AFRL-RI-RS-TR-2020-159

QUANTUM ALGORITHMS AND APPLICATIONS

BOOZE ALLEN HAMILTON INC.

SEPTEMBER 2020

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, N**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2020-159 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
KRISTI T. MEZZANO
Work Unit Manager

/ S /
GREGORY HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* SEPTEMBER 2020	**2. REPORT TYPE** FINAL TECHNICAL REPORT	**3. DATES COVERED** *(From - To)* MAR 2019 - MAR 2020

4. TITLE AND SUBTITLE

Quantum Algorithms and Applications

5a. CONTRACT NUMBER FA8750-19-C-0044

5b. GRANT NUMBER N/A

5c. PROGRAM ELEMENT NUMBER 62788F

6. AUTHOR(S)

Melinda Andrews, JD Dulny, Andrew Glaudell, Taylor Hernandez, Michael Jarret, Vaibhaw Kumar, Emily Morgan, Isabella Martinez, and Casey Tomlin

5d. PROJECT NUMBER CYDT

5e. TASK NUMBER BA

5f. WORK UNIT NUMBER H2

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Booz Allen Hamilton Inc.
8283 Greensboro Dr
Mclean VA 22102-3830

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITQ
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI

11. SPONSOR/MONITOR’S REPORT NUMBER AFRL-RI-RS-TR-2020-159

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA# 88ABW-2020-2703
Date Cleared: 8/31/20

13. SUPPLEMENTARY NOTES

14. ABSTRACT

While it will likely be decades before a fully fault-tolerant quantum computer is engineered, the use of hybrid, quantum-classical protocols promise near-term, perhaps even immediate, applicability in particular cases. The purpose of the research described in this report has been to explore these quantum algorithms with potential, near-term impact. To explore various dimensions of quantum algorithms used for subroutine improvement, the team conducted research on three different sub-topics: quantum walks, obstructions to classical computers, and the quantum linear systems problem (QLSP).

15. SUBJECT TERMS

Quantum algorithms, quantum walks, quantum linear systems

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					KRISTI MEZZANO
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (*Include area code*)**
U	U	U	UU	32	N/A

TABLE OF CONTENTS

LIST OF FIGURES .... *ii*

1 SUMMARY *1*

2 INTRODUCTION *2*

2.1 Introduction to Quantum Walks 2

2.2 Introduction to Quantum/Classical Separation 3

2.3 Introduction to Quantum Linear Systems 4

3 METHODS, ASSUMPTIONS, AND PROCEDURES *6*

3.1 Methods for Quantum Walks 6

3.1.1 Setting. 6

3.2 Methods Classical Separation 7

3.2.1 Related Work. 7

3.2.2 Setting. 8

3.3 Methods for Exploring QLSP 8

3.3.1 Implementation and Exploration of the Subasi et al. Algorithm in QuTiP. 8

3.3.2 Implementation and Exploration of the Subasi et al. Algorithm in Qiskit. 10

3.4 Detailed Implementation of the Lin and Tong Algorithm 12

3. 4. 1 Block-encoding H1 13

3.4.2 Singular Value Transformation 15

4 RESULTS AND DISCUSSION *17*

4.1 Results of Quantum Walks 17

4.2 Results of Classical Separation 17

4.3 Results from Exploring QLSP 18

4.3.1 Results from Implementation and Exploration of the Subasi et al. Algorithm in QuTiP. 18

4.3.2 Results from Implementation and Exploration of the Subasi et al. Algorithm in Qiskit. 20

4.3.3 Results from the Detailed Implementation of Lin and Tong Algorithm. 21

5 CONCLUSIONS *21*

5.1 Conclusion from Quantum Walks 21

5.2 Conclusion from Classical Separation 21

5.3 Conclusions from QLSP 22

6 REFERENCES *23*

APPENDIX A: NOTE ON SOURCE CODE *26*

LIST OF ACRONYMS *27*

LIST OF FIGURES

Figure 1: Evolution From (a) the Original Bipartite Graph to (b) the Augmented Graph ... 6

Figure 2: Circuit of Hadamard Test ... 11

Figure 3: Creating U_H from Block-encodings U_b and U_A ... 13

Figure 4: (1,1,0)-Block-encoding of Q_b ... 13

Figure 5: Block-encoding of Controlled Q_b ... 14

Figure 6: Block-encoding of A ... 14

Figure 7: Block-encoding of $\boldsymbol{X \otimes A}$ ... 15

Figure 8: Singular Value Transformation of a Block-encoded Matrix ... 15

Figure 9: Comparison of RM and Algorithm Variations for Several Choices of Number of Repetitions ... 19

Figure 10: Performance of Individual RM Runs Compared to the Average of All Runs.... 20

1 SUMMARY

The recent demonstration of quantum supremacy has placed us squarely within the era of noisy, intermediate-scale quantum (NISQ) devices [1]. Nonetheless, it remains unclear which problems are most amenable to quantum speedups. In this report, we describe the Booz Allen Hamilton's quantum research team's progress towards distinguishing where a quantum computer can confer an advantage over classical computation. We explore where quantum algorithms can expedite important classical methods through three fundamental questions:

(1) Which common subroutines of important algorithms can be made more efficient with quantum computation?
(2) When can one avoid these common subroutines altogether?
(3) When is our knowledge of classical algorithms simply too limited and better classical subroutines exist?

We investigate these questions by exploring the following three research topics.

(a) First, we examine *quantum walk* algorithms for finding marked states within local query models.
(b) Second, we research known *obstructions to classical computers* using Monte Carlo techniques to simulate various quantum Hamiltonians.
(c) Finally, we explore near-term implementable quantum algorithms for the *quantum linear systems problem* (QLSP).

Our approach to items (a) and (b) proceeds as a pure theoretical study, whereas we approach item (c) theoretically and computationally. For (a), we have found ways that one might advance quantum walk algorithms to explore arbitrary, unknown spaces using purely local queries and also have adapted very new techniques to expedite solving Boolean satisfiability (and optimization) problems. For (b), we have introduced a new classical, quasipolynomial-time algorithm for handling obstructions that arise due to exponential separations between L^1 and L^2 norms of particular quantum states. For (c), we are currently developing code to implement QLSP in Qiskit, as well as investigating the possibility of improving the asymptotic scaling of the recent advancement by Lin and Tong [2].

Each of these research efforts will be introduced separately, with designated sections for their methods and results. Conclusions of the three efforts, and their overarching themes, will be presented together. Additionally, through the performance of this contract, Booz Allen has provided extensive support to experimental groups at AFRL, including the LaHaye superconducting qubit lab and the Soderberg-Tabakov Ytterbium (Yb) ion trapping effort.

2 INTRODUCTION

Recent advances in the field of quantum computing, which seeks to leverage the unique resources provided by quantum mechanics to perform information processing tasks, have propelled this field into the limelight. It has attracted global attention with enticing prospects, such as advancing personalized medicine, training the next generation of artificial intelligence, and upending contemporary cryptosystems. While it will likely be decades before a fully fault-tolerant quantum computer is engineered, the use of hybrid, quantum-classical protocols promise near-term, perhaps even immediate, applicability in particular cases. The purpose of the research described in this report has been to explore these quantum algorithms with potential, near-term impact.

To explore various dimensions of quantum algorithms used for subroutine improvement, the team conducted research on four different sub-topics: quantum walks, obstructions to classical computers, and the quantum linear system problem (QLSP). The latter of these efforts will be presented in two parts: the theoretical study of scaling improvements to QLSP and the Qiskit implementation of QLSP.

Each of these research efforts will be introduced separately, with designated sections for their methods and results. Conclusions of the four efforts, and their overarching themes, will be presented together.

2.1 Introduction to Quantum Walks

While there exist numerous quantum algorithms that provide speedups over classical methods [3], there are only a few key quantum processes from which these advantages derive. One such ubiquitous process is the quantum walk [4,5]. A quantum walk is a quantum analog of a random walk which explores a space, typically conceived of as a graph, faster than the corresponding classical walk. Though these quantum processes usually only enjoy a quadratic advantage over their classical counterparts, multiple classes of examples exist in which the advantage can become exponential. In one famous example, a continuous time quantum walk is able to explore a graph with properties that severely limit classical random walks [6].

Presently, although there exists a great deal of work on quantum processes on graphs [7,8], there is scant understanding of "local" query models [9]. Let $G = (V, E)$ be a graph with vertex set V and edge set E. The local query model assumes access to an oracle that, given input $v \in V$ returns the set of vertices adjacent to v (and subsequently the degree of v). This local query model is more natural for analogies to random walks where, although the existence of a particular vertex may be known, the full sets V, E are not as readily accessible. In this setting, one can perform a quantum walk on the graph G with no *a priori* knowledge of V, E.

The restriction from global information to local information can drastically change both the time and space complexity of algorithms used to explore such spaces. One striking application is to Boolean satisfiability (SAT), where the standard Davis-Putnam-Logemann-Loveland (DPLL) algorithm builds a tree data structure on-the-fly [10,11]. In this context, and in more modern algorithms built on DPLL, the space searched is never larger than necessary to solve the problem at hand. For "easy" problems, this can exponentially reduce the space to be searched. The difference in power between local and global approaches for quantum walks was realized by Montanaro in [12], where he provided a local algorithm for solving constraint satisfaction problems (a generalization of SAT). His algorithm explores a backtracking tree structure, like that used in DPLL, and achieves scaling $(T^{1/2}n^{3/2})$ where T is the size of the full backtracking tree of all partial solutions, and n is its depth. This can be quite faster than DPLL which has a worst-case runtime of (T).

Unfortunately, the full backtracking tree is often much larger than the tree constructed by DPLL, which ceases to construct the tree after a solution is discovered. Noticing that classical heuristics can sometimes run faster than Montanaro's algorithm if the actual tree size $\hat{T}$ explored classically is much less than T, Ambainis and Kokainis (which we will refer to below as AK) provided an algorithm that estimates $\hat{T}$ and solves both problems in time $\mathcal{O}(\hat{T}^{1/2}n^{3/2})$ [13].

Although an improvement, this difference comes at the cost of abandoning the original intuition (in terms of electrical networks) and its elegant mathematical foundations [14]. Recent work by Jarret (now one of the contributing Booz Allen team members) and Wan in [15] showed that this intuition can be retained with some advantage over [12]. By providing a routine capable of using local operations to estimate the effective resistance $R < n$ of the tree (where the root is treated as a source and all k solutions are treated as sinks), an algorithm is obtained that runs in time $(\sqrt{TR})$ to decide the existence of a solution and $(\sqrt{TR}\log^4(kn))$ to return the assignment [5]. Not only can R be significantly smaller than n, but this result appears somewhat more fundamental than that of [13]. Indeed, in [16], Piddock adapted the approach of Jarret and Wan to prove a similar quantum speedup on general graphs.

Our ongoing work (described below) aims to further improve upon AK using the effective resistance estimate portion of the Jarret and Wan/Piddock (JWP) approach, which provide for potentially optimal backtracking algorithms. We are also investigating adapting the results of AK to more general graphs, much like Piddock did for the Jarret and Wan approach. This would provide faster graph search algorithms on graphs of unknown size and structure.

2.2 Introduction to Quantum/Classical Separation

While studying improved quantum walks is valuable since they provide known advantages over classical walks [4,5], typically they are limited to quadratic improvements, and it is likewise crucial to determine when a quantum algorithm or device is capable of obtaining a *significant* advantage. Recently, Ewin Tang achieved notoriety by showing that a classical computer can tackle the problem of recommendation systems nearly as efficiently as known quantum machine learning algorithms [17]. In this work, Tang pursued the question of finding *some* classical competitor for a known quantum algorithm, but often the question of what separates quantum from classical presents itself more concretely.

Monte Carlo methods are widely employed to investigate quantum systems, and folklore theorems posit that there are particular cases in which they are genuinely efficient. Properties of quantum systems which can be *efficiently* modeled by classical techniques are referred to as *simulatable*. If a property is simulatable, then an algorithm leveraging this property cannot provide a substantial quantum advantage. That is, if one can efficiently simulate the quantum system with a classical computer, then that simulation in itself provides a classical competitor for the quantum algorithm it simulates. In turn, this proves a barrier to obtaining a quantum advantage out of both algorithms based on the quantum system, and quantum computers based on that system's architecture.

As an example, Bravyi and Gosset demonstrated in [18] that the partition function of certain ferromagnetic models is efficiently simulatable. Also, Crosson and Slezak very recently demonstrated that quantum Monte Carlo methods are efficient for simulating the partition function of high-temperature systems [19]. Both of these results provide barriers to quantum advantage in NISQ (or general) quantum devices. Of course, we do not expect all quantum systems to be simulatable, and as early as 2013 Hastings demonstrated the existence of "obstructions" to classically simulating adiabatic processes with particular Monte Carlo methods [20].

In [21], Jarret and coauthors built on Hastings' work, creating an obstruction based on an exponential separation between the L^1 and L^2 norms of quantum systems that have hidden, but otherwise strong symmetries. For a system of n qubits, as long as a quantum state is reasonably well distributed, the inequality $||\psi||_1 \leq \sqrt{2^n}||\psi||_2$ is near-saturated. This separation causes crucial difficulties when sampling from the classical probability distribution defined by ψ. Though contrived and easily bested by alternative classical algorithms, additional work extending this example's strategy demonstrated more realistic obstructions providing compelling evidence against classical simulations [22,23].

A natural question thus presents itself: Does the exponential separation in norms actually provide a fundamental obstruction, or have we neglected to explore classical algorithms that exploit hidden symmetries?

In the present work we investigate whether a symmetry existence promise is sufficient to construct a classical algorithm that competes with its quantum counterpart. The strength of a symmetry can be measured in terms of the size of an appropriate automorphism group, and large norm-divergences correspond to large automorphism groups. Specifically, we develop a classical algorithm that can sample from the ground state of a Hamiltonian with hidden symmetry and large automorphism group in quasipolynomial time, thereby overcoming the exponentially diverging norms obstruction. Whether this algorithm is *necessarily* quasipolynomial may reveal if there exists a fundamental separation provided by this obstruction, even if less problematic than previously expected.

2.3 Introduction to Quantum Linear Systems

As mentioned above, Tang's 2019 result [17] dashed hopes that an exponential separation may exist between quantum and classical recommendation systems by providing an improved classical algorithm in relevant cases. Though classical, Tang's algorithm was inspired by an earlier quantum algorithm [24], which has helped draw attention to quantum-inspired algorithms and made a stronger case for the utility of studying quantum machine learning algorithms generally. Similarly, Crosson and Harrow found a quantum-inspired classical algorithm which can be exponentially faster than classical simulated annealing [25]. As simulated annealing and its variants feature prominently as optimization subroutines in many machine learning applications, the approach in [25] could accelerate such applications.

These examples suggest that studying quantum machine learning (QML) serves both as a use case for quantum computers in applications for artificial intelligence, and to advance classical methods. Understanding the quantum algorithms which cannot be simulated classically provides insight into when QML routines might confer an advantage over their classical counterparts. In particular, subroutines of existing algorithms, such as eigensolvers, are called frequently in many machine learning pipelines. Improving these subroutines necessarily expedites both quantum and classical algorithms that call them. These themes guided our theoretical research into QLSP.

Recently, our team member Jarret proposed an algorithm for obtaining optimal runtimes from an adiabatic protocol [26]. The method, although initially presented as a means of controlling adiabatic evolutions, relies on a classical strategy. At around the same time, Subasi *et al.* [27] proposed a quantum algorithm for solving linear systems of equations, based on adiabatic optimization. Notably, Subasi *et al.*'s algorithm is suboptimal in some complexity parameters (the condition number of the matrix defining the linear system and the state preparation error) and uses a "randomization method" to determine variation rates. Standard usage of the randomization method requires multiple runs, which are then averaged together to obtain a final result. This adds runtime and results in a mixed state, rather than a pure state, thus limiting its use as a subroutine in a larger quantum algorithm.

Our initial goal was to adapt the methods of [26] to bring Subasi's protocol closer to optimal in the condition number. However, [26] requires the adiabatic Hamiltonian to be "nearly" stoquastic, which is not easy to achieve for general linear systems. We investigated ways in which non-stoquastic Hamiltonians may be "stoquasticized," but these methods introduce other undesirable features (e.g. ground state evolution becomes excited state evolution, and eigenvalue crossings may be introduced). Instead, we shifted attention to the error scaling.

Given the usefulness of adiabatic inspiration in the original algorithm, we identified the "adiabatic expansion" method [28] as a promising route to better scaling due to its ability to exponentially suppress error. In general, this improved error scaling comes at the cost of poorer scaling in the minimum eigenvalue gap, which notably may be preferred in some situations. Though we found slightly improved gap dependence in the explicit example of adiabatic Grover search, asymptotic behavior remained unchanged. Nevertheless, as discussed further in 2.3 Methods for Exploring QLSP, this instructive example provides some intuition about the nature of the adiabatic expansion method compared with "vanilla" adiabatic protocols.

The original workstream again shifted in October 2019 when, building upon the work of Subasi *et al.*, Lin and Tong introduced a nearly optimal algorithm for solving QLSP [2]. Their algorithm enjoys optimal dependence on the desired level of error ϵ, and is only sub-optimal by log factors in the condition number. This condition number scaling is achieved by means of an adiabatic state preparation step, which is then followed by an eigenstate filtering step making use of quantum signal processing that improves the error scaling. However, this algorithm introduces an additional parameter, the degree ℓ of the filtering polynomial used to perform eigenstate filtering, which grows with both condition number and desired accuracy of the solution. The filtering polynomial is implemented by means of a set of phase factors. The classical subroutine used by Lin and Tong to calculate these phase factors requires $\mathcal{O}(\ell \log(\ell/\epsilon))$ bits of precision, and thus is numerically unstable.

In our current work, we adapt to this new development and aim to refine Lin and Tong's results. Specifically, we are attempting to avoid the issue of numerical instability by obtaining an analytic expression for the phase factors or an improved, numerically stable algorithm for approximating them. This improvement is broadly applicable not only to the linear systems problem, but to any problem which involves eigenstate filtering. The approach taken requires the creation of a new algorithm to determine relevant spectral quantities, and it remains unclear whether the effort will be successful; however, preliminary results (described further below) look promising.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

This research effort theoretically explored the use of quantum walk algorithms for finding marked states within local query models and known obstructions to classical computers using Monte Carlo techniques to simulate various quantum Hamiltonians. Near-term implementable quantum algorithms for the quantum linear systems problem was approached both theoretically and computationally, with methods for implanting QLSP in QuTiP and Qiskit presented in this report.

Each of these research efforts will be described separately, with a detailed implementation of the implementation of Lin's and Tong's algorithm for QLSP [2] presented in its own section.

3.1 Methods for Quantum Walks

Ambainis and Kokainis [13] showed how to construct a quantum algorithm which estimates the size of the tree T within a factor of $1 \ \pm \delta$ for constant δ in $\tilde{O}(\sqrt{nT})$ steps. In [16] Piddock considers the problem of finding a marked vertex in a general weighted graph from an arbitrary starting distribution, using a quantum walk-based algorithm in the framework of Belovs [29], generalizing and improving on the approach of Jarret and Wan [15]. Here, we use the machinery of Ambainis and Kokainis and apply it in the setting of general weighted graphs to construct an algorithm which estimates total weight.

3.1.1 Setting.

Let $\mathcal{G} = (V, E)$ be a graph with edge weights $\{w_e\}_{e\in E}$, total weight $W = \Sigma_{e\in E} w_e$, and an initial probability distribution on V given by $\sigma = \{\sigma_x\}_{x\in V}$. The graph is assumed bipartite with vertices partitioned as $V \ = \ A \sqcup B$, and $\text{supp}(\sigma) \subset A$. When $\mathcal{G}$ is not bipartite, for the purposes of walks, one can instead consider $\mathcal{G}' = (V', E')$ where $V' = V \ \times \{0,1\}$ and $E' = \{\{(x,0),(y,1)\}: (x,y) \in E\}$, which is now bipartite. The bipartite structure supports Szegedy quantum walks [30] and allows the corresponding quantum circuit to be performed using operators defined on their respective subspaces, which are easier to specify than generic walk operators.

The graph $\mathcal{G}$ can be augmented in such a way which allows one to restrict attention to an initial distribution concentrated at a single vertex: $\sigma_x = \delta_{s'x}$ for a vertex $s' \in B$. As can be seen in Figure 1, the original bipartite graph (a) can be augmented (b) so that each $\text{v} \in \text{supp}(\sigma)$ gets an extra edge (with weight σ_v/η, where η is a global parameter) connecting it to a single new vertex s'.

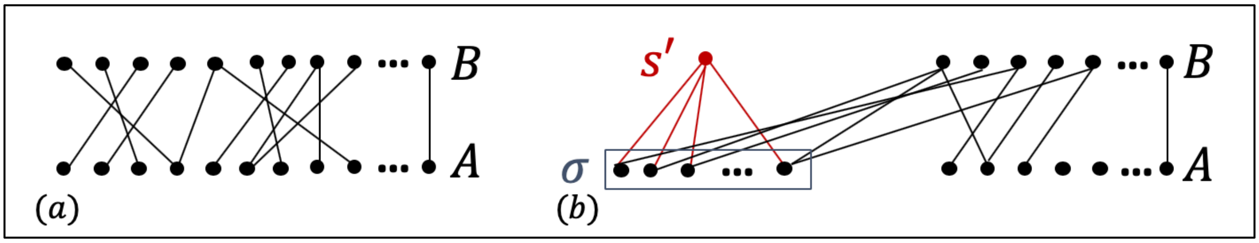

Figure 1: Evolution From (a) the Original Bipartite Graph to (b) the Augmented Graph

From the standard edge-based Hilbert space $\mathcal{H} = span(\{|e\rangle\}_{e\in E})$ one may define orthonormal vertex states

$$|x\rangle := \frac{1}{\sqrt{d_x}}\sum_{y\sim x}\sqrt{w_{xy}}\,|xy\rangle \tag{1}$$

Where

$$d_x := \sum_{y\sim x} w_{xy} \tag{2}$$

and diffusion operators

$$D_x := \begin{cases} \mathbf{1}, & x \in M \cup \{s'\} \\ \mathbf{1} - 2|x\rangle\langle x|, & x \notin M \cup \{s'\} \end{cases} \tag{3}$$

Here M denotes a set of marked vertices, which is not immediately relevant for the present application. The walk operators are then given by $U_A := \bigoplus_{x\in A} D_x$, and $U_B := \bigoplus_{y\in B} D_y$.

The direct sum notation is slightly abusive, but standard: The diffusion operators in each sum are mutually commutative because $\mathcal{G}$ is bipartite and only neighboring vertex states have overlap:

$$\langle x|y\rangle = \frac{1}{\sqrt{d_x d_y}}\sum_{z'\sim x}\sum_{z'\sim x}\sqrt{w_{xz}w_{yz'}}\langle xz|yz'\rangle = \frac{w_{xy}}{\sqrt{d_x d_y}} \tag{4}$$

In this setting we note that U_A is a reflection over the set $\widetilde{\mathcal{H}}_A := span(\{|x\rangle : x \in A\backslash(M \cup \{s\})\})$. Similarly, U_B is a reflection over $\widetilde{\mathcal{H}}_B$, which is defined analogously to $\widetilde{\mathcal{H}}_A$.

Our goal is to use the spectral theorem to bound the output of a phase estimation circuit. This method, inspired by AK's total tree size estimation, requires that we:

a) Verify that the starting state $|\psi_{s'}\rangle$ is orthogonal to all $U_B U_A$-invariant states.
b) Determine bounds on the eigenvalue of $U_B U_A$ that is closest to (a). To do this, we
    i. Exploit a correspondence by Szegedy [30] for which we require the principle eigenvalue of the discriminant matrix associated with the subspaces bases $\{|x\rangle : x \in A\}, \{|y\rangle : y \in B \setminus \{s'\}\}$.
    ii. Construct the fundamental matrix of an absorbing random walk on a related graph, that can be viewed as an electrical network, and as such allows us to bound the resistance between each vertex pair and characterize the largest eigenvalue of the altered discriminant matrix, subsequently bounding the eigenvalues of $U_B U_A$.
c) Verify that there are no obstructions to implementing the circuit locally, as in [16].

3.2 Methods Classical Separation

3.2.1 **Related Work**.

We apply the graph-theoretic characterization of Hamiltonians from [31] to create algorithms that can efficiently sample from symmetric subspaces, contingent upon our ability to solve graph isomorphism. Also, we apply Babai's graph isomorphism algorithm to show that this subroutine can be performed in quasipolynomial time [32]. Finally, we use the Davis-Kahan $\sin\theta$ theorem [33] to generalize to nearly symmetric systems.

3.2.2 **Setting.**

Let H be a k-local Hamiltonian that can be written as an Ising model in the form

$$H = \sum_{|b|\leq k} \alpha_b \otimes_i (I - X_i^{b_i}) + \sum_{|b|\leq k} \beta_b \otimes_i Y_i^{b_i} + \sum_{|b|\leq k} \kappa_b \otimes_i Z_i^{b_i} \tag{5}$$

Above, b is a bitstring with Hamming weight $|b|$. We let the set $\mathrm{K} = \left\{\otimes_i X_i^{b_i} | \alpha_b \neq 0 \right\}$ form the edge-generating set of an Abelian graph $G = (V, E)$. Additionally, we impose stoquasticity, or that $|\beta_b| \leq \alpha_b \ \forall b$. Since in our algorithms the $\sum_{|b|\leq k} \beta_b \otimes_i Y_i^{b_i}$ terms ultimately add additional distinguishing information that eases our task, we ultimately assume that $\beta_b = 0$.

To each vertex $v \in V$ we assign a color $\sum_{|b|\leq k} \kappa_b \langle v| \otimes_i Z_i^{b_i} |v\rangle$. For the obstructions of the type mentioned above to exist, the color-preserving automorphism group of G must be large.

3.3 Methods for Exploring QLSP

Linear systems of equations ($Ax = b$ where A is a matrix and x and b are vectors) are ubiquitous. Thus, quantum algorithms developed to solve such systems are found as subroutines of many other algorithms, including in machine learning routines [34], network resistance [35], and in the analysis of classical Markov chains [36]. Under certain assumptions, one finds an exponential speedup over the best known classical algorithms [27]. However, the phase estimation and amplitude amplification routines used therein substantially increase the number of ancillary qubits required.

Recently, an adiabatic-inspired quantum linear systems algorithm was proposed by Subasi *et al.* [27]. The adiabatic schedule interpolates between a Hamiltonian H_o which has $|b\rangle$ encoded as the lowest eigenstate, a Hamiltonian H_1 whose lowest eigenstate encodes the solution to the problem $|x\rangle$.

In contrast to HHL, Subasi *et al.*'s algorithm uses significantly fewer ancillary qubits. Moreover, it is easy to conceptualize, inspiring the development of schemes which can potentially be implemented on near-term quantum devices. Already a few works have appeared that provide scaling improvements over [27,37,38].

Our research, some ongoing, attempts to build upon this work in different ways:

a) Implementing a version of the Subasi *et al.* algorithm in QuTiP, detailed in Section 3.3.1
b) Implementing a version of the Subasi *et al.* algorithm in Qiskit, detailed in Section 3.3.2
c) Implementing and improving the numerical stability of a version of the Lin and Tong algorithm in QuTiP, detailed in Section 3.4

3.3.1 Implementation and Exploration of the Subasi et al. Algorithm in QuTiP.

The Subasi *et al.* algorithm, while suboptimal in some complexity parameters, is a prime candidate for near-term implementation because (unlike HHL or the nearly-optimal Lin and Tong algorithm, described below in 3.4) it does not rely on large ancillary systems. Implementing this adiabatic-inspired algorithm on a gate-based quantum computer requires Hamiltonian simulation, which can increase the computational complexity and/or re-introduce large ancillas. We chose to initially investigate the numerical performance using QuTiP, which is designed for simulating Hamiltonian dynamics using classical differential equation solvers.

We build upon our results from this implementation in order to implement the same algorithm in Qiskit, using Hamiltonian simulation to transform the adiabatic evolution to gate-based evolution (see Section 3.3.2). The QuTiP implementation allows us to explore and streamline the structure of the Subasi *et al.* algorithm but cannot be run on a gate-based quantum computer. The Qiskit implementation explores the additional challenges that arise when implementing this algorithm on near-term devices.

The distinctive features of the Subasi *et al.* algorithm are as follows:

1. The choice of Hamiltonian family $H(s) \equiv A(s)\, P_b^{\perp}\, A(s)$, where $A(s) \equiv (1-s)Z \otimes I + sX \otimes A$ and $P_b^{\perp} \equiv I - |+,b\rangle\langle +,b|$. By stepping through $s \in [0,1]$ slowly enough to ensure adiabaticity, this evolves the initial ground state $|-,b\rangle$ at $s = 0$ to the desired final state $|+,x\rangle$ at $s = 1$. The single ancilla qubit ensures that $A(s)$ remains non-singular throughout the evolution.
2. The choice of schedule $s(v)$ for evenly-spaced v. The chosen Hamiltonian family has a monotonically decreasing gap between the ground state and the first excited state, so choosing to take smaller steps in s as s approaches 1 minimizes the chances of adiabatic transition occurring.
3. Usage of the randomization method (RM). At each step the system is evolved under the Hamiltonian $H_i = H(s(v_i))$ for a time Δt_i chosen randomly from $[0, \frac{2\pi}{\text{spectral gap}}]$. The entire evolution is run multiple times and averaged to obtain an output density matrix approximating the target state.

We and others [38] suspect that the value of the Subasi *et al.* algorithm lies entirely in the first two points, and RM adds little to no benefit other than easing analysis. Indeed, one can show that in terms of quantum state fidelity, RM is only as good as its best individual run.

In order to explore the value of RM, we implemented the Subasi *et al.* algorithm and several variations using QuTiP. An overview of this implementation is included here; for further details see delivered supplementary information.

a) Generate a linear system $A\vec{x} = \vec{b}$ to solve. The project code randomly generates a $2^n \times 2^n$ Hermitian matrix A that is sparse, has norm 1, and has a user-specified condition number. Similarly, $\vec{b}$ is a randomly generated sparse vector of length 2^n.
b) Produce a solution strategy $H(t)$ according to the chosen algorithm variation. The resulting time-dependent Hamiltonian is a stepwise constant function of time. The total runtime depends on the choice of algorithm variation and on random step length selection. The algorithm variations we considered are:
    i. Unmodified RM with n_{rep} repetitions.
    ii. Match the average total RM runtime: a single long run with deterministic time steps $\Delta t_i = \frac{n_{rep}\pi}{\text{spectral gap}}$.
    iii. Match the maximum single-run RM runtime: $\Delta t_i = \max\limits_{\text{RM runs}} \Delta t_{i,RM}$.
    iv. A single run of RM.
c) Evolve the state under $H(t)$ using the QuTiP master equation solver.

d) Investigate how the chosen variation impacts the distance of the output state from the target state.

3.3.2 **Implementation and Exploration of the Subasi et al. Algorithm in Qiskit**.

The quantum linear system problem is the analog of a classical linear system problem, in which one aims to, given a matrix A and state $|b\rangle$, prepare a state $|x\rangle$ such that $A|x\rangle = |b\rangle$**.** Here A $\in$ $\mathrm{C}^{\mathrm{NXN}}$, $|b\rangle \in \mathrm{C}^{\mathrm{N}}$, and $N = 2^n$.

This task can be framed as setting up an appropriate adiabatic evolution, with respect to some time-dependent Hamiltonian:

$$H(s) = \big(1 - f(s)\big)H_o + f(s)H_1 \tag{6}$$

where $f(s) \in [0,1]$ when $s \in [0,1]$. A popular choice is $f(s) = s$. We consider A to be Hermitian, positive definite, with $\|A\|_2 = 1$, and $|b\rangle$ is assumed to be a unit vector.

QLSP has

$$H_o = \sigma_x \otimes Q_b \tag{7}$$

where

$$Q_b = I_N - |b\rangle\langle b| \tag{8}$$

and

$$H_1 = \sigma_+ \otimes AQ_b + \sigma_- \otimes Q_bA \tag{9}$$

where

$$\sigma_\pm = \frac{1}{2}\big(\sigma_x \pm i\sigma_y\big) \tag{10}$$

The time evolution of $H(s)$ is governed by Schrodinger equation

$$\frac{1}{T} i\partial_s|\psi_T(s)\rangle = H(s)|\psi_T(s)\rangle,\ \ |\psi_T(0)\rangle = |0, b\rangle \tag{11}$$

where $0 \leq s \leq 1$ and T is the evolution time.

Here we provide a brief overview of the QLSP adaptation to the quantum approximate optimization algorithm (QAOA). Note that one can always divide the evolution range $s \in [0,1]$ into small enough bins Δs. For each bin, $H(s)$ can be considered a constant. Hence, the total time evolution can be decomposed into smaller pieces $e^{-iT\Delta sH(\Delta s)}$ using Trotter formula. QAOA style algorithms use this well-known decomposition approach to carry out the evolution using p steps.

$$|\psi\rangle = e^{-i\beta_p H_o} e^{-i\gamma_p H_1} e^{-i\beta_{p-1} H_o} e^{-i\gamma_{p-1} H_1} \dots\dots e^{-i\beta_1 H_o} e^{-i\gamma_1 H_1}\ |0, b\rangle \tag{12}$$

where $\{\gamma_i, \beta_i\}_{i=1}^{P}$ is the set of rotations employed at each step i.

To implement such a scheme on the short depth circuits available to us on near-term devices, usually p is set equal to 1. Under this situation, one attempts to find optimal angles $\beta_{opt}, \gamma_{opt}$, using a classical algorithm, which minimizes a suitable objective function encoding the proximity of $\psi(\beta, \gamma)$ to the solution $|x\rangle$. For the QLSP problem, different choices are available for the objective function [38]. Our implementation uses $|\langle\psi(\beta, \gamma)|H_1|\psi(\beta, \gamma)\rangle|^2$ as the objective function to minimize which is equivalent to maximizing fidelity $|\langle\psi(\beta, \gamma)|b\rangle|^2$ [38].

Implementation of such a scheme involves four basic components:

a) A circuit for the preparation of state $|b\rangle$

b) Circuits for the two rotations $e^{-i\beta H_o}$ and $e^{-i\gamma H_1}$

c) A circuit for the computation of objective function $|\langle\psi(\beta, \gamma)|H_1|\psi(\beta, \gamma)\rangle|^2$

d) A classical optimization routine to find optimal angles β_{opt} and γ_{opt}

Steps (a), (b), and (d) can be easily implemented using the tools provided in Qiskit and are detailed in the project supplementary information. Circuit implementation of the objective function is somewhat non-trivial and is described in detail here.

A popular algorithm, the Hadamard test [39] is used to compute expectation values of the form $\langle\psi|U|\psi\rangle$ using a quantum circuit, where U is an arbitrary unitary, as represented by the circuit in Figure 2 .

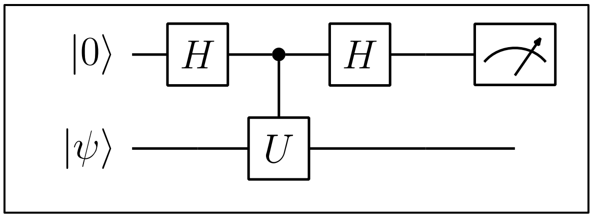

Figure 2: Circuit of Hadamard Test

The top wire is the control wire and the first H gate applied to the state $|0\rangle$ prepares the state $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Then, considering both wires, one has the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi\rangle$.

The controlled unitary implements the transformation:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle(U|\psi\rangle)) \qquad (13)$$

The final Hadamard gate on the control wire, gives:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle\, U|\psi\rangle) \mapsto \frac{1}{2}|0\rangle(U + I)|\psi\rangle + \frac{1}{2}|1\rangle(U - I)|\psi\rangle \qquad (14)$$

Measurements on the control wire produce $|1\rangle$ with P(1), allowing us to calculate $\Re(\langle\psi|U|\psi\rangle) = 1 - 2P(1)$. Similarly, if instead of the initial Hadamard gate we introduce a unitary on the control wire that maps $|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$, the same procedure yields $\Im(\langle\psi|U|\psi\rangle)$. With both real and imaginary components in hand, one can determine $\langle\psi|U|\psi\rangle$.

Our implementation requires us to compute $\langle\psi(\beta,\gamma)|H_1|\psi(\beta,\gamma)\rangle$. We can do so by expressing $H_1 = \sum_{j=1}^{m} c_j \otimes_{i=1}^{n} g_i^k$ where $g^k \in \{I, \sigma_x, \sigma_y, \sigma_z\}$ are Pauli matrices and $c_i \in \mathbb{R}$. We can measure each unitary $U_i = c_i\{\otimes\, g\}$ separately with a Hadamard test. All the expectation values are then summed together

$$\langle\psi(\beta,\gamma)|H_1|\psi(\beta,\gamma)\rangle = \sum_{i=1}^{n} c_i \langle\psi(\beta,\gamma)|U_i|\psi(\beta,\gamma)\rangle \tag{15}$$

to compute

$$\langle\psi(\beta,\gamma)|H_1|\psi(\beta,\gamma)\rangle \tag{16}$$

3.4 Detailed Implementation of the Lin and Tong Algorithm

Lin and Tong's algorithm (LT) [2] combines two independent steps, adiabatic evolution and eigenstate filtering, in such a way as to eliminate the sources of sub-optimal scaling in each approach. A fully adiabatic approach, such as Subasi *et al.*'s, scales sub-optimally with ϵ, a measure of how close the output state is to the correct "target" state. LT introduces eigenstate filtering to improve scaling with ϵ. Although able to strengthen convergence, eigenstate filtering on a generic initial state uses amplitude amplification for the algorithm to succeed with $\mathcal{O}(1)$ probability, which scales poorly in the condition number κ. LT avoids both issues by using an adiabatic process to prepare a state with $\mathcal{O}(1)$ overlap with the target state, then using eigenstate filtering to decrease the error below ϵ.

As a first step towards a full gate-based implementation, and in order to numerically explore the performance of LT, we implement it in QuTiP, the code for which was presented to AFRL with this deliverable. This allows us to leverage QuTiP's state-of-the-art classical time evolution for the adiabatic step of LT.

Generation of problem instances and the adiabatic part are quite similar to the implementation of Subasi *et al.*'s algorithm discussed in Section 3.3.1. Note in particular that the matrix A to be inverted is d-sparse. We focus here on the implementation of the eigenstate filtering phase.

The target state is a null eigenstate of the final Hamiltonian of the adiabatic step, $H_1 = \begin{pmatrix} 0 & AQ_b \\ Q_bA & 0 \end{pmatrix}$ with $Q_b = (I - |b\rangle\langle b|)$. We can filter out the unwanted components of the prepared state with an appropriately chosen polynomial $P(x)$, concentrated near the origin, such that $P(H_1)$ approximates a projector onto the null eigenspace. The optimal filtering polynomial satisfying all necessary conditions is a function of Chebyshev polynomials $T_\ell(x)$:

$$R_\ell(x;\Delta) = \frac{(-1)^\ell}{T_\ell\left(\frac{1+\Delta^2}{1-\Delta^2}\right)} T_\ell\left(2\left(\frac{x^2-\Delta^2}{1-\Delta^2}\right) - 1\right) \tag{17}$$

This is a degree-2ℓ even polynomial such that $R_\ell(H_1;\Delta)$ approximates a projector onto the null eigenspace for large enough ℓ. That is, $\left\|R_\ell\left(\frac{H_1}{d};\frac{1}{\kappa d}\right)-P_0\right\|_2 \le e^{-\sqrt{2}\ell/\kappa \mathrm{d}}$.

Calculating a function of a quantum operator, like $R_\ell(H_1;\Delta)$, can be accomplished by means of singular value transformation [40]. The first step of this procedure is to block-encode the operator as the top-left block of a unitary matrix, which can then be used as a quantum gate.

3.4.1 **Block-encoding $\mathbf{H_1}$.**

An $(m+n)$-qubit unitary matrix U is an (α, m, ϵ)-block-encoding of an n-qubit operator M if the upper-left block of U is ϵ-close to M/α. That is, at the cost of m ancilla qubits and error ϵ, a non-unitary matrix M can be used as an $(m+n)$-qubit quantum gate. Using various techniques to block-encode and combine block-encodings of the parts of H_1, we produce a $(d, n+4, 0)$-block-encoding U_H of the final Hamiltonian.

Note that $H_1 = (\text{controlled } Q_b)(X \otimes A)(\text{controlled } Q_b)$, so U_H can be created from block-encodings U_b of controlled Q_b and U_A of $X \otimes A$ using the rule for block-encoded matrix products as shown in Figure 3.

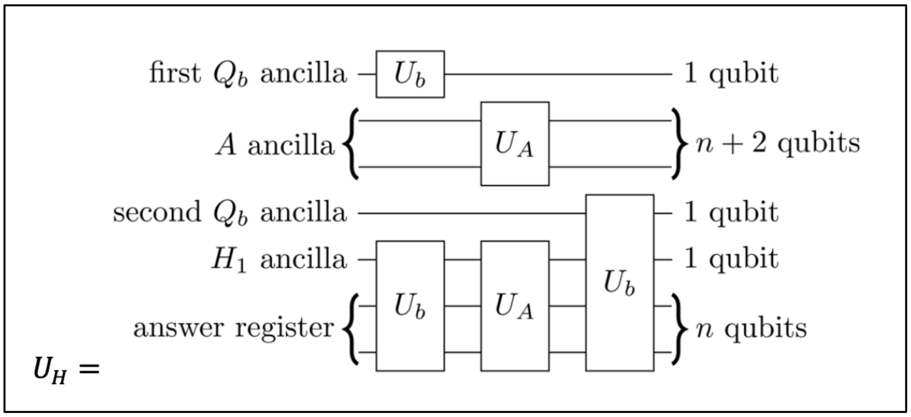

Figure 3: Creating U_H from Block-encodings U_b and U_A

The operator Q_b can be written as a unitary matrix $R_b = I - 2|b\rangle\langle b|$ plus a constant times the identity matrix, so the following circuit gives a (1,1,0)-block-encoding of Q_b as shown in Figure 4.

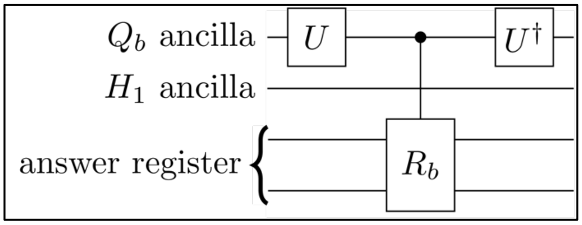

Figure 4: (1,1,0)-Block-encoding of Q_b

This can be modified to block-encode controlled Q_b by controlling all gates in the circuit on the appropriate qubit as shown in Figure 5.

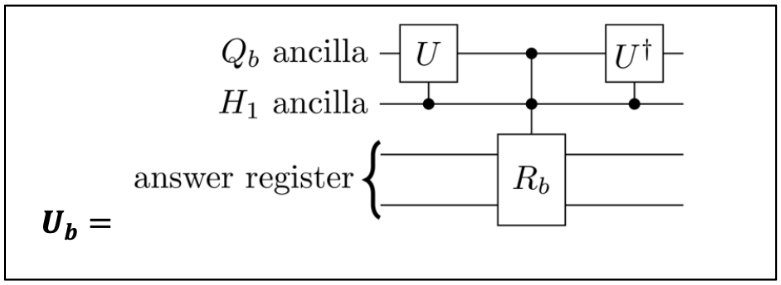

Figure 5: Block-encoding of Controlled Q_b

Since A is a d-sparse matrix, it can be block-encoded using the method for sparse matrices, yielding a $(d, n+2, 0)$-block-encoding with the use of a sparse-access oracle O_R as shown in Figure 6.

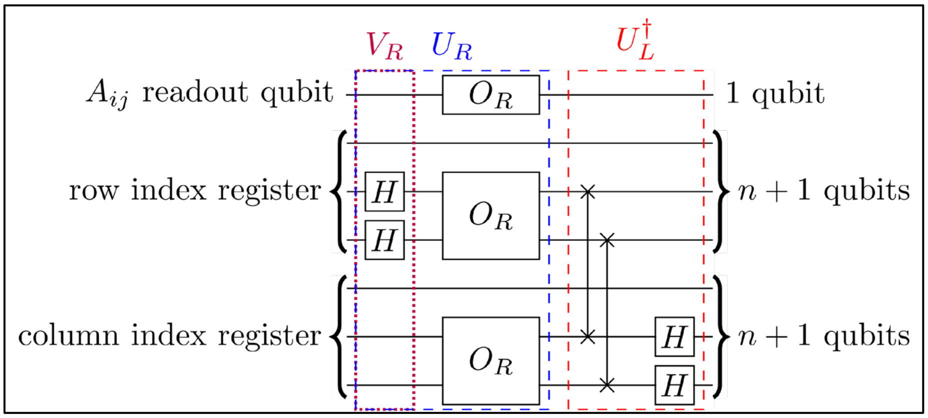

Figure 6: Block-encoding of A

We can modify this circuit to get a block-encoding of $X \otimes A$ by simply acting an X gate on the appropriate qubit, as shown in Figure 7.

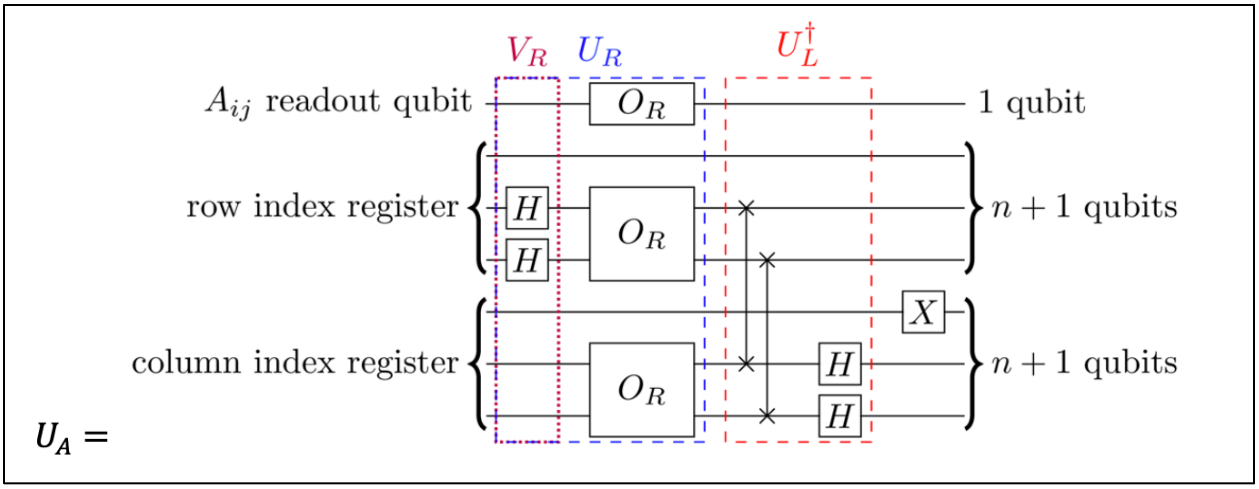

Figure 7: Block-encoding of $X \otimes A$

Thus, at the cost of subnormalizing H_1 by a factor of the density of the matrix and introducing $n + 2$ ancilla qubits, we can calculate the desired filtering polynomial using singular value transformation.

3.4.2 Singular Value Transformation.

We can calculate a degree-d polynomial $P(M)$ of a Hermitian matrix M using d applications of the block-encoded version U_M. It requires finding the appropriate phase factors $\phi_1, \dots, \phi_d$ corresponding to the chosen polynomial with Π a projector onto the zero state of the ancillas as shown in Figure 8.

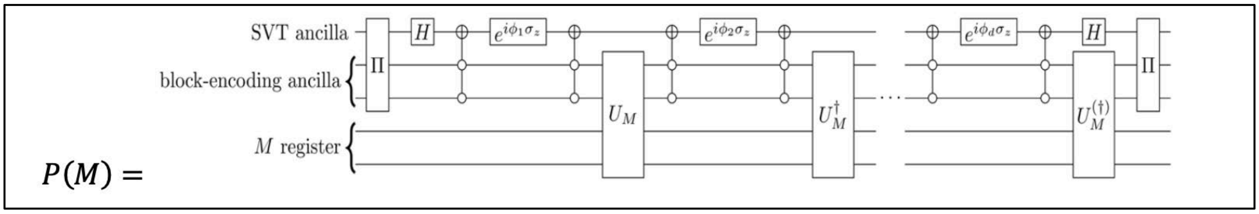

Figure 8: Singular Value Transformation of a Block-encoded Matrix

LT approximate these phase factors with a numerically unstable classical subroutine. We seek an analytic expression for the phase factors resulting from the filtering polynomial $R_\ell(x;\Delta)$. In order to find the exact phase factors, the steps are as follows:

Step 1*:* Find the roots of the degree-4ℓ polynomial $A(x) \equiv 1 - R_\ell^2(x;\Delta)$. $A(x)$ is an even, real polynomial so it suffices to list only the roots with non-negative real and imaginary parts. These can be split into the multisets:

$$S_0 = \{0, 0\}, \tag{18}$$

$$S_{[1,\infty)} = \left\{\sqrt{1+\Delta^2}\right\}, \tag{19}$$

$$S_C = \{a_n + ib_n : n = 1, \dots, \ell - 1\} \tag{20}$$

with

$$a_n^2 = \frac{1}{2}(1+\Delta^2)\sin\left(\frac{n\pi}{2\ell}\right)\left(\sin\left(\frac{n\pi}{2\ell}\right) + \sqrt{1-\left(\frac{1-\Delta^2}{1+\Delta^2}\right)^2\cos^2\left(\frac{n\pi}{2\ell}\right)}\right) \tag{21}$$

and

$$b_n^2 = \frac{1}{2}(1+\Delta^2)\sin\left(\frac{n\pi}{2\ell}\right)\left(-\sin\left(\frac{n\pi}{2\ell}\right) + \sqrt{1-\left(\frac{1-\Delta^2}{1+\Delta^2}\right)^2\cos^2\left(\frac{n\pi}{2\ell}\right)}\right) \tag{22}$$

Step 2: We can use these roots to generate a function $W(x)$ such that $A(x) = W(x)W^*(x)$. Using our results from Equations 18 - 22:

$$W(x) = Kx\left(\Delta x + i\sqrt{1-x^2}\sqrt{1+\Delta^2}\right)\prod_{n=1}^{\ell-1} Q_n(x) \tag{23}$$

where K is a real constant and

$$Q_n(x) \equiv c_n x^2 - (a_n^2 + b_n^2) + i\sqrt{1-x^2}\sqrt{c_n^2 - 1}\,x \tag{24}$$

with

$$c_n \equiv a_n^2 + b_n^2 + \sqrt{(a_n^2 + b_n^2)^2 - 2(a_n^2 - b_n^2) + 1} \tag{25}$$

$W(x)$ has the form $B(x) + i\sqrt{1-x^2}C(x)$, where $B(x)$ and $C(x)$ are polynomials.

Step 3: Define complex polynomials $P(x) = R_{\ell(x;\Delta)} + iB(x)$ and $Q(x) = iC(x)$. $P(x)$ is then a degree-2ℓ even polynomial, $Q(x)$ is a degree-$(2\ell-1)$ odd polynomial, and $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1$.

Step 4: Complex polynomials satisfying the conditions in step 2 allow us to calculate modified phase factors $\tilde{\phi}_0, \dots, \tilde{\phi}_{2\ell}$. For $P^{(0)}(x) = P(x)$ and $Q^{(0)}(x) = Q(x)$, the procedure is:

a. Starting from $m = 0$, define $\tilde{\phi}_{2\ell-m}$ by

$$e^{2i\tilde{\phi}_{2\ell-m}} = \frac{\text{coefficient } p_{2\ell-m} \text{ of highest-order term in } P^{(m)}(x)}{\text{coefficient } q_{2\ell-m-1} \text{ of highest-order term in } Q^{(m)}(x)} \tag{26}$$

b. Eliminate the highest-order terms in $P^{(m)}(x)$ and $Q^{(m)}(x)$ by defining:

$$P^{(m+1)}(x) = e^{-i\tilde{\phi}_{2\ell-m}}\left(xP^{(m)}(x) + e^{2i\tilde{\phi}_{2\ell-m}}(1-x^2)Q^{(m)}(x)\right) \tag{27}$$

and

$$Q^{(m+1)}(x) = e^{-i\tilde{\phi}_{2\ell-m}}\left(e^{2i\tilde{\phi}_{2\ell-m}}xQ^{(m)}(x) - P^{(m)}(x)\right) \tag{28}$$

c. Repeat steps a and b for $m = 0,\ 1, \ldots, 2\ell - 1$.

d. Finally, define $\tilde{\phi}_0$ by $P(x) = e^{i\tilde{\phi}_0}$.

Step 5: The correct phase factors are related to the modified phase factors by $\phi_1 = \tilde{\phi}_0 + \tilde{\phi}_{2\ell} + (2\ell - 1)\frac{\pi}{2}$ and $\phi_{j\neq 1} = \tilde{\phi}_{j-1} - \frac{\pi}{2}$.

4 RESULTS AND DISCUSSION

Our results for the theoretical exploration of quantum walks and known obstructions to classical computers can be found in sections 3.1 and 3.2. In reflection of the three, separate, computational efforts with regards to QLSP – the QuTiP implementation of the Subasi *et al* (Section 3.3.1), algorithm, the Qiskit implementation of the Subasi *et al.* algorithm (Section 3.3.2), and the QuTip implementation of the Lin and Tong algorithm (Section 3.4) – results and discussion on the theoretical and computational examination of QLSP will be divided into three parts.

4.1 Results of Quantum Walks

Presently, we have utilized Piddock's approach, which builds upon our own, to design a faster backtracking method based on effective resistance estimates. Additionally, we have improved upon these results by reducing the dependence upon the number of marked states. Although our improvement provides us with a significant advantage in the case that there are exponentially many marked states, it is presently insufficient to improve asymptotic scaling beyond Piddock's $O\left(\sqrt{TR}\log(M)\right)$ when that restriction is relaxed. Improving upon this logarithmic term using Piddock's method, which with our present understanding is critical to using local operations to search unknown graphs, still seems possible. We believe that using similar methods to the classical/quantum hybrid approach we currently pursue will be able to eliminate the logarithmic factor, but a specific algorithm remains elusive.

An additional roadblock to the development of optimal local query algorithms is the requirement for the upper bound T. A method for estimating T was previously developed by Ambainis and Kokainis [13]. We have made substantial progress in understanding their method and have shown that many of their statements (slightly generalized) are still true on general graphs. At present, we are generalizing this method and aim to merge it with the JW/Piddock approach to obtain what is ultimately an optimal algorithm. We have reason to believe that, since $2TR$ corresponds to the classical commute time between two vertices in a graph and the classical random walk algorithm for estimating the commute time is somewhat trivial, there may be a quantum algorithm for directly estimating commute time as well. Such an algorithm, which we are presently pursuing, may provide a provably optimal quantum algorithm for local query type search.

4.2 Results of Classical Separation

We mapped H to the graph sample from the ground state of H. In particular, we rely on three subroutines that combine to produce appropriate samples of the ground state of H. These subroutines will be featured in a manuscript that is pending review and will be made available to AFRL upon project completion.

Algorithm 1 takes as a seed a random vertex $v \in V$ and prepares an empty list of vertices $L = \emptyset$. For each neighbor u of v, *Algorithm 1* compares u to each member of $l \in L$ and uses a method

FindRepresentative to determine whether there exists a color-preserving automorphism f such that $f(u) = l$. If there does not exist such an f, then *Algorithm 1* adds u to L. *Algorithm 1* calls itself recursively and, eventually, returns the list L.

If the automorphism group is sufficiently large and **FindRepresentative** is efficient, then L returned by *Algorithm 1* consists of at most $O(poly(n))$ many vertices which are unique representatives of each equivalence class defined over color-preserving automorphisms. (That is, for all pairs $(l, l') \in L \times L$, there does not exist a color-preserving automorphism such that $f(l) = l'$.)

Algorithm 2 uses *Algorithm 1* to find an effective subspace graph. By beginning with L returned by *Algorithm 1*, *Algorithm 2* builds an effective graph by querying the weights in G of all edges connected to each member of L. *Algorithm 2* then returns the effective graph G'.

Finally, *Algorithm 3* takes as input $G' = (V', E')$ and, using standard polynomial-time algorithms, produces the ground state $f: V' \to [0,1]$ of a corresponding Laplacian matrix with appropriate boundary constraints. Then, *Algorithm 3* returns a vertex $v \in V'$ with probability $f(v)^2$.

The most difficult subroutine in this process is **FindRepresentative**. To do this, we convert our color mapping to a clausal theory, which in turn has a natural graph representation, and apply Babai's graph isomorphism algorithm [32] to implement **FindRepresentative** in quasipolynomial time. We are presently checking that our analysis in terms of clausal theories is indeed correct and intend to publish this work in the near future.

4.3 Results from Exploring QLSP

4.3.1 Results from Implementation and Exploration of the Subasi et al. Algorithm in QuTiP.

Numerical results indicated minimal value added by the RM, as we suspected. Figure 9 shows trace distance from the target state versus step number (each change in Hamiltonian is one step) for the unmodified RM and two other variations dependent upon the total number of RM repetitions. The three algorithms reach similar final errors, with the randomization method consistently performing slightly better. However, a larger number of repetitions does not appear to improve the performance of the RM. Further investigation showed that the slightly better performance of the RM versus its variations vanishes when we use fidelity, rather than trace distance, to measure performance. The performance improvement can be shown analytically to be simply a feature of the difference between fidelity and trace distance when comparing an average of multiple runs to a pure target state.

Consider a set of N runs that produces output states $|\psi_i\rangle$ which approximate a target state $|t\rangle$. The fidelity between the i^{th} run and the target state is $F_i = |\langle t|\psi_i\rangle|$ and similarly the trace distance is $D_i = \sqrt{1 - F_i^2}$, since both states are pure. The trace distance between individual runs can also be defined as follows:

$$D_{ij} = \sqrt{1 - |\langle\psi_i|\psi_j\rangle|^2} \tag{29}$$

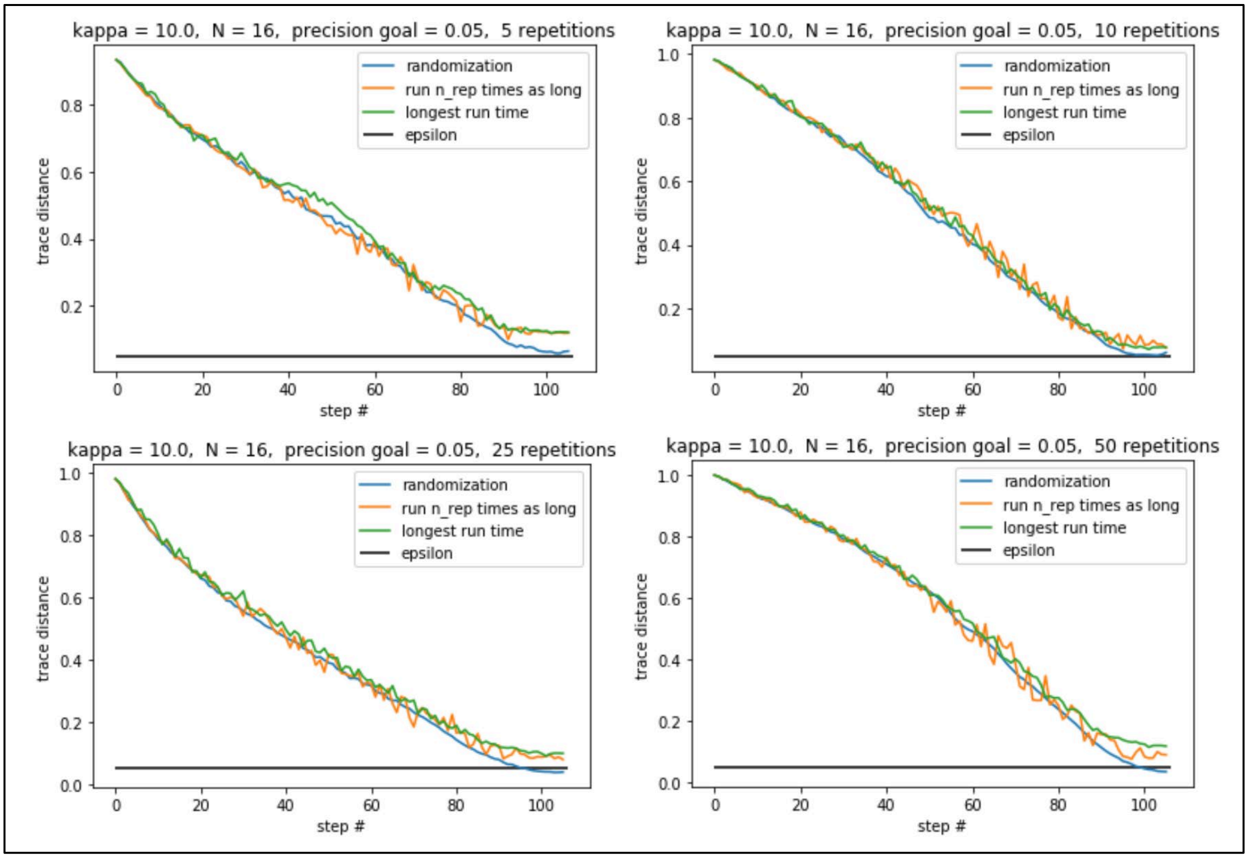

Figure 9: Comparison of RM and Algorithm Variations for Several Choices of Number of Repetitions

The average of all runs, defined by $\rho_{\mathrm{avg}} = \frac{1}{N}\sum_{i=1}^{N} |\psi_i\rangle\langle\psi_i|$, can also be compared to the target state using fidelity $F_{\mathrm{avg}} = \sqrt{\langle t|\rho_{\mathrm{avg}}|t\rangle}$ and trace distance $D_{\mathrm{avg}} = \frac{1}{2}\mathrm{Tr}\big|\rho_{\mathrm{avg}} - |t\rangle\langle t|\big|$. The distance between the average state and the target state is then related in a simple way to the distances of the individual runs from the target state and each other:

$$F_{\mathrm{avg}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} F_i^2} \tag{30}$$

$$D_{\mathrm{avg}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} D_i^2 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}D_{ij}^2} \tag{31}$$

The important thing to note here is that there is an extra term in D_{avg} that makes the average state closer to the target state when there is more scatter between individual runs. In terms of fidelity, however, it is impossible for the average state to attain better fidelity with the target state than the best individual run does: $F_i \leq F_{\mathrm{best}}$ and thus

$$F_{\text{avg}} \leq \sqrt{\frac{1}{N}(NF_{\text{best}}^2)} = F_{\text{best}} \tag{32}$$

These results are demonstrated in Figure 10. Note that the fidelity of the average state is consistent with the average fidelity of a single run, while the trace distance of the average state is better than the best run. Tracing out the ancilla qubit makes all states less distinguishable in terms of both distance measures. This decreases the scatter between individual runs, thus bringing the trace distance of the average state closer to the average trace distance of a single run. Thus, we can conclude that the averaging procedure associated with RM does not add value to the algorithm.

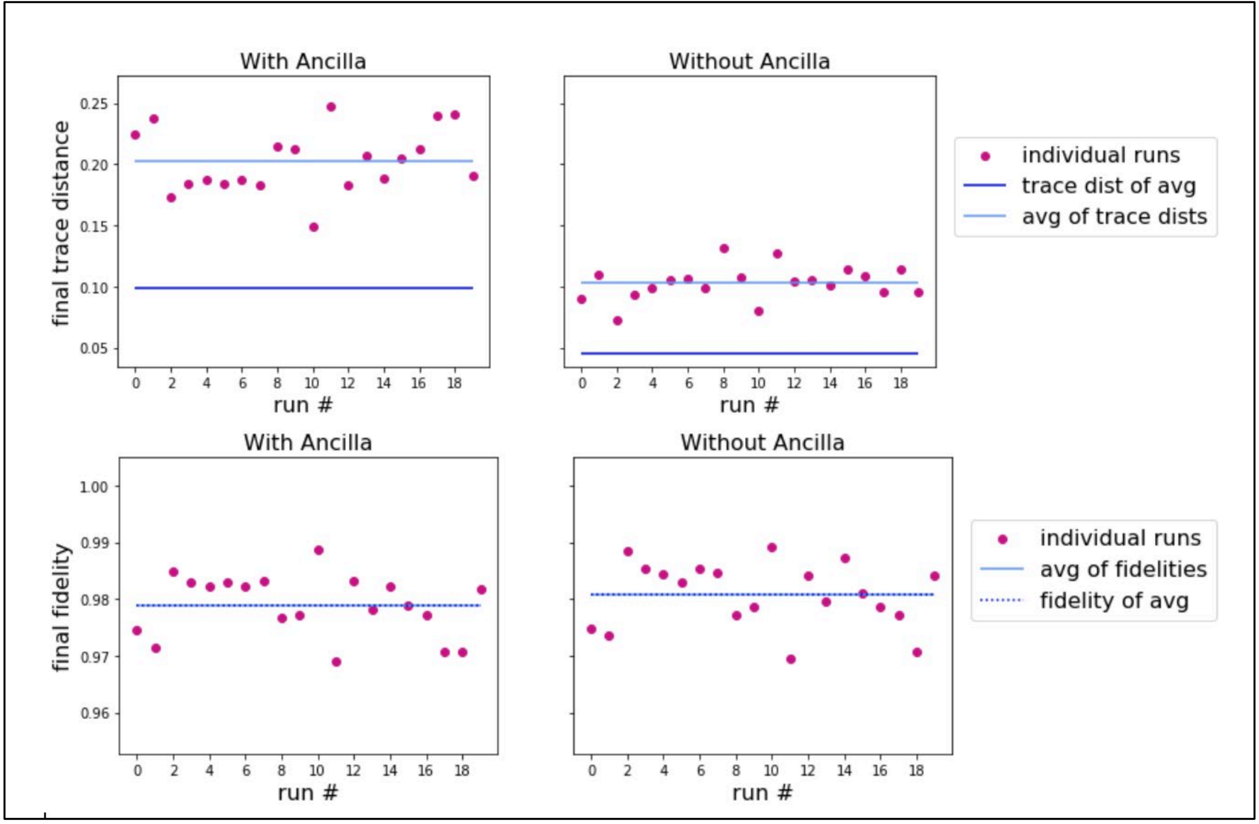

Figure 10: Performance of Individual RM Runs Compared to the Average of All Runs

4.3.2 Results from Implementation and Exploration of the Subasi et al. Algorithm in Qiskit.

We ran simulations on select example problems; one such case is included in the supplementary information. We noticed that setting number of QAOA steps $p = 1$ is not always sufficient for the convergence of the classical optimization scheme. One must find a suitable $p \;>= 1$ for a given problem instance. We also found that the algorithm, in some cases, requires multiple runs with different initializations in order to converge to the minima. Such aspects of QAOA in the context of QLSP require further research in order to develop a better understanding of the issue.

4.3.3 Results from the Detailed Implementation of Lin and Tong Algorithm.

We produced a detailed implementation plan for LT, which can be used for future exploration of the algorithm's performance at varying levels of abstraction. We obtained an analytic expression for the roots of a high-degree polynomial, the calculation of which contributes to numerical instability in the original LT. We are currently using this result to pursue an analytic expression or approximation for the phase factors used in eigenstate filtering, thus potentially producing a speedup polynomial in the degree $\boldsymbol{\ell}$ and/or improvement in numerical stability of LT.

5 CONCLUSIONS

Over the course of this research effort, we have made exciting progress on several fronts of quantum algorithm development and implementation. This includes generalizing local query model quantum walk algorithms, and further illuminating the quantum-classical divide in the adiabatic model, as purely theoretical efforts, and extensive investigations of recent algorithms for the QLSP, which include both theoretical and implementation components.

As is customary in research, in addition to our progress there have been necessary dead ends and pivoting, and we have been diligent about reporting these ups and downs to the AFRL algorithms team to advance the collective knowledge base. This has taken the form of regular seminars and conference calls.

In the following subsections we offer more detailed concluding remarks on each of our research efforts.

5.1 Conclusion from Quantum Walks

Our research has led to an improved local query search algorithm, as well as numerous new avenues of investigation. In particular, we are presently investigating the possibility of either applying the methods of [13] to generic graphs or, more promisingly, designing a quantum algorithm that can directly measure commute time.

5.2 Conclusion from Classical Separation

The divergence in the L^1 and L^2 does not seem to provide a genuine obstruction to simulations of the adiabatic algorithm and, further, to generic simulations of the ground state of the Hamiltonians under consideration. In fact, our algorithm is efficient in cases where known quantum algorithms are expected to fail.

Another interesting open question is whether the quasipolynomial time algorithm is necessary. That is, it remains unclear whether solving this problem is fully equivalent to solving graph isomorphism (GI). It does not appear that general GI reduces to solving our Hamiltonian problem, however it is indeed possible that the hardest cases of GI do. If indeed this is true, it may demonstrate a quasipolynomial separation between classical and quantum systems of this variety and partially solve a longstanding open question. We are aggressively looking into whether this is provable or, alternatively, whether a more efficient algorithm that precludes more than a polynomial time separation might exist.

5.3 Conclusions from QLSP

Our work on circuit level QLSP simulations provides natural segue to actual implementation of such a scheme on the hardware. While developing the framework, we noticed several places where the gate set provided by Qiskit can be replaced by a more economical gate set, hence leading to the possibility of reduced circuit depths. Such gate customizations were avoided in the current work but would be necessary if one wants to run on the actual hardware.

Current work opens up the possibility of using our implementation to study Hamiltonian simulations encountered in areas like chemistry and statistical physics. It will be important to understand the circuit size requirements and modifications needed to fit such problem instances on the hardware. Note that feasibility of such calculations on the hardware will depend heavily on hardware level noise and control errors and is another direction to be explored.

6 REFERENCES

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G.S.L. Brandao, D.A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M.P. Harrigan, M.J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T.S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J.R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M.Y. Niu, E. Ostby, A. Petukhov, J.C. Platt, C. Quintana, E.G. Rieffel, P. Roushan, N.C. Rubin, D. Sank, K.J. Satzinger, V. Smelyanskiy, K.J. Sung, M.D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z.J. Yao, P. Yeh, A. Zalcman, H. Neven, J.M. Martinis, Quantum supremacy using a programmable superconducting processor, Nature. 574 (2019) 505–510. doi:10.1038/s41586-019-1666-5.

[2] L. Lin, Y. Tong, Solving quantum linear system problem with near-optimal complexity, (2019). http://arxiv.org/abs/1910.14596.

[3] S. Jordan, Quantum Algorithm Zoo, (2019). https://quantumalgorithmzoo.org/ (accessed August 30, 2019).

[4] S.E. Venegas-Andraca, Quantum walks: A comprehensive review, Quantum Inf. Process. (2012). doi:10.1007/s11128-012-0432-5.

[5] P. Kaye, R. Laflamme, M. Mosca, V. Vedral, J.R. Friedman, An introduction to quantum computing and introduction to quantum information science, Opt. Eng. (2008). doi:10.1063/1.2883912.

[6] A.M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, D.A. Spielman, Exponential algorithmic speedup by a quantum walk, in: Conf. Proc. Annu. ACM Symp. Theory Comput., 2003. doi:10.1145/780551.780552.

[7] C. Dürr, M. Heiligman, P. Høyer, M. Mhalla, Quantum query complexity of some graph problems, SIAM J. Comput. (2006). doi:10.1137/050644719.

[8] A. Berzina, A. Dubrovsky, R. Freivalds, L. Lace, O. Scegulnaja, Quantum query complexity for some graph problems, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). (2003). doi:10.1007/978-3-540-27836-8_42.

[9] S. Apers, A. Gilyén, S. Jeffery, A Unified Framework of Quantum Walk Search, (2019). http://arxiv.org/abs/1912.04233.

[10] M. Davis, H. Putnam, A Computing Procedure for Quantification Theory, J. ACM. (1960). doi:10.1145/321033.321034.

[11] M. Davis, G. Logemann, D. Loveland, A machine program for theorem-proving, Commun. ACM. (1962). doi:10.1145/368273.368557.

[12] A. Montanaro, Quantum speedup of branch-and-bound algorithms, Phys. Rev. Res. 2 (2020) 013056. doi:10.1103/PhysRevResearch.2.013056.

[13] A. Ambainis, M. Kokainis, Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games, (2017). http://arxiv.org/abs/1704.06774.

[14] A. Belovs, Quantum Walks and Electric Networks, (2013) 1–10. http://arxiv.org/abs/1302.3143.

[15] M. Jarret, K. Wan, Improved quantum backtracking algorithms using effective resistance estimates, Phys. Rev. A. (2017). doi:10.1103/PhysRevA.97.022337.

[16] S. Piddock, Quantum walk search algorithms and effective resistance, (2019). http://arxiv.org/abs/1912.04196.

[17] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in: Proc. Annu. ACM Symp. Theory Comput., Association for Computing Machinery, 2019: pp. 217–228. doi:10.1145/3313276.3316310.

[18] S. Bravyi, D. Gosset, Polynomial-Time Classical Simulation of Quantum Ferromagnets, Phys. Rev. Lett. 119 (2017) 1–12. doi:10.1103/PhysRevLett.119.100503.

[19] E. Crosson, S. Slezak, Classical Simulation of High Temperature Quantum Ising Models, (2020). http://arxiv.org/abs/2002.02232.

[20] M.B. Hastings, M.H. Freedman, Obstructions To Classically Simulating The Quantum Adiabatic Algorithm, (2013). http://arxiv.org/abs/1302.5733.

[21] M. Jarret, S.P. Jordan, B. Lackey, Adiabatic optimization versus diffusion Monte Carlo methods, Phys. Rev. A. 94 (2016) 042318. doi:10.1103/PhysRevA.94.042318.

[22] J. Bringewatt, W. Dorland, S.P. Jordan, A. Mink, Diffusion Monte Carlo approach versus adiabatic computation for local Hamiltonians, (2017). doi:10.1103/PhysRevA.97.022323.

[23] J. Bringewatt, W. Dorland, S.P. Jordan, Polynomial time algorithms for estimating spectra of adiabatic Hamiltonians, Phys. Rev. A. (2019). doi:10.1103/PhysRevA.100.032336.

[24] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, J. Dean, Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, (2017) 1–19. http://arxiv.org/abs/1701.06538.

[25] E. Crosson, A.W. Harrow, Simulated Quantum Annealing Can Be Exponentially Faster Than Classical Simulated Annealing, in: Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS, 2016: pp. 714–723. doi:10.1109/FOCS.2016.81.

[26] M. Jarret, B. Lackey, A. Liu, K. Wan, Quantum adiabatic optimization without heuristics, (2018). http://arxiv.org/abs/1810.04686.

[27] Y. Subasi, R.D. Somma, D. Orsucci, Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing, (2018). doi:10.1103/PhysRevLett.122.060504.

[28] Y. Ge, A. Molnár, J.I. Cirac, Rapid adiabatic preparation of injective PEPS and Gibbs states, (2015). doi:10.1103/PhysRevLett.116.080503.

[29] A. Ambainis, A. Gilyén, S. Jeffery, M. Kokainis, Quadratic speedup for finding marked vertices by quantum walks, (2019).

[30] M. Szegedy, Quantum speed-up of Markov Chain based algorithms, in: Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS, 2004. doi:10.1109/focs.2004.53.

[31] M. Jarret, Hamiltonian surgery: Cheeger-type gap inequalities for nonpositive (stoquastic), real, and Hermitian matrices, (2018) 1–28. http://arxiv.org/abs/1804.06857.

[32] L. Babai, Graph Isomorphism in Quasipolynomial Time, 7443327 (2016) 1–89.

[33] R. Bhatia, Matrix Analysis, Springer New York, New York, NY, 1997. doi:10.1007/978-1-4612-0653-8.

[34] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. 113 (2014) 1–5. doi:10.1103/PhysRevLett.113.130503.

[35] G. Wang, Efficient quantum algorithms for analyzing large sparse electrical networks, Quantum Inf. Comput. 17 (2017) 987–1026.

[36] A.N. Chowdhury, R.D. Somma, Quantum algorithms for gibbs sampling and hitting-time estimation, Quantum Inf. Comput. 17 (2017) 41–64. doi:10.26421/QIC17.1-2.

[37] S. Boixo, E. Knill, R. Somma, Eigenpath Traversal By Phase Randomization, Quantum Inf. Comput. 9 (2009) 833–855. papers2://publication/uuid/45F0C81D-87DE-4D78-8026-2785A39D462D.

[38] D. An, L. Lin, Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, (2019). http://arxiv.org/abs/1909.05500.

[39] J. Smith, M. Mosca, Algorithms for Quantum Computers, in: Handb. Nat. Comput., Springer Berlin Heidelberg, Berlin, Heidelberg, 2012: pp. 1451–1492. doi:10.1007/978-3-540-92910-9_43.

[40] A. Gilyén, Y. Su, G.H. Low, N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, (2018). doi:10.1145/3313276.3316366.

[41] H. Paik, SYSTEMAND METHOD FOR QUANTUM INFORMATION TRANSFER BETWEEN OPTICAL PHOTONS AND SUPERCONDUCTIVE QUBITS, US20140314419A1, 2014.

[42] A. Wallraff, D.I. Schuster, A. Blais, L. Frunzio, R.-S. Huang, J. Majer, S. Kumar, S.M. Girvin, R.J. Schoelkopf, Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics, Nature. 431 (2004) 162–167. doi:10.1038/nature02851.

[43] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D.G. Cory, Y. Nakamura, J.S. Tsai, W.D. Oliver, Noise spectroscopy through dynamical decoupling with a superconducting flux qubit, Nat. Phys. 7 (2011) 565–570. doi:10.1038/nphys1994.

[44] Q. Ficheux, S. Jezouin, Z. Leghtas, B. Huard, Dynamics of a qubit while simultaneously monitoring its relaxation and dephasing, Nat. Commun. 9 (2018) 1926. doi:10.1038/s41467-018-04372-9.

[45] P. V. Klimov, J. Kelly, Z. Chen, M. Neeley, A. Megrant, B. Burkett, R. Barends, K. Arya, B. Chiaro, Y. Chen, A. Dunsworth, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, T. Huang, E. Jeffrey, E. Lucero, J.Y. Mutus, O. Naaman, C. Neill, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T.C. White, S. Boixo, R. Babbush, V.N. Smelyanskiy, H. Neven, J.M. Martinis, Fluctuations of Energy-Relaxation Times in Superconducting Qubits, Phys. Rev. Lett. 121 (2018). doi:10.1103/PhysRevLett.121.090502.

APPENDIX A: NOTE ON SOURCE CODE

All source code has been submitted as a separate document to the client.

LIST OF ACRONYMS

AFRL – Air Force Research Laboratory

AI – Artificial intelligence

AK – Ambainis and Kokainis

DPLL – Davis-Putnam-Logemann-Loveland algorithm

FPGA – Field programmable gate array

GI – Graph isomorphism

HHL – Quantum algorithm for linear systems problem developed by Aram Harrow, Avinatan Hassidim, and Seth Lloyd.

JWP – Jarret and Wan/Piddock

LT – Lin and Tong's algorithm

NISQ – Noisy intermediate-scale quantum

PCB – Printed circuit board

PMT – Photomultiplier

QAOA – Quantum approximate optimization algorithm

QED – Quantum electrodynamics

QLSP – Quantum linear systems problem

QML – Quantum machine learning

RH – Relative humidity

RM – Randomization method

SAT – Boolean satisfiability

VoIP – Voice over internet protocol

Yb – Ytterbium, number 70 on the periodic table of the elements