perspective. Classical computers are deeply integrated into our society and economy. We are adept at engineering incredibly small and power-efficient consumer devices as well as enormous massively parallel and distributed systems. Thus there is presently a mismatch in maturity and scale between classical and quantum computation, both from algorithmic and engineering perspectives. Any near-term approach to harnessing quantum computation must address this mismatch.

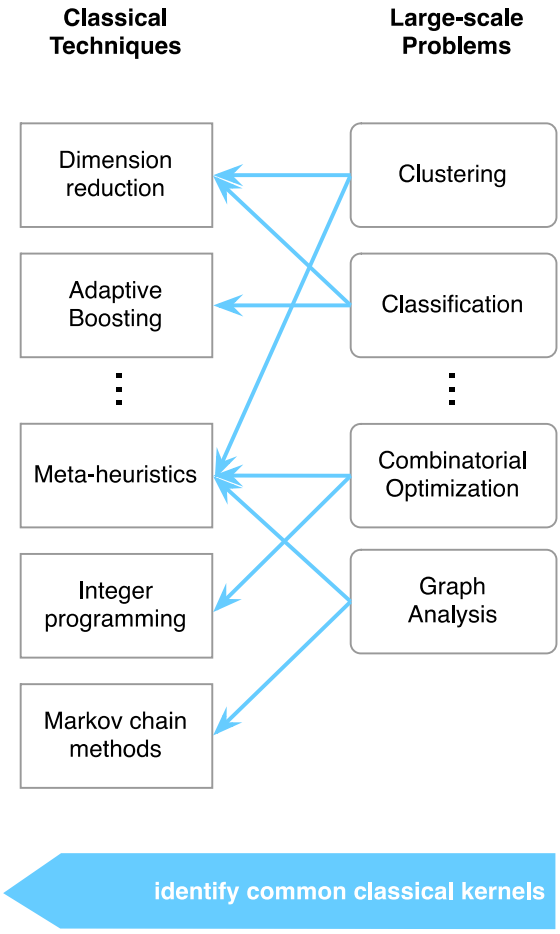

Figure 2: Identifying commonly used classical kernels for large-scale problems

Large-scale classical systems are currently heterogeneous in the sense that they are typically composed of a massive network of interconnected classical processors or cores. Our perspective is that the potential for impact lies in viewing a quantum system as a specialized core or kernel to be used in conjunction with a classical system, rather than as a replacement for an entire classical system. This type of viewpoint is not uncommon in the design of classical systems; for example consider the floating accelerators that accompanied Intel CPUs in the late 80s and early 90s. A more modern example are the highly parallel but specialized GPUs available to augment graphics-related (and increasingly, more general) kernels. Thus we may view a moderate-sized quantum system as a specialized accelerator available to an otherwise classical large-scale system.

There are distinct advantages to such an approach. For one, such a hybrid system allows a quantum system to naturally delegate mundane classical pre- and post-processing tasks. Qubits are a potentially powerful but severely limited resource. Why waste this resource on tasks on which classical systems already excel? Another advantage is that hybrid systems allow us to reuse the expertise and algorithmic techniques developed in attacking big-data problems. These problems push the envelope of classical computation, and, as a result, there has been much fruitful research into classical decomposition techniques for big-data problems; recall that no matter how big our data, the actual work is performed on individual cores with modest amounts of memory and power. Hybrid systems enable us to apply these very same decomposition techniques to produce bite-sized chunks of data suitable for digestion by a quantum system. The key is to ensure that we are realizing the potential of the quantum components of the system. We must ensure that the decomposed problems posed to the quantum system are those on which it truly shines.

We also note that hybrid systems come equipped with a natural expansion strategy. If quantum computation proves itself practical, then as the scale and quantity of quantum systems grow, one may tune a hybrid system to leverage more quantum accelerators and fewer classical processors. Other strategies for the utilization of emerging large-scale quantum systems may eventually present themselves, but hybrid systems are a compelling near-term option.

3. BENCHMARKING METHODOLOGY

3.1 Benchmarking challenges

There are several layers of challenges in comparing the performance of two algorithms. For the purposes of benchmarking, an algorithm, be it quantum or classical, is simply a black box. When given an instance of a problem and a maximum execution time, the black box either terminates and produces a feasible (but not necessarily optimal) solution or fails to do so in the allotted execution time. Our general goal is to estimate the performance of each algorithm being compared as we consider problem instances of increasing size. Two questions are immediate: (i) what is an appropriate measure of performance; (ii) how do we select a family of instances in order to derive a meaningful and valid comparison.

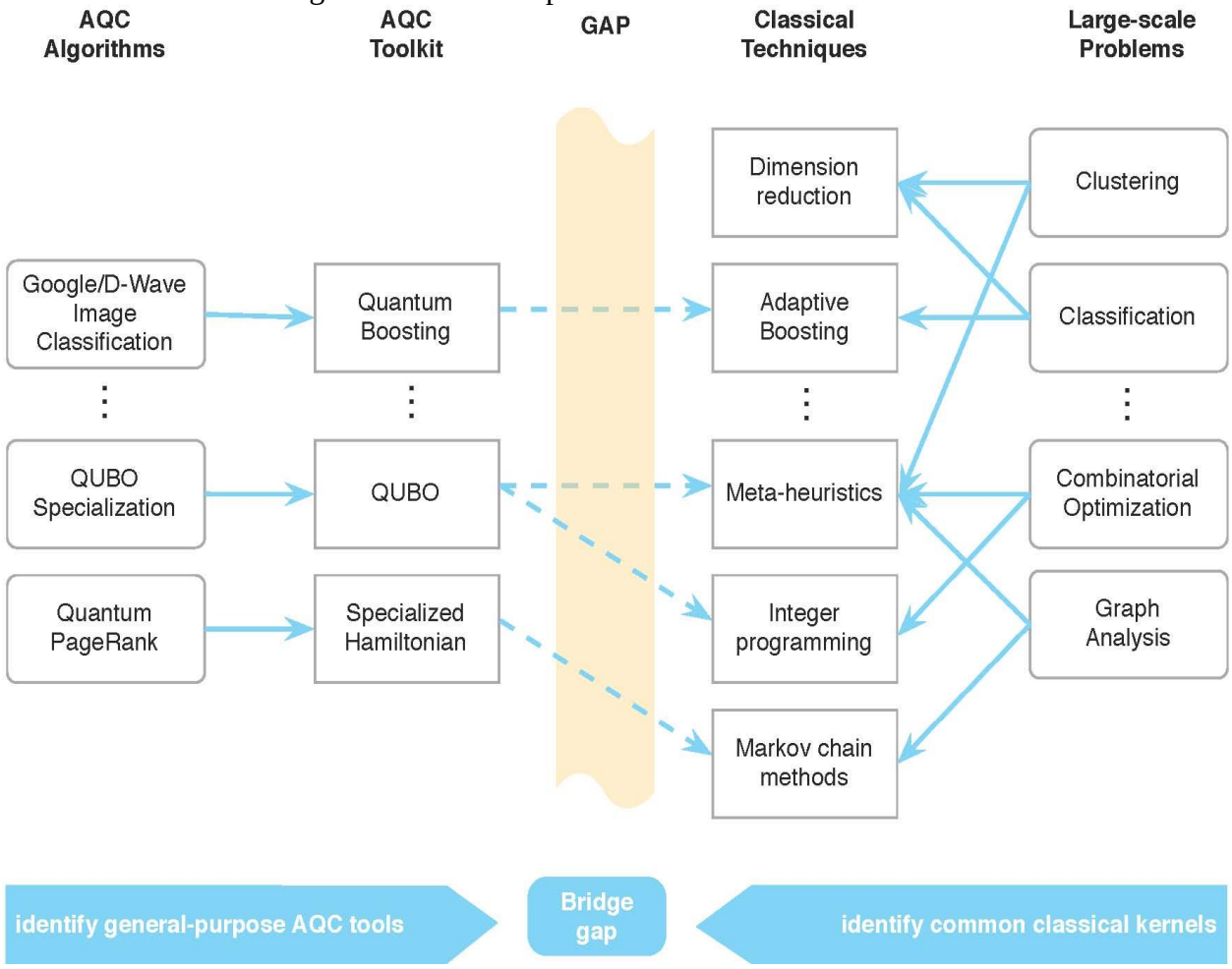

Figure 3: Plan for devising AQC algorithms by bridging the gap between a collection of powerful and reusable AQC tools and classical techniques

The notion of the performance of an algorithm depends on some implicit resource being measured. The usual notion of performance is with respect to overall running time; however, care must be taken to define "overall running time" properly. Do we care about CPU time expended or wall-clock time, where the former is a fairer comparison metric but the latter may

better reflect practical scenarios? There are cases when running time may not be an appropriate metric at all. Consider comparing a quantum algorithm versus a classical one. The notion of an algorithm is distinct from that of a code or software that may be readily executed. The former is a more abstract notion of a recipe for solving a problem, whereas the latter is an implementation of such a recipe on a specific platform. In the classical world we have many examples of both algorithms and software implementations, whereas in the quantum world, we essentially have only algorithms. One manifestation of this issue is that overall running time may be a red herring in comparing a highly optimized classical code against a seminal implementation of a quantum algorithm on an immature quantum device. In the short run, a well-engineered implementation of a poor algorithm may outperform a poor implementation of an algorithm based on a powerful and clever idea; however, as our data sets grow larger we expect the latter to dominate the former. A benchmark that considers this also allows for the possibility of better-engineered versions of algorithms in the future. *Thus we focus on relative performance, as growth curves indicating scalability over instances of increasing size, rather than absolute performance.*

Instance selection is another critical issue. Real-world instances are often a poor choice. One reason is that in order to sketch performance curves for our algorithms, we need families of related instances rather than one-off data sets. Indeed, corpuses of data of varying sizes are often available; however, the issue here is that there is possibly no relation between the individual data sets in the corpus. For example, size may not correlate with complexity, and we may have a corpus where all the large data sets are "easy," while the small ones are "hard." Our goal is to capture some inherent notion of complexity that is independent of size. Consider a text corpus consisting of Shakespeare's plays and a collection of e-mail messages. This corpus certainly has both large and small instances, but these instances are unrelated. On the other hand consider one consisting of Shakespeare's plays as well as letters he composed. Now we have a relation among the instances that transcends size – namely, there were all written by a single individual. We may be able to find idealized or synthesize families of related instances; however, they may not capture key characteristics of real-world data. On the other hand, real-world data sets tend to be unrelated to one another. This sort of tradeoff between rich but complex versus simper but less accurate models is not uncommon; we saw another example of this type of tradeoff above in CPU versus wall-clock time. The latter is a simpler and more natural model but fails to capture how a computer truly operates.

3.2 Community Detection Problems

Many real-world phenomena can be easily and usefully represented as node-edge graphs: personal interactions become social networks; webpages linking to each other become Internet graphs; protein interactions in biology become biochemical networks; etc. These graphs can quickly grow to be larger than millions of nodes – making them much too large for human reasoning, and even too large for any but the simplest algorithmic analyses.

Fortunately, most real-world graphs contain naturally "clumpy" groups. These groups are often referred to as communities. By reducing the massive graph from millions of nodes to many thousands of communities, some human reasoning can begin, and complex algorithmic analyses become more tractable. Moreover, since these communities are formed due to real group

characteristics, reasoning about the communities can truly lead to better comprehension of the overall graph.

Kleinberg and Lawrence used community detection to identify web pages that shared topics on the Internet [Kleinberg, 2001a]. They used this analysis to organize Internet content and proposed that this organization could help identify "hubs" (index pages that identify many pages in a topic) and "authorities" (leader pages in topic areas). They suggest such pages could help improve Internet search engines.

Colbaugh and Glass leveraged community detection in their early warning system for predicting which large-scale movements will gain traction [Colbaugh, 2010a; Glass, 2011a]. Their work is based on a principle from social sciences: if one hears of an event at various locations in ones life (work, church, sporting events, etc.), then one is more likely to think it important than if one heard of it in only one location. They use blogs and news sites on the Internet as proxy for real-world discussions. They create a graph of the blogs and news sites where edges are based on links between the sites. The communities identified on this graph serve as proxy to the different real-world communities where one might hear of an event. They then search for early distribution of the topic across multiple communities. They found that if the spread occurs before the overall volume on the topic becomes too high, it serves as a good predictor of real-world action.

Ravasz et al. use similar graph metrics to analyze metabolic pathways in cellular organisms [Ravasz, 2002a]. They demonstrate that metabolic pathways create sparse graph structures that are different from the small-world graphs built by social networks. They show that community detection can be used on the metabolic graph to separate largely disparate metabolic processes from each other.

As previously mentioned, graphs such as those described above quickly grow into millions of nodes. Therefore, community detection algorithms must function on massive-scale graphs. Many types of community detection optimization problems map to the Max Cut problem: given the inverse of the current graph (creating an edge at each location where none exists and no edge where any currently exists), partition this graph into two where the most edges are cut. In the original graph, these are the two highest-level communities. One may then recursively apply this procedure on each of the obtained communities to obtain additional finer communities. Unfortunately, the Max Cut problem is NP-hard, and such problems are unlikely to have efficient algorithms. However, since community detection is a critical problem, many sub-optimal algorithms exist for finding acceptable-quality partitions of the graph into community.

Thus our interest in community detection problems:

1) It is a widely applicable problem.
2) Several classical-computer based algorithms have been created. Moreover, optimized implementations of many of these algorithms are freely available.
3) Many of the core problems can be readily cast as Max Cut and other types of QUBO instances for comparison with AQC implementations.

3.2.1 Classical Heuristics

Clauset-Newmann-Moore (CNM) [Clauset, 2004a]: Clauset, Newmann, and Moore proposed an efficient algorithm for a greedy approximation to community detection. They begin by assigning each node its own community. Then, at each iteration, they merge the two communities that will most improve modularity. Modularity is high when the number of edges within a community is significantly greater than would be expected in a random sampling of the graph. This algorithm requires no user inputs. By intelligently choosing their data structures, this algorithm runs in $O(n \log^2 n)$ time.

Walktrap [Pons, 2006a]: Pons and Latapy's algorithm begins from the observation that random walks on a graph starting from any node are more likely to stay within that node's community than to leave it. This is due to the fact that most edges should stay within the same community, and fewer leave it. They assign each node u in the input graph a probability distribution, over the nodes in the graph, indicating the likelihood of reaching each node in t steps when starting at u. The assumption is that members of the same community should have similar probability distributions. They begin by assigning each node its own community. Then, during each iteration, they merge the two communities that minimize the change in the probability distribution when merged. This algorithm requires the user to specify the number of steps to take in the random walk. Their algorithm runs in time $O(mn^2)$.

Louvain [Blondel, 2008a]: Blondel, Guillame, Lambiotte, and Lefebvre's algorithm performs a variant of CNM's algorithm. They begin by assigning each node its own community. During the first step, they identify which communities can be merged to increase modularity. This continues as long as modularity increases. When modularity can no longer increase, they create a new graph from the communities created in step one. They repeat these two steps until no communities are merged in the first step. This algorithm requires no user inputs. While they do not provide an algorithmic analysis, they demonstrate that Louvain functions faster than previous techniques and continues to run on massive datasets when previous algorithms fail to complete.

Even though the Walktrap algorithm merges based on a different metric than modularity, all of these algorithms compute their best result and compare to each other based on modularity. Mathematically, modularity (Q) is defined as follows:

$$Q = \frac{1}{2m}\sum_{v}\sum_{w}\left[A_{vw} - \frac{k_v k_w}{2m}\right]\delta(c_v, c_w),$$

where m is the number of edges, v and w are any two nodes, A_{vw} is 1 if an edge exists between nodes v and w (else 0), k_v is the degree of node v (the number of edges incident to v), and $\delta(c_v, c_w)$ is 1 if v and w are in the same community (else 0). In words, this equation sums across the whole graph, and for all nodes in the same community, it adds to the modularity if the two nodes are connected by an edge and subtracts from the modularity if they are not.

However, maximizing modularity is not necessarily the best way to solve community detection. Fortunato and Barthélemy found that the smallest community that could be found via modularity

maximization is bounded by the number of edges in the graph [Fortunato, 2007a]. Specifically, the smallest community that can be found is bounded by

$$m_c \geq \sqrt{\frac{M}{2}},$$

where m_c is the number of edges in a community, and M is the total number of edges in the graph. This means that, even when the graph is obviously partitioned into much smaller communities, modularity maximizing methods will merge the smaller "true" communities until the within-community internal edge count reaches this threshold.

wCNM [Berry, 2011a]: Berry, Hendrickson, LaViolette, and Phillips recognize Fortunato and Barthélemy's limitation, but note that it only holds true for unweighted edges. The equation becomes more complicated when edge weights are included, and the resolution limit becomes based on the overall weight on the graph and the weight of the edges within a community. They propose a pre-weighting algorithm that weights edges based on their likelihood of being within-community edges: By adding weights to an edge based on the number of triangles and rectangles it is a member of, they increase the weight of edges with strong interconnections – thus likely within a community. This algorithm requires the user specify the number of pre-weighting iterations to run.

We expect to compare the four algorithms described above. We found implementations of each[1].

3.3 Producing realistic datasets

To realistically assess these algorithms, we need to evaluate them on real-world-like graphs. To test how they scale, we must apply the algorithms against multiple graphs of widely varying size. However, graphs have many different properties: number of nodes, number of edges, per-node degree distribution, clustering coefficients, average path length, etc. If we select graphs solely based on number of nodes, the many other features may vary – leading to skewed results. Thus, we would like to generate graphs where critical features remain largely constant while changing the graph size.

However, we still want to preserve the reality inherent in the graphs. Real-world graphs are generally sparse, power-law (or similar) degree distributed, with relatively high clustering coefficients, and low average path length. These features generally result from external features in the graph (such as political affiliation, relative age, education level, etc. in social networks).

Our proposed procedure for analyzing community detection algorithms on classical computing is as follows:

1) Identify real-world graphs to use as bases for generated graphs.

[1] CNM – From the SNAP codebase http://snap.stanford.edu/snap/index.html
Walktrap – http://www-rp.lip6.fr/~latapy/PP/walktrap.html
Louvain – http://perso.uclouvain.be/vincent.blondel/research/louvain.html
wCNM – From the MTGL codebase https://software.sandia.gov/trac/mtgl.

2) Measure various characteristics for each graph.
3) Create graphs derived from the basis graphs at various scales (from hundreds to millions of nodes).
4) Run each of the above community detection algorithms against these graphs.
5) Report the results of each of the community detection algorithms for computational efficiency (execution time, memory usage, disk usage, etc.) and quality of results (modularity and other metrics).

3.4. AQC Benchmarking

Benchmarking simulated Quantum Annealing relative to classical approaches has become a recent theme in the AQC community [Boixo, 2013a; Hen 2011a; McGeoch, 2013a]. Several groups have presented results comparing Quantum Monte Carlo (QMC) simulations of Quantum Annealing (QA) to Simulated (Thermal) Annealing (SA), as well as problem-specific code such as WalkSAT for the well-known Boolean Satisfiability Problem (SAT). Currently there is no strong evidence that Quantum Annealing is superior, although there is certainly evidence that, as expected, QA is an inherently different beast than SA.

We feel that there are shortcomings to the current benchmarking approaches, and our goal is to develop an infrastructure that would allow us to perform simulations that address these shortcomings. Mathias Troyer's group at ETH Zürich has produced a tightly engineered parallel and GPU-based SA code [Boixo, 2013a]. Their group consists of expert software engineers, and they have apparently consulted with NVidia as well. Thus it is no surprise that their SA code is orders of magnitude faster than simulated QA, as well as the D-Wave One machine (Rainer chip, 108 qubit). We understand that one of the goals of the Troyer effort was to produce a clear data point addressing D-Wave's claims that their machines outperform classical SA; however, we feel that comparing a marvel of engineering against a potential marvel of nature is perhaps not a fair comparison. If AQC proves indispensable, skilled engineers will come.

Boixo and colleagues have produced interesting results that clearly differentiate the performance of QA from that of SA; moreover, their results show that the D-Wave machine's performance profile matches that of QA, perhaps lending evidence to the claim that it is a bona fide quantum device. Hen and Young have produced performance curves indicating that both simulated QA and exact state-of-the-art WalkSAT code scale exponentially for a variety of SAT problems [Hen, 2011a]. One of the problems considered is actually solvable in polynomial time via Gaussian Elimination, but neither the simulated QA nor WalkSAT "figured out" that they were solving an "easy" problem. We feel that such sanity checks are essential. Is it reasonable to expect stellar performance on hard problems if AQO/AQC cannot even provably solve easy problems efficiently? Admittedly the matter is not as simple as stated, since perhaps there is a way of expressing polynomial solvable problems as AQO/AQC instances that solve efficiently. This is critical area of research in rendering AQC a viable practical technology, since the same issue applies to any problem: different representations of a problem as AQC instances likely lead to differences in performance, so finding a "good" representation is critical

As noted earlier, instance selection is riddled with subtleties beyond just obtaining a real-world data set. The foremost of which is that in order to assess the scalability of algorithms, we need a

family of related instances of varying sizes. Another issue is that although the state-of-the-art classical algorithms can directly handle large real-world data sets, our quantum simulations, of course, require more computational resources and can only handle small instances. Moreover, an instance of a community detection problem, such as computing modularity, must be reduced to a suitable instance of something like Max Cut, which our simulations can more easily handle. Sometimes such reductions are straightforward and other times there are a plethora of choices, with some resulting in better performance than others. This issue is exacerbated if we restrict ourselves to Hamiltonians admissible to specific quantum devices such as D-Wave's, in which case we must also find an efficient embedding. Thus producing an accurate comparison is challenging, and our simulations must address these issues.

Both Boixo et al. and McGeoch and Wang use random Ising instances (over -1 or +1 coupler weights) on the D-Wave Chimera Graph (avoiding the embedding issue). Hen and Young also use SAT instances with a random component [Hen, 2011a]. Random families of instances of varying sizes are easy to generate, but it is well known in combinatorial optimization that such instances do not necessarily reflect either real-world or hard instances.

4. RESULTS

4.1 AQC Toolkit

We have abstracted and selected three general-purpose classes of algorithmic tools currently offered by adiabatic quantum algorithms. These tools are each of a different algorithmic flavor, and our goal is that this diversity will enable us to address a range of problems.

1) Machine Learning: Quantum Boosting

   QBoost (Quantum Boosting) is an adiabatic algorithm developed collaboratively by Google and D-Wave and is inspired by the classical AdaBoost (Adaptive Boosting) algorithm [Neven, 2012a]. These algorithms solve the binary classification problem, which is a general-purpose machine learning problem useful in a variety of settings including text analysis and feature recognition. The high level idea driving these algorithms is the iterative construction of a strong binary classifier from a potentially large collection of weaker classifiers. The latter are essentially black boxes from the point of view of boosting algorithms, and can be existing generic classical classifiers or specialized classifiers for the problem at hand.

   QBoost is appealing from a scalability perspective as the quantum resources required primarily depend on the number of weak classifiers employed. One may expect reasonable results even with a modest collection of weak classifiers. Such instances may be solved directly on an AQC with hundreds of qubits in a single run. For larger instances, which may call for thousands of weak classifiers for a sufficient level of accuracy, a hybrid version of QBoost may be performed in stages consisting of many AQC runs on smaller instances. On the other hand, as pointed out in [Smelyanskiy, 2012a], although QBoost offers increased accuracy over AdaBoost for some classification problems, "it takes longer just to set up the problem prior to the quantum annealing step than it does to run the Adaboost algorithm."

   However, one may imagine applications for which one is willing to wait longer for more accurate results. Streaming algorithms are those to which input is presented in a sequence of blocks of data, and the algorithm does not necessarily have random access to the entire data stream. Typically such algorithms have a very small amount of memory available to them, and they are only allowed a few passes over the stream, in many cases just one. An example is real-time analysis of network traffic. A streaming algorithm for this example may live in a network switch or node with a small high-speed buffer. A goal, for instance, may be to identify an attack as it is occurring. Another example is real-time text analysis of e-mail. One may want to classify each e-mail message as suspicious or not in real time, for example in the context of insider threat. E-mail volume is relatively low compared to general network traffic, and performance rather than time may be a more critical objective. Moreover, each message itself is relatively small, hence quantum machine learning algorithms like QBoost may be suitable for such scenarios.
2) Combinatorial Optimization: Maximum Cut

   Max Cut is a cornerstone combinatorial optimization problem that arises in a variety of

contexts including analysis of spin glass models, image segmentation, clustering, community detection in social networks, and VLSI circuit design. It is NP-complete, hence in some sense it is a universal representation of hard combinatorial optimization problems. Instances of other common NP-hard problems such as SAT and Maximum Independent Set are relatively easily cast as Max Cut instances with a modest overhead.

Our selection of Max Cut as a prototypical combinatorial optimization is motivated by several factors. Max Cut is almost trivially expressed as an instance of Quadratic Unconstrained Binary Optimization (QUBO), and the two problems are closely related. The latter is the focus of Adiabatic Quantum Optimization (AQO), and in particular the problem natively solved by D-Wave machines.

Another motivation for Max Cut is that it has been studied widely and approached from different perspectives for over 30 years. Researchers have developed exact algorithms (requiring super-polynomial time in the worst case), heuristics, and approximation algorithms; they have applied metaheuristics including Tabu search, Simulated Annealing, and GRASP. Max Cut has inspired one of the most beautiful and celebrated approximation algorithms, *i.e.*, Geomans' and Williamson's semidefinite-programming random projection algorithm [Goemans, 1995a]. This efficient algorithm is guaranteed to construct a solution whose weight is at least 0.878 of the maximum possible weight; moreover, it is conjectured that no polynomial-time algorithm can offer performance better than 0.878 (on arbitrary instances).

Finally, for many well-studied combinatorial optimization problems, curated libraries of instances exist. Such libraries typically contain instances of varying sizes and difficulty and offer a systematic approach to benchmarking. For Max Cut, we have the Biq Mac (**Bi**nary **q**uadratic and **Ma**x **c**ut) library [Wiegele, 2007a].

Scalability is a challenge with AQO algorithms such as those for Max Cut. Hard instances over several hundred nodes do exist, and Biq Mac includes several such instances. In standard QUBO formulations for Max Cut, each node of the graph is represented by a QUBO variable, which in turn is represented by a qubit (ignoring embeddings). Thus there is potential for impact; however, addressing big data is an issue. As mentioned earlier, we feel that hybrid algorithms are the most promising approach for leveraging traditional combinatorial optimization problems, such as Max Cut, in solving big data problems. The main hurdle is devising classical decomposition techniques for Max Cut that effectively harnesses the potential power of an AQ accelerator.

3) Eigenvector optimization: Quantum Eigenvector Sampling

An intriguing AQ algorithm for solving the well-known PageRank problem is proposed in [Garnerone, 2012a]. The PageRank problem amounts to finding a steady-state distribution of a random walk on the (directed) link graph of the web. This can be reduced to finding the eigenvector corresponding to the smallest eigenvalue of a suitably defined n by n Hermitian matrix, where n corresponds to the number of nodes (i.e., pages) in the web graph. An entry

of this eigenvector gives the PageRank of the corresponding page, which is roughly the amount of time an infinite random walk would spend on that page. This is a useful measure since it gives a measure of importance to a page based only on the structure of the link graph and not the contents of the pages.

Finding an optimal eigenvector is a well-studied classical problem, and algorithms such as the power method deliver a solution in $O(n \log(n))$ worst-case time. An advantage of the proposed Quantum Page Rank (QPR) algorithm is that, once the problem instance is loaded, the algorithm offers a speedup in sampling the largest $O(\log(n))$ entries of the page rank vector for certain types of graphs. This is relevant for page rank since one is typically interested in identifying a small number of important pages. Such an algorithm is useful in scenarios where one wishes to repeatedly process a single data set. Web search is an example, for once indexing has occurred, each subsequent query takes a very small amount of time. One may also expect that small changes to the data set do not require starting from scratch.

The QPR algorithm may be easily extended to a general algorithm for sampling from an optimal eigenvector of a Hermitian matrix. This has applications in graph analysis, including analyzing social networks.

4.2 Realistic Graph Generation

One of the primary contributions of our work is infusing the benchmarking of both classical and AQC algorithms with more realistic problem instances. As previously mentioned, this is challenging for several reasons: (i) families of real-world instances of widely varying scale are difficult to obtain, (ii) real-world instances are not directly consumable by AQC algorithms and may require significant preprocessing or transformation, and (iii) a particular set of real-world instances may be too easy or too hard and not accurately represent the range of instances of interest. To overcome these hurdles we propose generating synthetic graphs that are designed to mimic certain features of real-world graphs. This gives us more control over the instances while still reflecting realistic data. Realistic graph generation has emerged as a topic of recent interest in the study of complex and social networks, which is one of our main motivations for focusing on community detection in such types of networks.

Creating realistic graphs given a basis graph is performed in three distinct ways in the literature:

- Given some characteristics, create random graphs based on those characteristics. Several algorithms have been proposed for this generative graph model.
- Given the basis graph, sample from it to create a subgraph that preserves specific qualities. Several algorithms have been proposed for this graph-sampling model.
- Given the data used to generate the basis graph, run the same algorithm from differing starting locations and differing durations to create similar graphs of different scales. We only found one instance of this data-sampling model in the literature.

Lancichinetti, Fortunato, and Radicchi (LFR) proposed a graph generation algorithm specifically designed for benchmarking community detection algorithms [Lancichinetti, 2008a]. The LFR algorithm accepts five parameters: number of nodes, degree distribution exponent, average node

degree, community size exponent, and a mixing parameter (between zero and one). Given these inputs, the algorithm assigns each node a degree ensuring the degree distribution and average degree are preserved. It then randomly assigns each node to a community, ensuring that the community is large enough to contain a node of this degree. Edges are randomly assigned within the community – preserving the assigned degree for all nodes. In a final step, random edges are removed from within communities and assigned outside the community until the mixing-parameter-defined fraction of edges occur between communities instead of within communities. While the LFR algorithm generates random graphs with some user-specifiable parameters – which can be set to match those observed in a real-world graph – the results are confounded by the non-natural mixing parameter.

Kim and Leskovec proposed that multiplicative attribute graphs (MAGs) can be used to represent external features like political affiliation, gender, etc. [Kim, 2012a]. Specifically, probability matrices for each feature represent the likelihood that two nodes are connected by an edge. Thus, if one node is assigned a 0 and another a 1 for a feature, the probability they connect based on this feature is stored at A(1, 0) in the matrix. Nodes are assigned a 0 or 1 for a feature based on a specified probability (p_i). They proposed an algorithm, called MagFit, to generate these probability matrices and feature probabilities from existing graphs; a companion algorithm, called MagGen, creates a random graph that follows those probability distributions. Implementations of both are available as part of the SNAP codebase.[2] Given this model, an attribute contributes an average probability (P_i) of an edge forming between two "average" nodes as follows:

$$P_i = p_i^2 A(0,0) + p_i(1-p_i)A(0,1) + (1-p_i)p_i A(1,0) + (1-p_i)^2 A(1,1)$$

All attributes are combined by multiplication to form the overall probability that an edge will be created between two nodes:

$$\mathcal{P} = \prod_{\forall i} P_i$$

After experimenting with this method, we realized something intrinsic about the MAG model: the above average edge-forming probability ($\mathcal{P}$) does not depend on the number of nodes at all. On average, this probability is applied at every opportunity, resulting in $\Theta(n^2)$ expected edges, where n is number of nodes. This means that the expected average degree (number of edges divided by n) scales with n – an average node connects to a fixed percentage of the nodes in the graph.

In their study of graphs growing over time, Leskovec, Kleinberg, and Faloutsos found that for real-world citation and affiliation graphs (authors-to-papers and actors-to-movies) average degree does increase with time [Leskovic, 2007a]. The amount of increase varied between $n^{1.08}$ to $n^{1.68}$ (average $n^{1.26}$). Even the most aggressively increasing degree fell well below MAG's n^2 rate.

[2] http://snap.stanford.edu/snap/download.html

Seshadhri, Kolda, and Pinar proposed a method similar to LFR, but ground their Block Two-Level Erdös-Rényi model (BTER) on several real-world-graph theories [Seshadhri, 2011a]. Much like LFR, BTER requires distribution parameters for both the degree distribution and the community size distribution. First, BTER randomly assigns each node to a community. Second, each community is self-connected as an Erdös-Rényi graph. Some number of connections for each node are reserved for cross-community edges, which are added next. These cross-community edges are added randomly. They demonstrate that BTER better matches average clustering coefficient by degree and the eigenvalues of the sample graph than a model proposed by Chung and Lu [Chung, 2002a; Chung, 2002b].

Both LFR and BTER should better match the degree distribution as graph size increases than MAG. Unfortunately, LFR and BTER both require community size distribution to generate graphs – a feature that cannot be known for the input graph without having definitive answers for the community detection problem.

4.2.1 Graph Sampling

Leskovec and Faloutsos examined several simple graph-sampling techniques including random node sampling, random edge sampling, and differing random walk techniques [Leskovic, 2006a]. Their research indicated that different sampling mechanisms biased the results in different ways: random node sampling preserved degree better, random edge preserved weakly connected components better, and random walks preserved clustering and size patterns better.

Hubler, Kriegel, and Borgwardt developed two Metropolis-based algorithms that sampled with emphasis on preserving clustering coefficient or degree distribution [Hubler, 2008a]. Maiya and Berger-Wolf further emphasized that sampling techniques can be developed for specific purposes when they developed Expansion Sampling to quickly sample from all communities in a graph [Maiya, 2010a]. They also argue that these sampling biases can be considered benefits to be exploited for specific purposes [Maiya, 2011a].

Due to the biasing caused by all sampling methods, we are concerned that we would have to run various sampling methods to generate subgraphs to identify the effects of the biases on the community detection algorithms before we could trust the results in any side-by-side comparison across different scales as we had hoped.

4.2.2 Data Sampling

When examining how graphs grow over time, Leskovec and Faloutsos returned to the original data and generated the graphs over time [Leskovic, 2006a]. This highlights that although there are many available datasets of graphs, those graphs were all sampled or derived from some other dataset. Although likely not proposing a new solution to the graph generation/sampling paradigm, we believe this may be the best technique for us to generate graphs at different scales that should be self similar: if we create a graph from different scales of the original data, we should obtain graphs of different sizes with similar properties. If we create the graphs from different starting points in the original data but for the same scale, we should obtain graphs that are approximately the same size but differ.

We will shortly begin generating graphs from the Border Gateway Protocol dataset.[3] If this data shows self-similarity across various metrics over different scales and different starting points, we will find other datasets until we have sufficient for our testing.

4.3 Simulation

In order to better understand the function of adiabatic quantum computers, such as the D-Wave device, we are developing a two-pronged simulation capability. The first of these capabilities is a purely classical simulation of the time-dependent behavior of the annealing of a corresponding classical spin system. This approach was inspired by [Smolin, 2013a], and will allow for a control to show whether the quantum behavior of the bits is necessary to explain the performance of the D-Wave machine. Currently we are testing small simulations on a single placquette of D-Wave style spins with arbitrary connections between the spins.

The second simulation capability we are developing is a quantum Monte Carlo algorithm to understand the Hamiltonians at any given point along the annealing. This should allow a detailed look at how different encodings of a given problem would affect the density of states and therefore the requirements on the speed of the evolution in the quantum annealing. As the capability becomes more efficient, the simulations will also be able to address how the system size affects the speed and thus can speak to the scaling prospects for D-Wave-style computers.

An additional theoretical question arises due to the specific connectivity of the D-Wave qubits. This restriction of the D-Wave Hamiltonians allows for quantum Monte Carlo algorithms to solve the problem without approximation in polynomial time without any complications due to the Fermion sign problem. We will seek to understand whether increasing the flexibility of the D-Wave design would allow for the efficient solution of problems that are thought to be impossible in polynomial time with quantum Monte Carlo or whether the current performance of the machine is intimately connected to the ability to map the problem onto something that can be simulated efficiently via classical means.

4.4 Critical Review of Recent Relevant Results

McGeoch and Wang announced the first experimental results that seem to demonstrate a remarkable performance advantage for D-Wave's latest chips over state-of-the-art classical approaches [McGeoch, 2013a]. The study considered several Computer-Science-flavored combinatorial optimization problems (e.g. Max 2-SAT); however, we will discuss the one on which the most striking results were obtained: random {-1,1}-weighted QUBO instances on the native D-Wave hardware graph.

[3] http://www.ripe.net/data-tools/stats/ris/ris-raw-data

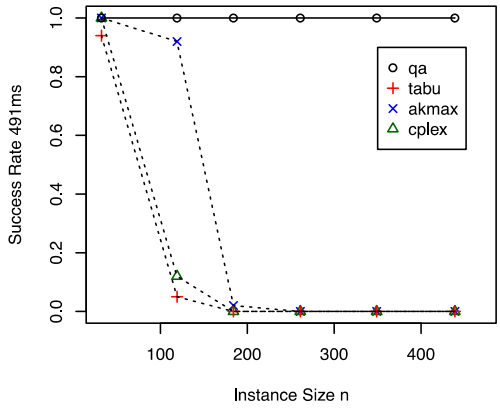

Figure 4: Success rates: proportion of best solutions found in 491 ms CPU time (tabu, amax, cples software) and exclusive access time (QA hardware).

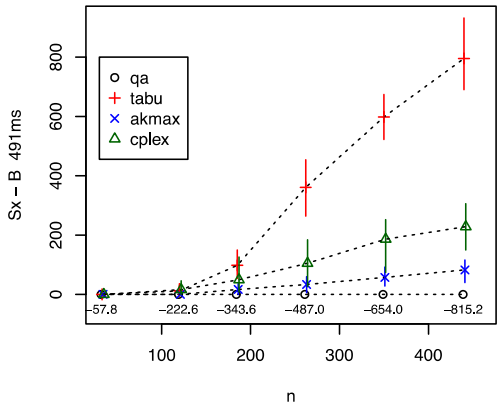

Figure 5: Differences between solution cost S_x in 491 ms runtime, and best solution B for each input. Dotted lines connect means for each solver; vertical bars show the range of observations. The numbers at bottom are means $\bar{B}$ for each problem size.

The graphs above [McGeoch, 2013a] indicate that the tested D-Wave machine is able to consistently obtain optimal solutions within 491ms, while the probability of the other solvers doing so, as well as the quality of the solutions they obtain, decays with instance size. Success in figure 1 is measured by finding the best solution among the four solvers, which is not necessarily the optimal solution; however, it turns out that the D-Wave machine did find the optimal solution in 97% of the instances on the largest, n=439, problem size. The classical CPLEX general-purpose integer program solver comes in next, requiring 10 minutes to obtain optimal solutions on 99% of the instances or 30 minutes to find all the optimal solutions. Comparing 491ms (D-Wave optimal on 97% of instances) and 10 minutes (CPLEX optimal on 99% of instances) gives a speedup of approximately 1222. Is this a fair assessment of the potential power of the D-Wave machine? Although the study is certainly intriguing, further investigation is warranted. We would not expect the D-Wave machine (or any machine) to consistently find optimal solutions to NP-hard problems in constant time on increasingly larger instances – this is impossibly good to be true.

It may be the case that we need larger instances to witness a performance degradation; however, this also begs the question whether the considered instances are actually (NP-)hard. In fact this is an apparent subtlety that seems to have been missed by Boixo et al. as well [Boixo, 2013a]. It is known that QUBO instances are NP-hard on the D-Wave graph (in fact even on a much simpler graph consisting of two "stacked" 2D-grids), even when weights are restricted to {-1,0,1} [Barahona, 1982a]. However, both the studies omit the linear term (magnetic field) and focus on weights in {-1,1}. Boixo et al. experimentally observe that "Local fields… give a bias to individual spins, tending to make the problem easier to solve for annealers," which may be true for the weights they selected, but is not true in general (at least from an NP-hardness point of view). For example, QUBO on a 2D-grid is NP-hard in general, but is solvable in polynomial time without the linear term [Barahona, 1982a]. Perhaps the more critical issue is the choice of weights in {-1,1} rather than in {-1,0,1}. Intuitively it makes sense to exclude 0-weights, since

this amounts to a wasted coupler; however, the presence of 0-weights allows modeling richer combinatorial structures. As noted above, the hardness of {-1,1} instances on the D-Wave graphs is not known, and we aim to reconcile this issue.

There is another issue worth highlighting. Boixo et al. also considered random {-1,1}-weighted QUBO instances without a linear term, albeit at smaller scales. In stark contrast to the above results, their results suggest an exponential scaling on the success probability, which is more plausible for a hard problem. How does one explain the discrepancy between these results? It turns out that the two studies are considering different problems. Boixo et al. define QUBO such that the variables take values in {-1,1} while McGeoch and Wang's variables takes values in {0,1}. With the presence of a linear term, these two versions of QUBO are equivalent in the sense that an instance of one may be transformed to an instance of another by a linear transformation in the variables that preserves solution values. However, in the absence of a linear term, the two problems are not equivalent via a direct linear transformation. Given McGeoch and Wang's results, it is conceivable that their version of the problem is significantly easier than Boixo et al.'s version.

We also feel that these studies may not be selecting the right kind of classical algorithms for comparison. Any quantum computer is inherently probabilistic, while the sorts of classical algorithms considered are designed to deterministically return an exactly optimal solution. A fair comparison would be against approximate or randomized classical heuristics. Though we do not have a precise theoretical explanation, it has long been observed that randomized (classical) algorithms seem more powerful than deterministic algorithms. Exact algorithms are by definition required to provide an optimal answer; this is a significant burden. For example, one might argue that the comparison between D-Wave and CPLEX in the McGeoch study is fair because the best solution that each produced in a fixed amount of time was selected. In this case why should it matter that one algorithm is guaranteed to eventually produce an optimal solution? One answer is that an exact algorithm allocates its resources with the goal of producing an exact solution. CPLEX in particular is a branch-and-bound-based code, and one may set parameters that determine how aggressive CPLEX is in searching for better solutions versus attempting to prove the optimality of its best-known solution. If CPLEX were run with a pre-specified time limit, it might work harder at finding good solutions rather than worrying about proving optimality; the default CPLEX strategy likely attempts to balance these tasks. Exact algorithms provide a tremendous amount of extra information in a proof of optimality, i.e. that no other solution could possibly be better, so we should expect that they perform more work.

Another issue in using CPLEX out-of-the-box is that although it is able to handle quadratic optimization problems over integer variables, such as QUBO, this is not necessarily the best way to present such a problem to CPLEX. One may argue that any algorithm or technique may have some preferred problem representation on which it performs well, and it may not be fair to cater to this preference. However, in this case we note that CPLEX is first and foremost a mixed integer linear program solver; the quadratic capability is in some sense an ancillary convenience feature. Moreover, it is relatively easy to recast integer quadratic programs as integer linear programs. Dash [Dash, 2013a] recently observed that by using a simple textbook integer linear formulation of McGeoch and Wang's version of QUBO (without a linear term), that CPLEX is able to solve McGeoch and Wang's instances as quickly as the D-Wave Two. We had

performed experiments independently arriving at this conclusion as well. Figure 3 compares the performance of CPLEX on the McGeoch-Wang {0,1}-variable QUBO instances with {-1,1}-variable QUBO instances, as employed by Boixo et al.

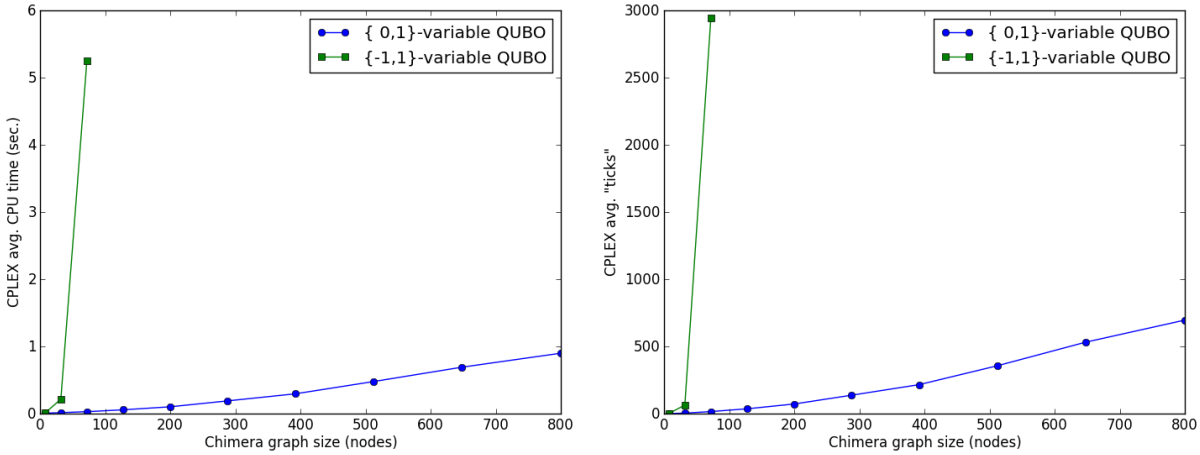

Figure 6: CPLEX v12.4 performance on random {-1,1}-weight instances of QUBO, without a linear term, on the Chimera graph; 100 instances of each size were generated and -1 weights occur with probability 0.5. Experiments were performed on a 16-core machine with 3GHz Intel Xeon cores. Green curves indicate the QUBO variant used by Boixo et al. while blue curves indicate those of McGeoch and Wang.

CPLEX is able to solve Chimera graph instances of up to 800 nodes within 1 second; moreover, it is able to solve instances of size up to 512 nodes within 0.5 second, which is comparable to the results McGeoch and Wang obtained on the D-Wave Two machine. There is a sharp contrast in terms of the scalability for the two types of instances. CPLEX on {0,1}-variable QUBO appears to scale linearly, while for {-1,1}-variable instances, even 128-node instances exhausted the assigned time limit of 10 minutes. This is consistent with the exponential scaling for such instances observed by Boixo et al. Our results clearly raise the question of whether the {0,1}-variable instances that McGeoch and Wang considered are NP-hard. The most striking conclusion is that not only does CPLEX exhibit very modest scaling on {0,1}-variable instances but that it also matches the performance of the D-Wave Two machine in an absolute sense, requiring under 1 second for the instance sizes considered. CPLEX is able to leverage multiple cores, and to address the impact of parallelism we considered both CPU time as well as measure of total work which CPLEX calls "ticks." The similarity in the graphs of Figure 3 suggests that running CPLEX on a single-core machine is likely to deliver comparable results.

The complexity of random instances is often highly sensitive to parameter changes. To ensure that the McGeoch-Wang instances are easy in some robust sense, we also considered random instances with varying probability, p, of selecting a weight of -1 for each edge. Both Boixo et al. and McGeoch and Wang consider a probability of $p = 0.5$, resulting in equal expected numbers of edges of weight -1 and 1. We found that even in this regime, the {-1,1}-variable instances are exhibit an exponential scaling and are empirically hard across all nontrivial probabilities (i.e. $0 < p < 1$). Our results for the {0,1}-variable McGeoch-Wang instances appear in Figure 4. We

report that random 800-node instances across all values of *p* are empirically easy and solve within 2 seconds. The hardest of the McGeoch-Wang instances occur for *p* somewhere in the range [0.65, 0.7]. Our results underscore the inherent challenge and plethora of pitfalls in instance selection.

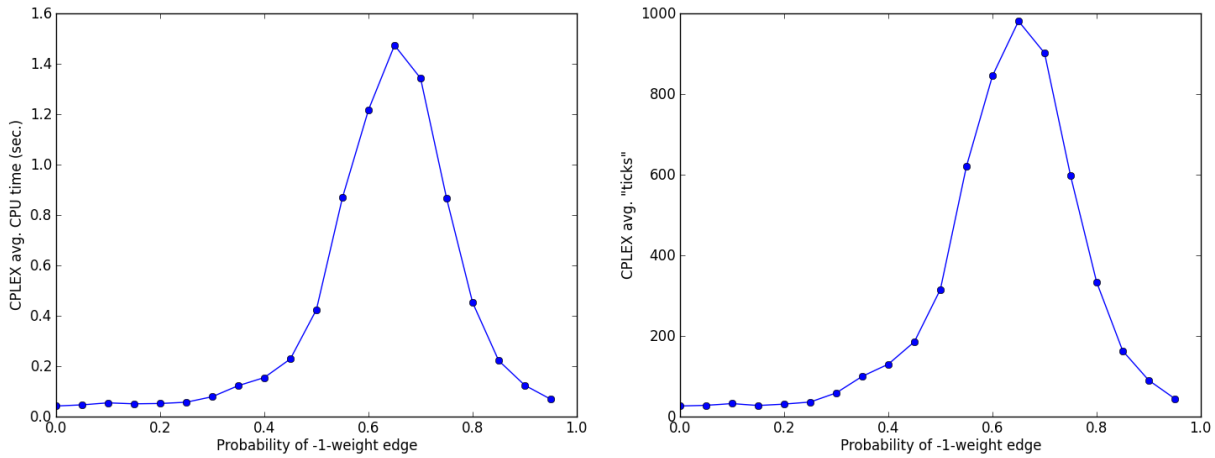

Figure 7: CPLEX v12.4 performance on random {-1,1}-weight, 800-node Chimera graph instances with specified probability of selecting an edge of weight -1; 100 instances of {0,1}-variable QUBO were generated at each probability value, and as above, a linear term is not present.

5. REFERENCES

[Aharonov, 2004a] D. Aharonov *et al., Adiabatic quantum computation is equivalent to standard quantum computation.* In Proc. 45th Symp. Found. Comp. Sci., pp. 42-51 (2004).

[Amin, 2008a] M. H. S. Amin, P. J. Love, and C. J. S. Truncik, *Thermally assisted adiabatic quantum computation*. Phys. Rev. Lett. **100**, 060503 (2008). doi:10.1103/PhysRevLett.100.060503. arXiv:cond-mat/0609332.

[Barahona, 1982a] Francisco Barahona, *On the computational complexity of Ising spin glass models*. *J. Phys. A: Math. Gen.* **15,** 3241 (1982). doi:10.1088/0305-4470/15/10/028.

[Berry, 2011a] Jonathan W. Berry, Bruce Hendrickson, Randall A. LaViolette, and Cynthia A. Phillips, *Tolerating the Community Detection Resolution Limit with Edge Weighting*. Physical Review, (2011).

[Blondel, 2008a] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, *Fast Unfolding of Communities in Large Networks*. Journal of Statistical Mechanics: Theory and Experiment, (2008).

[Boixo, 2013a] Sergio Boixo, Troels F. Rønnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer, *Quantum annealing with more than one hundred qubits.* arXiv:1304.4595, (2013).

[Bravyi, 2009a] S. Bravyi and B. Terhal, *Complexity of stoquastic frustration-free Hamiltonians*. SIAM J. Comput. **29**, 1462 (2009). doi:10.1137/08072689X. arXiv:0806.1746.

[Carroll, 2013a] M. S. Carroll *et al.*, *Semiconductor qubits for adiabatic quantum annealing and quantum computing*. Provisional US Patent, application number 61757609, (2013).

[Chung, 2002a] F. Chung, and L. Lu, *The Average Distances in Random Graphs with Given Expected Degrees*. In Proceedings of the National Academy of Sciences **99**, (2002).

[Chung, 2002b] F. Chung, and L. Lu, *Connected Components in Random Graphs with Given Degree Sequences.* Annals of Combinatorics **6**, (2002).

[Clauset, 2004a] Aaron Clauset, Mark Newmann, and Christopher Moore, *Finding Community Structure in Very Large Networks*. Physical Review, (2004).

[Colbaugh, 2010a] Richard Colbaugh and Kristin Glass, *Early Warning Analysis for Social Diffusion Events*. In The IEEE Conference on Intelligence and Security Informatics (ISI), (2010).

[Dash, 2013a] Sanjeeb Dash, *A note on QUBO instances defined on Chimera graphs.* arXiv:1306.1202, (2013).

[Dickson, 2013a] N. G. Dickson *et al*., *Thermally assisted quantum annealing of a 16-qubit problem*. Nature Comm. **4**, 1903 (2013). doi:10.1038/ncomms2920.

[Farhi, 2000a] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *Quantum computation by adiabatic evolution*. arXiv:quant-ph/0001106, (2000).

[Farhi, 2001a] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem.* Science **292**, 472 (2001). doi:10.1126/science.1057726. arXiv:quant-ph/0104129.

[Feynman, 1986a] R. P. Feynman, *Quantum mechanical computers*. Found. Phys. **16**, 507 (1986). doi:10.1007/BF01886518.

[Fortunato, 2007a] Santo Fortunato and Marc Barthélemy, *Resolution Limit in Community Detection*. Proceedings of the National Academy of Sciences of the USA **104**, (2007).

[Fuhrer, 2009a] A. Fuhrer *et al*., *Atomic-scale, all epitaxial in-plane gate donor quantum dot in silicon*. Nano Lett. **9**, 707 (2009). doi:10.1021/nl803196f.

[Gaitan, 2012a] F. Gaitan and L. Clark, *Ramsey numbers and adiabatic quantum computing*. Phys. Rev. Lett**. 108**, 010501 (2012). doi:110.1103/PhysRevLett.108.010501. arXiv:1103.1345.

[Gaitan, 2013a] F. Gaitan and L. Clark, *Graph isomorphism and adiabatic quantum computing*. arXiv:1304.5773, (2013).

[Garnerone, 2012a] S. Garnerone, P. Zanardi, and D. A. Lidar, *Adiabatic quantum algorithm for search engine ranking.* Physical Review Letters **108**, 230506 (2012). doi:10.1103/PhysRevLett.108.230506. arXiv:1109.6546

[Glass, 2011a] Kristin Glass and Richard Colbaugh, *Web Analytics for Security Informatics*. In The European Intelligence and Security Informatics Conference (EISIC), (2011).

[Goemans, 1995a] M. X. Goemans and D. P. Williamson. *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*. Journal of the ACM (JACM) **42**, 6:1115-1145 (1995).

[Hayashi, 2003a] T. Hayashi *et al*., *Coherent manipulation of electronic states in a double quantum dot*. Phys. Rev. Lett. **91**, 226804 (2003). doi:10.1103/PhysRevLett.91.226804. arXiv:cond-mat/0308362.

[Hen, 2011a] Itay Hen and A. P. Young, *Exponential Complexity of the Quantum Adiabatic Algorithm for certain Satisfiability Problems*. Phys. Rev. E **84**, 061152 (2011).

[Hollenberg, 2004a] L. C. L. Hollenberg *et al.*, *Charge-based quantum computing using single donors in semiconductors*. Phys. Rev. B **69**, 113301 (2004). doi:10.1103/PhysRevB.69.113301. arXiv:cond-mat/0306235.

[Hubler, 2008a] C. Hubler, H.-P. Kriegel, K. Borgwardt, and Z. Ghahramani, *Metropolis Algorithms for Represntative Subgraph Sampling*. In Proceedings of the IEEE International Conference on Data Miniming, (2008).

[Jordan, 2013a] S. Jordan, *Quantum Algorithm Zoo*. URL: http://math.nist.gov/quantum/zoo/, (last updated 2013).

[Kim, 2012a] Myunghwan Kim and Jure Leskovec, *Multiplicative Attribute Graph Model of Real-World Networks*. Internet Mathematics **8**, (2012).

[Kitaev, 2002a] A. Yu. Kitaev, A. Shen, and M. N. Vyali, *Classical and quantum computation*. Graduate Studies in Mathematics, translated from the Russian edition by L. J. Senechal **47**, (2002).

[Kleinberg, 2001a] Jon Kleinberg and Steve Lawrence, *The Structure of the Web*. Science **294**, (2001).

[Lancichinetti, 2008a] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi, *Benchmark graphs for testing community detection algorithms*. The American Physical Society **78**, (2008).

[Leskovic, 2006a] Jure Leskovec and Christos Faloutsos, *Sampling from Large Graphs*. In Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2006).

[Leskovic, 2007a] Jure Leskovec, Jon Klenberg, and Christos Faloutsos, *Graph Evolution: Densification and Shrinking Diameters*. In Proceedings of ACM Transactions on Knowledge Discovery from Data (TKDD) **1**, (2007).

[Maiya, 2010a] Arun S. Maiya, and Tanya Y. Berger-Wolf, *Sampling Community Structure*. In Proceedings of the World Wide Web Conference, (2010).

[Maiya, 2011a] Arun S. Maiya, and Tanya Y. Berger-Wolf, *Benefits of Bias: Towards Better Characterization of Network Sampling*. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, (2011).

[McGeoch, 2013a] Catherine C. McGeoch and Cong Wang, *Experimental evaluation of an adiabiatic quantum system for combinatorial optimization*. In Proceedings of the ACM International Conference on Computing Frontiers (CF '13), (2013). doi:10.1145/2482767.2482797.

[Messiah, 1961a] A. Messiah, *Quantum Mechanics* (Dover, Mineola, NY, 1999), translation from the French *Mechanique Quantique* by G. M. Temmer. Previously published in English by (North-Holland, New York, 1961).

[Mizel, 2007a] A. Mizel, D. A. Lidar and M. Mitchell, *Simple proof of equivalence between adiabatic quantum computation and the circuit model.* Phys. Rev. Lett. **99**, 070502 (2007).

[Neven, 2008a] H. Neven, G. Rose, W. G. Macready, *Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization*. arXiv:0804.4457, (2008).

[Neven, 2012a] H. Neven, V.S. Denchev, G. Rose & W.G. Macready**.** *QBoost: Large Scale Classifier Training with Adiabatic Quantum Optimization*. Journal of Machine Learning Research W&CP **25**, pp. 333-348 (2012).

[Petersson, 2009a] K. D. Petersson *et al.*, *Microwave-driven transitions in two coupled semiconductor charge qubits*. Phys. Rev. Lett. **103**, 016805 (2009). doi:10.1103/PhysRevLett.103.016805. arXiv:0907.2409.

[Petta, 2005a] J. R. Petta *et al.*, *Coherent manipulation of coupled electron spins in semiconductor quantum dots*. Science **309**, 2180 (2005). doi:10.1126/science.1116955.

[Pons, 2006a] Pascal Pons and Matthieu Latapy, *Computing Communities in Large Networks using Random Walks*. Journal of Graph Algorithms and Applications (JGAA) **10**, (2006).

[Pudenz, 2011a] K. L. Pudenz and D. A. Lidar, *Quantum adiabatic machine learning*. Quant. Inf. Proc. **12**, 2027 (2013). doi:10.1007/s11128-012-0506-4. arXiv:1109.0325.

[Rahman, 2011a] R. Rahman *et al.*, *Stark tuning of the charge states of a two-donor molecule in silicon*. Nanotech. **22**, 225202 (2011). doi:10.1088/0957-4484/22/22/225202. arXiv:0907.3929.

[Ravasz, 2002a] E. Ravasz, A. L. Somera, D.A. Mongru, Z. N. Oltvai, and A.-L. Barabási, *Hierarchical Organization of Modularity in Metabolic Networks*. Science **297**, (2002).

[Seshadhri, 2011a] C. Seshadhri, Tamara G. Kolda, and Ali Pinar, *Community Structure and Scale-free Collections of Erdös-Rényi Graphs*. arXiv:1112.3644, (2011).

[Shinkai, 2009a] G. Shinkai *et al.*, *Correlated coherent oscillations in coupled semiconductor charge qubits*. Phys. Rev. Lett. **103**, 056802 (2009). doi:10.1103/PhysRevLett.103.056802. arXiv:0905.0805.

[Shor, 1994a] P. W. Shor, *Discrete logarithms and factoring*, in Proc. 35th Ann. Symp. on Found. Comp. Sci. **124,** (1994). doi:10.1109/SFCS.1994.365700

[Smelyanskiy, 2012a] Vadim N. Smelyanskiy, Eleanor G. Rieffel, Sergey I. Knysh, Colin P. Williams, Mark W. Johnson, Murray C. Thom, William G. Macready, and Kristen L.

Pudenz, *A Near-Term Quantum Computing Approach for Hard Computational Problems in Space Exploration*. arXiv:1204.2821, (2012).

[Smolin, 2013a] John A. Smolin and Graeme Smith, *Classical signature of quantum annealing*. arXiv:1305.4904, (2013).

[Tracy, 2010a] L. A. Tracy et al., Double quantum dot with tunable coupling in an enhancement-mode silicon metal-oxide semiconductor device with lateral geometry. App. Phys. Lett. **97**, 192110 (2010). doi:10.1063/1.3518058. arXiv:1011.0034.

[Weber, 2012a] B. Weber *et al*., *Engineering independent electrostatic control of atomic-scale (similar to 4 nm) silicon double quantum dots*. Nano Lett. **12**, 4001 (2012). doi:10.1021/nl3012903.

[Wigele, 2012a] A. Wigele. *Biq Mac Library (Binary quadratic and Max cut Library)*. URL: http://biqmac.uni-klu.ac.at/biqmaclib.html, (last updated 2007). Retrieved on December, 14th 2012.

[van der Wiel, 2003a] W. G. van der Wiel *et al*., *Electron transport through double quantum dots*. Rev. Mod. Phys **75**, 1 (2003). doi:10.1103/RevModPhys.75.1. arXiv:cond-mat/0205350.

[van Weperen, 2011a] I. van Weperen *et al*., *Charge-state conditional operation of a spin qubit*. Phys. Rev. Lett. **107**, 030506 (2011). doi:10.1103/PhysRevLett.107.030506. arXiv:1101.5654

DISTRIBUTION

1	MS1322	John B. Aidun	1425 (electronic copy)
1	MS1326	Ojas D. Parekh	1465 (electronic copy)
1	MS0899	Technical Library	9536 (electronic copy)


Sandia National Laboratories