Berkeley CS191x: Quantum Mechanics and Quantum Computation
Optional Class Project

This document describes the optional class project for the Fall 2013 offering of CS191x. The project will not be graded.

In this project, we will explore the computational power of D-Wave's alleged quantum computer, the D-Wave One. To understand where it stands in comparison to classical computers, we will implement a simple classical algorithm for the optimization problem that the D-Wave One is specifically designed to solve and compare the performance of the two.

This project will require a good command of the C programming language. You may use other low-level programming languages such as Fortran if you wish, but please note that high-level languages such as Python or Java will not be able to compete since we want to run highly optimized numerical computation.

You can think of the project as a discussion question with a programming component. There will be an official thread on the forum for discussion of the project, and we encourage you to openly exchange ideas, results, and even codes. Even if you do not have the required programming skills, you can still participate by contributing conceptual ideas on how to optimize the code.

Of course, we will also provide you with feedback whenever appropriate. We hope you enjoy working on it!

Note: If you want to jump right into the project, you can start reading from Section 4. But we strongly recommend that you first read the first three sections, which will provide a nice context in which this project is scientifically meaningful and relevant.

1 Brief History

On May 11, 2011, D-Wave Systems announced what they claimed to be the world's first commercial quantum computer, the D-Wave One, which they described as a quantum annealer with 128 qubits [1]. The device, whose "quantumness" has been called into question, was purchased by Lockheed Martin and installed at the Lockheed Martin Quantum Computing Center at the University of Southern California. In April 2013, a research group that had been conducting experiments with this device published a paper claiming evidence of quantum behavior in the device [2]. In May 2013, McGeoch and Wang published a paper [3], which claimed that the D-Wave Two solves certain optimization problems 3,600 times faster than the best classical software solvers. They failed to disclose in their paper that their study was sponsored by some combination of D-Wave and Google, and the specific tests they performed were determined by their sponsors [4].

Soon after the publication of that paper, it was announced that a collaboration between Google, NASA, and Universities Space Research Association was to purchase a new D-Wave computer, the D-Wave Two, a quantum annealer with 512 qubits [5]. It was said that they intended to use it for their research on artificial intelligence and machine learning. D-Wave Systems is now claiming that they have a machine that is "better at something than any other option available," [6] and that they will have a 2,000-qubit quantum annealer by 2015 [7].

The coverage of these events in the news media and popular press was quite breathless, and promoted the view that D-Wave has successfully created quantum computers that are orders of magnitude faster than the best classical computers. The goal of this project is to give you an opportunity to explore and understand precisely what computational problems the D-Wave machine solves and whether there are any scientific grounds on which to believe the claims made by the company and the news media.

2 Quantum Annealing

First, we briefly discuss what a quantum annealer is. D-Wave's machines are built upon the same idea as that of adiabatic quantum computing, but they are not true adiabatic quantum computers. The reason is that their system does not satisfy the two conditions of the theoretical model of adiabatic quantum optimization: a) the system should be isolated from the environment and b) it should be run at zero temperature. Instead, D-Wave's machines do interact with the environment and run at a nonzero temperature. The resulting physical process is a somewhat noisy version of adiabatic quantum optimization whose performance is even less well understood theoretically than adiabatic quantum optimization. The hope that this noisy "quantum annealing" may still yield decent solutions to optimization problems in practice.

3 State of the Art

In this section, we briefly summarize the two main scientific results about D-Wave's machines that have been repeatedly quoted by the media as evidence in favor of D-Wave [2, 3].

1. Quantum annealing with more than one hundred qubits, Boixo et al., 2013 [2]

   This paper reports experimental results about the D-Wave One installed at USC's Lockheed Martin Quantum Computing Center. They run three different algorithms on a certain optimization problem (this problem, which we will define precisely in the next section, will be referred to as the "D-Wave problem" from this point on, because it is the problem that the D-Wave machines are specifically designed to solve) and compare the results. The following are the three algorithms compared in the paper.

   (a) **Simulated Annealing**: Simulated annealing is a classical probabilistic algorithm for optimization problems. It is in many cases one of the first choices when dealing with a computational problem for which we do not know a provably efficient algorithm.
   (b) **Quantum Monte Carlo**: Despite its slightly misleading name, Quantum Monte Carlo is a classical algorithm that attempts to simulate quantum dynamics.
   (c) **The D-Wave One**

   The experiment that they run is as follows; first, they randomly generate 1,000 instances of the D-Wave problem. Then they run each of the above three algorithms 1,000 times on each one of those 1,000 instances and record how many times it found the optimal answer. Thus, they obtain the success probabilities for each algorithm on each of the 1,000 instances. (That is, we will have 3,000 of these success probabilities.)

   Based on these data, the paper initially exhibited two pieces of evidence to refute the claim that the D-Wave machine was a purely classical device. We will only discuss one of them here because the other one has been successfully refuted by Smolin and Smith [8]. The surviving evidence is that the correlation between the success probabilities of the D-Wave One and the success probabilities of Quantum Monte Carlo is higher than that between D-Wave and Simulated Annealing. The paper puts this forth as evidence that the D-Wave machine must be performing some kind of quantum annealing and therefore must be using entanglement. The jury is still out on whether this constitutes convincing evidence of "quantumness" of the D-Wave machine. In particular, it is not clear whether the paper has ruled out other possible classical dynamics according to which the D-Wave machine might be evolving.

   The other result of this paper, which the news media mostly failed to pick up on, is that their simulated annealing code was in fact about 6 times faster than the D-Wave One.

2. Experimental Evaluation of an Adiabatic Quantum System for Combinatorial Optimization, McGeoch and Wang, 2013 [3]

   This paper contains the results of experiments with the newer D-Wave Two machine as performed by Catherine McGeoch and Wang. In this paper, they compare the D-Wave machine to three off-the-shelf classical solvers on a slightly differently phrased version of the D-Wave problem. Also, since they are using the D-Wave Two which has 512 qubits, the size of their test instances is much larger than in the Boixo et al. paper, using as many as 439 qubits as opposed to 108 qubits of Boixo et al.

   They report that the D-Wave Two was as much as 3,600 times faster than the classical solvers on the D-Wave problem. However, here also remain some questions, especially whether they are comparing to the right classical solvers. The three classical solvers that they use in this experiment are general-purpose solvers that can handle any combinatorial optimization problem, whereas the D-Wave Two is designed specifically for the D-Wave problem (although it can also handle other problems by encoding them in the form of the D-Wave problem). In fact, within a month, Alex Selby claimed to have written a classical code that was 160 times faster than the D-Wave Two [9], based on an algorithm that he devised specifically for the D-Wave problem. But it is not possible to compare his algorithm to McGeoch et al.'s results directly, as the data from McGeoch et al. have not been made public and thus we cannot test the algorithm on the same data set.

McGeoch may have been aware of these problems from the beginning, as she said in an interview in May 2013 that "the speed tests are also not quite fair, because generic computers will always perform less well than a device dedicated to solving a specific problem" [10]. She also commented later, "our tests were never meant to be used to compare platform speeds, and it is wrong to use our data to support arguments either way. . . . Fundamentally, I think the experiments in our paper . . . are far too small in scope to support any conclusions about the larger picture. . . . I think the results are not nearly as exciting as the press does" [4].

Concluding this section, we bring to your attention that we are summarizing 30 pages worth of material on one page here. If you want to get a complete picture, you can read the above papers yourself, but that is out of the scope of this project.

4 Project

In this project, we will implement the simulated annealing algorithm and run it on the same test instances of the D-Wave problem which were used in the experiments of Boixo et al. [2] to see how it compares to the performance of the D-Wave One. Ideally, we would want to do this experiment with the newer D-Wave Two machine, but the data regarding that machine have not been made public at this time.

4.1 The D-Wave Problem

The combinatorial optimization problem solved by the D-Wave machine, as defined in [2] is the following Ising spin glass ground state problem:

1. **Input:** Number of particles n, coupling constants $J_{ij} \in \{-1, 1\}$ for each $1 \leq i < j \leq n$.
2. **Goal:** Find an assignment $z_i \in \{-1, 1\}$ that minimizes the following function:

$$H = -\sum_{i<j} J_{ij} z_i z_j$$

(The actual combinatorial optimization problem solved by D-Wave imposes some very strict constraints on the structure of the interaction graph, which we are suppressing in this description. For further detail see the appendix of [2].)

Note that we can easily formulate this problem in the formalism of the quantum adiabatic algorithm using the following initial and final Hamiltonians.

$$H_0 = -\sum_{1\le i\le n} \sigma_i^x$$
$$H_f = -\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z$$

If we start in the ground state of H_0 which is just $|+\rangle^{\otimes n}$ and then evolve our Hamiltonian slowly from H_0 to H_f, we will arrive in the ground state of H_f which will give us the optimal solution to the D-Wave problem.

4.2 The Data

The research group that conducted the experiments of [2] have generously shared their test instances as a service to the community. The data can be downloaded at http://arxiv.org/src/1305.5837v1/anc.

Their input files are formatted as follows.

```
# name: /Users/.../Benchmarking--...-13-55-11.mat with energy -173
1 5 -1
1 6 -1
1 7 -1
1 8 -1
2 5 1
2 6 -1
...
...
```

As one can see, the first line of the file contains the optimal solution for that particular instance. From the second line, the coupling constants are given in the form "$i\ j\ J_{ij}$". All unspecified J_{ij}'s are assumed to be zero. For instance, in the above example $J_{12} = J_{13} = J_{14} = J_{19} = \cdots = 0$. The number of particles n is fixed to be 108 for all their test instances.

In `success.txt`, they provide the success probabilities of the D-Wave One for each of the 1,000 instances, i.e. the probability that it finds the optimal solution.

4.3 Goal

The goal of the project is to write a classical code that beats the D-Wave One all-around. That is, (a) it should run faster and (b) it should achieve higher success probabilities on average.

According to the Methods section of [2], the D-Wave One takes 2.5 seconds to cool down to a sufficiently low temperature and then performs annealing for 5 microseconds, taking a total of about 2.5 seconds per instance. Ideally, we would like to beat the pure annealing time of 5 microseconds, which the authors of [2] actually did using high-end classical chips, but we expect that this would be difficult to achieve on a personal computer. So in this project, we will attempt to beat their total running time of 2.5 seconds. Note that, if achieved, this will already demonstrate that personal computers can readily outperform the D-Wave One, which was presumably priced at millions of dollars, on a specific problem that it was designed to solve.

With careful optimization, we expect that it should be possible to write codes that run in less than a hundred milliseconds and still exhibit much better success probabilities than the D-Wave One.

4.4 Algorithm

You are free to solve the problem any way you want, but one guaranteed way to achieve our goal is to use simulated annealing, as demonstrated by [2]. We will only briefly sketch the algorithm here, but you can read [11] and [12] if you want more detail.

Simulated annealing is an algorithm that borrows its inspiration from a technique in metallurgy, in which a material is heated and then cooled slowly in a controlled manner. This controlled cooling helps the material explore the state space and find the lowest energy state, thus forming large crystals. If the material is cooled too fast, it will get stuck at a local minimum and yield defective material.

Simulated annealing is basically a local search algorithm that starts with some randomly generated solution (that is, randomly assign z_i's) and then explores its neighboring solutions by making some local change to it (for instance, flip the sign of z_i for some random i). A greedy local search algorithm would accept the new solution if and only if it is better than the old solution, and thus would very likely get stuck at a local minimum. In simulated annealing, we try to overcome this limitation by using the notion of temperature. The "temperature" is set to be a high value at the beginning of the algorithm and is gradually decreased down to zero as the algorithm progresses. When the temperature is high, the algorithm will not only accept better solutions, but also accept worse solutions with some probability. When the temperature is zero, the algorithm will only accept better solutions.

At a high level, the simulated annealing algorithm has the following structure.

```
Generate a random state z_i's.
temperature = some high constant
for some fixed number of steps do the following
    pick random i
    new state = old state except the sign of z_i is flipped
    if H_new <= H_old, accept the new state
    if H_new > H_old, accept with probability exp(-(H_new-H_old)/temperature)
    decrease the temperature
```

Note that this is a mere guideline and you should feel free to modify and optimize this skeleton to obtain better performance. One obvious degree of freedom comes from the choice of the annealing schedule, i.e. the pace at which we decrease the temperature. Also note that you can control the runtime of the algorithm by choosing the number of steps appropriately.

4.5 Analysis

Once you have a working code, you have to compare its success probabilities to the D-Wave One's. A good way to visualize your results is to produce a histogram of the success probabilities as in Figure 1.

5 Final Words

We hope that in the process of working on this project you will come to better understand the combinatorial optimization problem encapsulated in the Ising spin glass ground state problem, as well as form an informed opinion about the D-Wave controversy.

As quantum computing researchers, we are excited to see these experiments being done which can potentially provide us with very useful insights on the power of quantum systems. Unlike the news media, as scientists our only duty is to get to the truth of the matter. To quote Bertrand Russell's advice to future generations "When you are studying any matter, or considering any philosophy, ask yourself only: What are the facts, and what is the truth that the facts bear out. Never let yourself be diverted, either by what you wish to believe, or what you think could have beneficent social effects if it were believed; but look only and solely at what are the facts."

We hope you enjoy working on the project!

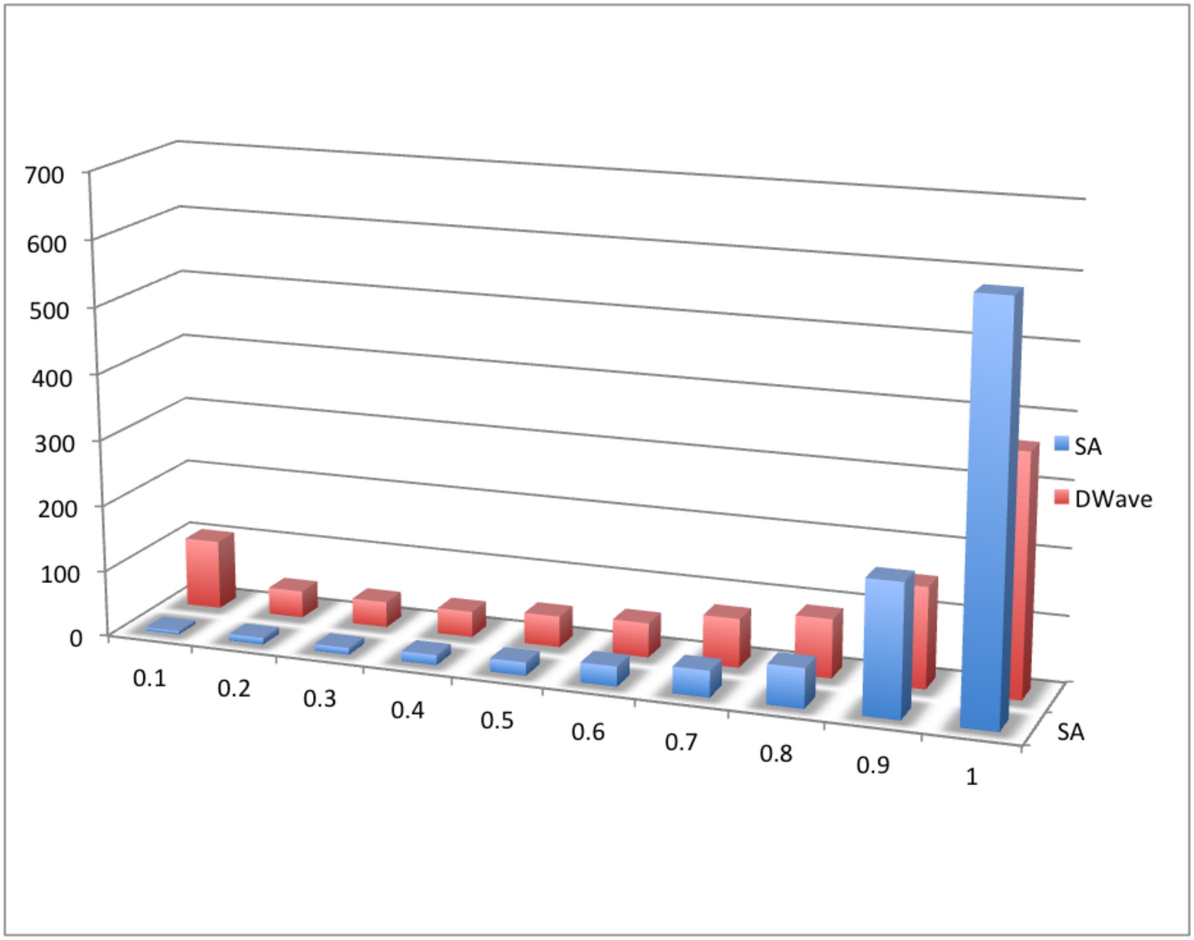

Figure 1: An example histogram

References

[1] http://dwave.wordpress.com/2011/05/11/learning-to-program-the-d-wave-one/.

[2] http://arxiv.org/pdf/1304.4595v2.pdf.

[3] http://graphics8.nytimes.com/packages/pdf/business/quantum-study.pdf.

[4] http://www.scottaaronson.com/blog/?p=1400.

[5] http://www.scientificamerican.com/article.cfm?id=google-nasa-snap-up-quantum-computer-dwave-two.

[6] http://blogs.scientificamerican.com/guest-blog/2013/08/27/quantum-computing-disentangled-a-look-behind-the-d-wave-buzz/.

[7] http://nextbigfuture.com/2013/09/dwave-systems-scales-up-production-of.html.

[8] http://arxiv.org/abs/1305.4904.

[9] http://www.archduke.org/stuff/d-wave-comment-on-comparison-with-classical-computers/.

[10] http://www.newscientist.com/article/dn23519#.ukpwngtwl9c.

[11] http://en.wikipedia.org/wiki/simulated%20annealing.

[12] http://en.wikipedia.org/wiki/metropolis%20algorithm.