The Theory and Effectiveness of Adiabatic Quantum Computation

A report submitted as the examined component of the Project Module SXP390

Steven Anderson (A8895844)

September 7, 2014

4927 words

Abstract

I present an overview of adiabatic quantum computation (AQC) from a computational perspective, and in comparison to classical computing and the circuit model. This report was produced via a review and analysis of the relevant literature. I first describe the AQC method and how it differs from the circuit model, then proceed to survey the most important circuit model algorithms and their computational complexity. I show that AQC can replicate their success - indeed both methods are computationally equivalent. The running time of AQC algorithms is then considered, particularly the effect of entanglement and the energy gap of the ground and first excited states of the Hamiltonian. Variants of AQC that exploit knowledge of the energy gap are also described. The role of AQC algorithms in efficiently solving NP-Complete problems is discussed, although as yet no algorithm is known to be efficient for all problem instances. Finally, I consider the advantages of AQC over the circuit model. These include the naturally quantum nature, making both reasoning and realisation more simple; robustness against decoherence and other errors; and the potential applications to NP-Complete problems. I conclude that AQC is a useful alternative to the circuit model, but given the equivalence to the circuit model, future research should focus on AQC optimisations and error correction techniques.

[216 words]

Contents

0.1 Introduction

Adiabatic quantum computation (AQC) is an alternative model for quantum computation to the traditional **circuit model**. Its operation is considered more 'naturally quantum' (Ahrensmeier, 2006, p.647), as the entire entangled quantum system evolves continuously subject to Schrödinger's equation, in contrast to the discrete transformations that the circuit model employs.

Many algorithms have been proposed that would solve problems using AQC, a number of which are of particular interest because of their potential to speed up a class of hard problems known as **NP-Complete**. This class of problems is unsolvable in an efficient way using any known classical or quantum algorithm, and many computer scientists believe this is an intrinsic property of the problems (e.g. Aaronson, 2008a). Thus, the tantalising suggestion of an efficient adiabatic algorithm is of great importance to computer science.

The nature of AQC also has inherent advantages such as its protection from decoherence (Childs *et al.*, 2002) and the ability to exploit the intuitions of quantum physicists (Aharanov *et al.*, 2008, p.757). However, there is controversy as to whether these theoretical advantages have much effect if one wishes to build a real world, fault-tolerant adiabatic quantum computer (Young *et al.*, 2013).

This report will look at AQC from a theoretical perspective, i.e. the theory behind its operation, the problems it can be applied to, and the speedups it can provide over classical algorithms. The realisation of adiabatic quantum computers is a wide and complex field, so this report restricts itself to an analysis from the computational perspective.

The objectives for this report are fourfold:

1. Describe the operation of adiabatic quantum computers in comparison to the traditional circuit model of quantum computing and classical computing.
2. Contextualise the role of entanglement in adiabatic quantum computing.
3. Summarise the problems for which adiabatic quantum computing provides speedups versus classical computing and versus the circuit model.
4. Explain and analyse the future direction of research into the theory of adiabatic quantum computers and provide conclusions about the usefulness of this method.

The search tools used for compiling this review were the Open University One Stop Search and Google Scholar. The One Stop Search was used for all initial search queries (the terms "Adiabatic quantum computation", "Adiabatic quantum computer", "Adiabatic quantum computers", and "Quantum annealing"). Where key papers were referenced by the papers found, but didn't appear in the original searches, the Open University search facilities were used to find specific papers. Where this failed, Google Scholar was used as a fallback. RefWorks was used as a reference store and for compiling the bibliography.

Chapter 1

An overview of AQC

1.1 Computing models

1.1.1 The classical computing model

The unit of information in classical computing is the **bit**, which can hold one of two values: 1 or 0 (sometimes referred to as **True** and **False**). Physically, a bit is represented by a property of an electrical circuit, e.g. the voltage of a circuit element: low voltage corresponds to 0 and high voltage corresponds to 1. Sequences of bits are transformed in predictable ways by **Boolean logic gates**. An OR gate, for example, takes two bits as input and outputs a 1 if either of the input bits is 1, a 0 otherwise. By combining gates into circuits, sequences of bits can be transformed in complex, deterministic ways, and thus computations can be performed.

1.1.2 The quantum circuit model

Early quantum computing was a close analogue of classical computing. Instead of bits, it deals with **qubits**, which model an entire quantum state. A qubit may be in the quantum state $|0\rangle$, $|1\rangle$, or a linear combination of the two, $\alpha|0\rangle + \beta|1\rangle$ (Nielsen and Chuang, 2010, p.13). Physically, a qubit can be represented by any quantum system that has exactly two basis states, which can be labelled $|0\rangle$ and $|1\rangle$, e.g. the polarisation of a photon or the spin of an electron.

In the **circuit model**, sequences of qubits are acted on by **quantum gates**. They differ from classical gates as they can operate on linear combinations of states, and may introduce entanglement. For example, the CNOT gate acts on a two qubit state, and reverses the state of the second qubit if the first qubit is 1, otherwise it leaves the second qubit unchanged. When applied to the two qubit state $(\alpha|0\rangle + \beta|1\rangle)|0\rangle$ (where the first qubit is in a superposition state), the CNOT gate produces the entangled state $\alpha|00\rangle + \beta|11\rangle$ (Ekert and Kay, 2014). Quantum gates are arranged in a **quantum circuit** - the qubits are transformed by each gate in turn, in discrete steps. As such, complex computations can be performed.

1.1.3 The adiabatic quantum computation model

Adiabatic quantum computation (AQC) is a more recently proposed model (Farhi *et al.*, 2000), which breaks with the circuit analogy and is arguably more 'naturally quantum' (Ahrensmeier, 2006, p.647). It relies on the fact that many computing problems can be reduced to finding the ground state of a certain **problem Hamiltonian**, H_P. Finding this ground state may be difficult, but AQC allows it to be discovered quickly.

First, a system of qubits is arranged in the ground state of an **initial Hamiltonian**, H_B, chosen such that it is easy to produce (Farhi *et al.*, 2000). Often this is a superposition of all possible basis states for the system of qubits. The applied Hamiltonian is then slowly evolved from H_B into H_P. The **quantum adiabatic theorem** states that if this evolution is 'slow enough', the system remains in its ground state (Roland and Cerf, 2002). Therefore, the system will finish in the ground state of H_P, which encodes the solution to the original problem. Measurements of the system in its final state can then reveal the solution.

1.2 An example AQC algorithm

1.2.1 The Boolean satisfiability problem (SAT)

An N-bit instance of the **Boolean satisfiability problem** (shortened to **SAT**) begins with:

- n Boolean variables, x_i, which can each hold the value True or False.
- m **Boolean clauses**, C_j each of which is True or False depending on the values of any of the n variables (e.g. C_1 might be $(x_1 \vee \neg x_2)$, which is True if x_1 is True, or x_2 is False).

The problem to solve is then:

Is it possible to assign values to each variable x_i, such that all m clauses are satisfied (i.e. evaluate to True)? If so, what value should each variable be given to satisfy all m clauses?

1.2.2 AQC applied to SAT

This description follows the outline from the paper that introduced AQC (Farhi *et al.*, 2000), but it has been modified and expanded for this readership.

Modelling as qubits

Firstly, we represent each variable by a qubit: a qubit in the state $|0\rangle$ represents an assignment to the corresponding variable of False, and $|1\rangle$ similarly represents True. There are 2^n basis states, corresponding to all permutations of every qubit being in the state $|0\rangle$ or $|1\rangle$. Each basis state thus represents a particular assignment of True or False to every variable. We represent one of these basis states by:

$$|z_1\rangle|z_2\rangle|z_3\rangle...|z_n\rangle$$

Where we use positional notation (i.e. the first state corresponds to variable x_1), and each of the z_i's is 0 or 1. It is important to note that x_i represents the ith variable, whereas z_i represents the value (0 or 1) of the ith variable in a specific basis state. For example, a particular basis state may be:

$$|0\rangle|1\rangle|1\rangle...|0\rangle$$

Which corresponds to variable x_1 being False, variable x_2 being True, variable x_3 being True, and variable x_n being False.

The problem Hamiltonian

We wish to define a Hamiltonian, H_P, whose ground state encodes the solution to the SAT instance. For simplicity we consider 3-SAT, where each clause depends on at most 3 variables. A given clause, C, is said to depend on the variables x_{p_C}, x_{q_C} and x_{r_C}.

We then define a Hamiltonian for each clause. This Hamiltonian acts on the basis states in the following way:

$$H_{P,C}(|z_1\rangle|z_2\rangle...|z_n\rangle) = h_C(z_{p_C}, z_{q_C}, z_{r_C})|z_1\rangle|z_2\rangle...|z_n\rangle$$

Where h_C is an energy function that depends on the clause C. Specifically:

$$h_C(z_{p_C}, z_{q_C}, z_{r_C}) = \begin{cases} 0 & \text{if the values } (z_{p_C}, z_{q_C}, z_{r_C}) \text{ satisfy clause C} \\ 1 & \text{if the values } (z_{p_C}, z_{q_C}, z_{r_C}) \text{ do not satisfy clause C} \end{cases}$$

The problem Hamiltonian is the sum of the individual Hamiltonians for each clause, i.e.

$$H_P = \sum_C H_{P,C}$$

Which acts on the basis states in this way:

$$H_P(|z_1\rangle|z_2\rangle...|z_n\rangle) = \left[\sum_C h_C(z_{p_C}, z_{q_C}, z_{r_C})\right] |z_1\rangle|z_2\rangle...|z_n\rangle$$

The ground state of H_P is evidently the state in which $\sum_C h_C(z_{p_C}, z_{q_C}, z_{r_C})$ is minimised. Since h_C is 0 if the clause is satisfied by the given values, the ground state of H_P is that in which the most possible clauses are satisfied. Thus H_P is the desired problem Hamiltonian.

The adiabatic evolution

Having defined H_P, the qubits are initiated in the ground state of an initial Hamiltonian, H_B, which is a superposition of all basis states (i.e. all possible assignments). A time dependent Hamiltonian is then applied to the system:

$$H(t) = (1 - t/T)H_B + (t/T)H_P$$

In other words, the Hamiltonian is slowly changed from H_B to H_P, over a timeframe T. At time T, the system is in the ground state of H_P (providing the evolution was slow enough to be adiabatic).

Measuring each qubit in the ground state of H_P gives the value of the corresponding variable in the assignment that satisfies most clauses. We can then quickly check if all clauses are satisfied by simply evaluating each clause with the measured values. We then have the solution to the SAT instance: if all clauses are satisfied, there is a possible assignment, which is given by the measured values; if any clause is not satisfied, then it is not possible to satisfy all clauses simultaneously.

Chapter 2

The speed of AQC

2.1 Time complexity

To understand quantum speedups, the computer science concepts of **time complexity** and **complexity classes** must be introduced.

2.1.1 Big-O notation

The **time complexity** of an algorithm determines how it scales with input size. For example, an $O(n^2)$ algorithm has a running time that scales with the square of the input size. For a formal definition see Nielsen and Chuang (2010, p.136). An algorithm is considered **efficient** if its running time is **polynomial** (i.e. $O(n^p)$ for some p).

2.1.2 Complexity classes

Problems are classified according to the time complexity of the best known algorithms to solve them. For **decision problems** (those with a yes/no answer, which are often studied), three important complexity classes are:

- **P** - Problems where a solution can be found in polynomial time (i.e. given an instance of a decision problem, we can give the correct yes/no answer in polynomial time).
- **NP** - Problems where a solution can be verified in polynomial time. This means that given an instance of a decision problem and a proposed "yes" answer, we can verify that indeed the answer is yes in polynomial time. For example, given an instance of 3-SAT, and a proposed assignment of values to variables, it is trivial to check that each clause is satisfied by those values. Note that if we can find a solution in polynomial time, we can also verify solutions in polynomial time, so NP is a superset of P.
- **NP-Complete** - Informally, the 'hardest' problems in NP. Problems where a solution can be verified in polynomial time, and all other NP problems can be reduced to them in polynomial time. If we could find a polynomial time algorithm for a single NP-Complete problem, we could use it to solve all NP problems.

2.2 Speedups

2.2.1 Quantum speedups

The circuit model has already shown quantum computing can beat the best classical algorithms.

Grover's algorithm solves the problem of finding a marked item in a database. The best classical algorithm is simply to check each item in turn, and thus the worst case running time is $O(n)$, where n is the number of items. Grover proposed a quantum algorithm which is $O(n^{1/2})$, by employing constructive quantum interference to pick out the desired element (Grover, 1996). Thus it provides a quadratic speedup over the classical counterpart. Grover's algorithm is probabilistic - it will only be correct with a finite probability < 1. The probability of success can be increased by repeating the algorithm.

The Deutsch-Jozsa algorithm is a deterministic quantum algorithm, i.e. it always produces the correct answer. It calculates whether an unknown black box function (i.e. the internals are hidden) is constant (returns the same answer for all inputs) or balanced (returns one answer for half the inputs and another for the other half). Classically, one needs to evaluate just over half of the possible inputs, but the Deutsch-Josza algorithm requires only a single evaluation of the black box function. Overall, the Deutsch-Josza algorithm runs polynomial time, versus exponential time for the best classical algorithm (Deutsch and Jozsa, 1992). The algorithm is of great theoretical interest due to it being deterministic, but practically the problem it solves is contrived and of little use (Neilen and Chuang, 2012, p.36).

The jewel in the crown of circuit model algorithms is Shor's algorithm, which calculates the prime factors of an integer in polynomial time (Shor, 1999). This is an NP, but not NP-Complete problem, for which the best classical algorithms run in **sub-exponential** time, i.e. slower than $O(n^p)$ (polynomial time) but faster than $O(p^n)$ (exponential time). Many modern encryption systems like **RSA** rely on the difficulty of prime factorisation for security. Therefore, if Shor's algorithm could be implemented on a large enough quantum computer, these systems could be threatened. Hence, Shor's algorithm has prompted much interest from computer scientists.

2.2.2 Adiabatic speedups

Adiabatic counterparts to these three important algorithms have been described; Grover's algorithm (Roland and Cerf, 2002); the Deutsch-Jozsa algorithm (Das *et al.*, 2002); Shor's algorithm (Peng *et al.*, 2008). In fact, AQC has been shown to be equivalent to the circuit model (Aharonov *et al.*, 2008; Mizel *et al.*, 2007). This means that we can run a circuit model algorithm using AQC, or an AQC algorithm using the circuit model, with only a polynomial number of steps used in the overhead of converting one to the other. Thus, any problem which can be solved in polynomial time with AQC can also be solved in polynomial time with the circuit model. AQC, therefore, can provide the same speedups as the circuit model, but cannot improve upon them.

This equivalence should not, however, deter research into AQC. There are practical advantages to the method (as discussed in Chapter 3), and its more naturally quantum nature provides a useful alternative viewpoint on problems.

2.2.3 NP-Complete speedups?

AQC has shown promise in applications to NP-Complete problems. Indeed, it was first described in relation to SAT, which is NP-Complete (Farhi *et al.*, 2000). The AQC algorithm ran in polynomial time for certain instances. Farhi *et al.* (2001) provided a polynomial time algorithm for random instances of another NP-Complete problem - Exact Cover. However, these algorithms have only been shown efficient for specific problem instances, not in the general case.

The application of AQC to NP-Complete problems is a controversial topic, as efficient algorithms have so far proved elusive. It has long been suspected by computer scientists that NP-Complete problems cannot be solved efficiently, hence there is much scepticism surrounding a quantum solution. Aaronson (2008a) even proposes the hardness of NP-Complete problems could be considered a fundamental law, akin to the 2nd law of thermodynamics.

Theoretical objections to the work of Farhi *et al.* (2000) include Altshuler *et al.* (2010), who claim that the adiabatic optimisation fails on larger instances due to a phenomenon known as Anderson localisation. Choi (2011), however, disputes this claim, showing an assumption made in the analysis of Altshuler *et al.* is not true in general. Dickson and Amin (2011) argue similarly, supporting the proposed speedups of AQC. The application to NP-Complete problems thus remains an open question, and these discussions will be analysed further in Chapter 3.

2.3 Running time of AQC

2.3.1 The role of entanglement

In AQC, the initial and final states are not entangled, but during the evolution the system does become entangled (Ahrensmeier, 2006).

Passante *et al.* (2007) found evidence that greater entanglement leads to faster running times for two different quantum algorithms. Wen and Qiu (2008, p.1) also found that entanglement "has a significant impact on the computational complexity" of Grover's search algorithm.

However, it is not clear whether entanglement is a resource of speedup in general or just for the algorithms studied. Indeed, Biham *et al.* (2004) showed that quantum speedups are possible in the absence of entanglement, and Bhattacharya *et al.* (2002) showed that the speedup of Grover's algorithm is possible with entirely classical waves. It seems then that entanglement can be a valuable resource, but not in all circumstances. Further research regarding the role of entanglement in AQC would therefore be useful.

2.3.2 The limiting energy gap

The quantum adiabatic theorem states that the evolution must be "slow enough" for the system to remain in its ground state. The specific limiting speed of evolution (and hence the speed of an AQC algorithm) depends on the energy gap between the ground and the first excited states of the time-dependent Hamiltonian. The running time, T, and the energy gap, g, are related by: $T \propto 1/g^2$ (Farhi *et al.*, 2000).

In general, g is not constant throughout the evolution, prompting a number of variants:

- **Global AQC** (Farhi *et al.*, 2000) uses a constant evolution speed, based on the minimum gap over the whole evolution, g_{min}. This was the original form of AQC proposed.
- **Local AQC** (Roland and Cerf, 2000) varies the evolution speed as the gap g varies. The algorithm can therefore run faster at times when g is larger, providing a speedup over global AQC.
- **Partial AQC** (Zhang and Lu, 2010) only ensures the evolution is adiabatic over a small time interval, outside of which the Hamiltonian changes instantaneously. This method only produces the correct result with a certain probability, but the time reduction is such that repeated runs can be incorporated. There is uncertainty as to whether partial AQC provides a speedup versus local AQC. For example, Sun *et al.* (2013, p.1) claim that partial AQC "does not improve the time complexity" over local AQC.

Chapter 3

Is AQC worthwhile?

3.1 AQC versus the circuit model

Although AQC and the circuit model are equivalent (in that they can solve problems with the same time complexity), there is still value researching AQC in addition to the circuit model.

3.1.1 The search for algorithms

One of the major advantages of AQC is in finding new quantum algorithms. This is one of the most important problems in quantum computation (Aharonov *et al.*, 2008, p.757). Simply having an additional method with which to approach a problem is a positive development. In addition, AQC can benefit from "well-developed physics intuition in the area of adiabatic evolution" (Aharonov *et al.*, 2009, p.757). To develop circuit model algorithms, one requires much greater knowledge of classical computer science and classical algorithm design. The circuit model thus poses a greater barrier for those from a purely physics background. The prerequisite knowledge for AQC is only some fairly basic quantum physics. Thus, AQC has great value in widening the search for new algorithms.

3.1.2 Robustness and error correction

In the theoretical world, all methods of computing are perfect, in that they reliably transform the state of bits or qubits as expected. In reality, computing systems interact with real world components, which are subject to an array of error sources (e.g. electrical noise from other circuit components, the failure of individual circuit elements etc.). AQC also provides benefits in this area.

Childs *et al.* (2002) showed that AQC is inherently robust against both decoherence (at low temperatures) and random unitary perturbations of the Hamiltonian. Decoherence is one of the major challenges to building large scale quantum computers, and the random perturbations were designed to represent outside noise. In some cases, the addition of perturbations even increased the probability of success. They note further that their approach to fault tolerance can be built in to the quantum hardware, whereas previously proposed general fault tolerance systems involve "substantial computational overhead" (Childs *et al.*, 2002, p.1). Amin *et al.* (2009) back up these findings, revealing that global AQC is robust against decoherence even when stronger than the minimum energy gap. However, they also note that local AQC is not robust to decoherence, suggesting we may have to choose between running time (for which local AQC would be preferred), and robustness (for which global AQC would be preferred).

The inherent robustness of AQC is not, however, sufficient for a truly fault tolerant real world computing device. Errors will inevitably occur, and **error correction** methods must be employed to correct those that do (Young *et al.*, 2013). Lidar (2008) looked at various error correction techniques for AQC, proposing a number of methods. He does, however, note that this is incomplete, and would not provide full fault tolerance for AQC. Young *et al.* (2013) were even less optimistic, and concluded that purely adiabatic computation is incompatible with fault tolerance. They suggest that a hybrid model might be used, implementing circuit model error correction techniques in an AQC algorithm. But then they rightly question the usefulness of AQC as it "begins to look more and more like the circuit model" (Young *et al.*, 2013, p.10).

These fault tolerance failures bring into question the usefulness of the AQC method. However, their importance should not be overestimated. Firstly, the field of quantum computation is still in its infancy, and large scale realisations are many years away. Thus, regardless of whether AQC becomes a viable method for large scale devices, it can offer theoretical advantages in the meantime. Secondly, in the future there will likely be new developments in error prevention and correction which have not currently been proposed. Indeed, Young *et al.* (2013, p.10) note themselves that their analysis depends on two key assumptions, which are motivated only by convention and simplicity. We should not discard AQC because of the failures of specific error correction techniques.

3.2 The future for AQC

3.2.1 NP-Complete problems

The chance of AQC providing efficient algorithms for NP-Complete problems is slim. This is based on the many years of fruitless searching for classical algorithms, and the failure to show efficiency of quantum algorithms in the general case. There is however, little concrete evidence for this conclusion, apart from the lack of success.

In the face of such pessimism, Aaronson (2008a) proposes promoting the hardness of NP-Complete problems to an assumption to guide future theories. That is, if a new physical theory had as a consequence the efficient solution of NP-Complete problems, that theory can be discarded. This would give us a useful tool with which to test new theories (similar to the 2nd law of thermodynamics - if a new theory implies a decrease in entropy in an isolated system, it can be discarded). While a noble goal, the level of existing evidence does not justify such an extreme viewpoint. Whilst efficient solutions seem unlikely, and have so far eluded us, there is no satisfactory way of testing the assumption that they don't exist. The 2nd law of thermodynamics, in contrast, has been tested empirically, among many different systems, and always found to hold. This puts the two principles on vastly different levels. In summary, we should treat claims of efficient NP-Complete algorithms with scepticism, but we should not rule out the possibility.

3.2.2 The D-Wave

Of particular interest to the AQC field is the controversy surrounding the supposed first commercial quantum computer (the **D-Wave One**). Its method of operation relies on AQC, so if the machine is verified to perform how its manufacturers claim, this proves AQC as a viable quantum computing model.

There has been much scepticism surrounding the claims made about the D-Wave. Firstly, it is unknown whether it is truly quantum in nature, or simply an "extremely expensive and inefficient classical computer" (Aaronson, 2008b). The company published experimental evidence in 2011, showing that the operation of the machine cannot be explained by a classical model (Johnson *et al.*, 2011). Other researchers have reached similar conclusions, for example Boixo *et al.* (2014). However, as the device is relatively new and has been subject to limited investigation, there is still uncertainly surrounding its quantum nature (Hsu, 2013).

Secondly, even if the D-Wave is shown to correctly implement AQC, it is not clear whether it outperforms classical computers. A recent study by Rønnow *et al.* (2014, p.420) finds "no evidence of quantum speedup" in the D-Wave, for certain problem classes. However, they also note that the problem of determining possible speedups is "subtle", and "do not rule out the possibility of speedups for other classes of problems" (Rønnow *et al.*, 2014, p.420). There will undoubtedly be many efficiency improvements in quantum computers in the coming decades, so it would be unfair to rule out AQC by comparing a very early implementation to highly optimised modern classical computers.

The most important lesson from the D-Wave controversy is to be vigilant and critical of the source of future research in this field. For example, Dickson and Amin (2011) published a rebuttal to the argument made by Altshuler *et al.* (2010) that AQC speedups would fail. Dickson and Amin are affiliated to D-Wave Systems, who obviously have a vested interest in proving the efficacy of AQC. We must therefore be careful to scrutinise the methods and conclusions of such papers thoroughly.

3.2.3 The direction of future research

As noted above, AQC provides a new and potentially more useful (for some people) way of presenting quantum algorithms. The future will therefore likely see more AQC algorithms described, solving a wider range of problems, including those of a more practical nature. A recent example is that of Garnerone *et al.* (2012), who describe an AQC algorithm for calculating the PageRank vector. PageRank is the principal tool by which the Google search engine ranks web pages, so is a significant and practical algorithm.

Since AQC is computationally equivalent to the circuit model, further research must focus on the unique aspects and advantages of AQC. The two areas requiring most attention are the optimisation of the running time of AQC algorithms (both through AQC variants and exploitation of entanglement); and error suppression and correction techniques. If the error correction problem can be solved, the simplicity of AQC may well make it more desirable than the circuit model in the large scale realisations of the future.

3.3 Conclusion

To conclude, adiabatic quantum computation (AQC) is an alternative, more 'naturally quantum' model for quantum computation than the traditional circuit model. AQC algorithms have been described for NP-Complete problems, including Boolean satisfiability (SAT), which was described above. This potential application makes AQC a very interesting method, because NP-Complete problems have to date not been solved efficiently. Three important quantum algorithms are Grover's database search algorithm; the Deutsch-Josza algorithm; and Shor's algorithm for prime factorisation. All were first described in relation to the circuit model, and all have had AQC alternatives proposed. Indeed, AQC and the circuit model have been shown to be equivalent in terms of computational complexity, meaning that either model can be implemented by the other in a polynomial number of steps. The two major factors contributing to the speed of AQC are: the degree of entanglement during evolution; and the energy gap between the ground and first excited states. The latter of these has prompted variants of AQC that provide speedups: local AQC which varies the evolution speed as the energy gap changes; and partial AQC which only ensures adiabatic evolution through part of the evolution. However, it is unclear whether partial AQC provides any speedup, and local AQC is not robust to decoherence as global AQC has shown to be.

I conclude that AQC is a useful method which has some advantages over the circuit model: the ease of algorithm development for those with no computer science background but quantum physics knowledge; inherent robustness against decoherence; and the potential shown in solving NP-Complete problems. Efficient solutions to NP-Complete problems are still unlikely, but should not be ruled out simply because they have yet to be found. The manufacturers of the D-Wave commercial quantum computer claim that it uses AQC, which if proven correct would be a boost for the potential of AQC as a method. As yet, however, there are open questions around both the quantum nature of the D-Wave, and whether it provides any speedup over classical computers. Future analysis must also be careful to scrutinise the affiliations of AQC research, as D-Wave Systems have a vested interest in proving its efficacy. Because of the computational equivalence of AQC and the circuit model, future research ought to focus on the unique properties of AQC. The key areas in need of more exploration are optimisations of AQC algorithms (through the exploitation of entanglement, and through AQC variants such as local and partial AQC); and error suppression and correction techniques.

References

Aaronson, S. (2008a) 'The Limits of Quantum', *Scientific American*, vol. 298, no. 3, pp. 62-69 [Online]. Available at `http://www.cs.virginia.edu/~robins/The_Limits_of_Quantum_Computers.pdf`

Aaronson, S. (2008b) 'Desultory D-Wave', *Technology review*, vol. 111, no. 3, pp. 11-11 [Online]

Aharonov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S. and Regev, O. (2008) 'Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation', *SIAM Review*, vol. 50, no. 4, pp. 755-787 [Online]. DOI: 10.1137/080734479

Ahrensmeier, D. (2006) 'Entanglement and adiabatic quantum computation', *Canadian Journal of Physics*, vol. 84, no. 6-7, pp. 645-651 [Online]. DOI: 10.1139/P06-033

Altshuler, B., Krovi, H. and Rolandb, J. (2010) 'Anderson localization makes adiabatic quantum optimization fail', *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 28, pp. 12446-12450 [Online]. DOI: 10.1073/pnas.1002116107

Amin, M., Averin, D.V. and Nesteroff, J.A. (2009) 'Decoherence in adiabatic quantum computation', *Physical Review A*, vol. 79, no. 2 [Online]. DOI: 10.1103/PhysRevA.79.022107

Bhattacharya, N., van Linden van den Heuvell, H.B. and Spreeuw, R.J.C. (2002) 'Implementation of quantum search algorithm using classical Fourier optics', *Physical Review Letters*, vol. 88, no. 13 [Online]. DOI: 10.1103/PhysRevLett.88.137901

Biham, E., Brassard, G., Kenigsberg, D. and Mor, T. (2004) 'Quantum computing without entanglement', *Theoretical Computer Science*, vol. 320, no. 1, pp. 15-33 [Online]. DOI: 10.1016/j.tcs.2004.03.041

Boixo, S., Rønnow, T.F., Isakov, S.V., Wang, Z., Wecker, D., Lidar, D.A., Martinis, J.M. and Troyer, M. (2014) 'Evidence for quantum annealing with more than one hundred qubits', *Nature Physics*, vol. 10, no. 3, pp. 218-224 [Online]. DOI: 10.1038/nphys2900

Childs, A.M., Farhi, E. and Preskill, J. (2002) 'Robustness of adiabatic quantum computation', *Physical Review A*, vol. 65, no. 1 [Online]. DOI: 10.1103/PhysRevA.65.012322

Choi, V. (2011) 'Different adiabatic quantum optimization algorithms for the NP-complete exact cover problem', *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 7, pp. E19-E20 [Online]. DOI: 10.1073/pnas.1018310108

Das, S., Kobes, R. and Kunstatter, G. (2002) 'Adiabatic quantum computation and Deutsch's algorithm', *Physical Review A*, vol. 65, no. 6 [Online]. DOI: 10.1103/PhysRevA.65.062310

Deutsch, D. and Jozsa, R. (1992) 'Rapid Solution of Problems by Quantum Computation', *Proceedings: Mathematical and Physical Sciences*, no. 1907, pp. 553 [Online]. DOI: 10.2307/52182

Dickson, N.G. and Amin, M. (2011) 'Does Adiabatic Quantum Optimization Fail for NP-Complete Problems?', *Physical Review Letters*, vol. 106, no. 5 [Online]. DOI: 10.1103/PhysRevLett.106.050502

Ekert, A. and Kay, A. (2014) 'Entanglement and entangling states' [Online]. Available at: `http://www.arturekert.org/sandvox/note3-2.pdf` (Accessed 6 July 2014)

Farhi, E., Goldstone, J., Gutmann, S. and Sipser, M. (2000) 'Quantum computation by adiabatic evolution', *arXiv preprint*

quant-ph/0001106 [Online]. Available at: `http://arxiv.org/abs/quant-ph/0001106`

Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A. and Preda, D. (2001) 'A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an Np-Complete Problem', *Science*, vol. 292, no. 5516, pp. 472-476 [Online]. Available at: Academic Search Complete

Garnerone, S., Zanardi, P. and Lidar, D.A. (2012) 'Adiabatic Quantum Algorithm for Search Engine Ranking', *Physical Review Letters*, vol. 108, no. 23, pp. 1-6 [Online]. DOI: 10.1103/PhysRevLett.108.230506

Grover, L.K. (1996) 'A fast quantum mechanical algorithm for database search', *arXiv preprint quant-ph/9605043* [Online]. Available at: `http://arxiv.org/abs/quant-ph/9605043`

Hsu, J. (2013) 'D-Wave's year of computing dangerously [News]', *IEEE Spectrum*, vol. 50, no. 12, pp. 11-13 [Online]. DOI: 10.1109/MSPEC.2013.6676982

Johnson, M.W., Amin, M.H.S., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P., Chapple, E.M., Enderud, C., Hilton, J.P., Karimi, K., Ladizinsky, E., Ladizinsky, N., Oh, T., Perminov, I., Rich, C. and Thom, M.C. (2011) 'Quantum annealing with manufactured spins', *Nature*, vol. 473, no. 7346, pp. 194-198 [Online]. DOI: 10.1038/nature10012

Lidar, D.A. (2008) 'Towards fault tolerant adiabatic quantum computation', *Physical Review Letters*, vol. 100, no. 16 [Online]. DOI: 10.1103/PhysRevLett.100.160506

Lloyd, S. (2008) 'Riding D-Wave', *Technology review*, vol. 111, no. 3, pp. 78-80 [Online]

Mizel, A., Lidar, D.A. and Mitchell, M. (2007) 'Simple proof of equivalence between adiabatic quantum computation and the circuit model', *Physical Review Letters*, vol. 88, no. 7 [Online]. DOI: 10.1103/PhysRevLett.99.070502

Nielsen, M.A. and Chuang, I.L. (2010) *Quantum computation and quantum information* [Online], Cambridge university press. Available at: `http://www.johnboccio.com/research/quantum/notes/QC10th.pdf` (Accessed 6 July 2014)

Passante, G., Choy, K., Ahrensmeier, D., Carrington, M.E., Fugleberg, T., Kobes, R. and Kunstatter, G. (2007) 'The dynamics of entanglement in the adiabatic search and Deutsch algorithms', *Canadian Journal of Physics*, vol. 85, no. 10, pp. 995-1021 [Online]. DOI: 10.1139/P07-084

Peng, X., Liao, Z., Xu, N., Qin, G., Zhou, X., Suter, D. and Du, J. (2008) 'Quantum adiabatic algorithm for factorization and its experimental implementation', *Physical Review Letters*, vol. 101, no. 22, pp. 220405-220405 [Online]. DOI: 10.1103/PhysRevLett.101.220405

Roland, J. and Cerf, N.J. (2002) 'Quantum search by local adiabatic evolution', *Physical Review A*, vol. 65, no. 4 [Online]. DOI: 10.1103/PhysRevA.65.042308

Rønnow, T.F., Wang, Z., Job, J., Boixo, S., Isakov, S.V., Wecker, D., Martinis, J.M., Lidar, D.A. and Troyer, M. (2014) 'Defining and detecting quantum speedup', *Science*, vol. 345, no. 6195, pp. 420-424 [Online]. DOI: 10.1126/science.1252319

Shor, P.W. (1999) 'Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer', *SIAM Review*, vol. 41, no. 2, pp. 303-332 [Online]. DOI: 10.1137/S0097539795293172

Sun, J., Lu, S. and Liu, F. (2013) 'Partial adiabatic quantum search algorithm and its extensions', *Quantum Information Processing*, vol. 12, no. 8, pp. 2689-2699 [Online]. DOI: 10.1007/s11128-013-0557-1

Wen, J. and Qiu, D. (2008) 'Entanglement in Adiabatic Quantum Searching Algorithms', *International Journal of Quantum Information*, vol. 6, no. 5, pp. 997-1009 [Online]. DOI: 10.1142/S0219749908004249

Young, K.C., Sarovar, M. and Blume-Kohout, R. (2013) 'Error Suppression and Error Correction in Adiabatic Quantum Computation: Techniques and Challenges', *Physical Review X*, vol. 3, no. 4 [Online]. DOI: 10.1103/PhysRevX.3.041013

Zhang, Y.Y. and Lu, S.F. (2010) 'Quantum search by partial adiabatic evolution', *Physical Review A*, vol. 82, no. 3 [Online]. DOI: 10.1103/PhysRevA.82.034304

Appendix A

Glossary

Adiabatic quantum computation - The model of quantum computation whereby an initial Hamiltonian is evolved adiabatically to a problem Hamiltonian that encodes the solution to the computational problem.

Bit - The unit of information in classical computing. Abstractly, a bit can hold the value 0 or 1 (equivalently True or False). In practice, a bit is represented by a high or low voltage in a circuit.

Boolean clause - A combination of one or more bits that apply Boolean logic to produce a Boolean value, e.g. $a \vee \neg b$ is True if a is True, or b is False.

Boolean logic gate - In classical computing, a circuit element which transforms one or more bits into an output bit, e.g. an OR gate produces True if either input is True, False otherwise.

Boolean satisfiability problem - An NP-Complete problem which is to assign values to n bits, such that a set of m Boolean clauses involving those bits are all satisfied. See main text for a more complete description.

Circuit model - The traditional model of quantum computing, where qubits are acted upon discretely by a number of quantum gates, arranged into a quantum circuit.

Complexity class - A way of classifying the difficulty (or complexity) of certain problems. A complexity class contains all problems where the best known algorithms to solve them have similar running time. For example, the P complexity class contains all problems that can be solved in polynomial time.

Decision problem - A problem which has a yes/no answer.

D-Wave One - The "first commercial quantum computer" (as claimed by its manufacturer).

Efficient (algorithm) - An efficient algorithm is one whose running time is polynomial in the input size *[see Polynomial (algorithm)]*.

Error correction - The prevention of errors caused by physical influences such as noise in transmission. Error correction techniques aim to reverse any errors that occur, thus retrieving the original data.

False - One of the two possible values for a classical bit *[see also True]*.

Global AQC - The original variant of AQC proposed. The evolution speed is constant, and is related to the minimum energy gap (between the ground and first excited states) throughout the whole evolution.

Initial Hamiltonian - In AQC, the Hamiltonian in whose ground state the system is initially prepared. It is chosen primarily to be easy to produce, and is often an equal combination of possible basis states.

Local AQC - A variant of AQC, where the running speed is increased at times when the energy gap (between the

ground and first excited states) is larger.

NP (complexity class) - The complexity class containing all decision problems which can be verified in polynomial time. This means given a decision problem, and a certificate (i.e. a proof the answer is yes for a given instance), one can verify in polynomial time that the answer is indeed yes. For example, finding a solution to a SAT problem cannot be done in polynomial time, but confirming that a given set of assignments solves the SAT instance is trivial.

NP-Complete (complexity class) - Informally, the 'hardest' of the NP problems. This class is a subset of the NP class, and contains problems that every other NP problem can be reduced to (in a polynomial number of steps). Thus an algorithm to solve a single NP-Complete problem can be used to solve every NP problem.

P (complexity class) - The complexity class containing all decision problems that can be solved in polynomial time. This means given an instance of a decision problem, a yes/no answer can be produced in polynomial time.

Partial AQC - A variant of AQC that applies the adiabatic condition only over a small part of the evolution. Beyond this the evolution is instantaneous. This means partial AQC becomes probabilistic, but is faster.

Polynomial (algorithm) - An algorithm whose running time scales with a polynomial of the input size, n, i.e. its running time scales with n^p for some p.

Problem Hamiltonian - In AQC, the Hamiltonian whose ground state encodes the solution to the problem instance under question. At the end of AQC, the system is in the ground state of the problem Hamiltonian.

Quantum adiabatic theorem - States that if a system is in an eigenstate, and the Hamiltonian is changed slowly enough, it will remain in the corresponding eigenstate of the new Hamiltonian. Specifically, this is applied to the ground state in AQC, where the system is evolved from the ground state of the initial Hamiltonian to the ground state of the problem Hamiltonian.

Quantum circuit - An arrangement of quantum gates which act together to perform a certain desired algorithm.

Quantum gate - Like a Boolean logic gate, but acts on qubits. It takes a number of qubits as input and produces one or many qubits. Differ from Boolean logic gates in that they can act on superpositions of states, and may introduce entanglement.

Qubit - The unit of information in quantum computing. A qubit models an entire quantum state, and may hold the value $|0\rangle$, $|1\rangle$, or a linear combination of both, $\alpha|0\rangle + \beta|1\rangle$. In practice they are represented by any quantum system with two basis states, e.g. the polarization of a photon.

RSA - One of the most popular and widely used encryption systems. Its security relies on the difficulty of generating prime factors of large numbers, hence Shor's algorithm threatens its security.

SAT - *[see Boolean satisfiability problem]*

Sub-exponential (algorithm) - An algorithm which is slower than polynomial time, but faster than exponential time. That is, its running time scales slower than $O(n^p)$ but faster than $O(p^n)$ (where p is an integer and n is the problem size).

Time complexity - Defines how an algorithm scales with input size. For example an $O(n^3)$ algorithm has a running time that scales with the cube of the input size. Formally, an algorithm has time complexity $O(f(n))$ if, for some values of A and n_0, the running time is less than $Af(n)$ for all input sizes $n > n_0$.

True - One of the two possible values for a classical bit *[see also False]*.