QUANTUM ALGORITHMS FOR EXACT AND APPROXIMATE OPTIMIZATION

by

Brandon A. Barton

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Computational and Applied Mathematics).

Golden, Colorado

Date ______________________________

Signed: ______________________________

Brandon A. Barton

Signed: ______________________________

Dr. Lincoln D. Carr
Thesis Advisor

Signed: ______________________________

Dr. Cecilia Diniz Behn
Thesis Advisor

Golden, Colorado

Date ______________________________

Signed: ______________________________

Dr. Gus Greivel
Department Head
Department of Applied Mathematics and Statistics

ABSTRACT

Following early breakthroughs for quantum algorithms in database search and integer factoring, over recent years, quantum computing hardware technologies and quantum algorithms have undergone rapid developments. Yet, the search for “killer” applications of quantum computing has remained elusive amidst the noisy intermediate scale quantum computing (NISQ) era, where noise from the environment provides a challenge for quantum algorithms requiring a fault-tolerant machine. Among prevalent applications such as quantum simulation and quantum chemistry, hard optimization tasks provide a promising class of problems where quantum algorithms hope to demonstrate advantage beyond quadratic speed-ups.

This work investigates the application of heuristic quantum algorithms to combinatorial optimization, where we consider the class of binary constraint satisfaction problems. We introduce two new quantum algorithms for exact and approximate optimization, and demonstrate our methods with extensive quantum simulations that exhaust the capabilities of current supercomputers for classical simulation of quantum dynamics. In particular, we investigate a well-known problem known as MAX-3-XORSAT, in the complexity class, NP-hard. First, we introduce a new method called *Spectrally Folded Quantum Optimization* (SFQO) which transforms the energy landscape of the problem, allowing approximate solutions to be readily obtained with guaranteed approximation ratios. Secondly, we introduce a new non-classical steering mechanism called *Iterative Symphonic Tunneling for Satisfiability problems* (IST-SAT) which uses macroscopic quantum tunneling effects to guide the sufficiently good approximations towards the true global optima.

The first work we present in this thesis, Spectrally Folded Quantum Optimization, investigates the ability of quantum algorithms to find approximate solutions to the MAX-3-XORSAT hypergraph problem class. We identify several distinct physical mechanisms associated to these problems, which make the task of finding exact solutions hard for all previously known classical and quantum algorithms. However, we find the same mechanisms that prevent quantum algorithms to find exact solutions, do not necessarily hold for approximate optimization. Spectrally folded quantum optimization implements a classical deformation of the constraint satisfaction problem energy landscape, which allows quantum algorithms to find constant fractions of the optimal solution with increasing problem size. We provide theoretical performance predictions for the algorithm, and benchmark our methods with extensive quantum simulations. Our results demonstrate that all of our numerical simulations agree with our predictions at the system sizes we can classically simulate using the best super-computing resources for classical simulation of quantum dynamics. This work suggests that quantum algorithms may be more powerful than previously thought for the task of approximate optimization.

In the second work in this thesis, Iterative Symphonic Tunneling for Satisfiability problems, we present a non-classical steering mechanism for quantum optimization algorithms based on the use of high frequency oscillating "AC" drives. To demonstrate this mechanism we introduce an iterative quantum algorithm, IST-SAT, which does not require computing gradients or extensive fine-tuning. Using an initial classical or quantum algorithm to approximate the MAX-3-XORSAT problem, IST-SAT sets parameters in single-qubit oscillating drives according to the bits in the initial solution, which induces further macroscopic tunneling effects towards the true ground state(s) of the problem. IST-SAT converges to the ground state in an iterative manner, measured by the number of spin flips away from the ground state(s), also known as Hamming distance. We identify what it means to have a sufficient initial approximation for the IST-SAT, which defines a radius of convergence for the algorithm. The numerical results we obtain demonstrate that IST-SAT monotonically improves in performance, when provided initial states that are closer and closer to the ground state of the problem. When provided with an initial approximation at or above the radius of convergence, our results suggest that IST-SAT converges in polynomial time.

Together, the results presented in this work obtain exponential speed-ups for obtaining approximate solutions, and polynomial speed-ups for exact problem solving, over the best known quantum and classical algorithms. While NP-hard optimization problems remain hard to solve exactly, by combining the methods in this work, we show that our algorithm can converge to the true ground state in polynomial time when provided a sufficiently good initial state. The novel mechanisms in this work thus pave new pathways for achieving quantum advantage on well-known hard problems, such as MAX-3-XORSAT. We expect the methods in this work to be amenable for experimental demonstrations on current or near-term quantum hardware, thus providing an exciting opportunity to demonstrate utility of quantum computers in the NISQ era.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Approximation ratio	q_a
Approximation target	A
Bra	$\vert v\rangle$
Classical bit	b
Clause	C
Complex field	$\mathbb{C}$
Constraint	ϕ
Constraint density	N_C/N
Constraint tuple	S
Density matrix	ρ
Edge set	$\mathcal{E}$
Final time	t_f
Fraction of incorrect phases	r
Fraction of unsatisfied constraints	ϵ
Fractional Hamming distance	d
Frequency	ω
Graph uniformity	p
Ground state	$\vert G\rangle$
Ground state bitstring	G
Ground state energy	E_{GS}
Hamiltonian	H
Hamming distance	D_H
Hilbert space	$\mathcal{H}$
Hypergraph	$G_{\mathcal{H}}$
Identity matrix	I

Imaginary unit	i
Ket	$\langle v|$
Literal	a
Local phase	φ
Logical AND	$\wedge$
Logical NOT	$\neg$
Logical OR	$\vee$
Matrix	A
Measurement operator	M
Number if unsat. constraints minus sat. constraints.	k
Number of constraints	N_C
Number of qubits	N
One-state	$|1\rangle$
Pauli-X	X
Pauli-Y	Y
Pauli-Z	Z
Phase pattern	P
Planted partial solution ground state	G
Probability	P, p
Probability of reaching ground state	P_{GS}
Problem energy	E
Quantum state	$|\psi\rangle$
Radius of convergence	r_c
Spectral gap	Δ
Time	t
Time-step	dt
Total evolution time	T

Tunneling rate	Ω
Unit vector	$\hat{\mathbf{e}}$
Unitary operator	U
Vector	$\mathbf{v}$
Vertex set	$\mathcal{V}$
Weighting function	w
Zero-state	$\|0\rangle$

LIST OF ABBREVIATIONS

Adiabatic quantum computation	AQC
Boolean satifiability problem	SAT
Bounded-error probabilistic polynomial time	BPP
Bounded-error quantum polynomial time	BQP
Conjugate normal form	CNF
Constraint satisfaction problem	CSP
High performance computing	HPC
Iterative symphonic tunneling for satisfiability problems	IST-SAT
Low autocorrelation binary sequence	LABS
Max-Cut	MC
Merlin-Arthur	MA
Noisy intermediate scale quantum	NISQ
Partial planted solution	PPS
Partial planted solution problem	PPSP
Quantum Monte Carlo	QMC
Quantum approximate optimization algorithm	QAOA
Quantum-Max-Cut	Q-MC
Quantum-Merlin-Arthur	QMA
Quasi-greedy classical algorithm	GC
Semi-definite programming	SDP
Sherrington-Kirkpatrick	SK
Spectrally folded quantum optimization	SFQO
Time to solution	TTS
Transverse field Ising model	TFIM
Transverse field chaos	TFC

Trial minimum annealing	TMA
Trotterized adiabatic quantum computation	TAQC
Variational quantum algorithms	VQE

ACKNOWLEDGMENTS

First, I would like to acknowledge my thesis advisors: Professor Lincoln D. Carr and Professor Cecilia Diniz Behn at the Colorado School of Mines. Professor Carr, thank you for reeling me in when I needed it most. You have encouraged me to study meaningful problems, taught me the value of scientific communication, and allowed me to develop as an independent researcher. Professor Diniz Behn, thank you always listening intently, and being with me since the beginning of my mathematical research journey. I will look back on our many journal club discussions with very fond memories.

This work would not be possible without my family and friends. Thank you for your continued support and love throughout the years.

To my professors, collaborators, and committee members at the Colorado School of Mines: Professor Eliot Kapit, thank you for the opportunity to work on the projects we've developed together over the past year. Your resiliance, creativity, and humility have inspired me to think deeper and enjoy life outside of research. Professor Zhe-Xuan Gong, thank you for teaching me so much about quantum many-body physics. Professor Dinesh Mehta, thank you for your sage-like advice, and for inspiring my initial love for graph theory and network science. Finally, thank you to the several other wonderful professors in the Department of Applied Mathematics and Statistics who have supported my learning.

For those fellow students who I am now glad to call my friends: Bora Basyildiz, George Grattan, Benjamin Krawciw, Mason Kugler, Kylee Shiekh, Jacob Sagal, Pratik Patnaik, Sean Feeney, Caleb Rotello, and Hakan Ayaz. You have made the days (and late nights) we spent together in the quantum theory lab so enjoyable.

Thank you to the Institute for Pure and Applied Mathematics (IPAM) for the opportunity to visit during the long program on *Mathematical and Computational Challenges in Quantum Computing* (CQC), and to my collaborators and friends in the 2023 IPAM CQC co-hort.

Many of the numerical simulations in this work were performed with a generous grant of HPC access from the Fujitsu Corporation. I would like to acknowledge the Colorado School of Mines HPC team as other simulations in this work were completed on the Mines Wendian HPC cluster.

Finally, I would like to acknowledge my funding through the National Science Foundation (NSF) Quantum Engineering National Research Traineeship (QENRT), which has supported this work.

For the curious minded, who inspire others to do science.

CHAPTER 1
INTRODUCTION

1.1 Key mathematics of quantum computation and many-body physics

To begin, we introduce some mathematical notation to bridge a gap between the notation commonly used in applied mathematics and the notation used in quantum physics and quantum computing. Throughout this thesis, we use the latter notation used traditionally in the quantum computing literature [1]. In this section, we introduce the four postulates of quantum mechanics, and the common notations used throughout the rest of this thesis. Throughout this work, we use the Dirac (bra-ket) notation, equivalent to matrix-vector notation by following relations

$$|v\rangle \equiv \mathbf{v}, \quad \langle v| \equiv \mathbf{v}^\dagger, \quad \langle u|A|v\rangle \equiv \langle \mathbf{u}, A\mathbf{v}\rangle = \mathbf{u}^\dagger A\mathbf{v}, \quad \langle u|v\rangle \equiv \langle \mathbf{u}, \mathbf{v}\rangle = \mathbf{u}^\dagger \mathbf{v}, \tag{1.1}$$

where † denotes the Hermitian conjugate, and the standard ℓ_2 scalar product and norm for vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{C}^d$ are defined by

$$\langle \mathbf{u}, \mathbf{v}\rangle = \sum_{j=0}^{d-1} u_j^* v_j, \quad \|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle}. \tag{1.2}$$

The state of a closed quantum system (Postulate 1) is a complex vector in a Hilbert space $\mathcal{H}$. We define the Hilbert space by

$$\mathcal{H} = \{\mathbf{v} \in \mathbb{C}^d, \mathbf{v} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{pmatrix} |\alpha_j \in \mathbb{C}\}, \tag{1.3}$$

where quantum mechanical states are represented by $\mathbf{v} \in \mathcal{H}$ with $\dim(\mathcal{H}) = d$ as the dimension of the Hilbert space. Throughout this work, we consider finite dimensional vector spaces, comprised of a set of d-dimensional vectors, where the elements α_i of the state vector are complex numbers. As $d < \infty$, we denote a basis for $\mathcal{H}$ as product of the standard basis (unit vectors), and complex amplitudes α_j

$$\hat{\boldsymbol{e}}_j = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad e_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{1.4}$$

$$\mathbf{v} = \sum_{j=1}^{d} \alpha_j \hat{e}_j, \quad \alpha_j \in \mathbb{C}. \tag{1.5}$$

Any vector $\mathbf{v} \in \mathcal{H}$ can be written as a linear combination, also known as a *superposition* of states in quantum mechanics. Furthermore, quantum states can easily be interpreted as probability distributions as that the coefficients (amplitudes) α_j are complex modulus squared $|\alpha_j|^2$ gives a probability about the quantum system being in a specific state. For closed quantum systems, quantum states also have unit norm and ℓ_2 scalar product for vectors $\mathbf{u}$, $\mathbf{v} \in \mathbb{C}^d$

$$\|\mathbf{v}\|^2 = \langle v, v\rangle = \sum_{j=1}^{d} |\alpha_j|^2 = 1, \tag{1.6}$$

$$\langle u, v\rangle = \sum_{j=1}^{d} \bar{u_j} v_j. \tag{1.7}$$

We remark that quantum states can also be formally described by *rays* in a projective Hilbert space, used in the C^* algebraic formulation of quantum mechanics, where Hilbert spaces are emergent rather than fundamental structures in the mathematical formulation of quantum mechanics [2]. In this work, we use the "state vector" terminology to be consistent with the language used in the standard literature in quantum computing [1].

Given two quantum systems with respective Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, the state space of the composite quantum system $\mathcal{H}$ (Postulate 2) is the tensor product of the subsystems,

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2. \tag{1.8}$$

We define the first subsystem $\mathcal{H}_1$ to have the span $\{u_i\}_{i=1}^{d_1}$ and the second subsystem $\mathcal{H}_2$ to have the span $\{v_j\}_{j=1}^{d_2}$, so the composite space $\mathcal{H}$ is equivalently the span of the tensor product $\{u_i \otimes v_j\}_{i=1,j=1}^{d_1,d_2}$. For two states $\psi \in \mathcal{H}_1$ and $\phi \in \mathcal{H}_2$, the composite state is given by

$$|\psi\rangle \otimes |\phi\rangle = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{d_1} \end{pmatrix} \otimes \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{d_2} \end{pmatrix} = \begin{pmatrix} \psi_1 \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{d_2} \end{pmatrix} \\ \vdots \\ \psi_{d_1} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{d_2} \end{pmatrix} \end{pmatrix}. \tag{1.9}$$

We note that the underlying Hilbert spaces can be completely different quantum systems and this formalism is standard for describing open quantum systems [3]. In this thesis, we exclusively study *closed quantum systems*, with absence of noisy system-environment interactions. For closed quantum systems, quantum states evolve in time (Postulate 3) under a unitary operator $U(t)$ that may be time dependent given by the equation,

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle, \quad t \geq 0. \tag{1.10}$$

In other terms, quantum states evolve according to Schrödinger's partial differential equation given by

$$\frac{\partial|\psi\rangle}{\partial t} = \frac{i}{\hbar} H(t)|\psi(0)\rangle. \tag{1.11}$$

Here, $H(t)$ is an operator called the *Hamiltonian.* For closed quantum systems, the Hamiltonian is a self-adjoint, Hermitian operator such that $H = H^\dagger$. Throughout this thesis, we take $\hbar = 1$, so the factor $1/\hbar$ cancels. For states that evolve under a time-independent Hamiltonian H, the state at time t is exactly given by

$$|\psi(t)\rangle = e^{iHt}|\psi(0)\rangle, \tag{1.12}$$

where $|\psi(0)\rangle$ is the initial state, and $U(t) = e^{iHt}$ is a unitary operator which can be computed by diagonalizing H. However, for n-qubit Hamiltonians, H has dimension $d = 2^n$, making computing $U(t)$ intractable for systems with even a moderate number qubits. This problem, also known as the *quantum many-body problem*, underlies much work in quantum many-body physics over the last several decades, where the goal is to compute the spectrum of a Hamiltonian H representing the energy levels of an arbitrary quantum system. Diagonalizing H produces the eigenvalues (energies) and eigenvectors (states) of the physical system, so we may understand physical properties including phases of matter that the system exhibits in ground state(s) and low energy configurations.

For very large matrices without structure, diagonalizing H on any classical computer with standard tools from computational linear algebra, such as Lanczos iteration methods [4] quickly becomes intractable for even moderately sized systems. This challenge provided much of the initial motivation for the field of quantum computing and it is widely believed that there exists no classical algorithm which is sufficient for the task. However, due to much progress on quantum hardware in the past decade, we can now ask ourselves the following question: provided access to a programmable quantum computer, where can quantum algorithms provide advantage over classical algorithms on hard optimization tasks? Much work over the last several decades has made progress on developing quantum algorithms for both classical and quantum problems that are believed to be intractable with only classical resources.

In this work, we use quantum algorithms to investigate hard classical problems believed to be intractable for all known classical algorithms. To utilize a quantum computer, we employ elementary quantum operations known as quantum gates to simulate the dynamics of a quantum system under the time evolution under a k-local Hamiltonian H [5], defined by the sum of m terms acting upon at most k qubits each

$$H = \sum_{i=1}^{m} H_i. \tag{1.13}$$

Here, k-local does not imply any notion of geometric locality. We only require that each H_i acts on at most k of the total n qubits not restricted to neighboring qubits. To be more explicit, each H_i is the tensor product of k non-trivial terms, padded by the tensor product of identity terms. For example, given an arbitrary operator A, a local term H_i may be given by

$$H_i = A_i \otimes I \cdots \otimes I \otimes A_j \otimes I \cdots \otimes I \otimes A_k \otimes I, \tag{1.14}$$

where A_i, A_j, A_k acts on *any* three qubits with respective indices $i, j, k \in [0, n-1]$.

To evolve time under a time-independent Hamiltonian, one may naively compute the matrix exponential e^{iHt}. When the Hamiltonian H is given as a sum $H = A + B$ of square matrices A and B representing time-independent Hamiltonian operators, we then wish to compute the matrix exponential $e^{i(A+B)t}$. Throughout this work, we extensively use the Suzuki-Trotter decomposition [6, 7], an *operator splitting* method commonly known as just the Trotter decomposition, or simply the exponential product formula

$$e^{\delta A} e^{\delta B} = e^{\delta(A+B)+O(\delta^2)}. \tag{1.15}$$

In other terms, the Trotter decomposition provides an approximation

$$e^{i(A+B)dt} = e^{iAdt} e^{iBdt} + O(dt^2), \tag{1.16}$$

to the exponential operator on the left-hand side of the equation with a correction term of second order in dt denoted by Big-Oh notation. Here, we use the definition of Big-Oh notation provided in standard texts on computational complexity [8]. Hence, the time evolution under H for a small time step dt becomes exact in the limit as $dt \to 0$. It is also clear that when A and B commute, the error term of order dt^2 is zero. However, when A and B do not commute such that $[A, B] = AB - BA \neq 0$, the correction terms add in the iterative application $U(t)$ for each discrete time step dt in the interval from $t = 0$ to the total evolution time T. When the Hamiltonian is time-independent, the Trotter decomposition we discuss here is straight forward to apply. However, when the Hamiltonian is time-dependent, the additional effect of time-ordering must be considered, as Hamiltonians at different times t and t' may not commute such that $[H(t), H(t')] \neq 0$. In Chapter 2, we discuss time-dependent Hamiltonian simulation under the gate-model of quantum computation.

Summarizing postulate four, quantum measurements are described by a set of measurement operators $\{M_k\}_{k=1}^{d}$ which act on the state space of the system that is being measured, where k is the number of measurement outcomes that can occur. Given a quantum system with a state $|\psi\rangle \in \mathcal{H}$, the probability that a measurement results in outcome k is given by

$$p_k = \| M_k |\psi\rangle \|^2 = \langle \psi | M_k^\dagger M_k | \psi \rangle. \tag{1.17}$$

If the outcome k was observed, that state of the system is transformed by

$$\psi \rightarrow \frac{M_k|\psi\rangle}{\sqrt{p_k}} \equiv |\psi_k\rangle. \tag{1.18}$$

As p_k is a probability, the sum over all such probabilities must be equal to one, given by $\sum_k p_k = 1$. Furthermore, as $|\psi\rangle$ is an arbitrary choice, measurement operators have the constraint

$$\sum_k^d M_k^\dagger M_k = I. \tag{1.19}$$

In classical computing, bits are the fundamental object of information, taking possible states 0 or 1. Quantum bits or *qubits* are mathematically represented by a complex vector in $\mathbb{C}^2$ provided by the notation

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}. \tag{1.20}$$

Hence, qubits can be in either state $|0\rangle$ or $|1\rangle$. After we introduced classical and quantum bits, one might ask the question: If bits and qubits both have two possible states, how are they different from each other? In quantum mechanics, qubits can be in a linear combination of its possible states at once. In other words quantum states can be in a *superposition* state, while classical bits are definitively just equal to zero *or* one. As an example, we provide a simple qubit state in Eq. (1.21) by,

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \tag{1.21}$$

where α_0 and α_1 are complex amplitudes.

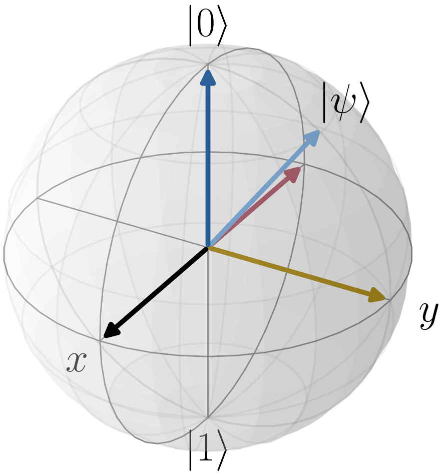


Figure 1.1 The geometric representation of a single qubit system known as the *Bloch sphere*. The terminal points of the z axis represent the $|0\rangle$ and $|1\rangle$ states respectively. The terminal points of the x and y axes represent the $\{|+\rangle, |-\rangle\}$ and $\{| - i\rangle, | + i\rangle\}$ states respectively. The vector of a single qubit pure state will extend to the surface of the sphere. The state $|\psi\rangle$ is a superposition state of $|0\rangle$ and $|1\rangle$. While the Bloch sphere provides the natural representation of single qubit states, the geometric representation for n-qubit states is more challenging to visualize.

Generalizing to more than one qubit, multi-qubit states are the tensor product of single qubit states. For example, the multi-qubit zero and ones states are given by

$$|00\dots0\rangle = |0\rangle \otimes |0\rangle \otimes \dots \otimes |0\rangle, \quad |11\dots1\rangle = |1\rangle \otimes |1\rangle \otimes \dots \otimes |1\rangle. \tag{1.22}$$

Similar to single qubits, multi-qubit states can be superpositions of their computational basis-states, or simply the linearly independent vectors that span $\mathbb{C}^{2\otimes n}$. For example, a two-qubit state

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle, \tag{1.23}$$

has 2^n computational basis states, where $n = 2$ in this case. It is clear that quantum states (in the qubit setting) in superposition are exponentially more information dense that classical multi-bit states.

One fundamental object studied throughout this work is the Hamiltonian of a quantum system. The Hamiltonian of a closed quantum system provides a description of the energies in terms of eigenvalues and associated states given by the eigenvectors. The Hamiltonians we work with throughout this thesis are defined by spin-$\frac{1}{2}$ degrees of freedom, commonly known as quantum spin-Hamiltonians. Spin Hamiltonians are commonly described using the set of Pauli matrices $\{X, Y, Z\}$ and the 2×2 identity matrix, given by

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{1.24}$$

The Pauli matrices, together with the identity matrix, form a basis for the real vector space of 2×2 Hermititan matrices. In quantum mechanics, Pauli matrices also represent observables, or linear operators that act on a complex Hilbert space. The Pauli matrices also form the Pauli group G_1, on one qubit

$$G_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\} \equiv \langle X, Y, Z\rangle. \tag{1.25}$$

For n qubits, the Pauli group G_n is the group generated by the operators above in the tensor product Hilbert space $(\mathbb{C}^2)^{\otimes n}$. For a quantum many-body system, the Pauli group is the generating set for the Hamiltonian which describes the energy of the composite system.

1.2 Computational complexity of quantum and classical problems

To introduce the problems considered in Chapters 3 and 4, we introduce the class known as Boolean satisfiability (SAT) problems. The primary object in SAT problems are *Boolean formulas*, which consist of a set of variables $\{x_i\}_{i=1}^m$ which are either TRUE or FALSE, and logical operators AND ($\wedge$), OR ($\vee$), and NOT($\neg$). The input to SAT problems are Boolean expressions, and the output is a YES/NO answer which tells which determines if a sequence of variables can be evaluated to TRUE. If there exists a sequence which outputs TRUE, the Boolean formula is deemed *satisfiable.* Otherwise, we say that the Boolean formula is *unsatisfiable.* However, the fact that a problem is satisfiable does not imply that finding such a sequence is computationally efficient.

The canonical 3SAT problem is a special case of a SAT problem in which the formula contains at most three variables between each pair of parentheses. In conjugate normal form (CNF), SAT problems are generally expressed as

$$\bigwedge_i \left(\bigvee_j a_{i_j} \right), \tag{1.26}$$

where a_{i_j} is a literal and $(\vee_j a_{i_j})$ are called *clauses.* For example, 3SAT takes the form

$$C_1 \wedge C_2 \wedge \cdots \wedge C_n, \tag{1.27}$$

where C_i is a clause exclusively consisting of OR logical operations between three literals $\{a_j\}_{j=1}^k \in C_i$. A k-CNF is a CNF in which each clause has at most k literals. Each literal is chosen from the set of m variables $\{x_i\}_{i=1}^m$. For example, the expression

$$(a_1 \vee a_4 \vee \neg a_2) \wedge (a_3 \vee \neg a_2 \vee a_1) \wedge (a_1 \vee \neg a_4 \vee a_3) \tag{1.28}$$

is a 3-CNF with four literals. In this work, we consider the optimization of *constraint satisfaction problems* (CSPs) on combinatorial objects, known as graphs. Classically, we define a constraint as a predicate in the form $\phi : \{\pm 1\}^k \to \{0, 1\}$. The constraint is said to be satisfied if it evaluates to 1, and unsatisfied otherwise. The CSP can be written as

$$\Phi(x) = \sum_{i=1}^{m} \phi(x_{S_i} \circ y_i), \tag{1.29}$$

where constraints are represented by tuples $S_1, S_2, \ldots, S_m \in [n]^k$ and $y_1, y_2, \ldots y_m \in \{\pm 1\}^k$ specify whether the variables will be negated in the constraint. CSPs provide a class of optimization problems, in which the goal is to find the configuration of variables that maximize the number of satified constraints.

$$\max_{x \in \{\pm 1\}^n} \varphi(x). \tag{1.30}$$

The CSPs we consider in this thesis are defined on random hypergraphs $G_{\mathcal{H}} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of N variables $\{x_0, x_1, \ldots, x_{N-1}\}$, and $\mathcal{E}$ is the set of hyperedges which encapsulate a many-variable interaction. Constraints in the problem are represented by the hyperedge which encapsulates k-variable interactions.

Throughout this thesis, we consider problems which are known to be in the complexity class **NP**. To review the essentials of computational complexity, let us define the complexity classes and provide examples problems in each class. A problem is said to be in complexity class **P** if there exists an algorithm which provides an exact solution in polynomial time. Formally, the class **P** is defined to consist of all decision problems which can be solved by a *Turing machine* in polynomial (poly) time. In less formal terms, **P** represents the class of problems in which "efficient" programs can be written to provide exact solutions. The term "efficient" refers to the asymptotic scaling of the algorithm in the same motivation as *Big-Oh* notation. However, even for problems known to be in **P**, large constant factors and many variables (at large n) may result in long computation times in practice.

Often in quantum computing, we wish to *diagonalize* a quantum many-body Hamiltonian to obtain the *exact* ground state, or some state that serves as an *approximation* to the ground state energy. The most naive strategy is to iterate through every possible state, compute its energy using the terms in the Hamiltonian and save the minimum energy state, therefore solving the problem. However, multi-qubit Hamiltonians may contain up to 2^n many basis states. Therefore, the *brute-force* strategy takes exponential time in n, which motivates the use of other, more efficient, strategies for computing ground states.

By formulating an optimization problem as a Hamiltonian, thereby *reducing* the problem to computing its spectrum, the complexity of the problem can analyzed by the time it takes to find ground state. The problems for which it is believed there exists no efficient algorithms for computing ground states belong in the class **NP**. As in the standard literature, we define the complexity class **NP** to contain all the decision problems for which a solution can be verified by a non-deterministic Turing machine in polynomial time. It is conjectured that the classes $\mathbf{P} \neq \mathbf{NP}$, suggesting that **P** does not contain *all* problems, although no proof has been found yet. However, among the theoretical computer science and mathematics communities, it is widely believed that $\mathbf{P} \neq \mathbf{NP}$, due to many decades searching for poly-time algorithms for **NP**-problems, with no success. Naturally, one should ask: if it's true that $\mathbf{P} \neq \mathbf{NP}$, can quantum algorithms at least provide a speed-up over classical algorithms with exponentially long run-times? In this work, we consider this question by considering the ability, or lack thereof, of a broad class of classical and quantum algorithms to solve **NP**-Hard problems.

Throughout the development of quantum computing, new complexity classes have been defined to encapsulate quantum problems and classical problems defined as quantum Hamiltonians. The quantum complexity class hierarchy lives in the setting of probabilistic computation, building on a branch of computer science which defines *bounded-error probabilistic polynomial time* (**BPP**) as the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability bounded by 1/3 for all instances. It is known that all problems in **P** are contained in **BPP**, although many problems in **BPP** are not known to be in **P**. However, it is still conjectured that $\mathbf{P} = \mathbf{BPP}$ [9].

The quantum analog of **BPP** is *bounded-error quantum polynomial time* (**BQP**), which is simply defined by replacing the Turing machine with a quantum algorithm which solves the problem successfully with a probability of 2/3. Finally, the complexity class *Quantum-Merlin-Arthur* (**QMA**) is the natural quantum analog of **MA**, for Merlin-Arthur which is the probabilistic extension of **NP**. Throughout this thesis, we make progress on problems which are known to be intractable for finding exact and approximate

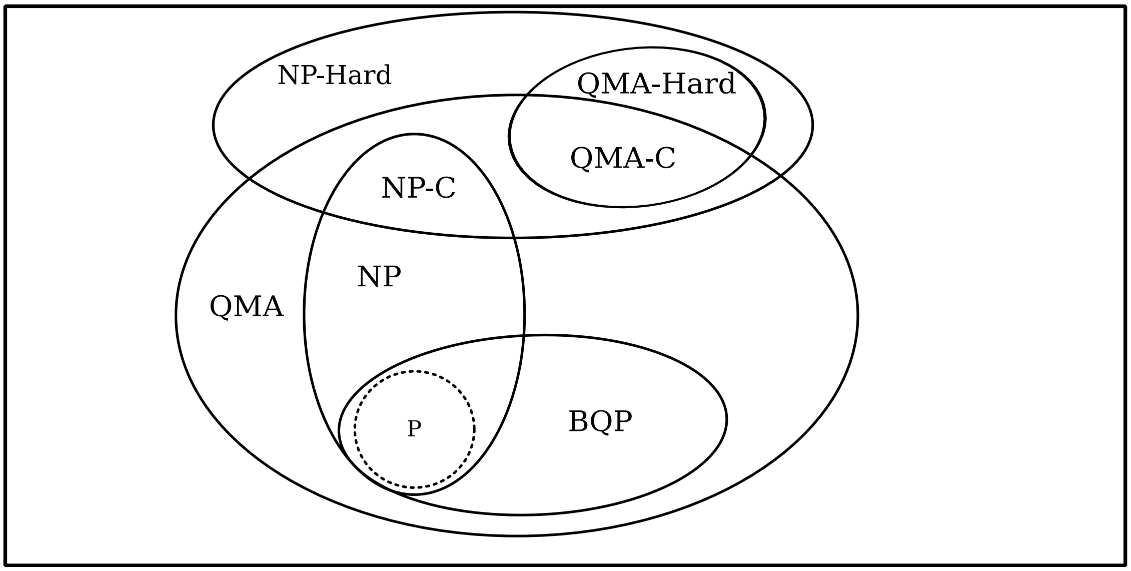


Figure 1.2 Complexity class Euler diagram showing the conjectured overlaps of classical and quantum complexity classes. The dashed lines around the class **P** represent the famous unresolved problem of whether $\mathbf{P} \overset{?}{=} \mathbf{NP}$.

solutions with a classical algorithm.

The k-local Hamiltonian problem is one such problem which encapsulates many efforts considered in this thesis. Formally, the k-local Hamiltonian is the problem of estimating the ground state energy of the a local Hamiltonian in which all interactions are k-local defined in Eq. (1.13). Given the k-local Hamiltonian H defined on n qubits, the problem is to determine (YES) whether H has an eigenvalues less than α, or (NO) all of the eigenvalues of H are larger than β. Hence, a YES-instance of the problem posits the existence of some state $|\psi\rangle$ such that $\langle\psi|H|\psi\rangle \leq \beta$, and a NO-instance is one such that for every $|\psi\rangle$, $\langle\psi|H|\psi\rangle \geq \alpha$. It was shown by [5] that the k-local Hamiltonian problem is in **QMA**. In satisfiability terms, k-local Hamiltonians are the quantum analog of MAX-k-SAT, the problem of finding a solution that maximizes the number of satisfied clauses in a k-CNF formula.

1.3 Quantum algorithms for discrete optimization

Given the hardness of finding exact or even approximate solutions to the local Hamiltonian problem, the natural question of whether quantum algorithms can provide any advantage arises. Beyond the quadratic speed-ups provided by amplitude amplification in Grover's algorithm [10], the search for quantum algorithms that provide exponential or super polynomial speed-ups is still an active area of research at the time of this work in 2024 [11].

Over recent years, two classes of quantum algorithms have been developed, known as *variational quantum algorithms* (VQE) [12] and *quantum adiabatic algorithms* [13, 14]. While many of the algorithms in these two classes lack rigorous performance gaurantees, heuristics commonly outperform the many of the best known classical algorithms, such as simulated annealing. One of the algorithms that has shown promise is known as the *quantum approximate optimization algorithm* (QAOA). Although QAOA has several variations, here we introduce the initial algorithm that was proposed by Farhi et al. in 2014 [15].

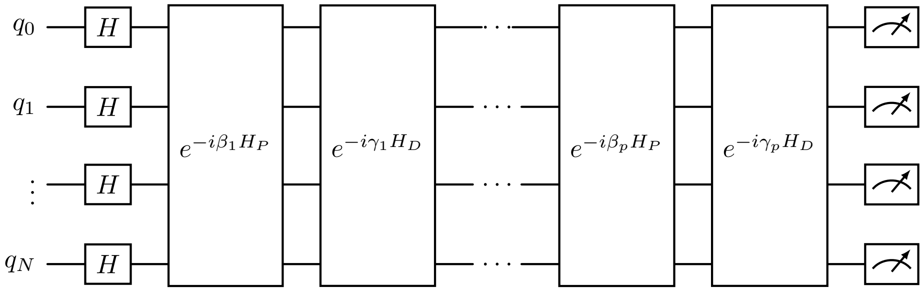


Figure 1.3 Quantum circuit diagram for p-layer QAOA, in the purest form of the algorithm. The N qubit state is initialized to the uniform superposition state using an initial layer of Haddamard gates. One layer of QAOA consists of the application of H_P and H_D layers respectively. After p layers, the final state is measured in the computational basis to obtain a solution in the form of a bitstring.

QAOA starts by initializing an n-qubit superposition state, which can be performed by a layer of Haddamard gates on each qubit. Following the layer of Haddamards, the algorithm then applies alternating rounds of a diagonal problem Hamiltonian denoted by H_P and a mixing Hamiltonian H_D. A common mixing operator is a transverse field in the X direction that acts on single qubits. By fixing the circuit depth, QAOA can be defined to a particular number of alternating problem and mixing Hamiltonian layers, which is often denoted as p. After obtaining the superposition state $|+\rangle$ and applying p layers of H_P and H_D, the state obtained can be written as

$$|\psi_{\gamma,\beta}\rangle = e^{-i\beta_p X} e^{-i\gamma_p H_P} \ldots e^{-i\beta_1 X} e^{-i\gamma_1 H_P} |+\rangle. \tag{1.31}$$

Supposing the parameters β and γ were chosen appropriately, QAOA aims to maximize the overlap between the state in Eq. 1.31 and the ground state or optimal solution to the problem Hamiltonian.

The Hamiltonian, H_P, provided to QAOA can be an arbitrary optimization problem expressed as a diagonal Hamiltonian. Combinatorial graph optimization problems are natural candidates for QAOA. Indeed, early work on QAOA has explored **NP**-hard graph problems such as Max-Cut [16], and maximum independent set [17]. Recent advances in reconfigurable atom arrays [17] and trapped ion quantum computing devices [18] have also provided practical routes to experimental demonstrations of adiabatic quantum algorithms such as QAOA [19].

Given a constant depth QAOA circuit, recent work has explored the task of finding optimal parameters by analyzing the performance of the algorithm with different total run-times discretized by various choices of time-steps dt [20, 21]. Although QAOA has not been rigorously proven to have advantage over classical algorithms, results have shown that by taking the limit of system size $N \to \infty$, optimal parameters can be found [21]. For circuit depths that are increasing with system size, finding optimal parameters is more challenging because taking $N \to \infty$ corresponds to infinite depth quantum circuits. We discuss this problem in more detail in Chapter 5.

A close relative to the QAOA algorithm is adiabatic quantum computation (AQC), also known as the adiabatic quantum algorithm. Here, AQC is constructed by defining H_P and H_D similar to QAOA. The algorithm runs by interpolating between the two Hamiltonians along a path such that the total Hamiltonian can be written as

$$H(s) = (1-s)H_D + sH_P, \tag{1.32}$$

where $H(0) = |+\rangle$ and $H(1) = |z^*\rangle$. The total runtime of the algorithm is dominated by the minimum spectral gap Δ_{min}^{-1} of $H(s)$. However, when Δ_{min} is on the order of $exp(-N)$, this induces an exponentially increasing runtime with the size of the problem.

While we keep $H(s)$ general here, one may define two functions $f(t)$ and $g(t)$ that act by defining the total Hamiltonian $H(t)$ in the following manner

$$H(t) = f(t)H_P + g(t)H_D. \tag{1.33}$$

The initial conditions at $t = 0$ and final conditions at total run-time T are defined as $f(0) = g(T) = 1$, and $f(T) = g(0) = 0$. We remark that more sophisticated routines can be used to optimize the way that AQC smoothly interpolates between these functions [22].

As the goal of QAOA is to optimize a cost function H_P over bitstrings, it is natural to ask whether starting QAOA from a good configuration increases the performance of the algorithm. Such *warm start* protocols for QAOA have been explored numerically and analytically at constant and near linear circuit depth. In the standard formulation of QAOA, the state is initialized in a uniform superposition $|+\rangle$. Starting QAOA using in the computational basis state that represents a good string showed no improvement in the algorithms convergence at constant and sub-linear circuit depths [23]. In Chapter 4, we show how to *improve* trotterized adiabatic quantum computation (TAQC) algorithms, using the bits in the classical strings to set parameters in a time-dependent Hamiltonian. We also discuss how our algorithm is fundamentally different from warm start methods in Chapter 4.

1.4 Quantum resources for large speed-ups

While we introduced the key mathematics of quantum mechanics in the previous sections, we emphasize that quantum algorithms are also based upon key physical phenomena which provide resources for attaining computational advantages over classical algorithms. Specifically, *quantum entanglement*, *interference*, exponentially large search spaces via *superposition* of states, and *quantum tunneling effects* are all fundamental resources which quantum algorithms can exploit to reach solutions using methods not otherwise possible for classical algorithms.

The first resource, entanglement, is a property of quantum information that has no classical analog. An entangled quantum state is a superposition of two or more variables, where the amplitudes corresponding to each variable are correlated such that a measurement of one variable changes the probability distribution of the other variable. In other terms, quantum entanglement is well-defined by states that are not separable. Given the composite state ρ_{AB} of two systems ρ_A, and ρ_B, we say the state is separable if and only if it can be written in the form

$$\rho_{AB} = \sum_i p_i |\varphi_{iA}\rangle\langle\varphi_{iA}| \otimes |\phi_{iB}\rangle\langle\phi_{iB}|. \tag{1.34}$$

If the state cannot be written in this separable form, we say it is entangled. The extent to which a pure bipartite state is entangled, is commonly measured by the von Neumann entropy of its subsystem

$$S(|\psi_{AB}\rangle) = -\mathrm{Tr}\rho_A \log(\rho_A) = -\mathrm{Tr}\rho_B \log(\rho_B). \tag{1.35}$$

To define many-body entanglement for greater than two systems, the definition of separable states generalizes to N parties. However, it may be the case that two subsystems are entangled with each other, but not the third party, for example, which makes defining mult-partite entanglement a challenging task. We leave further discussion on many-body entanglement in the following reference: [24]. Quantum entanglement has also been characterized as a hardness mechanism for classical simulation methods such as density matrix renormalization group (DMRG) [25], tensor networks [26], and other variational methods based on neural networks [27]. We emphasize that quantum algorithms that exploit highly entangled dynamics are often not classically simulable[1].

The second resource that makes quantum algorithms fundamentally distinct from classical algorithms is the phenomena of quantum interference. Observed in early works such as the double slit experiment [28], quantum interference is the strange ability of the quantum wavefunction to interfere not only with other wavefunctions representing the dynamics of other particles, but itself. In other words, even the

[1] Classicaly simuable in this context refers to the ability of classical methods to simulate the dynamics of the quantum algorithm to large system size N. Any quantum algorithm that is classically simuable, is therefore not expected to provide quantum advantage. We note that classical simulable does not refer to the small system size simulations which can almost always be performed, provided N is sufficiently small.

wavefunction of a single particle can interfere with itself, thereby manipulating the probabilities of observing the particle into high and low density areas. These high and low density of probabilities represent the concepts of constructive and destructive interference respectively. When quantum algorithms exploit these effects, more favorable solutions can be found with higher probability as a result of constructive interference [29]. Destructive interference, on the other hand can be burdensome to the dynamics of a quantum algorithm. We note the latter resource, quantum tunneling, is intimately connected to interference, where constructive interference provides a mechanism that allows for macroscopic quantum tunneling through large energy barriers [29].

Lastly, exponentially large Hilbert spaces provide a natural exponential advantage in the ability of an n-qubit state to encode 2^n complex amplitudes that act upon a basis of dimension 2^n. Conversely, a classical n-bit state only has the representational power of a single state. The power to represent exponentially many more possible states, can be interpreted as the power to search an exponentially larger space of solutions, when the quantum state is not confined to a particular symmetry sector. Superposition is related to the quantum parallelism, which we can interpret in accordance to the application of quantum algorithms searching across many states, or possible solutions, at once.

While these fundamentally-quantum resources provide mechanisms for quantum algorithms to achieve advantage, *decoherence* on real devices provides a challenge for demonstrating the full capabilities of quantum algorithms in the NISQ era. Decoherenece occurs as a result of unwanted noise from the environment effecting the dynamics of the algorithm. While the state of an isolated system maintains perfect coherence, quantum simulations performed on any physical hardware are inevitably effected by noise from the external environment. The dynamics of the quantum algorithm can be treated as a system-environment interaction, where the desired dynamics are represented as the primary system, and the external environment is treated as a bath which is coupled to the primary system. This interaction results in a loose coupling to the environment, where unwanted entanglement can be generated between the two systems, thus effecting the solutions obtained in the final measurement of the circuit. The effect of decoherence therefore limits the circuit depths we can run on real quantum hardware, where error proliferates throughout the dynamics eventually resulting in a completely de-cohered state. In the NISQ era especially, but also in the development of future quantum algorithms, low circuit depths may avoid the inevitable effects of decoherence. We emphasize that there exists a trade-off, where quantum algorithms with a small time-step dt may avoid error terms from Trotter decomposition, while at the same time, too small of a time-step dt results in deep circuits which are more vulnerable to the effects from noise. While all the simulations done in this work are performed on a classical computer with no real or simulated noise, we note that future experimental demonstrations of our algorithms would benefit from the consideration of quantum error correction and quantum error mitigation. In the next section, we outline the contributions in this thesis, and the author's specific role in each work.

1.5 Contributions and thesis outline

In Chapter 2, we introduce methods for time-dependent Hamiltonian simulation under the gate model of quantum computing. We provide an accessible example, which is relevant to constructing gate-based simulations of the quantum algorithms later in the thesis. Following these technical methods, in Chapters 3 and 4, we include two primary results by the author, Brandon A. Barton, and his colleagues.

In Chapter 3, we make progress on the approximation of the MAX-3-XORSAT problem with a quantum adiabatic inspired algorithm. We introduce a new method called *spectrally folded quantum optimization* (SFQO) which delivers a new method for approximating **NP**-Hard p-spin glass problems which are known to be exponentially hard to solve exactly or even approximately. In this work, the author, Brandon Barton developed software for quantum simulation, performed extensive benchmarking for the SFQO algorithm and previous methods – trotterized adiabatic quantum computation (TAQC), and contributed to a journal article currently in review at Physical Review X (PRX) Quantum. Chapter 3 is the based off the original manuscript submitted to PRX Quantum. The analytic performance predictions performed by co-authors Eliot Kapit and Vadim Oganesyan in this work are included in Appendix B for the reader's reference.

In Chapter 4, we further explore the MAX-3-XORSAT CSP. We introduce a new quantum algorithm called *Iterative Symphonic Tunneling* for Satisfiability problems (IST-SAT) which provides a new quantum steering mechanism for adiabtic quantum algorithms such as QAOA to find exact and approximate solutions to traditionally hard problems. In this work, the author's Brandon A. Barton contributions consist of: conceptualization the IST-SAT algorithm in collaboration with co-author Eliot Kapit, development of software for quantum simulation alongside co-first-author Jacob C. Sagal, execution of quantum simulations on high performance computing (HPC) classical devices for the IST-SAT algorithm and it's input model–TAQC. The results for the classical input algorithms to IST-SAT were performed by Jacob C. Sagal. In addition, the author lead the writing of the initial manuscript, which is currently under revision to be submitted to *Physical Review Letters* (PRL).

Following the two main results presented in this thesis, in Chapter 5, we conclude by connecting the results from Chapters 3 and 4. We discuss how the novel mechanisms presented in SFQO and IST-SAT may be combined together in order to demonstrate exponential speed-ups over the best known quantum and classical algorithms for the problem we consider. Finally, we conclude and discuss opportunities for future work.

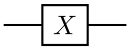
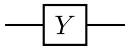
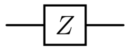
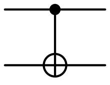
CHAPTER 2

TECHNICAL METHODS

2.1 Gate-based quantum computation

In gate-based quantum computation, the time evolution of a quantum state is described by a *quantum circuit*, consisting of many qubits $q_0, q_1, \ldots q_n$. In this work, qubits are represented as "wires" and unitary operations are represented by gates (denoted by boxes) acting on the qubit wires. In quantum circuits, time marches forward to the right of the page. We note that alternative formulations of gate-based time-evolution exist such as graphical ZX calculus [30], however, we use the circuit wire diagrams used most commonly in the literature. We provide examples of the common quantum gates used throughout this work in Table 2.1.

Table 2.1 Commonly used quantum gates, with one and two-qubit matrix representations, and example input and output quantum states. The state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ is used for example in action column for single qubit gates.

OPERATOR	GATE NOTATION	MATRIX	ACTION
Pauli-X (X)		$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$	$X\|\psi\rangle \to \alpha_1\|0\rangle + \alpha_0\|1\rangle$
Pauli-Y (Y)		$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$	$Y\|\psi\rangle \to -i\alpha_0\|0\rangle + i\alpha_1\|1\rangle$
Pauli-Z (Z)		$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$	$Z\|\psi\rangle \to \alpha_0\|0\rangle - \alpha_1\|1\rangle$
Identity (I)	—I—	$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$	$I\|\psi\rangle \to \alpha_0\|0\rangle + \alpha_1\|1\rangle$
Haddamard (H)		$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$	$H\|\psi\rangle \to \alpha_0\frac{\|0\rangle+\|1\rangle}{\sqrt{2}} + \alpha_1\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$
Controlled-NOT (C-NOT)		$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$	$CNOT\|00\rangle \to \|00\rangle,$ $CNOT\|01\rangle \to \|01\rangle,$ $CNOT\|10\rangle \to \|11\rangle,$ $CNOT\|11\rangle \to \|10\rangle$

Unitary evolution of the state $|\psi\rangle$ has equivalent in mathematical and quantum circuit diagram representations. For example, given two arbitrary unitary matrices U_1, U_2 applied to the state $|\psi\rangle$, we denote this operation as

$$U_2 U_1 |\psi\rangle = U_2 |\phi\rangle, \quad |\phi\rangle = U_1 |\psi\rangle. \tag{2.1}$$

The corresponding circuit diagram for Eq. (2.1) is provided in Figure 2.1.

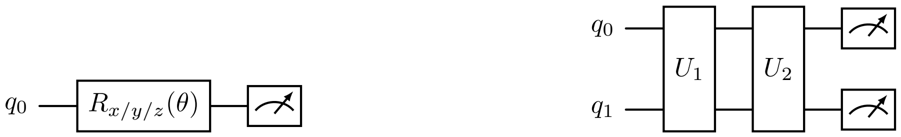


Figure 2.1 Examples of two quantum circuits. (left) A single qubit rotation gate in the x, y or z basis with angle θ, followed by a measurement gate. (right) a general two-qubit quantum circuit for two unitary applications U_1 and U_2, followed by two single qubit measurement gates.

2.2 Numerical methods for time-dependent Hamiltonian simulation

Throughout the work presented in Chapters 3 and 4, we consider quantum algorithms governed by a time-dependent Hamiltonian $H(t)$. Here, we introduce the numerical methods for simulating $H(t)$ using gate-based quantum computing. We showed in Chapter 1 that when H is time-independent, the solution to the Schrodinger equation is given by

$$|\psi(t)\rangle = \exp(-iHt)|\psi(0)\rangle. \tag{2.2}$$

Therefore, we might expect that when $H(t)$ varies with time, the solution is given by the integral

$$|\psi(t)\rangle = \exp\left(-i \int_0^t dt' H(t')\right) |\psi(0)\rangle. \tag{2.3}$$

However, this solution is not correct unless $[H(t), H(t')] = 0$ for all times $\{t, t'\} \in T$. If we consider the limit as $dt \to 0$, we can still write the solution in the form of Eq. 2.2 as

$$|\psi(t + dt)\rangle = \exp(-iH(t)dt)|\psi(t)\rangle = (1 - iH(t)dt + O(dt^2((|\psi(t)\rangle. \tag{2.4}$$

By slicing the total evolution time T into S slices, and taking the limit as $S \to \infty$, we can find a general solution given by

$$|\psi(t)\rangle = \left[\lim_{S\to\infty} \prod_{k=1}^{S} \exp\left(-iH\left(\frac{kt}{S}\right)\frac{t}{S}\right)\right] |\psi(0)\rangle, \tag{2.5}$$

given an initial state $|\psi(0)\rangle$ at $t = 0$. The expression in Eq. 2.5 provides a *time-ordered product*, which can be interpreted as an iterative application of $H(t)$. We note that this method can be useful when numerically integrating the Schrodinger equation for a small number of particles. More sophisticated numerical integration techniques such as Runge-Kutta methods can also make numerical integration more accurate.

In the gate-based model of quantum computing, simulating the dynamics of a time-dependent Hamiltonian is often straight-forward when there exists a sequence of gates that construct the Hamiltonian operator $H(t)$ over the total evolution time $(t = 0) \to (t = T)$. To provide a concrete example, we can consider the transverse-field Ising model, a standard work-horse in quantum many-body physics, given by the Hamiltonian

$$H_{TFIM} = J \sum_{\langle i,j \rangle}^{N} Z_i Z_j + \kappa \sum_{i=1}^{N} X_i. \tag{2.6}$$

Here, $\langle i, j \rangle$ denotes interactions between nearest neighboring spins, J is the Ising interaction or spin-coupling strength, and κ is the strength of the transverse-field along X. It is clear that the terms in H_{TFIM} do not commute and the ordering of operators matters. Of course, this Hamiltonian is time-independent as well. To make the H_{TFIM} time-dependent in this example, we consider the addition of an oscillating transverse field along Y given by $H_{ST} = \alpha \sum_i \sin(\omega t) Y_i$. Combining these terms, we obtain the total time-dependent Hamiltonian

$$H(t) = H_{TFIM} + H_{ST}(t). \tag{2.7}$$

We use H_{ST} to denote a *symphonic tunneling* drive term as considered in previous work [31, 32]. We outline the steps for applying simulating a time-dependent Hamiltonian using the example in Eq. 2.7.

Before applying any Hamiltonian operator, qubits are initialized in the all-zero state $|0\rangle^{\otimes N}$. In the algorithms we consider in this work, we then initialize the state $|\psi\rangle$ to the uniform superposition by applying a Haddamard gate to each qubit. To simulate the dynamics of Eq. 2.7, we then apply a sequence of quantum gates in a layered manner by applying each term in the Hamiltonian at each time-step.

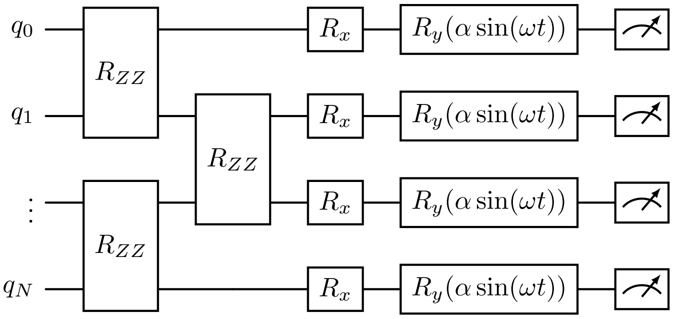

q_0
q_1
$\vdots$
q_N
R_{ZZ}
R_{ZZ}
R_{ZZ}
R_x
R_x
R_x
R_x
$R_y(\alpha \sin(\omega t))$
$R_y(\alpha \sin(\omega t))$
$R_y(\alpha \sin(\omega t))$
$R_y(\alpha \sin(\omega t))$

Figure 2.2 Quantum circuit for simulating a time-dependent transverse field Ising Hamiltonian in one-dimension, with an oscillating Y perturbation. This circuit initializes all the qubits to zero, applies the Ising interaction R_{ZZ} to all nearest neighboring qubits, the transverse field R_x, and the oscillating perturbation $R_y(\alpha \sin(\omega t)$. This sequence is repeated for every time step, until a measurement is applied to all qubits at the end of the simulation.

The following quantum circuit looks similar to that of "vanilla" QAOA in Figure 1.3, without the Haddamard gates to initialize the uniform superposition. When the terms of the Hamiltonian are native gates, the gate-based model for quantum computation is rather straight-forward. When the Hamiltonian contains more complex terms outside of the gate set in Table 2.1, more sophisticated techniques must be used to decompose the complex operator into smaller gates. However, the work presented in Chapters 3 and 4 comprise of Hamiltonians native to the gate set of most quantum computers. More sophisticated quantum simulation techniques are relevant, though beyond the scope of this thesis.

2.3 Implementation of Trotterized adiabatic quantum computation

In the two main works in this thesis, extensive computational simulations were performed on a combination of high-performance computing resources. Small simulations were performed on the author's laptop[2], while larger system-size simulations were performed on a combination of group work-stations, the Mines Wendian[3] HPC, and the *Fujitsu Quantum Simulator*[4] (FQS). To simulate the dynamics of the quantum algorithms presented in this work, we use exact state-vector simulations following the principles we present in the example of the transverse field Ising model with an oscillating field (see Figure 2.2). For the problems we consider in this work, we use similar methods for quantum simulation where the total Hamiltonian H can be decomposed into a sum of terms given by

$$H = H_D + H_P. \tag{2.8}$$

Here, H_D is the driving or "mixing" Hamiltonian, H_P is the problem Hamiltonian of interest to optimization. In Chapter 4, we consider the introduction of a time-dependent term $H_{ST}(t)$ to the total Hamiltonian, which makes the total Hamiltonian, therefore, also time-dependent. In a similar manner to AQC, we also consider the addition of time-dependent interpolation functions giving a total general Hamiltonian

$$H(t) = f(t)H_D + g(t)H_P. \tag{2.9}$$

[2] An Acer laptop with 16GB of random access memory (RAM), and 512GB of storage. Large amounts of data were stored on external devices including 4TB external hard-drives, and cloud storage service platforms.

[3] The Mines Wendian HPC is a high performance computer consisting of 87 compute nodes. These nodes are comprised of nine Nvidia GPU nodes, two OpenPower nodes, while the remaining nodes are SkyLake Intel processors. Simulations submitted to Wendian were submitted as jobs using the SLURM scheduler.

[4] The results in this thesis were obtained from a super-computer developed by Fujitsu Limited. The Fujitsu Quantum Simulator is a 39-qubit CPU-based state vector quantum simulator system consisting of 512 FX700 nodes using A64FX processors. The simulator utilizes a software package called *Qulacs* [33] developed by Osaka University and the QunaSys Corporation. This work was performed using Fujitsu's enhanced version of Qulacs, which uses message passing interface (MPI), and scalable vector extension (SVE) to execute operations in a parallel and distributed manner. This enhanced version of Qulacs optimizes memory bandwidth by distributing the state vector over several nodes, allowing for large-system-size simulations. These specifications were provided by Fujitsu Limited during the 2023 Fujistu Quantum Simulator Challenge. Some of the results presented in this thesis were obtained during and in the months following the competition. We thank Fujitsu for the computational resources the author was granted to after the competition, where valuable simulations were performed with additional parameter sets.

To benchmark previous quantum methods on the problems we consider in this work, we use a hybrid version of AQC and QAOA, called *Trotterized Adiabatic Quantum Computation* (TAQC). This method works by simulating the dynamics of a Trotterized Hamiltonian in the form of Eq. 2.9 through gate-based simulation methods. In the results we present in this work, we perform the simulation of TAQC through exact state-vector simulation using a software for gate-based quantum simulation called *Qulacs* [33]. Quantum circuits can be constructed in Qulacs using the following code written in the Python programming language:

Listing 2.1: Example of a basic quantum circuit that creates the uniform superposition state $|+\rangle^{\otimes N}$ using the Qulacs package written in the Python programming language.

```python
from qulacs import QuantumState
from qulacs import QuantumCircuit

# Construct a basic four-qubit circuit
num_qubits = 4
circuit = QuantumCircuit(num_qubits)
state = QuantumState(num_qubits, use_multi_cpu=False)

gate_list = []  # Empty list of gates that comprise circuit

# Construct a basic circuit which applies a H-gate to each qubit
for i in range(num_qubits):
    gate_list.append(H(i))

# Concatenate list of gates into the circuit
for gate in gate_list:
    circuit.add_gate(gate)

# Apply circuit to the state
circuit.update_quantum_state(state)

# Sample the state in the z-basis
samples = 1000
sample_result = state.sampling(sampling_count=samples, random_seed=1234)
```

In Listing 2.1, the state-vector is the primary mathematical object of interest, and quantum circuits are a sequence of quantum gates that act on the state as a unitary matrix. This matrix operation can be completed as a sparse matrix-vector multiplication, where the sparsity of the unitary matrix is determined by the gates in the circuit. For exact state-vector simulation, we can consider measurements of a quantum state as sampling of state vector, since the square-magnitude of all the complex amplitudes $\sum_{j=1}^{2^N} |\alpha_j|^2 = 1$ is a probability distribution given by the Born rule. For other software packages, not specifically designed for exact state-vector simulation, measurements are typically denoted by measurement *gates* (see Figure 2.1). By drawing a sufficient amount of samples from the state-vector simulation, the discrete probability distribution of samples provides an approximation to what measurement(s) are expected when a measurement is applied to an evolving quantum state.

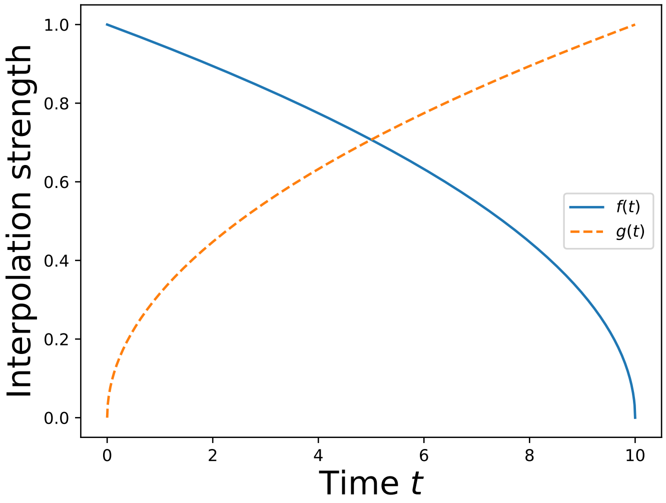


Figure 2.3 Adiabatic ramping functions $f(t)$ and $g(t)$ in the solid and dashed lines respectively. The driving Hamiltonian interpolation function starts at $f(0) = 1$, and is ramped down to the final condition $f(T) = 0$, while the problem Hamiltonian is ramped up from $g(0) = 0$ to $g(T) = 1$. In this figure, we use the interpolation functions $f(t) = \sqrt{1 - t/T}$ and $g(t) = \sqrt{t/T}$, although other suitable choices may also be used depending on the problem.

Listing 2.2: Functions to construct layers of Hamiltonian operations written in Qulacs. The functions implement the interactions from the spin Hamiltonians (H_P, H_D) in the Pauli basis. Here, we use a transverse field Ising model in one dimensional open-boundary condition chain as the problem Hamiltonian H_P with $Z_i Z_{i+1}$ nearest neighbor interactions in 2.6. The full circuit for TAQC can be constructed using the "build full circuit" function, which implements the application of the total Hamiltonian for each time-step.

```
# Haddamard layer
def Hlayer():
    gate_list = []
    for i in range(self.num_qubits):
        gate_list.append(H(i))
    return gate_list

# Problem Hamiltonian H_P layer
def Hp_layer(ang):
    gate_list = []
    for i in range(num_qubits):
        gate_list.append(RZZ([i,(i+1)%(num_qubits-1)], ang))
    return gate_list

# Driving Hamiltonian H_D layer
def Hd_layer(ang):
    gate_list = []
    for i in range(num_qubits):
        gate_list.append(RX(i, ang))
    return gate_list

def build_full_circuit():
    gate_list = []
    gate_list.extend(Hlayer())
    while(t<T):
        gate_list.extend(self.Hp_layer(g(t)))
        gate_list.extend(self.Hd_layer(f(t)))
    for gate in gate_list:
        circuit.add_gate(gate)
         t = t+dt
```

We can also use Python to implement the interpolation functions $f(t)$ and $g(t)$ which are used to ramp up/down the problem Hamiltonian H_P and the driving Hamiltonian H_D. At each time-step, we can apply a layer of gates which comprise H_P, followed by a layer of single qubit rotations which comprise H_D.

Listing 2.3: Full circuit for TAQC. The algorithm can be implemented by choosing a total evolution time T, a time-step dt, a total number of qubits. Using Qulacs, we construct a quantum QuantumCircuit, and initialize the quantum state using the QuantumState function. For large number of qubits, the state-vector can be distributed over multiple CPUs. To run the TAQC algorithm, the full circuit is constructed, and applied to the state using the "update quantum state" function. Following the application of the circuit to the state, measurements are obtained by sampling the circuit with the "sampling" function.

```
# Construct a TAQC circuit for a 4-qubit TFIM
T = 10 # Total evolution time
dt = 0.1 # Time step
num_qubits = 4
circuit = QuantumCircuit(num_qubits)
state = QuantumState(num_qubits, use_multi_cpu=False)

gate_list = []  # Empty list of gates that comprise circuit

gate_list.extend(build_full_circuit())
for gate in gate_list:
    circuit.add_gate(gate)

# Apply circuit to the state
circuit.update_quantum_state(state)

# Sample the state in the z-basis
samples = 1000
sample_result = state.sampling(sampling_count=samples, random_seed=1234)
```

Once measurements are obtained, the distribution of samples can be analyzed. The expectation value of the state obtained with the problem Hamiltonian gives an energy $E = \langle\psi|H_P|\psi\rangle$. When measuring in the z-basis, the measurements of $|\psi\rangle$ will be a binary string (bitstring). By sampling the state produced from exact state-vector simulation, average energies $\langle E\rangle$ can be obtained. By normalizing the average energy by the ground state energy $\langle E\rangle/E_{\text{GS}}$, system size effects are accounted for when the problem energy of H_P is normalized to $-N$ (see Chapter 3).

In Chapter 3 and 4, we simulate our novel quantum algorithms using similar methods as we present here for TAQC. We note that these simulation methods could be improved by considering more sophisticated methods of gate placement that reduce circuit depth. Quantum gates can be implemented in parallel at the same time step, or layer of the circuit. For example, a Z_1Z_2 gate can be completed in parallel to a Z_3Z_4 gate, as these two gates act on a disjoint set of qubits. To experimentally implement the circuits for TAQC, or the quantum algorithms we present is this thesis, specific hardware platform constraints should be further considered.

CHAPTER 3

ON THE APPROXIMABILITY OF RANDOM-HYPERGRAPH MAX-3-XORSAT PROBLEMS WITH QUANTUM ALGORITHMS

This chapter is based on a manuscript under review at *Physical Review X (PRX) Quantum.* The excerpts below reflect my primary contributions to the work. Additional materials from the manuscript are included in Appendix B. The authors of the manuscript are listed below.

Eliot Kapit[1,2,], Brandon A. Barton[2,3], Sean Feeney[2], George Grattan[2,4], Pratik Patnaik[3],
Jacob Sagal[2], Lincoln D. Carr[1,2,3], Vadim Oganesyan[5,6]

3.1 Abstract

Constraint satisfaction problems are an important area of computer science. Many of these problems are in the complexity class NP which is exponentially hard for all known methods, both for worst cases and often typical. Fundamentally, the lack of any guided local minimum escape method ensures both exact and approximate classical optimization are hard, but the intuitive mechanism(s) for approximation hardness in quantum algorithms are poorly understood. For algorithms simulating Hamiltonian time evolution, we explore this question using the prototypically hard MAX-3-XORSAT problem class. We conclude that the mechanisms for quantum exact and approximation hardness are fundamentally distinct. We qualitatively identify why traditional methods such as quantum adiabatic optimization are not good approximation algorithms. We propose a new spectral folding optimization method to escape these issues, and study it analytically and numerically. We consider random rank-3 hypergraphs including extremal planted solution instances, where the ground state satisfies a much higher fraction of constraints than truly random problems. We show that, if we define the energy to be $E = N_{\mathrm{unsat}} - N_{\mathrm{sat}}$, spectrally folded quantum optimization will return states with $E \leq q_a E_{\mathrm{GS}}$ (where E_{GS} is the ground state energy) in polynomial time, where conservatively, $q_a \simeq 0.6$. This is in contrast to $q_a \to 0$ for the hardest instances with classical searches. We thoroughly benchmark variations of spectrally folded quantum optimization for random classically approximation-hard (planted solution) instances in simulation, and find performance consistent with this prediction. We do not claim that this approximation guarantee holds for all possible hypergraphs,

[1] Department of Physics and Quantum Engineering Program, Colorado School of Mines, 1523 Illinois St, Golden CO 80401

[2] Quantum Engineering Program, Colorado School of Mines, 1523 Illinois St, Golden CO 80401

[3] Department of Applied Mathematics and Statistics, Colorado School of Mines, 1500 Illinois St, Golden CO 80401

[4] Department of Computer Science, Colorado School of Mines, 1500 Illinois St, Golden CO 80401

[5] Department of Physics and Astronomy, College of Staten Island, City University of New York, Staten Island, NY 10314, USA

[6] Physics program and Initiative for the Theoretical Sciences, The Graduate Center, City University of New York, New York, NY 10016, USA

though our algorithm's mechanism can likely generalize widely. These results suggest that quantum computers are more powerful for approximate optimization than had been previously assumed.

3.2 Introduction

In combinatorial optimization, the class of constraint satisfaction problems (CSPs) are well-studied, and known for being hard [34]. These problems instances are capable of representing a large class of problems in applications ranging from from machine learning, cybersecurity, portfolio optimization, and other high dimensional problems where one does not expect to find optimal solutions, but can rather be satisfied with a good approximation. In the large N limit of these problems, providing an algorithm that demonstrates a speed-up over the best known algorithms is thus highly sought after.

Constraint satisfaction problems can be understood as a set of N variables that participate in a large collection of M few-body constraints. The energy landscape of these problems tend to be rough, with an abundance of many local-minima that scale exponentially with N. Furthermore, the cost functions of these problems are often equivalent to the Hamiltonian *disordered spin glass*, where the hardness of the problem lies in the inability to escape local minima by only changing a few variables at each step of optimization [35–46]. Thus, finding the global optima, or any low-energy solution is said to be an exponentially difficult task for optimization algorithms, both for classical and quantum methods to date.

In the hardest of cases, CSPs can be both hard to solve exactly and approximately, where an approximate solution is defined as any constant fraction of the global optima [47]. Thus, when there is no hope to find exact solutions, attempts to improve such *approximation ratios* can characterize a large body of approximate algorithm works. Some examples include approximation ratios which have been proven theoretically for problems such as Max-Cut and quantum Max-Cut (QMC). These approximation ratios can be theoretically guaranteed using classical approximation methods such as semi-definite programming (SDP) based on Lasserre's and sum-of-squares hierarchies [48]. Naturally, many works have asked the question of whether quantum approximate algorithms can provide speed-ups over classical methods? Progress over recent years have introduced quantum methods including adiabatic quantum computing [49, 50], analog quantum annealing [51–57], and a diverse set of quantum approximate optimization algorithms (QAOA) [15]. Recent work has also combined classical methods with variational quantum algorithms, by using the classical method to set parameters in the quantum algorithm, thus providing a lower-bound on performance.

While previous methods discussed here have made promising progress, the exponential difficulty scaling of these hard spin glass problems remains, and is widely believed to hold true, yet no guarantees can be made unless the **P≠NP** conjecture can be proven. In previous work, the spin glass Hamiltonian which is typically diagonal in the z-basis, is combined with a transverse field, typically written as a uniform field along the x direction. The transverse field term, turns the problem into a quantum Hamiltonian, which can be used to induce macroscopic quantum tunneling events that guide the system to good solutions by escaping large energy barriers, typically insurmountable by a classical algorithm that makes local steps along the energy landscape. In experimental demonstrations, polynomial speed-ups have been observed for locally connected graphs, with short range connections [17, 57–60]. However the search for quantum algorithms that demonstrate beyond quadratic speed-ups for optimizations problems in **NP** still remains a outstanding problem in the NISQ era of quantum computing [61].

In this work, we investigate the approximation hardness of CSPs from a physically motivated point of view, with the goals of identifying the mechanisms for hardness, and opportunities to obtain exponential quantum advantage. In particular, we choose to investigate a well-known problem known as MAX-3-XORSAT defined in Section (3.3), and present the following arguments in this work:

- The hardness characteristics associated with finding the ground state of a given problem Hamiltonian H_P can be identified with heuristic quantum methods. Such mechanisms of hardness are furthermore not likely to be surmounted in the worst case.

- Classical algorithms that optimize by local steps are distinct from way quantum algorithms approach solutions via tunneling effect. Hence, mechanisms for hardness are distinct between classical and quantum methods.

- Quantum algorithms such as AQC and QAOA may still not be effective methods for returning low-energy approximations in polynomial time, in the worst case.

- The method presented in this work called *spectrally folded quantum optimization* (SFQO) provides intuition to the hardness of approximation for MAX-3-XORSAT. For random hypergraphs, SFQO provides an exponential speed-up for returning approximate solutions.

The arguments presented in this work are supported with theoretical analysis and numerical simulations, out to the large N, where asymptotic exponential scaling is observed in classical algorithms. We thoroughly benchmark classical methods based on local updates, trotterized adiabatic quantum computation based (TAQC) methods, and SFQO, the new method presented in this work. We show that classical methods and TAQC do not provide meaningful quantum advantage for finding exact or approximate solutions on the hard MAX-3-XORSAT instances we consider in this work. Our results demonstrate that SFQO significantly outperforms these methods, by finding approximate solutions in a linear number of cost function evaluations.

The structure of this Chapter is as follows: In Section 3.3, we review previous work, discuss hardness of approximation for MAX-3-XORSAT, and introduce the problem with a detailed construction. In Section 3.4, we introduce the spectrally folded quantum optimization and propose two variations of the algorithm. We include analytical performance predictions performed by the author's colleagues (E.K. and V.O.) in Appendix B, that establish a rigorous approximation ratio for SFQO. Our benchmarking results are presented in Section 3.5.3 for all methods we consider in this work. Finally, in Section 3.7, we conclude and provide an outlook for future work.

3.3 Problem definition and previous work

In this Chapter, we discuss the inability of previous methods to find exact or approximate solutions at large system size N, identify mechanisms for quantum solution hardness, and review the approximation hardness for AQC and QAOA-like algorithms.

We first establish the qualitative reasons for why classical and quantum approaches to the MAX-3-XORSAT and other **NP**-hard problems are unable to efficiently produce exact or even approximate solutions. A characteristic of hardness for both classical and quantum methods for these problems lies in the highly non-convex energy landscape, which contains a high density of poor quality local minima. This characteristic creates a high "clustering" of local minima, which we describe as a rough energy landscape, with several neighboring local minima and elusive ground states which are not easy to find, even when given a good initial starting point. For classical methods, there does not exist a known mechanism to escape local minima as it is not possible to know how far the optimization process is away from the true ground state(s). Methods such as stochastic gradient descent and simulated annealing furthermore, do not overcome this issue due to the exponential scaling of poor local minima. For quantum methods, this rough problem energy landscape does provide a challenge for exact optimization, although we argue for the existence of a hardness separation for approximate optimization between classical and quantum approaches.

In this work, we are concerned primarily with an approximation guarantee, defined by a fraction of the optimal solution, also known as an *approximation ratio.* We are interested in producing an approximation ratio in addition to observing the performance of the algorithm in-practice. For random problems, the ground state energy can often be predicted using arguments from statistical physics, such as the Parisi solution to the Sherrington-Kirkpatrick (SK) model [62]. However, this prediction is not sufficient for predicting extremal behavior of random problems, which might contain global optima far away from the predicted value in the worst case. Thus, producing a solution close to a predicted ground state may be good in practice or on average, but this performance does not satisfy all problem instances. Furthermore, the existence of an efficient classical algorithm for predicting this extremal problem characteristic would imply **P**=**NP** [47].

Due to the exponential scaling of poor local minima producing a “clustering” of sub-optimal solutions, the basin of attraction to the ground state(s) in **NP** combinatorial optimization problems represents an exponentially small fraction of the total energy landscape. This phenomenon introduces a what has been referred to as “entropic barriers” [45] to solving the problem, where it becomes more efficient to start the optimization process over from the beginning, once caught in a local minima where local steps in simulated annealing or parallel tempering[63] do not produce any meaningful improvements. We find this mechanism of hardness to be a generic feature of the hardest CSPs, which applies not only to finding the optimal solution(s), but to approximations as well. For classical algorithms, we thus emphasize that approximate and exact problem solving for a given CSP class, are both exponentially difficult in the number of variable (system size). Given the hardness mechanisms we’ve discussed for classical methods, we now ask which of these mechanisms hold true for quantum problem solving methods?

We consider a class of heuristic quantum algorithms derived from quantum annealing, AQC and QAOA. These type of algorithms have been well-studied in the literature, where gate model simulation techniques have provided accessible methods for experimental demonstrations at small N on NISQ devices. These algorithms can also gain polynomial speed-ups by supplementing with Grover type amplitude amplification [19]. In the barest form of these algorithms, the system is initialized in the ground state of a trivial Hamiltonian, usually in the form of a transverse field. Then, the system is slowly evolved via interpolation functions into the problem Hamiltonian H_P, followed by a multi-qubit measurement, producing a desired solution. Other variations of these algorithms can be initialized using a prior known low-energy states of H_P or a planted solution, where collective tunneling is used to find more optimal solutions.

The quantum methods discussed here are fundamentally distinct from classical algorithms in the following two ways: First, the mechanism for quantum advantage, multi-qubit tunneling where energy barriers that trap classical algorithms in local minima can be effectively surmounted. Second, while classical algorithms start at high-energy (sub-optimal) solution and try to optimize via local updates, quantum algorithms start below the energy of the problem Hamiltonian and attempt to tunnel into the global minimum or other low-energy states through the interpolation process.

However, in the search for promising applications of quantum technologies, quantum optimization algorithms based on macroscopic tunneling tend to provide only modest speed-ups, leaving the applications of Hamiltonian simulation a leading candidate for quantum advantage. In the worst case, the macroscopic tunneling rate into the ground state of **NP** problems decays exponentially with system size. This expectation implies an exponential run-time of such algorithms, which removes the quantum advantage obtained from such methods. Often times, these tunneling rates can be computed via Nth order perturbation theory providing analytic expressions for the convergence of the algorithm. For p-spin problems, this exponential decay may sometimes be circumvented by introducing additional terms into the time-dependent Hamiltonian, such as ramping down transverse field terms one-by-one [64], although this method is not expected to generalize other realistic disordered problems [56, 65–69].

For a given class of problems, if one can show the gap at the paramagnet-to-spin-glass transition decays only as a polynomial with the size of the problem, as recent work on the SK model [70, 71] and Max-Cut [72, 73] problem have shown, unfortunately this friendlier decay still does not guarantee the algorithm will find the ground state in polynomial time. Here lies a second mechanism of hardness for quantum algorithms, known as *transverse field chaos* (TFC) [74, 75], where energetic corrections from the transverse field push the system into a quantum spin-glass phase, thus creating false local minima below the ground state of the true problem H_P. This effect from TFC consequently steers optimization algorithms toward false optima first, weakening the performance to the true solution [46, 76–81]. Furthermore, the effect of TFC is seen even at weak transverse field strengths, where transitions are generically exponentially slow, which has allowed some to even engineer this effect as way of crafting hard problems for benchmarking quantum optimization algorithms [82]. The combination of exponentially small gaps and present two challenges that make quantum algorithms of this type unlikely to produce the global optima for **NP**-complete problems.

While we have demonstrated that finding exact ground states in challenging for quantum algorithms of the AQC/QAOA type, we argue that finding approximate solutions may be easier. For many **NP**-hard problems, it turns out that getting any approximation ratio better than random guessing is also **NP**-hard [47, 83, 84]. We ask the following: do the mechanisms for hardness in exact optimization explain the hardness for approximate problem solving? While TFC can effect the ability to find exact solutions by creating pseuo-ground states below the true optima of H_P, we don't expect this mechanism to effect approximate solutions as much. To provide an example, if we consider a class of problems where random guessing produces an average energy of zero, and the ground state energy is normalized to be $-N$, an algorithm that returns solutions in polynomial time within an energy $\leq r(-N)$ for some constant $r > 0$ for all instances, this would imply an exponential speed-up over classical algorithms. Thus, such an algorithm that reliably produced solutions with an energy $\leq -N/3$ would still provide an exponential speed-up if TFC reduced the constant to $r = 1/4$, for example. We conclude that TFC cannot be a mechanism for approximate hardness when the problem exhibits a clustering phase of exponentially many states close to the ground state $|G\rangle$, though well separated in Hamming distance D_H.

Given the severe overhead of fault tolerance [61], quantum hardware is expected to exhibit enormous prefactor disadvantages as compared to parallel silicon, particularly when the comparison is made to hardware with equivalent financial value (e.g. millions of USD). Quantum algorithms thus have the most promise when the problem is hard or outright impossible for classical machines. NP-hard constraint satisfaction problems are no exception, so when benchmarking a proposed quantum algorithm it is important to ensure that the problem classes we consider are sufficiently hard for classical machines, and in the present NISQ era, exhibit their exponential difficulty at small enough N that numerical simulations of quantum algorithms can demonstrate meaningful improvements. MAX-3-XORSAT problems are ideal for benchmarking quantum algorithms because their exponential difficulty scaling is obvious at small N for both classical and prior quantum approaches, in contrast to other problems where the asymptotic exponential scaling often does not set in until system sizes that are prohibitively large for simulation.

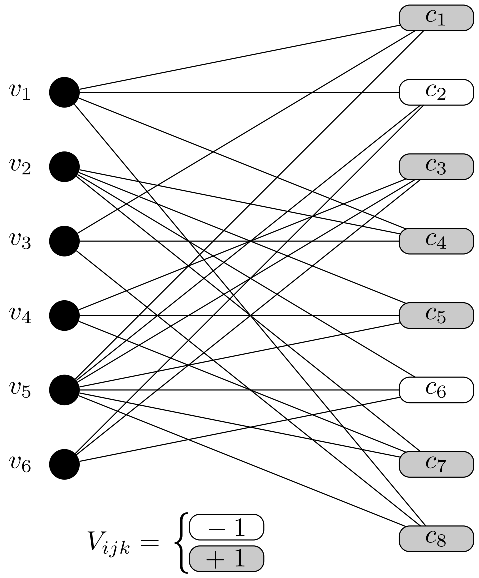


Figure 3.1 Graphical representation of a random 3-uniform hypergraph $G_{\mathcal{H}} = (\mathcal{V}, \mathcal{E})$ used in the MAX-3-XORSAT problem. The set of vertices $\mathcal{V}$ labelled $\{v_1, \ldots, v_6\}$ are connected to a hyperedge (i.e., constraint) from the set $\mathcal{E}$ on the right labeled $\{c_1, \ldots, c_8\}$ if the vertex is in constraint. Each constraint necessarily contains a random set of three unique vertices. The legend specifies the value of the variable $V_{ijk} = \pm 1$ where the grey and white filled c_j boxes denote a $(+1)$, and (-1) valued constraint respectively. This example has a constraint density of $N_C = \frac{4}{3}N$ for 6 vertices and 8 constraints, although we use other values of N_C throughout this work.

To prototype our quantum algorithms, we thus consider MAX-3-XORSAT [47], a well-studied problem that consists in the Pauli basis of a hypergraph of N_C three-body constraint terms, as sketched in Figure 3.1:

$$H_P = -\sum_{ijk}^{N_C} V_{ijk} Z_i Z_j Z_k, \;\; V_{ijk} = \pm 1. \tag{3.1}$$

A constraint is said to be satisfied if, for a given bitstring, $\langle V_{ijk} Z_i Z_j Z_k \rangle = 1$, and unsatisfied otherwise. This is called a hypergraph because V_{ijk} has three indices rather than the usual two found in graph theory. Thanks to the linearity of the problem, one can use Gaussian elimination to check if a solution exists that satisfies all the constraints in $O\left(N^3\right)$ time, but if the problem is not fully satisfiable, finding the lowest energy state(s) is NP hard. A random state satisfies half the constraints on average and is thus energy zero. Further, it was shown by Håstad [84] that if the true ground state satisfies a fraction $(1-\epsilon)$ of the constraints, then finding any configuration that satisfies more than $(1/2+\epsilon)$ of the constraints is also NP-hard (see also [85–87]). The hardest instances are thus those with small but finite ϵ, e.g. almost satisfiable problems, as both finding the true ground state, and even finding an approximate solution, is exponentially difficult. Note that if the problem graph is sparse (e.g. N_C/N is on the order of 1) finding approximate solutions can still be easy, since one can randomly select a fraction of the constraints $< cN$ (for some $O(1)$ constant c), and solve that new, much easier problem; solutions to this sub-problem will satisfy half the remaining constraints, on average.

To ensure that we are studying problems that are both hard to solve and hard to approximate for all known methods, inspired by Refs. [85–88] we consider a family of instances we call planted partial solution problems (PPSPs). To construct a PPSP, we choose a small unsatisfied fraction ϵ and pick a random hypergraph of $N_C \gg N$ unique triplets; we use $\epsilon = 0.1$ in all simulations here. We then pick a random bitstring G and randomly select $(1-\epsilon)\, N_C$ of the constraints to be satisfied in G, by picking the sign of V_{ijk} appropriately, with the rest unsatisfied. If ϵ is small and $N_C/N \gg 1$, G will be the problem ground state with very high probability, as the SAT/UNSAT transition for this problem is at $N_C/N \sim 0.92$ [89] and at densities much higher than the SAT/UNSAT threshold ground states for random graphs satisfy $N_C/2 + O\left(\sqrt{N \times N_C}\right)$ constraints. This property also ensures that G is a unique ground state with high probability at large N. When we refer to random hypergraph problems throughout this work, we refer to this construction rule: a random, potentially fairly dense, hypergraph where one can optionally randomly chose an anomalously large fraction of constraints to be satisfied by matching the signs to a randomly chosen ground state bitstring. We note that [87] recently discovered a hardness threshold at $N_C \propto N^{3/2}$, where at higher densities the problem again becomes amenable to classical optimization.

Our PPSP construction is necessary because truly approximation hard problems–where the practical polynomial time approximation difficulty approaches the random guessing limit of the complexity class separation–are rare in the space of all possible instances. Sufficiently sparse problems are approximation-easy, and for denser random problems one can always find strings that satisfy $N_C/2 + O\left(\sqrt{N \times N_C}\right)$ constraints in polynomial time [88, 90], with a smaller prefactor in front of the $\sqrt{N \times N_C}$ than the prefactor in the average satisfied in the ground state. We formulated our PPSP construction to ensure our algorithm was being benchmarked on instances with a plausible claim to true classical approximation hardness. We note that commonly studied 3-regular problems [45, 91] do not display strong approximation hardness, as they are sparse, and can be solved efficiently if satisfiable. And intriguingly, we show that, by some metrics, the performance of our novel algorithm progressively improves with increasing N_C/N in this regime, at constant ϵ.

3.4 Spectrally folded quantum optimization

The core idea of spectral folding (and related, more general spectral deformations) is to modify how the Hamiltonian is applied to the quantum state through the introduction of a filter function, somewhat analogous to the methods used in [92, 93], though our goals and filtering choices are very different. Specifically, the algorithms we consider solve problems through simulating the time evolution of a quantum state, as $|\psi\rangle \to e^{-iH(t)dt}|\psi\rangle$, with the exponentiated Hamiltonian discretized as a series of layers e.g. $e^{iaH_D}e^{ibH_P}$. The driver Hamiltonian, and any other additional Hamiltonian terms, are not changed by spectral folding, so we will ignore them for now and focus on the problem Hamiltonian itself. Specifically, we write $|\psi\rangle$ in the computational basis as $|\psi\rangle = \sum_{m=0}^{2^N-1} c_m |m\rangle$, where m is the decimal integer representation of a given bitstring. Then, for (arbitrary) control angle γ:

$$e^{i\gamma H_P}|\psi\rangle = \sum_{m=0}^{2^N-1} e^{i\gamma E(m)} c_m |m\rangle\,, \tag{3.2}$$

$$E(m) = \langle m| H_P |m\rangle = -\sum_{ijk}^{N_C} \langle m| V_{ijk} Z_i Z_j Z_k |m\rangle\,. \tag{3.3}$$

In other words, the phase of each component state advances proportionally to its energy under the problem Hamiltonian, and that energy is computed at each step by applying a sequence of gates to implement each constraint.

In spectral folding, the phase of each component instead advances proportional to an arbitrary function f of the diagonal H_P,

$$|\psi\rangle \to e^{i\gamma f(H_P)}|\psi\rangle = \sum_{m=0}^{2^N-1} e^{i\gamma f(E(m))} c_m |m\rangle\,. \tag{3.4}$$

This can be accomplished by introducing a register of auxiliary qubits, applying a gate sequence that maps the sum of the constraint terms to a fraction of that register to store E, using a second fraction of that register to compute $f(E)$, applying a sequence of controlled-phase gates to advance the phase by $f(E)$, and then uncomputing the previous steps to return the register to its initial state. The entire process is sketched in Figure 3.2. Provided f is a relatively simple function, this adds a multiplicative overhead which is polylogarithmic in N, since $E(m)$ is bounded by a polynomial in N and each arithmetic operation takes $O(\log N)$ steps.[5]

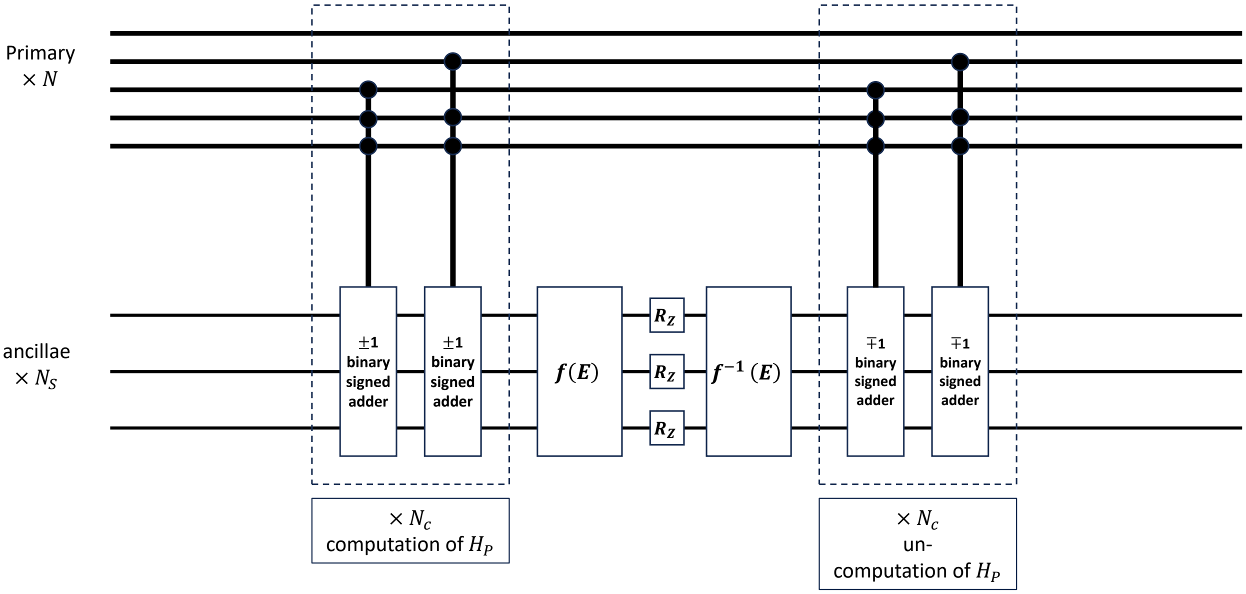


Figure 3.2 A schematic for implementing spectrally deformed time evolution $U = \exp(if(H_P)dt)$. A set of gates maps the value of each of the N_C constraint to adding or subtracting 1 to a register of ancilla qubits, using binary signed adder circuits controlled by the value of $V_{ijk}Z_iZ_jZ_k$. A sequence of additional gates computes $f(E)$ from E (likely using more ancilla qubits), and then a set of local Z rotations is applied to the register storing $f(E)$ to advance the phase of each component of $|\psi\rangle$ proportionally. The computation of $f(E)$ and entangling with "constraint-controlled" gates are then uncomputed, returning the ancillas to their initial state and disentangling them from the N primary qubits over which the problem is defined. The net result of this entire process is to enact the operation in Eq. (), evolving time under an arbitrary function of the diagonal Hamiltonian (transverse field layers and other operations on the primary qubits are not shown). For relatively simple functions, the net overhead of this entire process (compared to enacting $\exp(iH_Pdt)$ directly) is polylogarithmic in N.

We define *spectral warping* as any f which applies a nonlinear rescaling of E, such as $f(E) = cE^2$, and *spectral folding* as a choice of f that mirrors E about a specific value, e.g. $f(E) = |E - E_t|$. The core idea is sketched in Figure 3.3, and these two methods can of course be composed. Incorporating this operation enormously expands the space of quantum optimization algorithms we can define; in this work we focus on two choices, *linear* and *quadratic* spectral folding. Specifically, if we choose our problem normalization via including a multiplicative constant[6] so that the ground state energy is $E_{\text{GS}} = -N$, and let $E_t = AN$ for a constant A, then we define linear and quadratic spectral folding as

[5] We note that for any choice of f more complex than multiplying E by a constant (something that does not require auxiliary qubits to begin with), any spatial locality the graph might have is lost in this step, since $E(m)$ is a global quantity which we are deforming with f.

[6] Formally, this choice assumes that we know the fraction of the N_C constraints which are satisfied in the ground state, something that we cannot know in advance of running our quantum optimization algorithm! However, we can simply repeatedly run the algorithm with different normalization choices to guess its value, a prefactor overhead of at most $O(N_C)$.

$$f_{\text{lin}}\left(E\right)=\frac{\left|E+AN\right|}{A},\quad f_{\text{quad}}\left(E\right)=\frac{\left(E+AN\right)^{2}}{A^{2}N} \tag{3.5}$$

Here, $A<1$ defines the approximation target. And critically, it is defined using the conventions that random states have energy zero, so returning a state with energy AE_{GS} approximates, by a factor of A, the degree to which the true ground state itself improves on random states. These normalization choices ensure that the energy difference between the new ground states of the folded problem, and random states, is N as in the original re-normalized problem. Making this choice simplifies the analysis significantly.

A good choice of A is important for spectrally folded quantum optimization to succeed; if A is chosen to be too close to 1, then we risk failing to well-approximate H_P due to the interference effects. A choice of A which is too small will return a sub-optimal approximation ratio, and if A is too close to zero, cause instabilities from having a poorly defined problem to solve. Fortunately, for random hypergraph MAX-3-XORSAT instances–and here random refers to the graph itself and not, critically, on how many constraints are satisfied in the ground state–we can predict the threshold A for which we expect a polynomial depth circuit to return states with $E\simeq AE_{\text{GS}}$ from first principles. The ideal value of A depends both on the problem class and on the variation of spectral folding employed; for MAX-3-XORSAT, $A\geq 0.6$ is achievable as derived below in section B.4. This is a significant leap over the best known classical approximation algorithm for this problem [88], which offers a weaker guarantee with much more restricted viability, in the worst case, equivalent to $A\to 0$ in our notation. It is also a significant leap over recent quantum approaches to this problem [94, 95]. We note that minimizing $\left(H-E\right)^{2}$ is not itself a novel idea and has been used in classical and quantum algorithms for finding states close to specific energies in chemical and many-body systems [96–102]. To our knowledge, however, the use of spectral folding for approximate optimization of CSPs is novel, both in concept and in the analysis we present below to choose A and understand at a deeper level why it presents significant advantages over optimizing the problem H_P directly.

From hereon, we let H_{fold} be the spectrally folded problem Hamiltonian. Having defined it, there are a number of ways we can attempt to find its ground states. The simplest choice, and one that performs well, is AQC/QAOA-inspired state preparation, where we interpolate in Trotterized evolution between the transverse field driver and the folded problem over a time t_f:

$$H\left(t\right)=f\left(t\right)H_D+g\left(t\right)H_{\text{fold}},H_D=-\sum_j X_j, \tag{3.6}$$

$$f\left(0\right)=g\left(t_f\right)=1,\quad f\left(t_f\right)=g\left(0\right)=0. \tag{3.7}$$

This prescription, with the quadratic folding choice in Eq. (3.5), is the most straightforward to benchmark using standard quantum simulation packages as each call requires $O\left(N_C^2\right)$ multiqubit $Z_i Z_j Z_k Z_l$... rotations. Classically simulating the linear folding prescription requires auxiliary qubits to implement the absolute value operation or, much more practically, saving the phase oracle as a pre-computed diagonal operator.

We can also consider trial minimum annealing (TMA), originally proposed in [103]. We explore the TMA formulation in depth because we can predict its scaling analytically. In this scheme, a simple classical algorithm is used to find an initial local minimum of H_P; the quality of the minimum does not particularly matter and for approximation-hard instances we assume it is far above the true ground state energy, in the worst case asymptotically approaching random guessing. Let this classical minimum state be $|L\rangle$. We will use the linear folding prescription in Eq. 3.5 for H_P itself, and add to it a new, diagonal *lowering Hamiltonian* H_L which has $|L\rangle$ as its ground state, and assign to it a time-dependent coefficient $C\left(t\right)$. Our total cost function Hamiltonian is

$$H_{\text{cost}}\left(t\right) = \frac{\left|H_P + AN\right|}{A} + C\left(t\right) H_L. \tag{3.8}$$

Recall that H_P is normalized so that its ground state energy is $-N$. To go further, we need to specify a form for H_L. For this analysis will choose a new random hypergraph of N_C triples which, critically, has no correlation to the hypergraph of H_P; we choose the same N_C as the problem for convenience here but any $O\left(N\right)$ quantity should be fine. We choose the signs of all constraints so that $|L\rangle$ satisfies all of them. H_L is not included in the folding procedure so applied separately in time evolution. We then choose $C\left(t=0\right)$ such that the initial energy of $|L\rangle$ (defined by $H_{\text{fold}} + C\left(t\right) H_L$) is well below $-N$ but remains $O\left(N\right)$. Our algorithm simulates appropriately discretized time evolution in the following sequence:

- Initialize $|L\rangle$, with H_{cost} always on, and evolve time smoothly ramping up the transverse field from 0 to κ in time t_r. We assume t_r increases linearly with N and choose $\kappa \leq \kappa_c$; $\kappa_c \simeq 1.3$ for MAX-3-XORSAT but can vary for other problems, and we expect weak variation from one instance to the next. We want to choose κ at or just below this value, so we remain in the dressed problem phase (DPP, defined in Appendix B) at all times. Leaving and then re-entering the DPP does not mean the algorithm will fail but makes predictions harder. This smoothly evolves the state to $|L_D\rangle$, the dressed version of $|L\rangle$.

- Evolve time for a total time T, likely also $O\left(N\right)$, where $C\left(t\right)$ is smoothly ramped down to zero, ensuring that $|L\rangle$ crosses the hyperspherical shell of ground states of $H_{\text{fold}} = \left|H_P + AN\right|/A$. Note that this crossing occurs when the ground state energy of $C\left(t\right) H_L$ is $O\left(-N\right)$, and if we assume the initial minimum was uncorrelated with the true ground state $|G\rangle$, the mean Hamming distance between $|L\rangle$ and any of the ground states of the folded Hamiltonian is $N/2$ flips. Consequently, H_L adds an $O\left(\sqrt{N}\right)$ energy uncertainty to these states that has no meaningful impact on the approximation ratio.

- Finally, ramp the transverse down to zero smoothly over t_r and measure the system in the z basis. For an appropriate $O(1)$ choice of A and a random problem hypergraph, this algorithm will return states with energies close to AE_{GS} with constant probability. We can optionally repeat the algorithm many times, starting from different choices of $|L\rangle$, to ensure a fairer sampling of states in that energy range.

The total gate count of this algorithm is as follows. We have a factor of $O\left(N + N_C \text{ polylog}\left(N\right)\right)$ per time-step for the layers of transverse field, H_{fold} and H_L terms, which we simplify to N_C polylog (N). We obtain, in the worst case, a factor of $O\left(N_C\right)$ for the number of guesses one needs to make to correctly set the normalization for a chosen A. We assume, on empirical grounds, that the total quantum evolution time is $O\left(N\right)$.This choice works well in practice in our simulations, and more intuitively, the very simplest classical optimization routine, steepest descent, requires $O\left(N\right)$ Hamiltonian calls to halt. We do not think it reasonable, ultimately, that a quantum algorithm should perform well with fewer steps per shot. Finally, in the worst case we expect a time-step $dt \sim 1/N$, for graphs where a small number of variables connect to significant fractions of the N_C constraints, but dt constant or increasing logarithmically is empirically and intuitively fine in the typical case. Taken together, and we emphasize assuming that the algorithm is capable of returning states with $E < AE_{\text{GS}}$ in constant probability, we estimate a total runtime between $O\left(N_C^2 N^2 \text{ polylog}\left(N\right)\right)$ in the worst case and $O\left(N_C N \text{ polylog}\left(N\right)\right)$ in more typical cases.

All that said, justifying the assumption of constant success probability is the central task of the paper. We now provide a theoretical analysis of the performance of this formulation of spectral folding on random hypergraph PPSPs, and a more qualitative analysis of the expected performance of other variations.

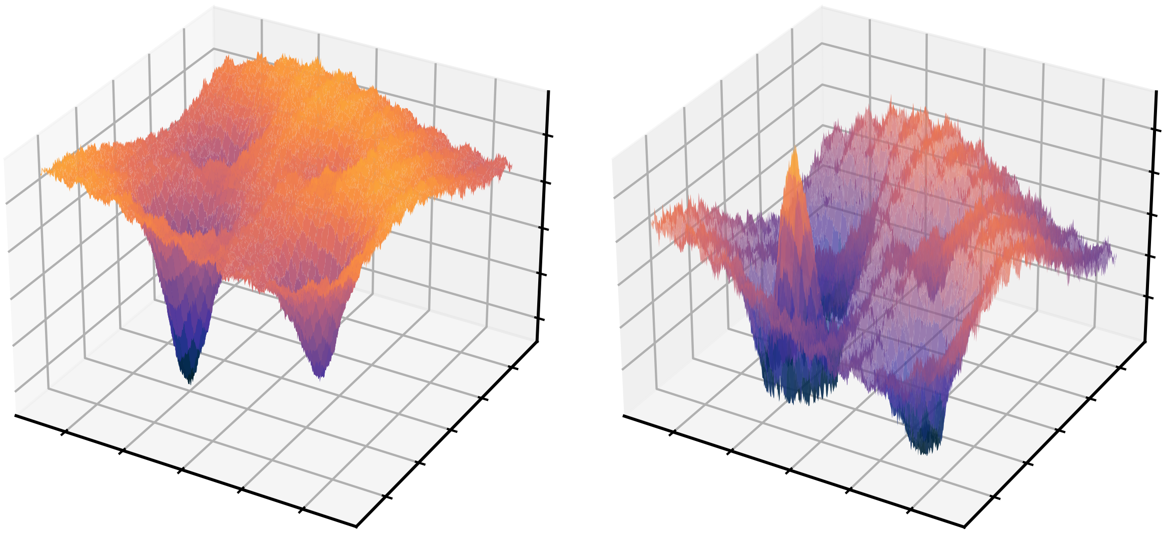


Figure 3.3 Illustration of the spectral folding procedure. (Left) Sketch of the rough energy landscape of an approximation-hard CSP, with a single deep minimum whose basin of attraction is an exponentially small fraction of the configuration space. Directly optimizing this cost function through quantum approaches often misses the deep minimum entirely, for reasons explained in the text. (Right) Spectrally folded energy landscape, where the problem Hamiltonian energies are mirrored around an approximation target $E = AE_{\mathrm{GS}}$. This can be implemented in a gate model algorithm with modest overhead, as shown in the text. Doing so promotes the states near the fold to an exponentially large ground state band while eliminating an interference effect that reduces tunneling into them from trivial initial states; for a wide range of problem instances (and likely, low-order problem classes), this works out to an approximation guarantee. Detailed performance predictions, and numerical benchmarking, are shown in the text.

3.5 Simulation results

To confirm our predictions–or at least, verify that any serious issues with our calculations and interpretation of the problem are subtle and not apparent at system sizes within reach of present or near-future classical simulations–we performed a series of numerical simulations of various classical and quantum algorithms applied to our PPSPs. For all quantum simulation tasks we used the Qulacs package [33]. For smaller systems and algorithm prototyping we ran our simulations on local workstations; this includes all the spectral folding TMA simulations. For all QAOA and spectral folding AQC simulations presented, we used the Fujitsu Quantum Simulator, a classical HPC system. This allowed us to probe larger system sizes while still averaging over enough instances to have reliable statistics for the problem class. Unless otherwise stated, each data-point represents the average over 960 or 1000 (for the QAOA and spectral folding AQC results) random problem instances constructed with the prescriptions outlined in Section 3.3. In all cases in this work we used an unsatisfied fraction $\epsilon = 0.1$ for our partial planted solutions; we expect similar phenomenological behavior for other constant ϵ values, though we expect the precise thresholds we measure will vary with ϵ for classical optimization algorithms and QAOA (but not folded optimization, below its predicted performance floor).

The results of our simulations are summarized in Table 3.1, which lists the best per-shot polynomial time approximation ratio achievable through each algorithm studied, both classical and quantum. To estimate these values, we measured the average per-shot probability $P_q(N) = P(E \leq qE_{\mathrm{GS}})$ of returning states with energies at or below qE_{GS}, where $q \leq 1$ is the approximation ratio and E_{GS} is negative in our conventions. These probabilities were computed by choosing bins of size 0.05 E_{GS}.

To determine the polynomial time hardness threshold, we applied a very simple rule where we fitted $P_q(N)$ to a simple exponential function, assuming any observed decay faster than $2^{-0.005N}$ corresponded asymptotically to exponential decay, and any positive exponents, e.g. exponential growth, represent small-N growth toward some constant saturation value. This threshold of $2^{-0.005N}$ was chosen to reduce the influence of uncertainty from fitting in a small handful of cases. For the greedy algorithm with $N_C/N = 1.5\sqrt{N}$ we fitted decay to an exponential in $\sqrt{N}$ as discussed below. The polynomial time approximation hardness threshold for a given algorithm, unsatisfied fraction ϵ and N_C/N scaling choice is defined to be q_a, the largest values of q for which we do not observe exponential decay. For $N_C/N = 3\sqrt{N}/2$ this threshold is decaying with system size, and the asterisk next to the result for QAOA is to highlight the fact that we only ran these simulations out to $N = 28$ so are likely not capturing the asymptotic threshold. As discussed elsewhere, the number of cost function calls is $O(N)$ for all approaches studied. Cases for which a range is quoted are where we felt there was some ambiguity to the fitting, and all values are the result of extrapolating fits to numerical simulations and are naturally somewhat approximate.

Table 3.1 Approximation hardness thresholds for the classical greedy search, high-depth QAOA and spectral folding variations. This table lists q_a, the largest value of the approximation ratio q before exponential decay is reported, drawn from the numerical experiments. Spectral folding results are labeled as protocol/A (where A is the approximation target); AQC is quadratic spectral folding in the AQC formulation, TMA-3 is trial minimum annealing with a linear folded Hamiltonian and 3-XORSAT lowering Hamiltonian, and TMA-L is the same with local Z biases for the lowering Hamiltonian. These results–where the threshold decays for traditional methods but not folded optimization–support the predictions that random hypergraph problems are efficiently approximable through spectrally folded quantum optimization.

N_C/N	Classical	QAOA	AQC/0.75	AQC/0.85	TMA-3/0.75	TMA-3/0.85	TMA-L/0.75
2	0.75-0.8	0.75	0.75	0.75	0.75	0.75	0.7
4	0.55	0.55	0.7	0.7	0.75	0.8	0.75
6	0.45	0.45	0.75	0.7	0.8	0.8	0.75
$3\sqrt{N}/2$	decay	0.25/decay*	0.75	0.75	0.8	0.8	0.75-0.8

3.5.1 Performance of quasi-greedy classical algorithms

To explore the classical difficulty of our PPSPs, we applied a greedy local search algorithm adapted from [45]. This algorithm is straightforward; we start with a random bitstring. Then, beginning at each step, we calculate $k =$ "number of unsatisfied constraints minus the number of satisfied constraints" associated with each bit. We then calculate the fraction of bits, f_k, belonging to each k value. Note that we only care when $k > 0$ because these are the cases where flipping a particular bit will lower the energy. Using some weight function, $w(k)$, we select a k value with normalized probabilities $\propto w(k) * f_k$ and flip a bit with that k value. If there are multiple bits belonging to the k value chosen, we choose a bit in this set with uniform probability to flip. This is repeated until the configuration finds itself in a minimum, and the algorithm halts.

We found the algorithm performed best with a weight profile quadratic with k, however this can be experimented with for different results. Notably, [45] found that when applying a highly optimized version of this search to 3-regular 3-XORSAT problems it performed well even when compared to more sophisticated algorithms such as simulated annealing and parallel tempering. The intuitive reason for this can be inferred from the typical energy landscapes of these problems, which are rough and contain exponentially many high energy local minima. Once one is found, it is more efficient to simply restart the algorithm from a new random configuration instead of attempting to “climb out” using penalized operations in simulated annealing or parallel tempering. As the locations of these minima are uncorrelated with the true ground state, finding one provides no useful information in a ground state search.

This expected inefficiency of simulated annealing/parallel tempering for this problem can easily be inferred from the results plotted in Figure 3.4 and Figure 3.5. Namely, in all cases the algorithm will find a single relatively deep minimum with high probability at each shot, leading to a super-polynomial cost to escape from it in algorithms simulating a thermal bath. Interestingly, as N_C/N increases for fixed unsatisfied ground state fraction ϵ, we find that the decay exponent of the per-shot probability of finding the ground state, $P_{\text{GS}}(N)$, monotonically decreases, suggesting that the basin of attraction of the true ground state is widening as the problem becomes more extremal. In fact, for $N_C \geq 2N$, we empirically observe that the per-shot probability of finding the planted ground state has the approximate scaling

$$\log\left(P_{\text{GS}}(N)\right) \simeq -c_g \frac{N^2}{N_C} . \quad \text{(PPSPs, } N_C \geq 2N\text{)} \tag{3.9}$$

However, the approximation ratio q_a–defined as the minimum energy for which the probability of finding states at or below it stays constant as N increase–steadily worsens. We attribute this to there being a high density of local minimia with energies $\geq q_a E_{\text{GS}}$ (recall E_{GS} is negative in our conventions), but below that

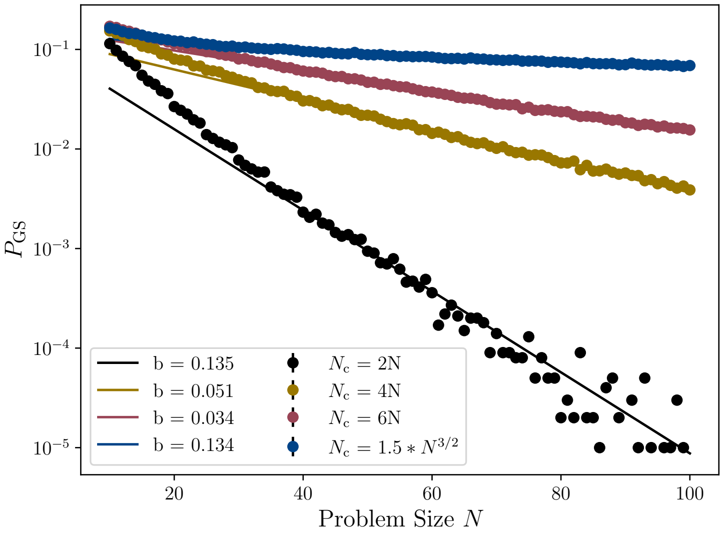


Figure 3.4 Per-shot probability of finding the true ground state with the quasi-greedy classical algorithm as adapted from [45], for the four constraint densities studied in this work and planted solution unsatisfied fraction $\epsilon = 0.1$. The probability decays superpolynomially with N; fits are to $a2^{-bN}$ for $N_C = \{2, 4, 6\} N$ and $a2^{-b\sqrt{N}}$ for $N_C = 1.5N^{3/2}$. Each data-point is the average of 10^5 shots. As the constraint density increases, the basin of attraction widens, and the problem becomes easier–though still exponentially scaling–for local update classical routines. Each shot consists of $O(N)$ local updates so the time to solution scales essentially as the inverse of this probability.

threshold the number of minima quickly decreases and the probability of finding one decreases exponentially. This results in the scaling collapse seen in Figure 3.5–for sufficiently low energy there are no minima aside from the ground state, so the approximation probability scales nearly identically to $P_{\mathrm{GS}}(N)$. This *high energy clustering phase* is a feature of our PPSP construction, and is responsible for its classical approximation hardness. We again contrast this to problems near the statistical SAT/UNSAT threshold such as three-regular instances, where the *clustering energy*, the lowest energy where there are still exponentially many local minima and they are thus easy to find, is close to E_{GS} and they are not approximation-hard in practice as a result. We conjecture that high energy clustering behavior is a generic feature of low-degree constraint problem classes that are approximation-hard for local update algorithms.

As shown in Figure 3.5, these construction rules yield in a set of instances which are hard to approximate in practice. If we let N_C/N grow slowly with N, e.g. as $\ln(N)$ or $\sqrt{N}$, then as $N \to \infty$, the probability of finding any states with energies any $O(1)$ fraction better than random guessing, decays superpolynomially[7] in N. And since the unsatisfied fraction ϵ in the ground state is small but nonzero, Gaussian elimination cannot be used to efficiently find the solution, forcing classical computers to rely on

[7] We note again the recent result of [87], who showed that for $N_C/N > O\left(N^{1/2}\right)$ the problem can be efficiently approximated classically; our PPSPs with $N_C/N = 1.5N^{1/2}$ are close to this threshold but do not cross it.

local update algorithms stymied by entropic barriers. It is of course possible that some clever algorithm could be written to exploit our PPSP structure to efficiently solve or approximate these instances classically; we merely claim hardness for generic methods based on local updates. Our PPSP construction rules can easily be generalized to other CSPs, and we suggest that they could prove to be a useful tool for exploring practical approximation hardness in other contexts.

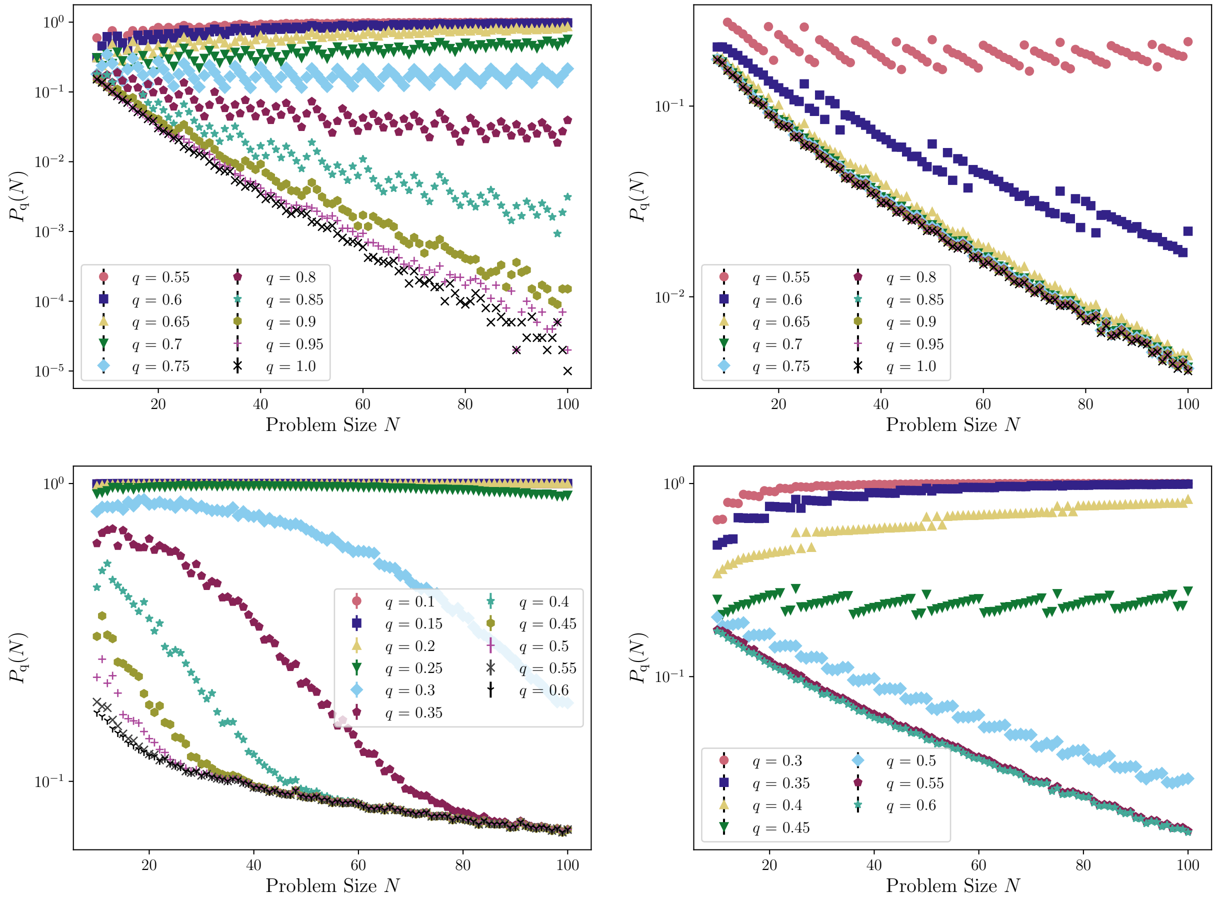


Figure 3.5 Classical approximation hardness of PPSPs using the quasi-greedy classical algorithm, for constraint densities (clockwise from top left) $N_C = \{2N, 4N, 6N, 1.5N^{3/2}\}$. In each figure we plot the probability that a given shot returns an energy below qE_{GS} for various choices of q. For all the fixed N_C/N problem classes we find an empirical approximation threshold $q = A_g$ below which finding states becomes superpolynomially hard, and that this value decreases as N_C/N increases. For the case $N_C = 1.5N^{3/2}$ (bottom left), this value steadily drops, as discussed in the text.

3.5.2 Performance of high depth TAQC

To make firm points of comparison, alongside the simulations of spectrally folded optimization itself we extensively benchmarked high-depth QAOA and quasi-greedy [45, 91] classical algorithms on these instances. For our high-depth QAOA simulations, we formulated our algorithm to mimic trotterized time evolution over a total time t_f, with:

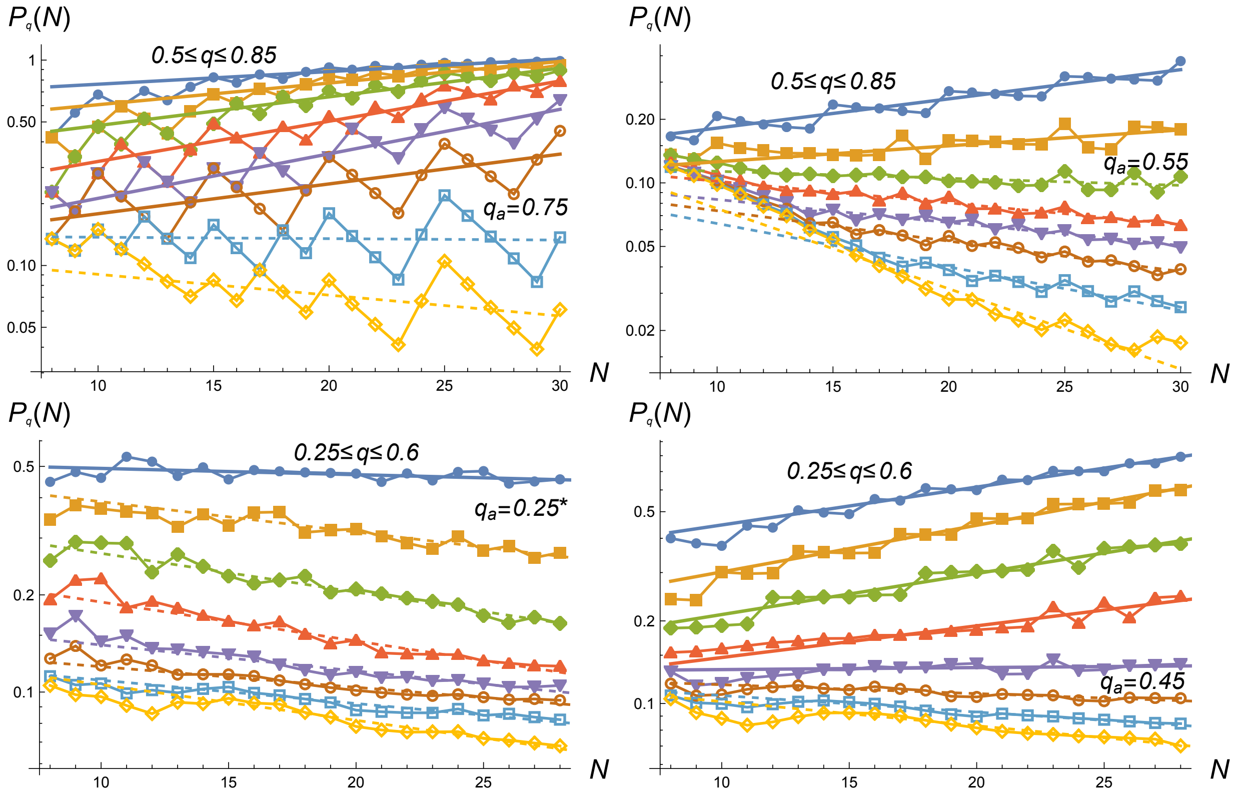


Figure 3.6 Performance of QAOA as an approximator for $N_C = \left\{2, 4, 6, 3\sqrt{N}/2\right\} N$ (clockwise from top left), with problem and algorithm parameters described in-text. Plotted are the probabilities $P_q\left(N\right) = P\left(E \leq qE_{\mathrm{GS}}\right)$ for q running from 0.5 to 0.85 in steps of 0.05 (top panels) and 0.25 to 0.6 (bottom panels). The results for the larger constraint densities thus plot a weaker approximation range. Thick straight lines correspond to simple exponential fits where $P_q\left(N\right)$ is not decaying, dashed straight lines correspond to exponential decay, and thin lines between points are included for visual clarity. As summarized in Table 3.1, these results do not represent a meaningful improvement over the classical greedy result (Figure 3.5), though in some cases where $P_q\left(N\right)$ decays exponentially for both approaches, the exponent for QAOA may be better. These results were obtained using the Fujitsu Quantum Simulator, a classical HPC system.

$$\begin{aligned} &|\psi(t+dt)\rangle = e^{-2\pi i f(t) dt H_D} e^{-2\pi i g(t) dt H_P} |\psi(t)\rangle, \\ &f(t) = \sqrt{1 - t/t_f},\ g(t) = \sqrt{t/t_f}. \end{aligned} \tag{3.10}$$

In all the presented data we used $t_f = N/32$ and $dt \simeq 0.05$. Individual shots use a random evolution time between $2t_f/3$ and $4t_f/3$; this runtime averaging produces substantially more reliable scaling, particularly when probabilities are small. These parameters were chosen by trial and error for smaller systems; we observed that the probabilities of finding the ground state and other low-energy states increased sublinearly with t_f beyond this point. The relative improvements of the probabilities of returning low-lying states were similarly sublinear. No sophisticated numerical or iterative optimization methods were used throughout, as we emphasize no fine tuning is necessary.

The results of our QAOA simulations, of 1000 random PPSPs for each choice of N running from 8 to 30, are shown in Figure 3.6. The probability of finding the ground state, shown in Figure A.1, decays exponentially with an exponent very close to that in Eq. (B.7), which we find remarkable given the simplifying assumptions in that derivation, and that it does not use the more sophisticated techniques used to compute tunneling rates between semi-classical minima. Further, this exponent displays only small variations with constraint density and is nearly identical in all four cases. Smaller system studies for other constraint densities all yielded very similar results for P_{GS}, as predicted by Eq. (B.7).

Turning to approximation hardness, being relatively sparse, the $N_C = 2N$ problems are fairly well-approximated by QAOA, with the algorithm returning strings within $q = 0.75$ with constant or saturating probability; we attribute this to the presence of many competing minima with energies not far from E_{GS}. In contrast, for $N_C = 4N$ the algorithm's performance for approximation degrades, with clear exponential decay for approximation ratios better than $q = 0.55$. For higher constraint densities approximation becomes even more difficult, decaying exponentially below $q = 0.45$ for $N_C = 6N$ and 0.25 for $N_C = 3N^{3/2}/2$. We expect decay at sufficiently large N for any constant fraction in that case, but cannot simulate larger system sizes. Crucially, the thresholds q_a we measure are nearly identical to those found by the classical greedy algorithm Figure 3.5, and no signatures of an exponential quantum advantage in these instances can be seen.

Interestingly, as N_C/N increases, the probabilities of finding states comparatively close to $|G\rangle$ in Hamming distance improve (see Figure A.1), but the probabilities of finding states close in energy worsen. We attribute this behavior to the high energy clustering phase. Empirically for our PPSPs there is a high density of local minima with energies $E \geq q_a E_{\text{GS}}$, and if q_a is relatively close to 1 it becomes harder for high-depth QAOA to find local excitations near the planted ground state, as the probability amplitudes will be spread over increasingly many competing minima and their own basins of attraction. Conversely, as q_a decreases with increasing N_C/N, the probability of finding local excitations relatively near $|G\rangle$ increases though still decays exponentially, as the low energy minima far from $|G\rangle$ are at proportionally higher energies and thus do not compete directly with few-flip states. That the thresholds q_a for QAOA match those of the greedy classical algorithm further supports the interpretation of an energy threshold above which local minima become common.

Comparing the classical and established quantum methods, we find that, for these approximation hard instances, QAOA performs very poorly for finding the ground state but less poorly for approximation below the classical hardness threshold q_a (see Table 3.1), with decay exponents that are much closer to the classical result. In some higher constraint density cases our fits produced favorable exponents for approximation with QAOA but our range of N here is smaller than we would prefer to claim any relative quantum advantage absent theoretical justification. Nonetheless, both methods show clear super-polynomial decay per-shot for approximate optimization below the energy range where local minima are dense. With these results in hand, we now turn to folded quantum optimization, which maintains an approximation guarantee regardless of the problem's constraint density.

3.5.3 Performance of SFQO and its variants

Having numerically confirmed the expectation that our PPSPs are superpolynomially hard for classical and prior quantum approaches, for both exact and approximate optimization, we now present the results of our folded quantum optimization simulations. We simulated both the trial minimum annealing and interpolation (e.g. AQC) variations. In all cases we chose run-times increased linearly in N, albeit with larger prefactors than in the QAOA simulations (which used $t_f = N/32$). A longer runtime further helps reduce the potential influence of diabatic local heating as the Hamiltonian parameters are varied. Run and ramp times that are too short can lead to artificially poor scaling, arising from the formation of local excitations as the transverse field is turned on or off too quickly. This is fundamentally a different, and much more prosaic, issue, than decaying collective tunneling rates, but can be difficult to distinguish when our only measures are energy and Hamming distance from $|G\rangle$.

We first present simulations of the AQC variation of spectral folding in Figure 3.7. For the AQC variation, we followed the procedure in Eq. (3.10), with a quadratic H_{fold} (Eq. 3.5) in place of H_P and $f(t) = (1 - t/t_f)^{1/4}$ instead of $\sqrt{1 - t/t_f}$, with an average runtime $t_f = N/24$. This schedule modification was found to improve scaling at higher approximation ratios. In all cases individual shots are runtime averaged between $2t_f/3$ and $4t_f/3$ as in our QAOA simulations.

All these parameter choices are the result of intuition, trial and error, and a desire for simulations to complete in reasonable amounts of time; none are particularly optimal. Nonetheless, as shown in Figure 3.7, in all but one case we were able to meet the approximation target $A = 0.75$, but not exceed it; for $N_C = 4N$. Our spectral folding methods are also much better at reliably returning states close to G in Hamming distance, consistent with the approximation guarantee. We likewise tested increasing A to 0.85 in simulations up to $N = 24$ with this variation, and found improved prefactors but no improvement in scaling. This suggests that we have found the performance ceiling for this approach. Interestingly, the worst case $q_a = 0.7$ observed for this method is very close to the approximation ratio of ~ 0.68 predicted, using a more simplified analysis than was employed for the TMA variation.

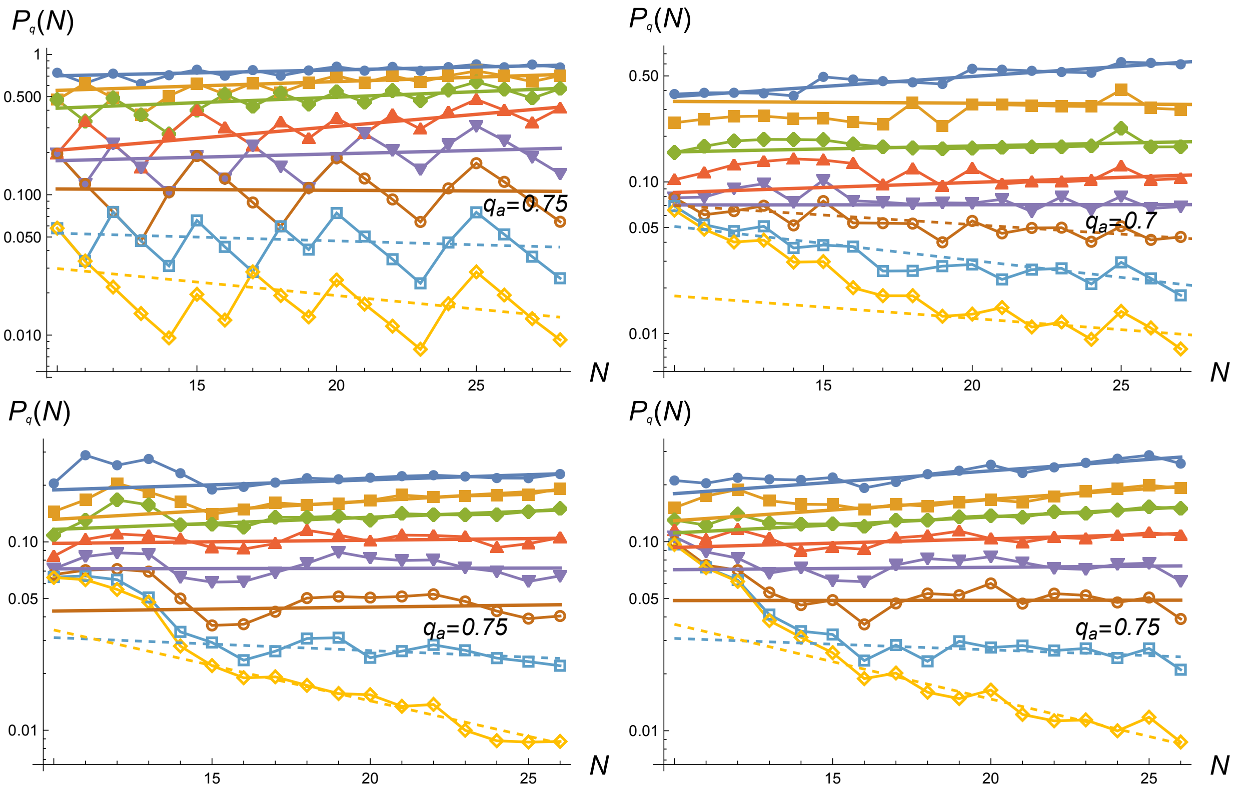


Figure 3.7 Performance of the quadratic AQC formulation of spectral folding, for $N_C = \left\{2, 4, 6, 3\sqrt{N}/2\right\} N$ (clockwise from top left), with N running from 10 to 27 (top row) or 26 (bottom row), $A = 0.75$, $dt = 0.0325$, $t_f = N/24$ and other parameters as stated in text. In each plot the 8 curves plot $P_q\left(N\right) = P\left(E \leq qE_{\mathrm{GS}}\right)$ for q running from 0.5 to 0.85 (top to bottom) in steps of 0.05. Thick straight lines correspond to simple exponential fits where $P_q\left(N\right)$ is not decaying, dashed straight lines correspond to exponential decay, and thin lines between points are included for visual clarity. In all four cases spectrally folded optimization is able to meet its approximation target of $A = 0.75$, returning states at or below this energy with constant probability in a linearly growing number of cost function calls. This is in stark contrast to our classical greedy algorithm (Figure 3.5) and QAOA (Figure 3.6) results, where the achievable polynomial time approximation ratio steadily worsens with increasing N_C/N, and supports the theoretical analysis of Appendix B. These results were obtained using the Fujitsu Quantum Simulator, a classical HPC system.

We also tested the TMA formulation of spectral folding–the formulation for which we can make the most reliable analytical predictions–as plotted in Figure 3.8. For these variations we used run-times $t_f = N/12$ and $dt = 0.025$; note that this choice of t_f is a factor of 8/3 larger than in our QAOA simulations but with the same scaling. In smaller system studies similar qualitative performance was observed for shorter t_f (such as $N/24$ or $N/32$). For this variation we used a 3-XORSAT H_L–the formulation for which we could predict performance in Appendix B–with the minimum energy set to $-2N$ via $C(t)$, which was linearly ramped down to zero by t_f, and simple sinusoidal ramp profiles with $t_r = N/24$; the transverse field strength κ during the main evolution was 1.3. The careful reader may note that choosing $C(t)$ to set the minimum energy to $-2N$ instead of $-N$ is naively sub-optimal, as all other things being equal sweeping over a larger energy range increases W in Eq. (B.21), and should reduce the returned probabilities $P_q(N)$ by an appropriate prefactor. However in our simulations this choice consistently improved both the prefactors and scaling, e.g. the value of q_a, as compared to choosing a minimum energy $-N$ for H_L. We suspect this has to do with the band structure considerations, but due to the complexity of the problem, we are unable to make a quantitative prediction.

The performance of the two approaches is qualitatively similar with subtle differences as we vary the returned approximation ratio q. At lower approximation ratios the AQC formulation returns higher probabilities, at lower total gate count since there are no ramping steps and no additional gates associated with adding H_L. However, at higher approximation ratios the TMA formulation appears to be better able to approach the approximation target $A = 0.85$. As discussed in the algorithm definition, folded optimization will definitionally fail to consistently return energies significantly below AE_{GS}, and we expect it to break down as A gets too close to 1 given that QAOA and similar methods fail to reliably approximate these problems. Choosing the best value for A is thus a subtle issue that depends on the problem class; for extensions of this method to hard CSPs it will necessarily change from one problem class to the next.

Likewise, as mentioned in Appendix B.5, one can replace the random 3-XORSAT lowering Hamiltonian in TMA for a simpler set of linear Z biases, reducing the gate count per time-step and, potentially, increasing the per-state tunneling rate by implementing a shallower cost-per-flip curve. In comparison to the 3-XORSAT variation discussed in the previous paragraph, to achieve good performance we needed to double the ramp time. This protocol seemed to be more sensitive to performance degradation from heating during ramps. The algorithm also benefited from adjusting $C(t)$ so that the minimum energy of H_L was $-3N$, as compared to $-2N$ for the 3-XORSAT H_L. Relative performance for $A = 0.75$ is comparable to the other variations, as illustrated in Figure 3.9, though the individual $P_q(N)$ show more significant non-monotonicity that makes reliable curve fitting challenging. This issue is even more pronounced for $A = 0.85$ (data not shown), to the point that we did not report q_a values for that variation in Table 3.1.

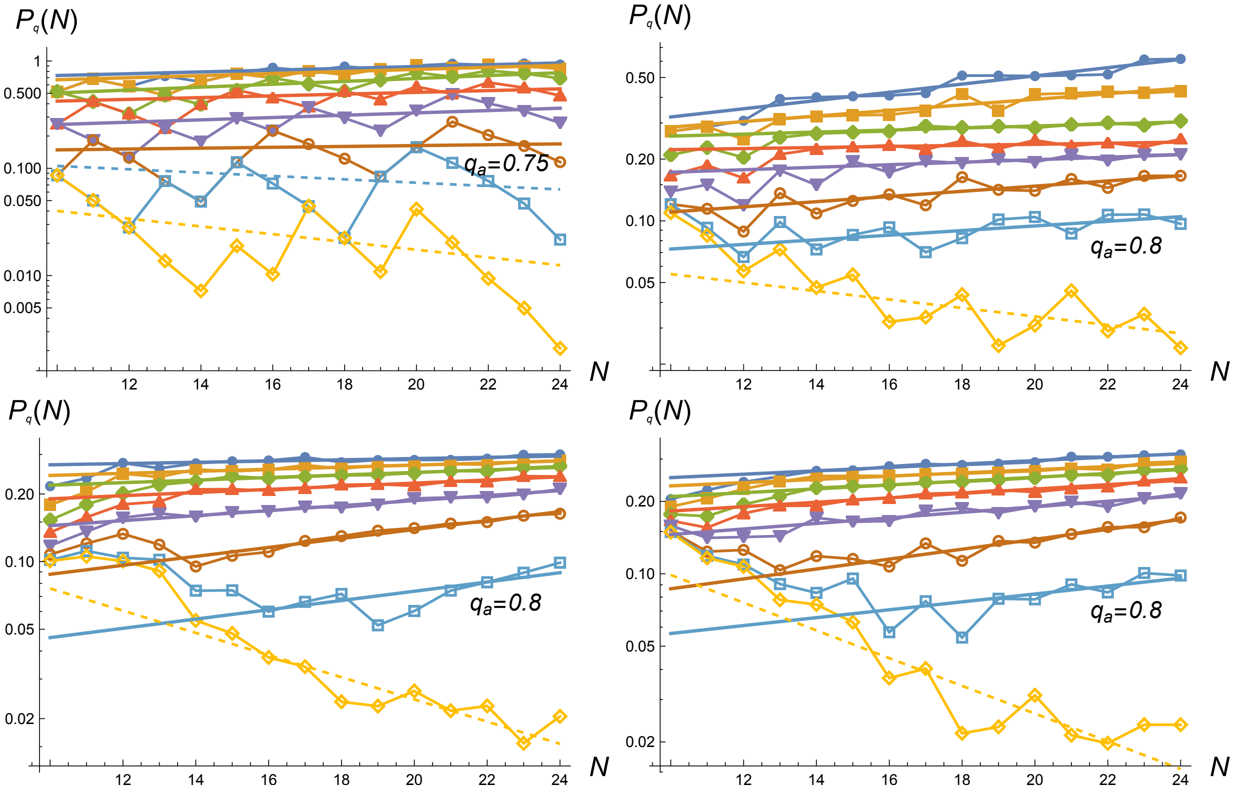


Figure 3.8 Performance of the trial minimum annealing formulation of spectral folding with a 3-XORSAT lowering Hamiltonian H_L, with $A = 0.85$, plotted for constraint densities (clockwise from top left) $N_C = \left\{2N, 4N, 6N, 1.5 \times N^{3/2}\right\}$ and approximation ratios between $q = 0.5$ and 0.85 with N running from 10 to 24, for the parameters detailed in the text. All data is derived from averaging over 960 random instances and choices of t_f. Compared to the AQC formulation shown in Figure 3.7, the achievable approximation ratio q_a is often slightly higher, though the total gate count in this formulation is larger by a constant prefactor. In all cases q_a well exceeds the value of 0.6 conservatively predicted for this formulation.

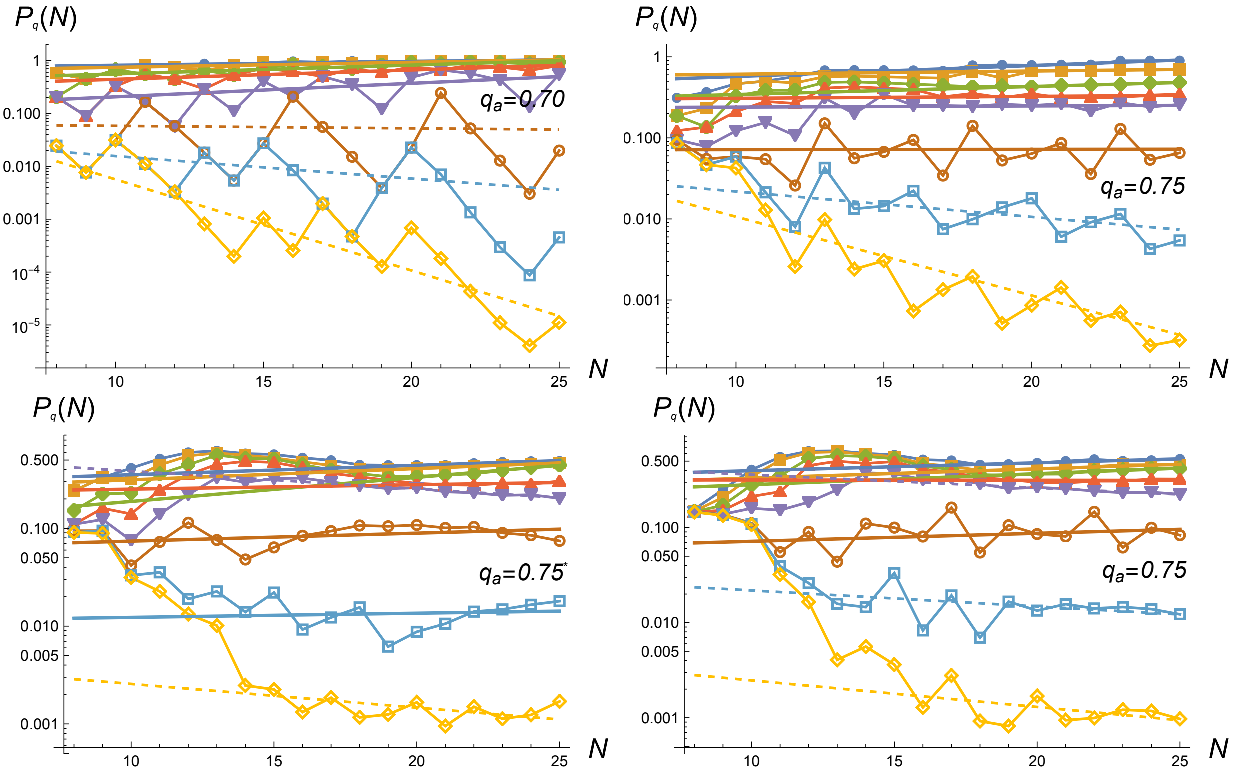


Figure 3.9 Performance of the trial minimum annealing formulation of spectral folding with a local Z bias H_L, with $A = 0.75$, for N running from 8 to 25. q_a in each case is comparable to other variations, though greater non-monotonicity in the individual $P_q\left(N\right)$ curves makes fitting more difficult.

3.6 Discussion

Our simulations of SQFO and its variants demonstrate that the algorithm targets approximate solutions in the range from $A = 0.75$ to $A = 0.85$, which is significantly larger than the theoretical prediction of $A \sim 0.6$. While not presented in this work, we did test $A = 0.65$ for small system size, which did not have any meaningful improvements over larger approximation targets of $A = 0.75$ or higher. While our simulations with $A = 0.75$ demonstrated constant probability of obtaining meeting the target, while increasing $A = 0.75 \rightarrow 0.85$, did not return constant probability. However, in the $A = 0.85$ case, we did observe some performance increases for achievable approximation ratios q_a. These results suggest that there exists an approximation target ceiling A_{ceil}. Below A_{ceil}, the algorithm is expected to reach the target approximation with constant probability, while above A_{ceil}, we expect the algorithm to fail on returning constant probabilities of reaching the target approximation. Our results demonstrate a minimum achievable approximation ration of at least 0.7 for the quadratic AQC formulation, and at least 0.75 for the 3-XORSAT TMA variation of SFQO.

In Table 3.1, we demonstrate that at high approximation ratios, SFQO returns constant probabilities of reaching the target approximation, while both classical and TAQC methods exhibit super-polynomial decay. This results for SFQO are shown in Figure 3.7-Figure 3.9, while the classical and TAQC results are presented in Figure 3.5, and Figure 3.6 respectively. Our results demonstrate no exponential separation between the quasi-greedy classical methods and TAQC algorithms tested in theis work. At low constraint densities, we expect that the clustering of poor local minima is relatively close the ground state. In this regime, SFQO does provides no benefits for approximation, although the convergence in Hamming distance (to the PPS) tends to scale well with increasing system size shown inA. However, at higher constraint densities, our results show that SFQO returns solutions that are close in energy and Hamming distance to the PPS, with constant probability. The variations of SFQO we present in this work perform similarly, with the 3-XORSAT TMA variation returning the highest approximation ratio. While the TMA variation performs best, it also has the highest prefactor cost in gate count.

Given the separation demonstrated by SFQO, how can we understand the mechanism for the observed speed-up? For established methods, polynomial time approximation is governed by the structure of the problem. In particular, we are interested in the relationship between the ground states (including the PPS) and the high energy clustering of local minima. As the PPS become more extremal, the relative energy difference grows, and the target threshold approximation ratio becomes a smaller. Our results illustrate the novelty of collective tunneling effects in spectral folding, which are not exploited in the previous methods we consider in this work. The search space for previous TAQC methods remains negatively affected by interference and TFC that result in a failure to produce the ground state or even approximate solutions, demonstrated by asymptotic exponential decay. However, we expect that SFQO is not limited by these mechanisms for hardness in the exponentially large hyper-spherical shell, provided by the spectral folding and spectral warping procedures. Thus, the search space that our method defines, allows the a more robust effect of collective tunneling to approximate solutions, where the probabilities of reaching the target approximate are constant for the system sizes we can simulate classically.

In the run-time cost analysis, all routines demonstrate an $O(N)$ complexity of Hamiltonian evaluations. In this linear scaling, our results demonstrate that exponential scaling for time to solution is not visible from the figures or numerical fits to our simulation data. Of course, it may be the case that beyond our simulation capabilities at large N, a small decay exponent may set in, so we cannot rule out this possibility from numeric results alone. Rather, these results suggest that the theoretical predictions derived from tunneling rates in Appendix B agree with our numerical results over the wide range of parameters we bechmarked in this work.

Lastly, would we expect classical methods to benefit from the spectral folding procedures in this work? As discussed in Section 3.3, classical methods attempt to cool towards the ground state by local optimization steps from an initial high-energy state. While the folding methods we present in this work are not quantum in nature, we only expect quantum algorithms to benefit from the procedure, due to entropic barriers that make reaching any approximation better than random guessing still NP-hard for the MAX-3-XORSAT problem we tested. Furthermore, we expect classical simulations of the procedure to be inefficient for current methods, due to the non-local interactions in the problem graph and volume-law entanglement we expect in the low-energy state of the quantum spin-glass. One exception that deserves further exploration is whether quantum Monte Carlo (QMC), could efficiently simulation the spectrally folded spin-glass, due to the problem being stoquastic. Although QMC is still based on local update rules, however, we conclude that such a discovery would be very interesting in it's own right. In what follows, we conclude and discuss possible extensions of these results for future work.

3.7 Conclusion and outlook

In this work, we investigated the MAX-3-XORSAT problem class, and explored opportunities for classical and quantum optimization methods to approximate the problem. We concluded that mechanisms of hardness make these problems hard to approximate classically, although these same mechanisms do not translate to quantum methods, due to the way in which quantum algorithms approach the ground state and low-lying approximate solutions to the problem. However, we found that previous quantum algorithms still face other mechanisms for hardness of exact optimization, most notably: exponentially small gaps and transverse field chaos. However, the same mechanisms that make the problems hard to solve exactly, may not contribute to hardness of approximation. Guided by these physical mechanisms, we proposed a class of partial planted solution problems (PPSPs), which are hard to approximate for all previously known classical and quantum methods.

Using these PPSPs, we proposed a novel mechanism called *spectral folded quantum optimization* (SFQO), which does not suffer from the same mechanisms of hardness we identify for previously known methods. The spectral folding procedure applies a classical deformation to the energy landscape of the problem, thus allowing the algorithm to find approximate solutions more readily.

Guided by theoretical predictions of the tunneling rate between the paramagent and spin-glass phases of the problem, derived by the author's colleagues (E.K. and V.O.), we showed that previously known quantum methods based on trotterized adiabatic quantum compution (TAQC) are bottle-necked by transverse field chaos and destructive interference. Furthermore, analytic predictions based on resummed Nth order perturbation theory allowed us to predict a constant fraction which serves as an approximation guarantee to random hypergraph PPSPs. This approximation gaurantee is an exponential speed-up for spectral folding in the classicaly hard regime. To test these predictions numerically, we performed extensive quantum simulations to large N. We simulated system sizes where we could gather sufficient statistics, with super-computing classical computing resources for trotterized evolution of quantum dynamics.

We find the approximation guarantee provided by spectral folding a profound result. This guides to ask the question: Are there types of extremal hypergraph instances in which we would expect SFQO to fail at reaching the approximation target? A notion of failure here would imply that the approximation target A that can be reached in polynomial time would be effectively zero. We note that a significant decrease in the target approximation would still not be entirely a failure, and our numerical simulations do not demonstrate this to be the case for the system sizes we simulated in this work. However, it may be the case that there exists a dense problem hypergraph that can be constructed in such a way that spectral folding does not provide an approximation guarantee. Such a PPSP would need to be constructed in a rather specific way. Such a hypergraph would need to be constructed in a dense way, such that it cannot be solved by partitioning into smaller problems. We leave the problem of identifying extremal instances to future work.

Lastly, we expect our methods to be general to other problems not explored in this work. By formulating problems with a normalized ground state energy, further analytical predictions of the tunneling rate which defines the time-to-solution may be found for other **NP**-Hard optimization problems. For example, one could try spectral folding on well-studied problems in the literature such as the Max-Cut, quantum Max-Cut, or low auto-correlation binary sequence (LABS) problems. We leave this exciting direction open to future work.

3.8 Acknowledgements

We would like to thank Matthew Jones, Chris Laumann, Gianni Mossi, Eleanor Rieffel, Luca Trevisan, Antonello Scardicchio and Davide Venturelli for valuable discussions of the issues in this work. We would like to thank Caleb Rotello for detailed feedback on the manuscript. We would also like to thank Takuto Komatsuki and Joey Liu for support with HPC calculations. This material is based upon work supported by the DARPA Reversible Quantum Machine Learning and Simulation program under contract HR00112190068, as well as by National Science Foundation grants CCF-1839232, PHY-1653820, PHY-2210566, DGE-2125899, and by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Superconducting Quantum Materials and Systems Center (SQMS) under contract number DE-AC02-07CH11359. The SQMS Center supported EK's advisory role in this project, as well as time improving and fine tuning the algorithms and writing the paper. Many of the numerical simulations in this work were performed with a generous grant of HPC access from the Fujitsu Corporation. Part of this research was performed while the one of the authors (BAB) was visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1925919). The Flatiron Institute is a division of the Simons Foundation.

CHAPTER 4
ITERATIVE OPTIMIZATION OF HARD SPIN GLASS PROBLEMS WITH HIGH FREQUENCY AC DRIVES

To be submitted to *Physical Review Letters*.

Brandon A. Barton[1,2,7], Jacob Sagal[2,7], Sean Feeney[2], George Grattan[2,3], Pratik Patnaik[4],
Vadim Oganesyan[5,6], Lincoln D. Carr[1,2,4], Eliot Kapit[2,4]

4.1 Abstract

Binary constraint satisfaction problems represent a promising opportunity to achieve practical quantum advantage in real-world problems. These problems are in the worst case, and often, the typical case, exponentially hard to solve or even approximate for classical machines. Therefore, an efficient solution mechanism would have broad applicability to many problems in optimization, artificial intelligence, and cryptography. So far however, demonstrating consistent quantum advantage with heuristic algorithms has remained elusive. In particular, many variational algorithms offer only modest speedups and are made less efficient in practice by the cost of calculating gradients to guide the algorithm. In this work, we introduce a new algorithm, called IST-SAT (where IST stands for Iterative Symphonic Tunneling), which solves optimization problems by simulating the dynamics of quantum spin glasses in high-frequency oscillating transverse fields. IST-SAT operates as a sequence of rounds where the individual bitstrings returned from one round are used to choose the phase pattern of the AC drives in the next, steering the system toward the problem ground state through a novel mechanism not seen in other algorithms. We thoroughly benchmark IST-SAT on sets of hard MAX-3-XORSAT instances and report polynomial speedups over both trotterized adiabatic quantum computation (TAQC) and the best known classical algorithm. We expect that combining IST-SAT with more sophisticated future classical or quantum approximation algorithms, larger gains may be achieved. The mechanism we present in this work thus presents a new path toward achieving quantum advantage in optimization.

[1] Department of Applied Mathematics and Statistics, Colorado School of Mines, 1500 Illinois St, Golden CO 80401
[2] Quantum Engineering Program, Colorado School of Mines, 1523 Illinois St, Golden CO 80401
[7] B.B. and J.S. share co-first-authorship on this work.
[3] Department of Computer Science, Colorado School of Mines, 1500 Illinois St, Golden CO 80401
[4] Department of Physics, Colorado School of Mines, 1523 Illinois St, Golden CO 80401
[5] Department of Physics and Astronomy, College of Staten Island, City University of New York, Staten Island, NY 10314, USA
[6] Physics program and Initiative for the Theoretical Sciences, The Graduate Center, City University of New York, New York, NY 10016, USA

4.2 Introduction

A wide array of heuristic quantum algorithms, such as analog quantum annealing [51–57], adiabatic quantum computing (AQC) [49, 50], QAOA [15], and more exotic variations such as ADAPT-QAOA [104], recursive QAOA [105], QAOA supplemented with amplitude amplification [19], and energy matching/population transfer algorithms [106–109], have been proposed for spin glass optimization. Their empirical performance, however, has been decidedly mixed, and the frequently observed quadratic speedups from schedule optimization (a scheme dating back to the adiabatic formulation of Grover's search [110]) are fragile and likely not feasible at large N [111]. Furthermore, the problem can often be simulated with quantum Monte Carlo [112–116] for small systems sizes in the quadratic (incoherent) scaling limit, provided that the underlying Hamiltonian is stoquastic. In the search for applications of quantum algorithms to machine learning, recently, variational quantum algorithms have also been applied to spin glass optimization problems [117]. Yet, large gains from these methods may be limited by the cost of computing gradients and may often encounter barren plateaus in optimization procedure [118, 119]. Thus, the full capabilities of quantum computing in this space, and potential asymptotic limits for very general algorithms in this class, deserve further exploration. Given that beyond-quadratic speedups are critical to achieve useful quantum advantage [61], new mechanisms that expand the quantum optimization toolbox for reaching exact solutions, or even approximate solutions therefore have significant impacts.

In this work, we propose a non-classical steering mechanism that guides quantum optimization algorithms towards the true ground state in spin glass problems. We demonstrate this mechanism by introducing a new heuristic quantum algorithm which we call IST-SAT, standing for *Iterative Symphonic Tunneling for Satisfiability problems*. The IST-SAT algorithm modifies quasi-continuous time Trotterized AQC (TAQC), with total evolution time increasing linearly with N, by adding a monochromatic fast oscillating field along Y to all spins. The addition of the oscillating field is inspired by a recent mechanism termed *symphonic tunneling* (ST) which has demonstrated exponential acceleration of macroscopic quantum tunneling (MQT) times between ground states of transverse field Ising ferromagnets [31, 32].

In this work, we asked the question: how can the accelerated tunneling provided by symphonic tunneling be used in quantum optimization algorithms? In particular, we identify disordered spin glass optimization as a well-known hard problem, and demonstrate its promise as a good candidate problem for extending extend symphonic tunneling. In particular, we study a combinatorial optimization problem, in the class of constraint satisfaction problems (CSPs), known as MAX-3-XORSAT. In our results, we empirically observe that when fast AC drives are applied to hard MAX-3-XORSAT instances, the probabilities of finding both the true ground state G, and classical excited states near to G (in Hamming distance), are very sensitive to the local relative phase offsets $\varphi_j \in \{0, \pi\}$ inside single-qubit AC drives oscillating by a sinusoidal function $\sin(\omega t + \varphi_j)Y_j$. As the pattern of phases $P = \{\varphi_1, \varphi_2, \ldots, \varphi_N\}$ more closely align with the bit values ($b_j \in \pm 1$) in $G = \{b_1, b_2, \ldots, b_N\}$, we observe monotonically increasing probabilities of reaching G and states nearby to the global optima. When the phase patterns are uncorrelated to the ground state's bit pattern, we observe no benefit, while a perfect match finds ground states in polynomial time with constant probability. This relationship suggests an iterative algorithm strategy, where bitstrings from one round of shots are used to seed phase patterns in the next round. To test this strategy, we benchmark IST-SAT over a range of problem parameters from the partial planted solution problem (PPSP) class [120]. The results we demonstrate obtain significant polynomial speedups in some regimes, over both TAQC and the best known classical algorithm for this problem. In addition to the speed-ups that may be obtained using IST-SAT, our results provide new intuition for understanding the classical energy landscape in the set of MAX-3-XORSAT PPSP instances we study in this work. Finally, our numerical results thus suggest a new avenue for achieving quantum speedups in hard spin glass optimization.

4.3 Methods

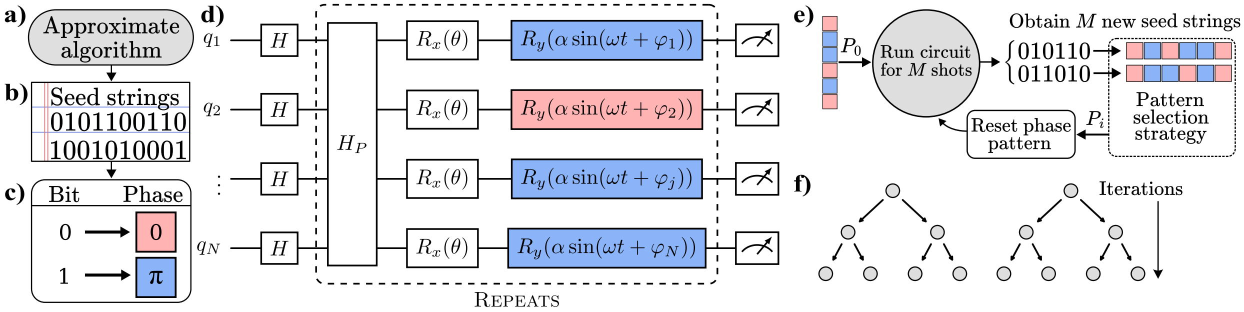

a)
Approximate algorithm
b)
Seed strings
0101100110
1001010001
c)
Bit
Phase
0
0
1
π
d)
q_1
q_2
⋮
q_N
H
H_P
$R_x(\theta)$
$R_y(\alpha \sin(\omega t + \varphi_1))$
$R_y(\alpha \sin(\omega t + \varphi_2))$
$R_y(\alpha \sin(\omega t + \varphi_j))$
$R_y(\alpha \sin(\omega t + \varphi_N))$
Repeats
e)
P_0
Run circuit for M shots
Obtain M new seed strings
010110
011010
Pattern selection strategy
P_i
Reset phase pattern
f)
Iterations

Figure 4.1 Schematic of the IST-SAT algorithm. An approximate algorithm–which can include random guessing and high-depth TAQC–in (a) is used to find "seed strings" (b) for the phase pattern in the H_{ST} drive Hamiltonian. The local phases $\{\varphi_1, \varphi_2, \ldots, \varphi_N\}$ in H_{ST} are set from bits of the seed states using the key (c), which we denote with colored R_y rotations in (d). After providing the algorithm an initial phase pattern P_0, the iterative process described in (e) continues, where the circuit is ran for M shots, generating new seed strings which are converted to phase patterns. A new pattern P_i is selected, which is then used in the next iteration. IST-SAT may be ran in a parallel branching process described in (f), to avoid local minima due to a poor initial seed string.

In contrast to many problems where asymptotic exponential scaling is not observed until prohibitively large system sizes for classical computers and NISQ era devices, 3-XORSAT problems have exponential scaling that is obvious at small N which makes them ideal for benchmarking quantum algorithms. We specifically work with the MAX-3-XORSAT [47] problem, defined on a random 3-uniform hypergraph consisting of N_C three-body constraint terms. The problem Hamiltonian is given by

$$H_P = -\sum_{ijk}^{N_C} V_{ijk} Z_i Z_j Z_k, \;\; V_{ijk} = \pm 1, \tag{4.1}$$

where for a given bitstring a constraint is satisfied if $\langle V_{ijk} Z_i Z_j Z_k \rangle = +1$, and unsatisfied otherwise. As the problem is linear, one can use Gaussian elimination to check if the problem is satisfiable in polynomial time. However, when not all the constraints can be satisfied, finding the lowest energy state(s) is known to be NP-hard [84].

This work utilizes a family of instances called planted partial solution problems (PPSPs), used in recent work on approximating MAX-3-XORSAT with quantum algorithms [120]. To construct a PPSP we first pick a random hypergraph with N variables that participate in N_C constraints with $N_C \gg N$, a fixed small fraction ϵ (we use $\epsilon = 0.1$ in this work), and a random bitstring G to be the planted ground state. We randomly select $(1-\epsilon) N_C$ constraints to be satisfied in G (by choosing the signs of the V_{ijk}) with the rest unsatisfied. For small ϵ and $N_C \gg N$, this makes G a unique ground state with high probability (the SAT/UNSAT transition here is at $N_C/N \sim 0.92$ [89]). These instances are more difficult than the 3-regular instances typically studied in the literature [45, 91], which are in practice easy to approximate and can be solved efficiently if they are satisfiable. Throughout this work, we studied constraint densities $N_C/N \in \{1.5, 2, 4\}$ to demonstrate the results of the algorithm over range of problem instances.

The schematic work-flow of the IST-SAT algorithm is shown in Figure 4.1, which begins by using an approximate algorithm to set parameters in the Hamiltonian. The initial approximation algorithm can be a greedy classical approach, simulated annealing, or a quantum algorithm such as AQC/TAQC or QAOA. The algorithm starts from the "standard" quasi-continuous time AQC method of interpolating between a transverse field "driver" Hamiltonian and the problem Hamiltonian with Trotterized evolution from $t = 0$ to $t = t_f$:

$$\begin{aligned} H(t) &= f(t) H_D + g(t) H_P, \;\; H_D = -\sum_j X_j, \\ f(0) &= g(t_f) = 1, \;\; f(t_f) = g(0) = 0. \end{aligned} \tag{4.2}$$

Throughout this work we use the the interpolation functions

$$f(t) = \sqrt{1 - t/t_f}, \;\; g(t) = \sqrt{t/t_f}. \tag{4.3}$$

which empirically outperforms simple linear interpolation for this problem, producing a better prefactor and modestly better scaling with N, though we expect the asymptotic scaling of the two schedules may converge. We let the total evolution time t_f grow linearly with N.

IST-SAT modifies the base Hamiltonian in (4.2) by adding a high-frequency monochromatic AC field $H_{ST}(t) = \alpha \sum_j Y_j \sin(\omega t + \varphi_j)$, producing the total Hamiltonian

$$H\left(t\right) = f\left(t\right) H_D + g\left(t\right) H_P + h\left(t\right) H_{ST}(t), \tag{4.4}$$

where $h\left(t\right)$ is a smooth function with initial and final conditions $h(t = 0), h(t_f) = 0$. Within $H_{ST}(t)$, single-qubit phases $\varphi_j = \{0, \pi\}$ form patterns $P = \{\varphi_1, \varphi_2, \ldots, \varphi_N\}$, and a drive strength $\alpha = \alpha_s \ln N$. In this work, we used $h\left(t\right) = 4\sqrt{\left(1 - t/t_f\right)\left(t/t_f\right)}$, $\alpha_s = 0.6$ and $\omega = 2\pi \times 6 \ln N$. The choice of α and ω both increasing logarithmically with N minimizes heating [121, 122] while ensuring that the novel low-frequency terms generated by the high frequency drive have constant magnitude.

The initial approximate algorithm in IST-SAT is used to find "seed" strings which are used to set phases in the high frequency AC drive $H_{ST}(t)$, which we call the symphonic tunneling Hamiltonian. We note that this set-up is distinct from *warm start* methods [23], which prepare a good initial state, defined by a large overlap with the ground state. IST-SAT still starts from the initial superposition state $|+\rangle^{\otimes N}$, and instead, sets parameters in the time-dependent Hamiltonian $H_{ST}(t)$. In one iteration of IST-SAT, the binary configuration of the "seed states", is used to form a phase pattern defined by the list of single qubit phases $\{\varphi_1, \varphi_2, \ldots, \varphi_N\}$, which is then ran using our Trotterized protocol to generate M new phase patterns, which can be used in further iterations of the algorithm.

We used a mean evolution time of $t_f = N/32$, with $dt = 0.4/\omega$ to ensure high frequencies are appropriately sampled; the total circuit depth thus scales as $N \ln N$ problem Hamiltonian applications. All data is averaged over 1000 random problem instances for each N. We also average over total run-times between $T = 2t_f/3$ and $T = 4t_f/3$, for all simulated cases. We have found [32] that this averaging effectively smooths out unpredictable diabatic effects that can make reliably estimating scaling difficult. The parameters we use in this work are the result of trial and error on small system sizes, with an eye toward testing on NISQ devices. Therefore we do not expect these parameters to be optimal, and emphasize that fine tuning is not necessary for IST-SAT to succeed.

4.4 Results

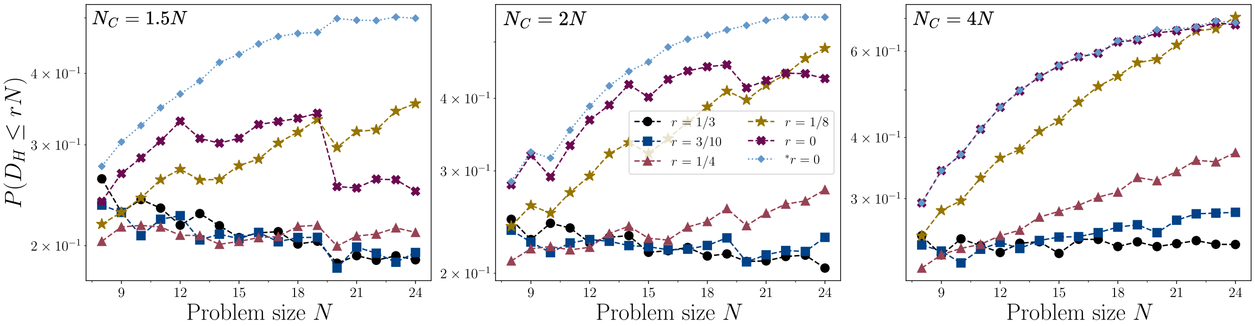


Figure 4.2 Convergence of IST-SAT. We seed the algorithm for one iteration with random phases chosen correctly (relative to the PPS) with probability $(1 - r)$, we plot the probability of returning bitstrings within $D_H \leq rN$ flips away from G. A "radius of convergence" r_c can be readily identified by starting from a single seed state $D_H \leq r_c N$. For the $^*r = 0$ case, we plot the probability to *any* ground state, along with the probability to the particular PPS in the $r = 0$ case.

Our results, shown in Figure 4.2, demonstrate that an appropriately chosen phase pattern will dramatically accelerate collective rearrangement at the transition from the paramagnetic initial state to the problem ground state. Of course, this choice of phase pattern is not obvious since the ground state G is generally not known, and the purpose of the algorithm is to find it. If we choose the phases φ_j randomly, the protocol in Eq. 4.4 shows no scaling advantage over the uniform field protocol in Eq. 4.2 with a modest prefactor disadvantage. Conversely, if we let $\varphi_j = \pi s_j$, where $s_j = \{0, 1\}$ is the value of bit j in G, $P_{GS}(t_f)$ is empirically *constant* with linear runtime $t_f \propto N$ (see the $r = 0$ data in Figure 4.2), as compared to exponential decay in the uniform field case. Making this choice requires knowledge of G, and thus the solution to the problem, so it's not obvious that this discovery will help us. The observation that motivates IST-SAT is the following: let's say we guess the phases φ_j correctly with some probability $r \geq 1/2$ (based on the values of each bit in G, and note that it's the relative phases that matter here). As r increases toward 1, the probabilities (at equivalent total evolution time) of finding both the ground state, and states comparatively close to it in Hamming distance, monotonically increase as shown in Figure 4.2. If the bitstrings returned from early iterations of IST-SAT are used to choose the phase pattern for subsequent iterations, then the solutions returned from those shots will be closer still and the algorithm can converge quickly to the ground state.

The results obtained from large scale quantum simulation in this work consistently presents an empirical *radius of convergence* r_c, which depends on N_C/N and the unsatisfied fraction ϵ in the ground state. We define r_c in the following way: let's say we guess each local phase correctly with probability $1 - r$. Then, for linearly growing t_f, r_c is defined to be the largest value of r such that the probability of returning states with a Hamming distance $\leq rN$ is constant with increasing N. For $N_C/N = \{1.5, 2, 4\}$ and $\epsilon = 0.1$, we observe respective $r_c \simeq \{1/8, 1/4, 1/3\}$ and simulations of smaller systems with larger N_C/N (data not shown) seem consistent with $r_c \simeq 1/3$ above this. When $r \leq r_c$ the probability of returning states substantially closer to the ground state, e.g. closer than $rN/2$, scales nearly identically to rN (see supplementary material). Of course, if a bitstring of distance rN from the true ground state is used to set the phase pattern, that is equivalent to guessing correctly with probability $1 - r$.

Table 4.1 Inferred time to solution (TTS) defined by the average number of trials (shots to run the circuit) to reach the ground state. In this table we report the exponent b obtained from fits to numerical data of the form $a2^{bN}$. We test different algorithm set ups including: quasi-greedy classical (GC)[45], TAQC[15], and spectrally folded quantum optimization (SFQO)[120]. For the algorithms respective use as seed algorithm for IST-SAT, we assume that the probabilities from IST-SAT are constant once the approximate state obtains a fraction in the radius of convergence such that $r \leq r_c$. Therefore, we report the exponent b at each seed algorithm's approximation distance corresponding to the radius of convergence $d = r_c$ (other distance fractions d are provided in Appendix C).

Algorithm set-up	$N_C/N = 1.5$	$N_C/N = 2$	$N_C/N = 4$
Quasi-greedy classical	-2.3e-1	-1.9e-1	-7.5e-2
TAQC	-3.0e-1	-2.5e-1	-2.8e-1
GC + IST-SAT	-1.5e-1	-8.3e-2	-4.5e-2
TAQC + IST-SAT	-1.1e-1	-5.5e-2	-2.6e-2
SFQO + IST-SAT	-7.0e-3	-5.6e-3	3.4e-3

Therefore, if the approximate "seed" strings are within some $r < r_c$, the bitstrings returned each round in IST-SAT will get closer and closer to the global optimum; with r_c defined in this way, we expect convergence from a state less than r_cN flips to the true ground state in a polynomial number of algorithm iterations. If we start from states which are more than r_c flips away, iterating phase patterns still is expected to converge to the true ground state faster (e.g. a smaller scaling exponent) than the "DC" TAQC algorithm IST-SAT uses as a starting point, as can be inferred from the smaller slopes for the curves with $r_c < r < 1/2$ in Figure 4.2 as compared to the TAQC data, though the quantitative analysis of scaling is more complex in that case.

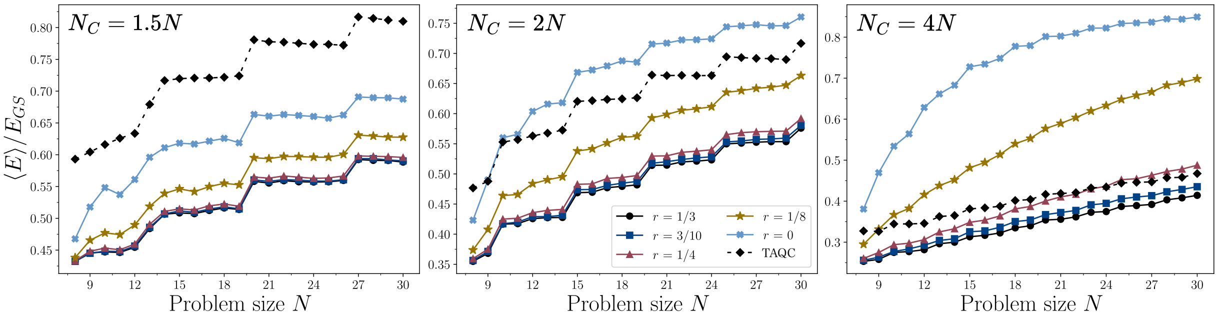


Figure 4.3 Average energy $\langle E \rangle$ returned from IST-SAT and TAQC, normalized by the ground state energy E_{GS}. Energy is obtained using H_P as the cost function. Here, we run a single iteration of IST-SAT, and use 10,000 samples (bitstrings) from the wave function in exact state vector simulation. The legend in the center figure specifies the fraction of incorrect bits rN used in the seed state protocol.

To examine the *quality of strings* returned from IST-SAT, we plot the average energy $\langle E \rangle$ (normalized by E_{GS}) in Figure 4.3. Interestingly, the rapid convergence to the ground state is based *only* on Hamming distance D_H and remains uncorrelated with the energies E of the returned states. However, we still observe monotonically increasing average string qualities as IST-SAT is seeded with better phase patterns. Our very first formulation of IST-SAT used a variation of the quasi-greedy algorithm in [45] to gather the initial seed states, and we saw no improvement over random guessing in making this choice. This is supported by the fact that, unless r is very close to one, the *average* energy returned by IST-SAT is somewhat worse than that of TAQC for all other parameters equal (see the supplemental information for figures). We attribute this fact to there being many local minima with energies close to E_{GS}, and the effect of the high frequency AC drive is to steer the evolving state away from them and toward $|G\rangle$ and its local excitations, which may be higher in energy even if they are much closer in Hamming distance.

It may be the case that other classical algorithms could prove better for the purpose of seeding IST-SAT; for example, an algorithm that broke the problem into sub-problems and solved those exactly to get sets of *relative* phase patterns could potentially yield more significant improvements, though we have not yet explored this possibility in detail. One could also try versions of IST-SAT which assigned each group of spins a different frequency (not just a different phase pattern), for better averaging, as in the earlier multi-frequency AC optimization schemes [32, 82, 111, 123] that inspired this work.

While naive classical seeding was unsuccessful, matching the observed $r_c \simeq \{1/8, 1/4, 1/3\}$ with the corresponding distance probabilities for TAQC yields significantly reduced asymptotic time to solution if strings from TAQC (or equivalently, random phase patterns) are used for initial seeding. Selecting fractions of the TAQC strings based on their energies yielded no benefit (reflecting the low correlation between distance and energy unless one is very close to the ground state, as expected from Eq. B.3 derived below) over randomly sampling the TAQC output distribution.

Interestingly our results demonstrate that at low constraint density $N_C/N = \{1.5, 2\}$, the performance for $r = 0$ degrades at large N. We computed the exact ground states over all 1000 problems at each system size through $N = 24$, and find that the decay in performance is likely due to more degeneracy in the ground state manifold, leading to an energy landscape with more ground states. In the Supplementary Material, we show that the probability that IST-SAT reaches any ground state does not decay with increasing problem size, and improves radii of convergence compared to those r_c we identified for reaching the planted ground state in particular. Thus, these results demonstrate that even when IST-SAT is seeded with the phase pattern from a different ground state, the solutions obtained maintain a high probability of reaching a state within the ground state manifold.

4.5 Conclusion

In conclusion, IST-SAT is a unique feedback-based quantum algorithm that does not require the calculation of any gradients or averages (as in the original variational QAOA proposal, or more recent versions such as recursive QAOA [105]), which are costly as they require hundreds to thousands of shots to calculate accurately. The core mechanism for how the phase pattern appropriately guides the optimization is only understood by analogy to other work in a simpler system [31], and we expect a careful analytical derivation of its speedup would undoubtedly lead to further algorithm innovations. While in this work, we choose the phase patterns as discrete offsets of $\{0, \pi\}$, we expect that more sophisticated phase pattern selection protocols would further improve IST-SAT. In light of recent work which use semi-definite programs to set the parameters of variational quantum circuits [124], future work may consider selecting phases in the H_{ST} drive in a continuous interval $\varphi_j = [0, \pi]$, which we expect may improve results for MAX-3-XORSAT and other optimization problems such as Max-Cut.

4.6 Acknowledgments

We would like to thank Ojas Parekh for valuable discussions. We would also like to thank Takuto Komatsuki and Joey Liu for support with quantum simulations on the Fujitsu Quantum Simulator. This material is based upon work supported by the National Science Foundation under Grant Numbers CCF-1839232 and DGE-2125899. Part of this research was performed while one of the authors (BAB) was visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1925919).

†B.B. and J.S. contributed equally to this work. E.K. conceptualized the project. We note the results obtained from quantum simulation in this work reach the capabilities of classical super-computing

resources for simulation of quantum dynamics, which may not be accessible for all research groups.

CHAPTER 5
CONCLUSION AND OUTLOOK

In this chapter, we recount on the main results in Chapters 3 and 4. In Section 5.1, we further elaborate the connection between the works in this thesis. Finally, in Section 5.2 we provide an outlook for the state of quantum approximation algorithms for constraint satisfaction problems, and provide several concrete problems that may be considered in future work.

5.1 Conclusion

The primary Chapters 3 and 4 in this thesis investigate quantum algorithms for a combinatorial optimization problem, known as MAX-3-XORSAT, in the class of constraint satisfaction problems. When the problem cannot be fully satisfied, both finding the ground state, or any α-approximation of the ground state better than random guessing ($\alpha = 1/2$) is **NP**-hard. We investigate the ability of quantum algorithms to do both tasks, including finding good approximations ($\alpha > 1/2$) and the optimal solution ($\alpha = 1$). We specifically work with hard partial planted solution problem instances, beyond the SAT/UNSAT threshold, where the ground state does satisfy all of the problem constraints.

In Chapter 3, we showed that the mechanism of spectrally folded quantum optimization increases the availability of approximate solutions, thereby significantly increasing the probability of obtaining such states. By applying a classical deformation to the energy landscape, the spectral folding mechanism allows the algorithm to obtain approximate solutions with a guaranteed approximation ratio. We introduced the algorithm in three variants: linear, and quadratic, and trial minimum annealing (TMA). We tested these variants of SFQO using extensive quantum simulation on a combination of super-computing, and high-performance computing resources. The numerical results obtained demonstrate that SFQO obtains approximate solutions in polynomial time, where previously known classical and quantum algorithms are expected to take exponential time in the size of the problem. Thus, we conclude that the mechanism provided by SFQO is an exponential speed-up for approximating the MAX-3-XORSAT problem class.

In Chapter 4, we introduced a new non-classical steering mechanism inspired by recent results of accelerated quantum tunneling [31] which we believe is applicable to a wide variety of optimization problems. We demonstrate this steering mechanism by introducing a new iterative quantum algorithm called Iterative Symphonic Tunneling for Satisfiability problems, or IST-SAT for short. IST-SAT uses an oscillating Y perturbation whose parameters are set using the bits from an approximate algorithm. The set-up for IST-SAT naturally motivates an iterative procedure, where the results from previous trials are used in the next iteration. We tested the IST-SAT algorithm using different trial seed states from several classical and quantum algorithms. Namely, we performed large numerical simulation to benchmark the performance of quasi-greedy classical approaches, adiabatic quantum algorithms, and spectrally folded quantum optimization procedures. In our simulations of MAX-3-XORSAT partial planted solution problem instances, over a set of constraint densities $N_C/N = \{1.5, 2, 4\}$, we find extensive numerical evidence of critical approximation ratios r_c in terms of Hamming distance to the planted solution. When IST-SAT is seeded with a sufficient approximation, as the iterations proceed, IST-SAT may converge to the ground state in a logarithmic number iterations, which implies a polynomial number of samples. A sufficient approximation is defined as a seed state that is a Hamming distance $D_H \leq r_c N$ away from the planted solution. We showed that when IST-SAT is seeded with string from an approximate algorithm, the distance of the solution we obtain to the ground state, measured in Hamming distance, improves monotonically with quality of the seed state. In other words, the better state we give IST-SAT, the better solution we obtain demonstrating a dependence on initial approximation.

Reflecting on the results in Chapters 3 and 4, we conclude and discuss the opportunity to combine the methods provided in both SFQO and IST-SAT algorithms, therefore obtaining significant speed-ups over the best known classical and quantum algorithms for the MAX-3-XORSAT problem. Shown in the quantum simulations performed in these work, we obtained exponential speed-ups for approximating MAX-3-XORSAT using spectrally folded quantum optimization. Then, using the novel steering mechanism of high frequency AC drives, we showed that when IST-SAT is provided a sufficiently good initial approximation, the algorithm can converge to the optimal solution in polynomial time. Thus, our two results motivate combining both algorithms in the following way: First, use spectrally folded quantum optimization to generate approximate solutions to the MAX-3-XORSAT problem. Then, using the approximate solutions as seed strings from spectral folding, run IST-SAT to converge to the optimal solution of the problem.

Provided that spectral folding returns states at or above the radius of convergence we identified in Chapter 4, we expect to converge to the optimal solution in polynomial time. Of course, we don't expect that these methods will always return polynomial-time exact solutions on the hardest extremal problems. Rather, the probabilities of reaching the global optima still demonstrated small exponential decay as expected for an **NP**-Hard problem, as summarized in the various algorithm set ups for IST-SAT in Chapter 4. However, this exponential decay is significantly smaller than previous classical and quantum algorithms, thus demonstrating the novel combination of the work in this thesis.

5.2 Outlook

In this section, we conclude by providing several directions that may be considered for future work.

For Chapter 3 in particular, future work may consist of the following:

- Define other variations of spectral folding. By defining a new class of deformation functions, one may obtain further performance enhancements. Furthermore, we expect the suitable function to depend on the problem one is wishing to solve. For example, if the problem is known to have a high clustering of local minima close to the ground state, the appropriate choice of approximation target A may need to be suitably chosen closer or further away from the ground state energy. New variations should be considered on a problem-by-problem basis.

- Use spectral folding as a seed algorithm for warm-start quantum and classical algorithms. We note that this direction is distinct from using SFQO as a seed algorithm for IST-SAT. It may be the case that for some problems, obtaining approximate solutions remains hard, although converging to the true ground state is relatively easier when provided with a sufficient initial approximation. This direction could consider the ability of both classical and quantum methods to find exact solutions using initial solutions from SFQO.

Similarly, for Chapter 4, we expect the following direction for future work to be promising:

- Refine the critical radius of convergence. In this work, we presented several radii of convergence for a range of MAX-3-XORSAT problem constructions. It would be worthwhile to do a study that refines the values of r_c across a larger set of constraint densities. This would allow the exact relationship between r_c and the constraint density to shine through, via numerical simulations. Of course, we expect these values to be dependent on the size of time-step and numerical simulation details that would need to be further considered.

- Try IST-SAT with new approximation algorithms. As new classical and quantum approximation algorithms arise in the literature, it would be interesting to test IST-SAT with new methods. While we chose discrete values of $\{0, \pi\}$ for the phase offsets in the AC drive, one could consider setting the single-qubit phase offsets according to a more sophisticated routine. For example, several improvements along the routes of semi-definite programming (SDP) may provide an interesting direction for future work. The way in which SDP algorithms relax the discrete optimization values to a continuous search space may allow one to choose phase offsets from a continuous interval $[0, \pi]$, allowing for more fine-tuning of the parameters. We expect further benefits may be obtained using this method.

- Characterize the effect of problem structure on collective tunneling mechanisms to the ground state. As discussed, the sparsity of the problem has a significant effect on the hardness of approximation and exact problem solving. We would like to understand the exact effect of problem structure, in terms of graph metrics such as density, regularity, and the girth (largest cycle in the graph). By studying this relationship, one may uncover the dynamical mechanisms of collective tunneling through large energy barriers that otherwise prohibit classical methods from finding the true ground state(s).

As discussed in the conclusions of Chapters 3 and 4, we expect the mechanisms proposed in this work to be general for a broad class of combinatorial optimization problems. Following promising work on approximation guarantees for Max-Cut and quantum Max-Cut, we expect both of these problems to be good candidates for further applying SFQO and IST-SAT. In both of these works, the question of classical simulability is an important question. While we expect the entanglement to be volume-law scaling in both algorithms, a further study of the entanglement dynamics would be illuminating.

Finally, as we expect these algorithms to be implementable on near-term quantum hardware, gate-based compilation techniques may be used to perform an end-to-end cost analysis. We expect such a cost analysis to inform future experimental demonstrations of SFQO and IST-SAT. As these problems require highly non-local gates, we expect a quantum device with all-to-all coupling would be best suited for the implementation of these algorithms. Namely, current neutral atom and trapped-ion devices would be interesting candidate hardware platforms to test the viability of SFQO and IST-SAT under the presence of noise.

REFERENCES

[1] Michael A. Nielsen and Issac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2011.

[2] Markus J. Pflaum Daniel Spiegel Frédéric Pierre-Jean Latrémolière, Paul Mitchener. Functional analysis and non-commutative geometry, 2024.

[3] Daniel A. Lidar. Lecture notes on the theory of open quantum systems, 2020.

[4] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra, Twenty-fifth Anniversary Edition.* Society for Industrial and Applied Mathematics, 2022.

[5] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem, 2005.

[6] H. F. Trotter. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 1959.

[7] Naomichi Hatano and Masuo Suzuki. *Finding Exponential Product Formulas of Higher Orders.* Springer Berlin Heidelberg, 2005.

[8] S. Arora and B. Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, 2006.

[9] Oded Goldreich. *In a world of P=BPP.* Springer-Verlag, Berlin, Heidelberg, 2011.

[10] Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.

[11] Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, and Fernando G. S. L. Brandão. Quantum algorithms: A survey of applications and end-to-end complexities, 2023.

[12] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 2021.

[13] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000.

[14] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, January 2018.

[15] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[16] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. Quantum approximate optimization algorithm for maxcut: A fermionic view. *Physical Review A*, 2018.

[17] Sepehr Ebadi, Alexander Keesling, Madelyn Cain, Tout T Wang, Harry Levine, Dolev Bluvstein, Giulia Semeghini, Ahmed Omran, J-G Liu, Rhine Samajdar, et al. Quantum optimization of maximum independent set using rydberg atom arrays. *Science*, 2022.

[18] Elijah Pelofske, Andreas Bärtschi, John Golden, and Stephan Eidenbenz. High-round qaoa for max k-sat on trapped ion nisq devices. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2023.

[19] Ruslan Shaydulin, Changhao Li, Shouvanik Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, Yuri Alexeev, Joan M. Dreiling, John P. Gaebler, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Nathan Hewitt, Chandler V. Horst, Shaohan Hu, Jacob Johansen, Mitchell Matheny, Tanner Mengle, Michael Mills, Steven A. Moses, Brian Neyenhuis, Peter Siegfried, Romina Yalovetzky, and Marco Pistoia. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem, 2023.

[20] Shree Hari Sureshbabu, Dylan Herman, Ruslan Shaydulin, Joao Basso, Shouvanik Chakrabarti, Yue Sun, and Marco Pistoia. Parameter Setting in Quantum Approximate Optimization of Weighted Problems. *Quantum*, 2024.

[21] Joao Basso, David Gamarnik, Song Mei, and Leo Zhou. Performance and limitations of the qaoa at constant levels on large sparse hypergraphs and spin glass models. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022.

[22] Constantin Brif, Matthew D. Grace, Mohan Sarovar, and Kevin C. Young. Exploring adiabatic quantum trajectories via optimal control. *New Journal of Physics*, 2014.

[23] Madelyn Cain, Edward Farhi, Sam Gutmann, Daniel Ranard, and Eugene Tang. The qaoa gets stuck starting from a good classical string, 2023.

[24] Bei Zeng, Xie Chen, Duan-Lu Zhou, and Xiao-Gang Wen. *Quantum Information Meets Quantum Matter: From Quantum Entanglement to Topological Phases of Many-Body Systems*. Quantum Science and Technology. Springer New York, 2019.

[25] Frank Verstraete, Tomotoshi Nishino, Ulrich Schollwöck, Mari Carmen Bañuls, Garnet K. Chan, and Miles E. Stoudenmire. Density matrix renormalization group, 30 years on. *Nature Reviews Physics*, 2023.

[26] Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell, 2017.

[27] Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nature Physics*, 2017.

[28] Thomas Young. I. The Bakerian Lecture. Experiments and calculations relative to physical optics. *Philosophical Transactions of the Royal Society of London*, 1804.

[29] R. Cleve, A. Ekert, L. Henderson, C. Macchiavello, and M. Mosca. On quantum algorithms, 1999.

[30] John van de Wetering. Zx-calculus for the working quantum computer scientist, 2020.

[31] George Grattan, Brandon A. Barton, Sean Feeney, Gianni Mossi, Pratik Patnaik, Jacob C. Sagal, Lincoln D. Carr, Vadim Oganesyan, and Eliot Kapit. Exponential acceleration of macroscopic quantum tunneling in a floquet ising model, 2024.

[32] Gianni Mossi, Vadim Oganesyan, and Eliot Kapit. Embedding quantum optimization problems using ac driven quantum ferromagnets. *arXiv:2306.10632*, 2023.

[33] Yasunari Suzuki, Yoshiaki Kawase, Yuya Masumura, Yuria Hiraga, Masahiro Nakadai, Jiabao Chen, Ken M Nakanishi, Kosuke Mitarai, Ryosuke Imai, Shiro Tamiya, et al. Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum*, 5:559, 2021.

[34] Bernhard H Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization.* Springer, 2011.

[35] Michael R Garey, David S Johnson, and Larry Stockmeyer. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, 1974.

[36] Jeffrey D. Ullman. Np-complete scheduling problems. *Journal of Computer and System sciences*, 1975.

[37] Gary S Grest, CM Soukoulis, and K Levin. Cooling-rate dependence for the spin-glass ground-state energy: Implications for optimization by simulated annealing. *Physical Review Letters*, 1986.

[38] Pierluigi Crescenzi, Viggo Kann, and M Halldórsson. A compendium of np optimization problems, 1995.

[39] Dorit S Hochba. Approximation algorithms for np-hard problems. *ACM Sigact News*, 1997.

[40] Rémi Monasson. Optimization problems and replica symmetry breaking in finite connectivity spin glasses. *Journal of Physics A: Mathematical and General*, 1998.

[41] Gerhard J Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial optimizationâeureka, you shrink!* Springer, 2003.

[42] Victor Bapst, Laura Foini, Florent Krzakala, Guilhem Semerjian, and Francesco Zamponi. The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective. *Physics Reports*, 2013.

[43] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 2013.

[44] Davide Venturelli, Salvatore Mandrà, Sergey Knysh, Bryan Gorman, Rupak Biswas, and Vadim Smelyanskiy. Quantum optimization of fully connected spin glasses. *Physical Review X*, 2015.

[45] Matteo Bellitti, Federico Ricci-Tersenghi, and Antonello Scardicchio. Entropic barriers as a reason for hardness in both classical and quantum algorithms, 2021.

[46] Chris Jones, Kunal Marwaha, Juspreet Singh Sandhu, and Jonathan Shi. Random max-csps inherit algorithmic hardness from spin glasses. *arXiv:2210.03006*, 2022.

[47] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[48] Lucas Slot. Sum-of-Squares Hierarchies for Polynomial Optimization and the Christoffel–Darboux Kernel. *SIAM Journal on Optimization*, 2022.

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

[50] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computing. *arXiv:1611.04471*, 2017.

[51] AB Finnila, MA Gomez, C Sebenik, C Stenson, and JD Doll. Quantum annealing: a new method for minimizing multidimensional functions. *Chemical physics letters*, 1994.

[52] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 1998.

[53] Arnab Das and Bikas K Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 2008.

[54] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, R Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. Quantum annealing with manufactured spins. *Nature*, 2011.

[55] Sergio Boixo, Troels F Rønnow, Sergei V Isakov, Zhihui Wang, David Wecker, Daniel A Lidar, John M Martinis, and Matthias Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 2014.

[56] Philipp Hauke, Helmut G Katzgraber, Wolfgang Lechner, Hidetoshi Nishimori, and William D Oliver. Perspectives of quantum annealing: Methods and implementations. *arXiv:1903.06559*, 2019.

[57] Andrew D King, Jack Raymond, Trevor Lanting, Richard Harris, Alex Zucca, Fabio Altomare, Andrew J Berkley, Kelly Boothby, Sara Ejtemaee, Colin Enderud, et al. Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature*, 2023.

[58] Tameem Albash and Daniel A Lidar. Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Physical Review X*, 2018.

[59] Andrew D King, Juan Carrasquilla, Jack Raymond, Isil Ozfidan, Evgeny Andriyash, Andrew Berkley, Mauricio Reis, Trevor Lanting, Richard Harris, Fabio Altomare, et al. Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature*, 2018.

[60] Humberto Munoz Bauza and Daniel A. Lidar. Scaling advantage in approximate optimization with quantum annealing, 2024.

[61] Ryan Babbush, Jarrod R McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum*, 2021.

[62] Giorgio Parisi. Infinite number of order parameters for spin-glasses. *Physical Review Letters*, 1979.

[63] David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 2005.

[64] Yuki Susa, Yu Yamashiro, Masayuki Yamamoto, Itay Hen, Daniel A. Lidar, and Hidetoshi Nishimori. Quantum annealing of the p-spin model under inhomogeneous transverse field driving. *Physical Review A*, 2018.

[65] Tobias Graß. Quantum annealing with longitudinal bias fields. *Physical Review Letters*, 2019.

[66] Dries Sels and Anatoli Polkovnikov. Minimizing irreversible losses in quantum systems by local counterdiabatic driving. *Proceedings of the National Academy of Sciences*, 2017.

[67] Seraph Bao, Silken Kleer, Ruoyu Wang, and Armin Rahmani. Optimal control of superconducting gmon qubits using pontryagin's minimum principle: Preparing a maximally entangled state with singular bang-bang protocols. *Physical Review A*, 2018.

[68] Zhi-Cheng Yang, Armin Rahmani, Alireza Shabani, Hartmut Neven, and Claudio Chamon. Optimizing variational quantum algorithms using pontryagin's minimum principle. *Physical Review X*, 2017.

[69] Ieva Čepaitė, Anatoli Polkovnikov, Andrew J. Daley, and Callum W. Duncan. Counterdiabatic optimized local driving. *PRX Quantum*, 2023.

[70] Andrea Montanari. Optimization of the sherrington–kirkpatrick hamiltonian. *SIAM Journal on Computing*, 2021.

[71] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. *Quantum*, 2022.

[72] Sami Boulebnane and Ashley Montanaro. Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit. *arXiv:2110.10685*, 2021.

[73] Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou. The quantum approximate optimization algorithm at high depth for maxcut on large-girth regular graphs and the sherrington-kirkpatrick model. *arXiv:2110.14206*, 2021.

[74] Boris Altshuler, Hari Krovi, and Jérémie Roland. Anderson localization makes adiabatic quantum optimization fail. *Proceedings of the National Academy of Sciences*, 2010.

[75] Sergey Knysh. Zero-temperature quantum annealing bottlenecks in the spin-glass phase. *Nature communications*, 2016.

[76] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 2002.

[77] Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 2005.

[78] Alexander K Hartmann and Martin Weigt. *Phase transitions in combinatorial optimization problems: basics, algorithms and statistical mechanics*. John Wiley & Sons, 2006.

[79] Florent Krzakala and Jorge Kurchan. Landscape analysis of constraint satisfaction problems. *Physical Review E*, 2007.

[80] Fabrizio Altarelli, Rémi Monasson, and Francesco Zamponi. Relationship between clustering and algorithmic phase transitions in the random k-xorsat model and its np-complete extensions. In *Journal of Physics: Conference Series*. IOP Publishing, 2008.

[81] Morteza Ibrahimi, Yashodhan Kanoria, Matt Kraning, and Andrea Montanari. The set of solutions of random xorsat formulae. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012.

[82] Zhijie Tang and Eliot Kapit. Unconventional quantum annealing methods for difficult trial problems. *Physical Review A*, 2021.

[83] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 1976.

[84] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 2001.

[85] Subhash Khot and Assaf Naor. Linear equations modulo 2 and the l1 diameter of convex bodies. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 2007.

[86] Sarah R Allen, Ryan O'Donnell, and David Witmer. How to refute a random csp. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 2015.

[87] Tommaso d'Orsi and Luca Trevisan. A ihara-bass formula for non-boolean matrices and strong refutations of random csps. *arXiv:2204.10881*, 2022.

[88] Boaz Barak, Ankur Moitra, Ryan O'Donnell, Prasad Raghavendra, Oded Regev, David Steurer, Luca Trevisan, Aravindan Vijayaraghavan, David Witmer, and John Wright. Beating the random assignment on constraint satisfaction problems of bounded degree. *arXiv:1505.03424*, 2015.

[89] Olivier Dubois and Jacques Mandler. The 3-xorsat threshold. *Comptes Rendus Mathematique*, 2002.

[90] Johan Håstad and Srinivasan Venkatesh. On the advantage over a random assignment. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002.

[91] Matthew Kowalsky, Tameem Albash, Itay Hen, and Daniel A Lidar. 3-regular 3-xorsat planted solutions benchmark of classical and quantum heuristic optimizers. *arXiv:2103.08464*, 2021.

[92] Matthew B Hastings. A short path quantum algorithm for exact optimization. *Quantum*, 2018.

[93] Alexander M Dalzell, Nicola Pancotti, Earl T Campbell, and Fernando GSL Brandão. Mind the gap: Achieving a super-grover quantum speedup by jumping to the end. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 2023.

[94] Anurag Anshu, David Gosset, Karen J Morenz Korol, and Mehdi Soleimanifar. Improved approximation algorithms for bounded-degree local hamiltonians. *Physical Review Letters*, 2021.

[95] Kunal Marwaha and Stuart Hadfield. Bounds on approximating max k xor with quantum and classical local algorithms. *Quantum*, 2022.

[96] Lin Wang Wang and Alex Zunger. Electronic structure pseudopotential calculations of large (. apprx. 1000 atoms) si quantum dots. *The Journal of Physical Chemistry*, 1994.

[97] Lin-Wang Wang and Alex Zunger. Solving schrödinger's equation around a desired energy: Application to silicon quantum dots. *The Journal of Chemical Physics*, 1994.

[98] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 2016.

[99] Raffaele Santagati, Jianwei Wang, Antonio A Gentile, Stefano Paesani, Nathan Wiebe, Jarrod R McClean, Sam Morley-Short, Peter J Shadbolt, Damien Bonneau, Joshua W Silverstone, et al. Witnessing eigenstates for quantum simulation of hamiltonian spectra. *Science advances*, 2018.

[100] Feng Zhang, Niladri Gomes, Yongxin Yao, Peter P Orth, and Thomas Iadecola. Adaptive variational quantum eigensolvers for highly excited states. *Physical Review B*, 2021.

[101] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 2022.

[102] Lila Cadi Tazi and Alex JW Thom. Folded spectrum vqe: A quantum computing method for the calculation of molecular excited states. *arXiv:2305.04783*, 2023.

[103] Eliot Kapit. Systems and methods for accelerated quantum optimization, March 25 2021. US Patent App. 17/027,146.

[104] Linghua Zhu, Ho Lun Tang, George S Barron, FA Calderon-Vargas, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. *Physical Review Research*, 2022.

[105] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Hybrid quantum-classical algorithms for approximate graph coloring. *Quantum*, 2022.

[106] CL Baldwin and CR Laumann. Quantum algorithm for energy matching in hard optimization problems. *Physical Review B*, 2018.

[107] Vadim N Smelyanskiy, Konstyantyn Kechedzhi, Sergio Boixo, Sergei V Isakov, Hartmut Neven, and Boris Altshuler. Non-ergodic delocalized states for efficient population transfer within a narrow band of the energy landscape. *arXiv:1802.09542*, 2018.

[108] Kostyantyn Kechedzhi, Vadim Smelyanskiy, Jarrod R McClean, Vasil S Denchev, Masoud Mohseni, Sergei Isakov, Sergio Boixo, Boris Altshuler, and Hartmut Neven. Efficient population transfer via non-ergodic extended states in quantum spin glass. *arXiv:1807.04792*, 2018.

[109] Vadim N Smelyanskiy, Kostyantyn Kechedzhi, Sergio Boixo, Hartmut Neven, and Boris Altshuler. Intermittency of dynamical phases in a quantum spin glass. *arXiv:1907.01609*, 2019.

[110] J. Roland and N. J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 2002.

[111] Eliot Kapit and Vadim Oganesyan. Noise-tolerant quantum speedups in quantum annealing without fine tuning. *Quantum Science and Technology*, 2021.

[112] Sergei V Isakov, Guglielmo Mazzola, Vadim N Smelyanskiy, Zhang Jiang, Sergio Boixo, Hartmut Neven, and Matthias Troyer. Understanding quantum tunneling through quantum monte carlo simulations. *Physical Review Letters*, 2016.

[113] Evgeny Andriyash and Mohammad H Amin. Can quantum monte carlo simulate quantum annealing? *arXiv:1703.09277*, 2017.

[114] Zhang Jiang, Vadim N Smelyanskiy, Sergei V Isakov, Sergio Boixo, Guglielmo Mazzola, Matthias Troyer, and Hartmut Neven. Scaling analysis and instantons for thermally assisted tunneling and quantum monte carlo simulations. *Physical Review A*, 2017.

[115] Zhang Jiang, Vadim N Smelyanskiy, Sergio Boixo, and Hartmut Neven. Path-integral quantum monte carlo simulation with open-boundary conditions. *Physical Review A*, 2017.

[116] Andrew D King, Jack Raymond, Trevor Lanting, Sergei V Isakov, Masoud Mohseni, Gabriel Poulin-Lamarre, Sara Ejtemaee, William Bernoudy, Isil Ozfidan, Anatoly Yu Smirnov, et al. Scaling advantage in quantum simulation of geometrically frustrated magnets. *arXiv:1911.03446*, 2019.

[117] Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 2021.

[118] Michael Ragone, Bojko N. Bakalov, Frédéric Sauvage, Alexander F. Kemper, Carlos Ortiz Marrero, Martin Larocca, and M. Cerezo. A unified theory of barren plateaus for deep parametrized quantum circuits, 2023.

[119] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 2021.

[120] Eliot Kapit, Brandon A. Barton, Sean Feeney, George Grattan, Pratik Patnaik, Jacob Sagal, Lincoln D. Carr, and Vadim Oganesyan. On the approximability of random-hypergraph max-3-xorsat problems with quantum algorithms, 2024.

[121] Dmitry A Abanin, Wojciech De Roeck, and François Huveneers. Exponentially slow heating in periodically driven many-body systems. *Physical Review Letters*, 2015.

[122] Tomotaka Kuwahara, Takashi Mori, and Keiji Saito. Floquet–magnus theory and generic transient dynamics in periodically driven many-body quantum systems. *Annals of Physics*, 2016.

[123] Eliot Kapit. Systems and methods for passive quantum error correction, March 23 2021. US Patent 10,956,267.

[124] Robbie King. An Improved Approximation Algorithm for Quantum Max-Cut on Triangle-Free Graphs. *Quantum*, 2023.

[125] Sergey Bravyi, David P DiVincenzo, and Daniel Loss. Schrieffer–wolff transformation for quantum many-body systems. *Annals of physics*, 2011.

[126] Francesca Pietracaprina, Valentina Ros, and Antonello Scardicchio. Forward approximation as a mean-field approximation for the anderson and many-body localization transitions. *Physical Review B*, 2016.

[127] CL Baldwin, CR Laumann, A Pal, and A Scardicchio. The many-body localized phase of the quantum random energy model. *Physical Review B*, 2016.

[128] CL Baldwin, CR Laumann, A Pal, and A Scardicchio. Clustering of nonergodic eigenstates in quantum spin glasses. *Physical Review Letters*, 2017.

[129] Antonello Scardicchio and Thimothée Thiery. Perturbation theory approaches to anderson and many-body localization: some lecture notes. *arXiv:1710.01234*, 2017.

[130] Bernard Derrida. Random-energy model: Limit of a family of disordered models. *Physical Review Letters*, 1980.

[131] Thomas Jörg, Florent Krzakala, Jorge Kurchan, Anthony C Maggs, and Justine Pujos. Energy gaps in quantum first-order mean-field–like transitions: The problems that quantum annealing cannot solve. *EPL (Europhysics Letters)*, 2010.

[132] Joao Basso, David Gamarnik, Song Mei, and Leo Zhou. Performance and limitations of the qaoa at constant levels on large sparse hypergraphs and spin glass models. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022.

[133] Benjamin F Schiffer, Dominik S Wild, Nishad Maskara, Madelyn Cain, Mikhail D Lukin, and Rhine Samajdar. Circumventing superexponential runtimes for hard instances of quantum adiabatic optimization. *arXiv:2306.13131*, 2023.

APPENDIX A

ADDITIONAL NUMERICAL RESULTS FOR SPECTRALLY FOLDED QUANTUM OPTIMIZATION

A.1 Convergence to the partial planted solution in Hamming distance

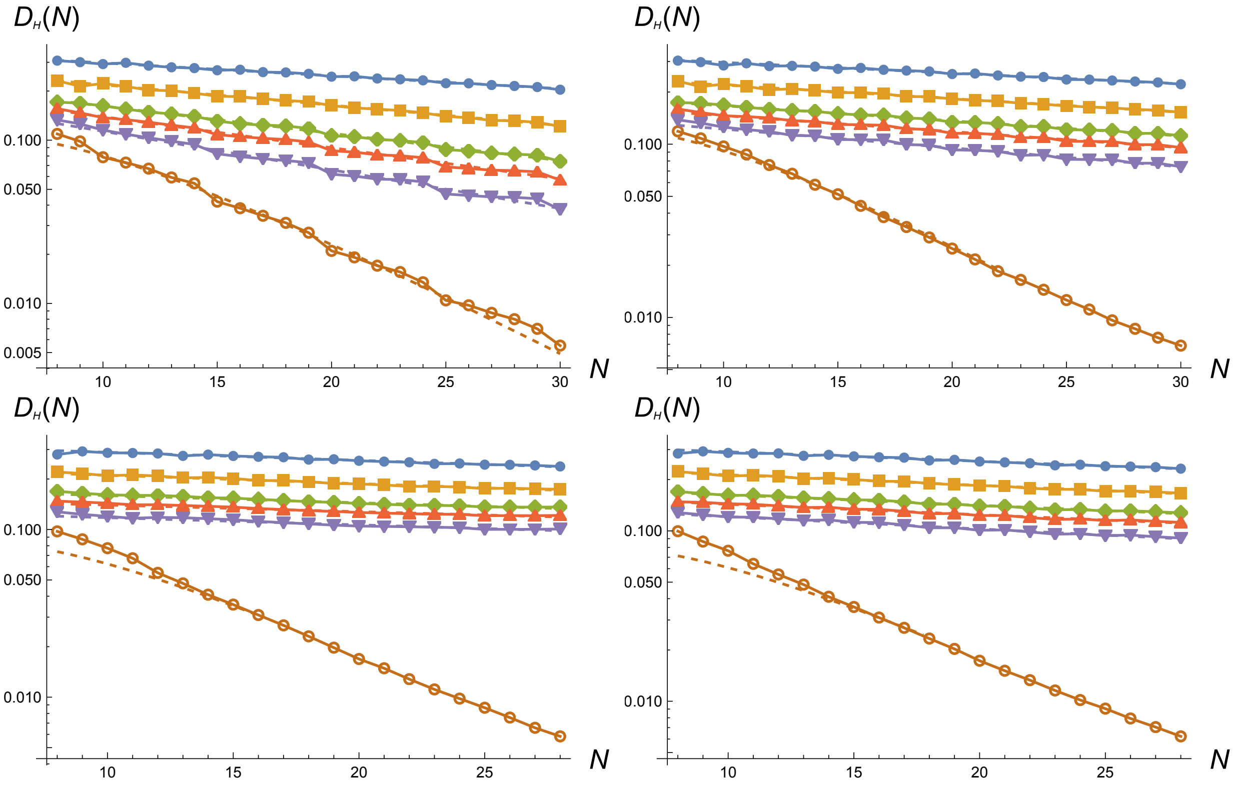
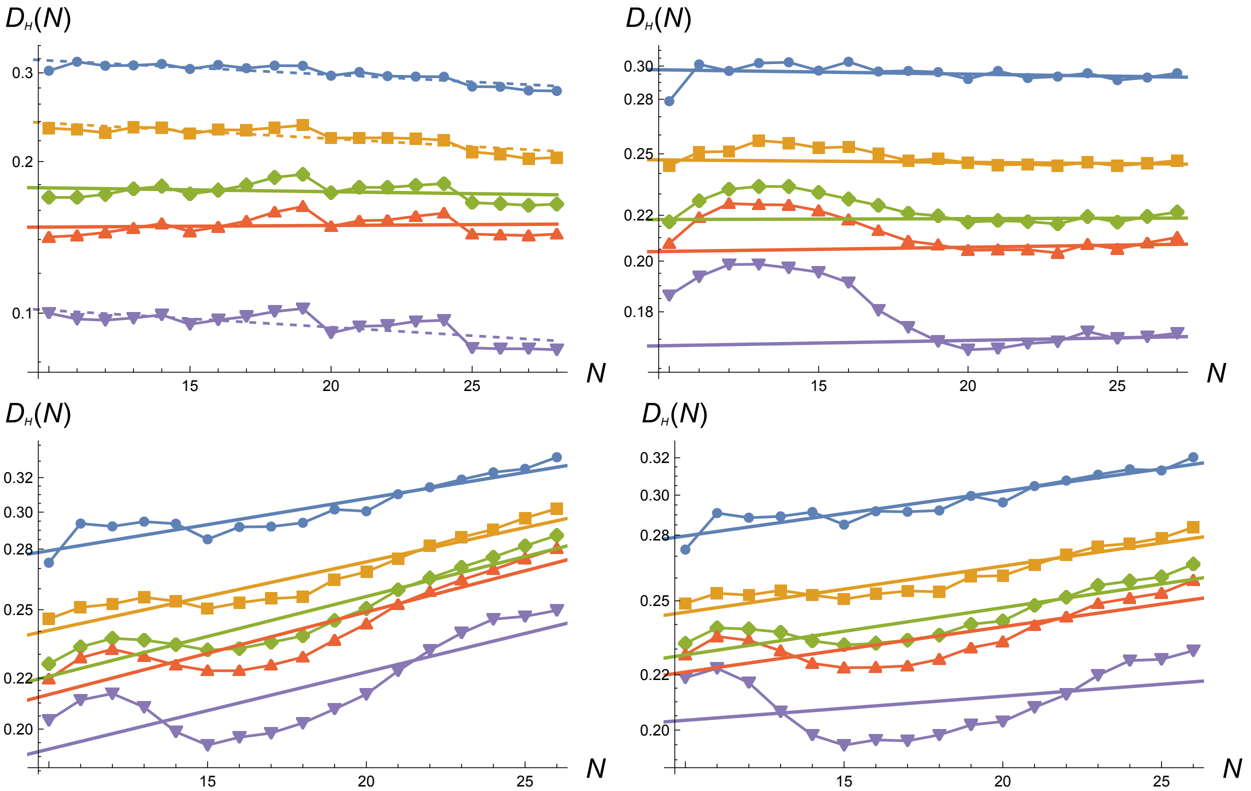


Figure A.1 Probabilities of returning states within Hamming distance $D_H = \{2/5, 1/3, 1/4, 1/5, 1/8, 0\}\, N$ flips from the ground state G for QAOA (top to bottom curves), for $N_C = \left\{2, 4, 6, 3\sqrt{N}/2\right\} N$ (clockwise from top left) and application parameters in text. As seen in the figures, the per-shot probability of finding the ground state is essentially independent of constraint density and tracks the prediction $N2^{-0.28N}$ in Eq. B.7. On the other hand, the probabilities of returning states at various extensive fractional Hamming distances, e.g. $N/4$ or fewer flips, decay much more slowly, and the task of finding states comparatively close to G becomes easier as N_C/N increases. That said, these probabilities all decay exponentially with N in all cases, consistent with the worsening approximation ratios observed.



of returning states within fractional Hamming distance approximations for AQC formulation of SFQO.]Probabilities of returning states within Hamming distance $D_H = \{2/5, 1/3, 1/4, 1/5, 1/8\}\, N$ flips from the ground state G for the AQC formulation of spectral folding (top to bottom curves), for $N_C = \left\{2, 4, 6, 3\sqrt{N}/2\right\} N$ (clockwise from top left) and application parameters in text. For $N_C = 2N$ some probabilities decay slowly with system size, due to competition with other minima (though non-monotonicity makes the fitting somewhat ambiguous here); for all other cases they are constant or increase toward some large N saturation value, indicating that this variation is finding states near the global minimum with constant probability. The probability of finding G is not plotted, as for spectrally folded optimization it's essentially zero.

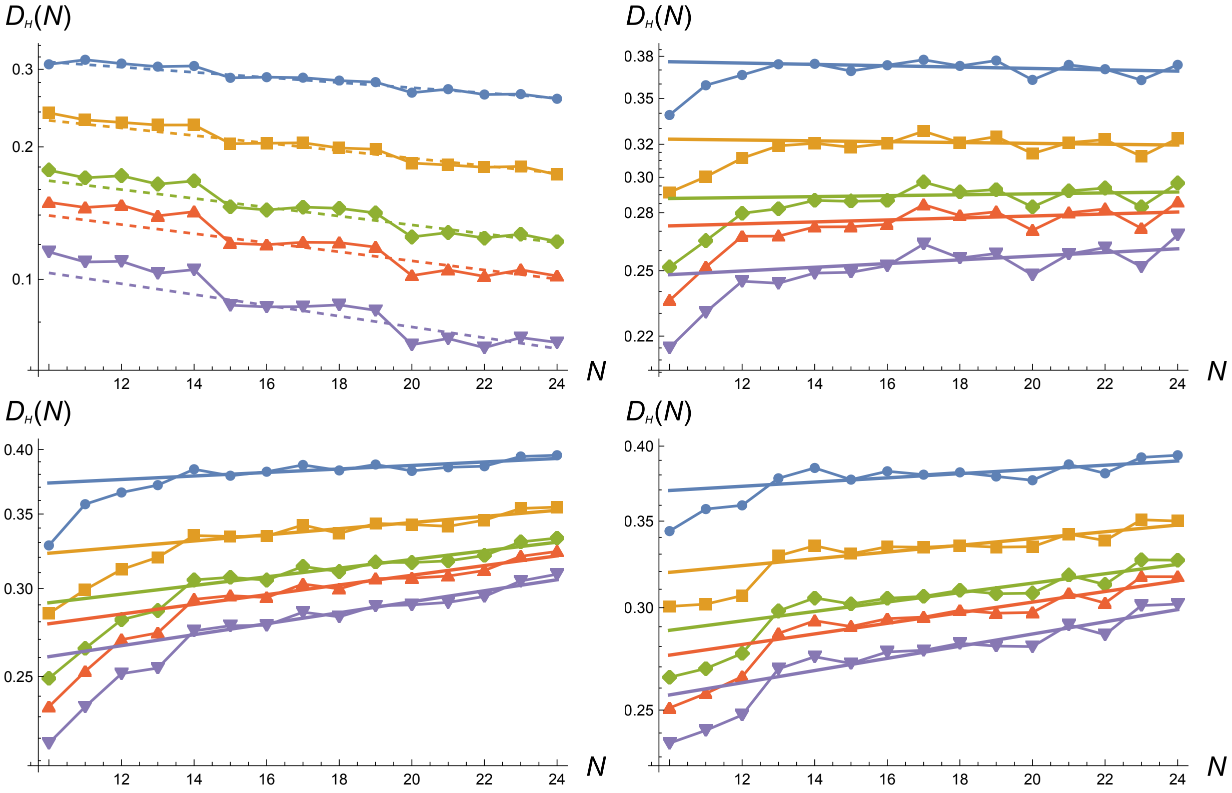


Figure A.2 Probabilities of returning states within Hamming distance $D_H = \{2/5, 1/3, 1/4, 1/5, 1/8\}\, N$ flips from the ground state G for the TMA formulation of spectral folding (top to bottom curves), for $N_C = \left\{2, 4, 6, 3\sqrt{N}/2\right\} N$ (clockwise from top left) and application parameters in text. For $N_C = 2N$ the probabilities decay slowly with system size, due to competition with other minima; for all other cases they are constant or increase toward some large N saturation value, indicating that this variation is finding states near the global minimum with constant probability. The probability of finding G is not plotted, as for spectrally folded optimization it's essentially zero.

APPENDIX B
THEORETICAL PERFORMANCE PREDICTIONS FOR SPECTRALLY FOLDED QUANTUM OPTIMIZATION

B.1 Preliminaries

In this Appendix, we include the analytic performance predictions obtained through an Nth order perturbation theory analysis done by co-authors (E.K. and V.O.). In this section, we will predict from first principles the average macroscopic quantum tunneling rate–and thus, achievable approximation ratio–for the TMA variation of spectrally folded quantum optimization. This version may not be the optimal choice, and there are intuitive reasons to believe that other variations could offer better performance, but being able to make direct analytical predictions enormously strengthens our argument and bolsters the scaling expectations one can infer from our numerical results. To do so, we have developed a somewhat novel resummed extensive order perturbation theory based on previous *forward approximation* results [31, 106, 125–129].

For the random hypergraph problems we study here, the two key factors in determining the macroscopic quantum tunneling rate are the transverse field strength κ_c where a phase transition occurs between the paramagnet and the quantum spin glass, which we call the dressed problem phase (DPP) in this work, and the energy cost $E\left(x\right)$ for x random flips away from the ground state. We first derive the energy cost, and remarkably, for random flip sequences it turns out to be graph independent [92]. Specifically, for MAX-3-XORSAT, our problem is defined as a hypergraph of N_C p-body constraints (e.g. $V_{ijk}Z_iZ_jZ_k$) over N variables, where $p = 3$ here, and each constraint returns ±1 and flips to the opposite value when any one of the spins flips. Let us say the system is in some classical configuration s; the energy is then given by $E\left(s\right) = N_C\left(n_{\text{unsat}} - n_{\text{sat}}\right)$, where a sat constraint returns 1 in this notation, and n implies a density.

Now we flip one spin at random. Each spin participates in, on average, pN_C/N constraints, and consequently, the average energy change for a single spin flip is

$$\Delta E_{\text{avg}} = +2p\frac{N_C}{N}\left(N_{\text{sat}} - N_{\text{unsat}}\right) = -2pE \tag{B.1}$$

Now imagine we have flipped y spins from our initial configuration. If we flip one more spin at random, once again $\Delta E_{\text{avg}}\left(y\right) = -2pE\left(y\right)\Delta y$. However, we have already flipped y spins, so when we flip one more at random, with probability $1 - y/N$ we have flipped a spin back and are computing the energy change associated with reducing y by 1. Consequently

$$\left(1-\frac{2y}{N}\right)\frac{\Delta E_{\text{avg}}\left(y\right)}{\Delta y}=-2pE\left(y\right) \tag{B.2}$$

If we interpret this as a differential equation, it has a straightforward solution: starting from the ground state G, for x unique random flips away the average energy is

$$E_{\text{avg}}\left(x\right)=E_{\text{GS}}\left(1-\frac{2x}{N}\right)^{p}. \tag{B.3}$$

Note that this statement is graph independent, and is only an average; individual trajectories will of course display substantial variations. It is a rederivation of a familiar result for dense hypergraphs with Gaussian distributed constraint energies [106, 130], but is applicable in much broader contexts and is easy to confirm numerically. Throughout this work, we rescale all problems by a multiplicative constant so that the ground state energy is $-N$.

B.2 Paramagnet to spin glass transition scaling for MAX-3-XORSAT

Before proceeding to our main calculation, we note that, from this result, we can predict the typical case difficulty scaling of finding the ground state with AQC or QAOA, for random hypergraphs as defined in section 3.3. To predict the phase transition rate, we will use high order perturbation theory to compute the perturbative dressings to the problem ground state; call this state $|G_D\rangle$. Using fourth order perturbation theory, for uniform transverse field strength κ, the energy of the dressed ground state is, again on average[8]

$$E_{\text{GS}}\simeq-N\left(1+\frac{\kappa^{2}}{2p}+\frac{\kappa^{4}}{8p^{3}}+...\right). \tag{B.4}$$

This crosses the energy of the paramagnetic state, which is $-N\kappa$ and minimally perturbed by the problem Hamiltonian, at $\kappa_c\simeq 1.29$ for our family of $p=3$ PPSPs. The splitting $2\Omega_0$ at the phase transition is expected to be proportional to the overlap of the uniform superposition state $|S\rangle$, the ground state of H_D, with the dressed state $|G_D\rangle$ [92]:

$$\Omega_0\propto\langle S|G_D\rangle. \tag{B.5}$$

To compute Ω_0, we must thus compute the perturbative corrections to $|G_D\rangle$, to high orders. Again using Eq. (B.3), if we let $\left|G^{(i,j,k...)}\right\rangle\equiv X_iX_jX_k...\left|G\right\rangle$, and $\tilde{E}_{\text{avg}}\left(k\right)\equiv E_{\text{avg}}\left(k\right)-E_G$, at the transition point we

[8] In this step the graph structure of H_P can potentially be important. This is because the energy denominator $2p$, the average cost per flip just a few flips away from G, is more sensitive to the details of the graph than the energy cost many flips away, which can in turn shift κ_c. We thus do not claim this result is valid for all graphs. However, we find that for our random PPSPs the transition is consistently near the κ_c we predict here and the scaling of the uniform field result closely matches our prediction.

have

$$\begin{aligned} \left|G_D\right\rangle \simeq \left|G\right\rangle + \tfrac{\kappa_c}{\tilde{E}_{\text{avg}}(1)} \textstyle\sum_j \left|G^{(j)}\right\rangle \\ +2!\tfrac{\kappa_c^2}{\tilde{E}_{\text{avg}}(1)\tilde{E}_{\text{avg}}(2)} \textstyle\sum_{i\neq j} \left|G^{(i,j)}\right\rangle \\ +3!\kappa_c^3 \textstyle\prod_{m=1}^{3} \tfrac{1}{\tilde{E}_{\text{avg}}(m)} \textstyle\sum_{i\neq j\neq k} \left|G^{(i,j,k)}\right\rangle + ... \end{aligned} \tag{B.6}$$

The factorials come from the combinatorics of ordering the m spin flips to reach each term. Now, since all states are present in $\left|S\right\rangle$ with equal amplitude $2^{-N/2}$ and all terms in $\left|G_D\right\rangle$ are positive definite, we can immediately conclude

$$\Omega_0\left(N\right) \simeq 2^{-N/2}\left(1 + \sum_{m=1}^{N} \kappa_c^m \binom{N}{m} m! \prod_{n=1}^{m} \frac{1}{\tilde{E}_{\text{avg}}\left(n\right)}\right). \tag{B.7}$$

For $\kappa_c = 1.29$, this function is well fit by $\Omega_0 = a\sqrt{N}2^{-bN}$, where $b \simeq 0.14$, in decent agreement with the result for a mean-field $p = 3$-spin ferromagnet derived in [131]. We note also the hardness equivalence between dense and random sparse graphs drawn in [132], though we emphasize that the results we derive here are not limited to completely random graphs. If we are in the diabatic regime for $t_f \ll 1/\Omega_0\left(N\right)$, the probability of finding the ground state after a sweep is $P_{\text{GS}}\left(t_f\right) \simeq \Omega_0^2 t_f/W$, where $W \sim O\left(N\right)$ is the energy range swept over in the evolution; for $t_f \propto N$ we thus have $P_{\text{GS}}\left(t_f, N\right) \simeq N2^{-2bN}$. Our numerical simulations for PPSPs, shown below, are in good agreement with this result. Note that this assumes a single avoided crossing and does not consider transverse field chaos, e.g. crossings late in evolution between states close in energy; such transitions can in principle be substantially more difficult [133], but are not relevant to approximation hardness in this problem as argued earlier.

B.3 Scaling of inter-valley tunneling in a $p = 3$ quantum spin glass

This calculation is less easy to generalize to a spectrally folded Hamiltonian, however, where we have exponentially many competing ground states, clustered in a thin hyperspherical shell around the true minimum, at least in approximation-hard problems. So instead we will consider tunneling between two p-spin wells in an N spin system, spaced $N/2$ flips apart, and show that the average per-state tunneling rate in the TMA formulation of spectrally folded optimization should have near-identical scaling. Our total N-spin Hamiltonian, consequently, is

$$\begin{aligned} H = -\frac{1}{N^{p-1}}\left[\left(\sum_j Z_j a_j\right)^p + \left(\sum_j Z_j b_j\right)^p\right] \\ -\kappa\sum_j X_j, \end{aligned} \tag{B.8}$$

where the a_j and b_j are all equal to ±1 and specify the minimum position for each of the two terms. We assume for the remainder of this writeup that a and b correspond to bitstrings $M = N/2$ flips apart; we let these states be $|0\rangle$ and $|1\rangle$. We will also define the bare classical energy, with no corrections from transverse fields. We start from one of the two minima, and consider a state which is $m+n$ random flips away from it. We let m of these flips be ones which move toward the other minimum (e.g. reduce the Hamming distance to it), and the n flips be flips that move away from it. Then the bare energy $E^{(0)}_{m,n}$ is given by:

$$E^{(0)}_{m,n} = -N\left[\left(1-2\frac{m+n}{N}\right)^p + \left(2\frac{m-n}{N}\right)^p\right]. \tag{B.9}$$

We assume that the transverse field strength κ is below κ_c, the p-dependent critical point where a transition to the paramagnetic state occurs. The ground states are thus the symmetric and antisymmetric combinations of the two dressed classical minima, with splitting $2\Omega_0$, where Ω_0 decays exponentially in N and our goal in this section is to predict its decay rate.

Computing Ω_0 proceeds through the following steps:

- We compute the renormalized cost per flip away from either minimum, incorporating transverse field corrections, which we will then use in the energy denominators of our Mth order perturbation theory. This step is analogous to commonly used resummation schemes in diagrammatic quantum field theory, where self-energy corrections are incorporated into the propagators used to compute higher order processes.

- We divide the system between primary spins, which flip between the classical minima, and secondary spins, which do not. We then compute the dressed states $|0_D\rangle$ and $|1_D\rangle$ that comprise all the primary flip sequences up to order $M/2$ away from each minimum. It is at this order that the two states have nonzero overlap.

- These dressed states are then normalized; incorporating this normalization, their overlap gives the primary spin contribution to the tunneling rate, $\Omega_0^{(p)}$.

- We then compute the secondary spin contributions to tunneling, which take two forms: an increase of the tunneling rate from the constructive contribution of many additional tunneling sequences in which secondary spins participate, and a decrease from normalization corrections and the spread of the classical minima away from the core classical configurations from which the tunneling calculation begins.

- Incorporating both sets of secondary spin contributions gives us a closed form expression for Ω_0 which can then be evaluated numerically and compared to exact diagonalization.

We first want to compute the energy shifts, in second order perturbation theory, to these states. These corrections arise from a single spin being flipped and flipped back, and are opposite in sign to the cost of the local flip. Let $u_{m,n}$ be the difference between energies of a state m, n and a ground state, incorporating these corrections. Then

$$u_{m,n} = E^{(0)}_{m,n} + N\left(1+\frac{\kappa^2}{2p}\right) - \left(\frac{N}{2}-2m\right)\frac{\kappa^2}{\partial_m E^{(0)}_{m,n}} \\ - \left(\frac{N}{2}-2n\right)\frac{\kappa^2}{\partial_n E^{(0)}_{m,n}} + O\left(\kappa^4\right). \tag{B.10}$$

Note that $\partial_m E_{m,n}|_{m,n=0} = \partial_n E_{m,n}|_{m,n=0} = 2p$, so $u_{0,0} = 0$. One can observe that if $p = 2$, $\partial_m E_{m,0} = 4\left(1-\frac{4m}{N}\right)$, and the transverse field corrections to state energies are m-independent, so that the energy barrier between the two competing ground states is not renormalized by the transverse field. But for $p = 3$ and higher these corrections are nontrivial and act to reduce the effective energy barrier between the states, increasing the tunneling matrix element. This process is effectively a resummation of higher order corrections and is necessary to obtain quantitatively accurate results.

We start by computing the dressed states, summing over primary spin corrections only. They take the form

$$|0_D\rangle \equiv |0\rangle + \sum_j \frac{\kappa}{u_{1,0}} X_j |0\rangle + \sum_{j,k(j\neq k)} \frac{2\kappa^2}{u_{1,0}u_{2,0}} X_j X_k |0\rangle \\ + \sum_{j,k,l(j\neq k\neq l)} \frac{3!\kappa^3}{u_{1,0}u_{2,0}u_{3,0}} X_j X_k X_l |0\rangle + ... \tag{B.11}$$

The expression for $|1_D\rangle$ is identical. We stop our expansion at order $M/2$, which is the lowest nontrivial order needed to connect the states. For simplicity we assume M is even though the argument is easy to generalize to odd M as well. Note that this state is not normalized, and in fact the norm of the state written above is exponentially large, so we will need to incorporate normalization corrections into the definition of the states. Thanks to the dressing of the states, we obtain a primary-spin energy splitting

$$\frac{1}{2}\left(\langle 0_D| + \langle 1_D|\right) H_P \left(|0_D\rangle + |1_D\rangle\right) = 2\Omega^{(p)}_0, \\ \frac{1}{2}\left(\langle 0_D| - \langle 1_D|\right) H_P \left(|0_D\rangle - |1_D\rangle\right) = 0. \tag{B.12}$$

Evaluating these expressions, the degeneracy splitting from only considering primary spins is:

$$\Omega_0^{(p)} = \kappa^M \binom{M}{M/2} \frac{1}{u_{M/2,0}} \left(\frac{M}{2}! \prod_{k=1}^{M/2-1} \frac{1}{u_{k,0}} \right)^2 \times \mathcal{N}^{(p)},$$

$$\mathcal{N}^{(p)} = \left(1 + \sum_{k=1}^{M/2} \binom{M}{k} \left(\kappa^k k! \prod_{j=1}^{k} \frac{1}{u_{j,0}} \right)^2 \right)^{-1}. \tag{B.13}$$

Here, $\mathcal{N}_p$ is the normalization correction. This covers the primary spin portion of the macroscopic quantum tunneling rate.

We now turn to the secondary spins. To introduce secondary spin corrections, consider a single secondary spin j out of the $(N-M) \sim N/2$ total, whose bit value is the same in both classical minima. Since the same transverse field is acting on it as all other spins, when we consider the sum of all processes that connect the two minima, we can now divide them between those where $M/2$ primary spins flip from each minima to meet in the middle, with secondary spin j unchanged, and a new set of processes where spin j flips starting from each minimum and the two wavefunctions overlap at the set of states where $M/2$ primary spins have flipped along with j. The first set of processes is what was considered in Eq. B.13; the second is new, and we want to calculate its matrix element. It is most useful to express these matrix elements as a ratio of the new term to the original, primary-spin-only process, since both decay exponentially in M. Let the primary spin perturbative matrix element to reach $M/2$ flips be $\xi_{M/2}$, so that

$$\xi_{M/2} = \kappa^{M/2} \left(\frac{M}{2}! \prod_{k=1}^{M/2} \frac{1}{u_{k,0}} \right). \tag{B.14}$$

To define the analogous process where j flips, we need to sum over all the points during the perturbative sequence when that can happen. We thus have:

$$\xi_{M/2}^s = \kappa^{M/2+1} \frac{M}{2}! \sum_{n=1}^{M/2} \prod_{k=1}^{n} \frac{1}{u_{k,0}} \prod_{k=n}^{M/2} \frac{1}{u_{k,1}}. \tag{B.15}$$

And noting that we have to make this insertion in the matrix elements from both minima, the total tunneling term is increased by

$$\Omega_0^{(p)} \to \left(1 + \left(\frac{\xi_{M/2}^s}{\xi_{M/2}} \right)^2 \right) \Omega_0^{(p)}, \tag{B.16}$$

$$\left(1 + \left(\frac{\xi_{M/2}^s}{\xi_{M/2}} \right)^2 \right) \equiv \gamma_T. \tag{B.17}$$

We now need to consider the rest of the secondary spins. Formally of course, the additional energy cost of each secondary spin flip changes as more secondary spins flip, but since these corrections are fairly weak (though they are appreciable and necessary for an accurate prediction of the scaling exponent) the total tunneling rate is going to be dominated by the set of processes where a comparatively small fraction of secondary spins have flipped, and we can thus approximate them as independent contributions. In this limit, since there are $N-M$ secondary spins,

$$\Omega_0^{(p)} \to \Omega_0^{(p)} \gamma_T^{N-M}. \tag{B.18}$$

Alongside this, the secondary spin corrections also spread the competing ground state wavefunctions out over Hilbert space, which exponentially reduces the weight of the core classical configurations from which the tunneling calculation begins. To be consistent with the independence approximation made above, we simply compute all the corrections to the ground state from each secondary spin independently and multiply them. Noting that we must apply this calculation to both competing ground states, this reduces the tunneling rate by

$$\Omega_0^{(p)} \to \Omega_0^{(p)} \left(\frac{\gamma_T}{\gamma_R}\right)^{N-M}, \quad \gamma_R \equiv 1 + \left(\frac{\kappa}{u_{0,1}}\right)^2. \tag{B.19}$$

Note that, if we set the cost per flip of a given secondary spin to some constant U, independent of the configuration of the other spins, that would imply it is disconnected from the primary spins as there are no couplings to shift the energy. In this limit a direct evaluation of the two functions shows that $\gamma_T = \gamma_R$ (for any choice of U) and this now disconnected spin plays no role in tunneling at all. This factorization of disconnected spins is reassuring, and lends support to the correctness of this approach. Taking into account all these effects, our total tunneling rate is

$$\Omega_0 = \kappa^M \binom{M}{M/2} \frac{1}{u_{M/2,0}} \left(\frac{M}{2}! \prod_{k=1}^{M/2-1} \frac{1}{u_{k,0}}\right)^2 \times \mathcal{N}^{(p)} \times \left(\frac{\gamma_T}{\gamma_R}\right)^{N-M}. \tag{B.20}$$

Taking all of these effects into account yields a highly accurate prediction of the minimum gap scaling for a wide range of values for p and κ, with only an $O(1)$ discrepancy in the prefactor and few percent discrepancies in the scaling exponent (empirically, Eq. B.20 tends to slightly overestimate the decay compared to the exponent extracted from numerical diagonalization).

B.4 Achievable approximation ratio with spectrally folded trial minimum annealing

With this result in hand, we will now predict the macroscopic quantum tunneling rate–and thus, achievable approximation ratio–for the spectrally folded Hamiltonian. From this, we can calculate our target value of A, assuming that there is a single deep minimum far below the energy of any local minima. Relaxing this assumption will improve the performance of the algorithm by virtue of there being many more target states, as confirmed in our simulations for lower constraint densities. We consider the protocol in section 3.4, with an initial state $|L\rangle$. We assume that the process of ramping the transverse field up and down is itself at least roughly adiabatic, i.e., we can assume approximate spectral continuity with respect to the folding and lowering Hamiltonians, noting that the lowering Hamiltonian will itself create $O\left(\sqrt{N}\right)$ shifts to the energies of states near the fold. It follows from our assumption that the ramping process itself does not meaningfully heat the system. We then consider the set $\mathcal{T}$ of all states within $O(1)$ shifts of $-AE_{\text{GS}}$ in H_P, the states closest to the fold, and compute, as a function of all our various algorithm parameters, the total probability of tunneling into any one of them.

Since the tunneling rate into any individual state is exponentially small, and the time over which we slowly turn off the lowering Hamiltonian is $T \propto O(N)$, we can assume that tunneling will be diabatic with respect to any individual state. A Fermi's Golden rule analysis as in [111] suggests that

$$P_{\text{tot}} \propto \frac{T}{W} \sum_{j \in \mathcal{T}} \Omega_{0,Lj}^2, \tag{B.21}$$

where $W \sim O(N)$ is the energy range swept over by reducing $C(t)$ to 0, and $\Omega_{0,Lj}$ is the tunnel splitting at degeneracy between $|L_D\rangle$ and the target state $|j_D\rangle$, which we assume are an average of $\sim N/2$ flips apart. To go further, we need to compute the average value of $\Omega_{0,Lj}^2$, noting that while of course there will be substantial state-to-state variations, given that there are exponentially many states in $\mathcal{T}$ the average value is going to dominate Eq. (B.21). As in the previous calculation, the most important quantity here is the average cost per flip away from the typical ground state in the folded Hamiltonian, which remarkably turns out to be A-independent.

To see this, we start from the average cost per flip away from $|G\rangle$, given by Eq. (B.3), and note that we can invert that equation to find the mean number of flips $x_A N$ for which $\langle E(x_A N)\rangle = -AN$. To be specific,

$$x_A = \frac{1 - A^{1/3}}{2}. \tag{B.22}$$

We can therefore assume that the typical state in $\mathcal{T}$ is $x_A N$ flips away from $|G\rangle$. If we consider the sequences of primary spin flips connecting $|j_D\rangle$ and $|L_D\rangle$, the typical flip sequence starts $x_A N$ flips away from $|G\rangle$, and notice that with probability $1 - x_A$ an additional random flip towards $|L\rangle$ will also move closer to $|G\rangle$. Taking all these effects and the division by A into account, so that the bare unperturbed energy of both $|j\rangle$ and $|L\rangle$ when they cross are both $\sim -N$, a bit of algebra shows that for y flips away from a ground state of the folded Hamiltonian, not only is the total average cost $\Delta E\left(y\right)$ A-independent, it is precisely equal to the cost given by Eq. (B.3).

This is again only an average, but noting that since it appears the denominators of equations like (B.20), variations about it are more likely to increase the tunneling rate than decrease it. And likewise, since H_L is a random 3-XORSAT problem itself the mean cost per flip away from $|L\rangle$ is going to be given by Eq. B.3 as well, so Eqns. B.8 through B.20 can faithfully predict the average tunneling rate between $|L\rangle$ and a randomly chosen ground state of the folded Hamiltonian.[9]

Of course, this rate decays exponentially; assuming the two states are $M \sim N/2$ flips away for $\kappa = 1.29$, $\Omega_0 \propto 2^{-bN}$ where $b \simeq 0.2$. But this is balanced by the fact that there are on the order of $\binom{N}{x_A N}$ target states. We can further note that out of these states, while the mean distance to $|L\rangle$ is $M \sim N/2$, ones which are k flips closer have tunneling rates which are larger by a factor of 2^{2bk} on average, and though those states are proportionally rare their increased weight is enough to meaningfully impact our choice of A. Since our total runtime is linear, simple diabatic scaling predicts that the probability of tunneling into the typical state $M - k$ flips away is proportional to Ω_0^2, e.g. $2^{-4b(M-k)}$.

We now take this result and plug it into Eq. (B.21), so that we can determine the choice of A where the returned P_{tot} provides an approximation guarantee. If we use Stirling's approximation to write the binomial coefficients as exponentials, and ignore slowly varying polynomial factors, the total number of states in $\mathcal{T}$ scales as:

$$N_{\mathcal{T}} \propto \exp\left(-\left[x_A \ln x_A + \left(1 - x_A\right)\ln\left(1 - x_A\right)\right] N\right). \tag{B.23}$$

Likewise, if the average probability of tunneling into a target state k primary spin flips closer to $|L\rangle$ is increased by a factor of at least 2^{4bk}, the weighted per-state average of the diabatic tunneling rate into states in $\mathcal{T}$ can be approximated as

$$\frac{\log\left\langle \Omega_0^2 \right\rangle}{-N} \approx 2b\ln 2 - x_A\left(\ln\left(1 + 2^{4b}\right) - \ln 2 - 2b\ln 2\right). \tag{B.24}$$

[9] We expect that using the average cost per flip in Eq. B.9 will if anything underestimate the per-state tunneling rate in real disordered problems. This is because all of these energy costs appear in denominators, which leads to likely small asymmetries in how much the deviations from the average in any individual flip sequence contribute to the total matrix element, giving lower energy sequences proportionally higher weight. We do not really expect this effect to be significant but rather highlight it as another point where our prediction is conservative by design.

Note that this average comes from considering only $x_A N$ flips away from $|G\rangle$ but varying Hamming distance from $|L\rangle$ and thus neglects the influence of comparatively rarer states larger distances from $|G\rangle$. Taking all these terms into account, the probability of returning a state with $E \simeq AE_{\mathrm{GS}}$ as measured relative to the original H_P becomes constant, or at least stops decaying exponentially, when

$$-2b\ln 2 - \left[x_A \ln x_A + (1-x_A)\ln(1-x_A)\right] + x_A\left(-\ln 2 - 2b\ln 2 + \ln\left(1+2^{4b}\right)\right) = 0. \tag{B.25}$$

The achievable approximation ratio is thus determined by the per-state decay exponent b, computed in section B.3 as a function of N and κ by fitting Eq. (B.20) to $\Omega_0(N) \propto \sqrt{N}2^{-bN}$, and then choosing x_A using Eq. (B.22) to solve Eq. (B.25). This analysis only counts states within $O(1)$ shifts of AE_{GS} (recall that $E_{\mathrm{GS}} = -N$ in our normalization) and ignores low-order polynomial prefactors; for $b = 0.2$, which again depends on $\kappa = 1.29$ in this calculation, this is solved when $x_A \simeq 0.08$, or $A \simeq 0.59$.

This means that if the true ground state satisfies a constraint fraction F beyond random guessing—e.g. a total fraction $1/2 + F$, so F is at most $1/2$ here—our algorithm will return states which satisfy a fraction $1/2 + AF$ with high probability. If we choose A to be too large compared to the target value established by expressions like Eq. (B.25), we risk failing to well-approximate the problem; conversely, choosing A below it will reduce the returned approximation ratio to A and thus perform suboptimally. And, we emphasize again, this prediction assumes a random, potentially dense hypergraph but is fundamentally independent of the fraction satisfied in E_{GS} itself and so applies to the planted partial solution instances we use for numerical benchmarking below.

B.5 Further Comments and Caveats

We expect that this analysis underestimates the choice of A that will return states with $E \leq AE_{\mathrm{GS}}$ with constant probability. This is because our counting here only counts states very close to the fold, when in reality the probability of tunneling into states a small extensive fraction larger than AE_{GS} is still going to be appreciable due to the continued exponential growth of the number of targets, even if the per-state tunneling rate does tend to decrease with increasing E due to the interference effect mentioned earlier, in which perturbative corrections that mix with states of lower energy have opposite sign. In addition to this consideration, because the target ground states and low lying excitations of the folded Hamiltonian in $\mathcal{T}$ very roughly form a hyperspherical shell, any individual target state will have other states in $\mathcal{T}$ that are relatively close to it in Hamming distance. Consequently we expect these states to have a band dispersion, centered around the mean energy given by the corrections in Eq. (B.10). Since we are already assuming off-resonant tunneling, i.e. a per-state tunneling rate $\propto \Omega_0^2$, and so summing over squared matrix elements, if we consider states near the band center where the density is highest this will alter the average tunneling rate by at most a prefactor. However, there are good reasons to suspect that tunneling into extremal states near the bottom of the band can be substantially enhanced, enough to increase the optimal value of A. This calculation is difficult to do quantitatively so we do not attempt it here; we instead see the approximation of considering only states near the band center of $\mathcal{T}$ as another choice that likely underestimates the achievable approximation ratio.

We also want to emphasize that this variation is not necessarily the optimal spectrally folded optimization algorithm, but instead merely the one where we were able to analytically compute the threshold A. For example, one can perform trial minimum annealing with a simple local Z bias lowering Hamiltonian (e.g. $H_L = \sum_j h_j Z_j$), or standard AQC interpolation using the quadratic folding procedure in Eq. (3.5) as the cost function. The linear lowering Hamiltonian is expected to have equal or better tunneling rates to a 3-XORSAT-based minimum as the overall cost-per-flip curve is shallower, though the local energy shifts to the ground states of H_{fold} from H_L are expected to be larger. The total gate count at each time step is lower. Empirical performance in testing up through $N = 25$ showed fairly similar performance to 3XOR-based H_L for all other parameters equal, but with more significant non-monotonic behaviors that made fitting difficult; see Section 3.5.3 for details. That the two schemes could asymptotically converge to the same achievable approximation ratios seems plausible to us but we cannot simulate large enough systems to be sure.

For quadratic folding AQC as in Eq. (3.6), if we choose $A = 1$ the gap is efficiently computable using the methods in [131] and decays as $\Omega_0 \sim 2^{-0.16N}$. Given that, like linear folding, the cost per flip curve is A-independent, if we assume that the tunneling rate per state for $A < 1$ is basically equal to this, then the total decay exponent vanishes if $x_A \simeq 0.06$ and $A \simeq 0.68$ using the arguments of the previous few paragraphs. We do not think that can be simply assumed as easily as with tunneling between semiclassical minima and a linearly folded problem Hamiltonian, in the DPP, and more theoretical work is needed here to analytically determine the optimal choice of A. Interestingly however, our simulation data in Section 3.5.3 supports this conclusion, with a worst case polynomial time approximation ratio of 0.7 found in our simulations. These simulations show that this method performs similarly to, or slightly worse than, the 3XORSAT-TMA algorithm, which is better able to outperform the approximation guarantee of ~ 0.6 derived here.[10] We also expect that the average tunneling rate–and thus, achievable approximation ratio–can likely be further increased by using other, potentially many-frequency, AC methods such as RFQA [31, 32, 82, 111, 123]. For simplicity, we do not incorporate these methods in this work, but they

[10] For smaller systems we also tested linear folding AQC and quadratic folding TMA. In very preliminary studies we found that quadratic AQC modestly outperformed linear AQC, and linear TMA more significantly outperformed quadratic TMA. So we chose not to pursue those methods for larger simulations and do not present those results here, but they may be viable or even superior for other problem classes.

could be a novel way to further improve the performance of this algorithm and are worth exploring in future research.

In summary, through a relatively novel resummed extensive order perturbation theory, we have shown that random hypergraph MAX-3-XORSAT instances, including extremal ones with planted partial solutions, are efficiently approximable to a fairly large constant fraction through the spectrally folded quantum optimization algorithms. We do not expect this to be the case for QAOA, for the reasons discussed earlier in Section 3.3 and supported by the numerical evidence we present below. We similarly do not expect such guarantees to be possible for directly finding global optima, for the reasons set forth in the introduction. Evidently, one cannot so easily summit a mountain in hyperspace, but one can reach the rim of a crater. We now present a series of numerical simulations to further support these claims.

APPENDIX C

ADDITIONAL NUMERICAL RESULTS FOR IST-SAT WITH VARIOUS ALGORITHM SET-UPS

C.1 Identifying empirical radii of convergence for IST-SAT

In this section, we report the results obtained for IST-SAT with lower frequency of $\omega = 2\pi \times 5 \ln N$ and shorter run-time that was simulated to larger system size ($N = 30$). While this parameter set under performs higher frequency drives and with longer run-times as presented in the main text, the empirical radii of convergence we identify do change significantly. One may consider a finer-grid search to identify how exactly r_c scales with the constraint density N_C/N and fraction ϵ of unsatisfied constraints in the ground state.

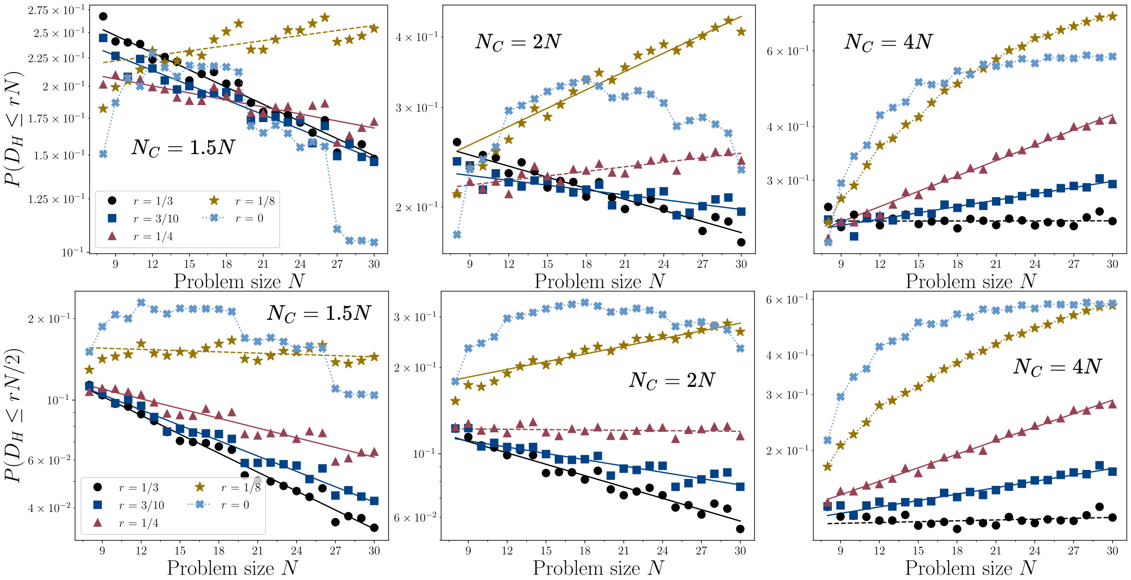


Figure C.1 Convergence of IST-SAT. Assuming we guess phases correctly with probability $(1 - r)$, in the top (bottom) row we plot the probability of returning strings within $D_H \leq rN$ ($D_H \leq r_c N/2$) flips away from G. A radius of convergence r_c can be readily identified in these plots by dashed lines, which are approximately the first fraction r to obtain constant probability with increasing N.

Shown in Figure C.1, we observe monotonically increasing probabilities of returning solutions that approximate the PPS at different approximation ratios rN in terms of Hamming distance D_H. We identify an approximate radius of convergence r_c for each constraint density $N_C/N = \{1.5, 2, 4\}$ respectively as $r_c = \{1/8, 1/4, 1/3\}$.

Table C.1 In this table, we report the exponent b obtained from fitting the function $a2^{bN}$ to the probabilities for constraint densities of $N_C = \{1.5, 2, 4\}N$. In (a), we report the fitting results to $P(D_H \leq rN)$, and in (b), we report the parameters fit to $P(D_H \leq rN/2)$ of finding states close in Hamming distance D_H for different distances defined by a fraction r of $\{N, N/2\}$ or fewer flips from the ground state. In both (a) and (b), initial phase lists were chosen from seed states $(1-r)N$ close in D_H. The exponential fit parameters reported here may be used to infer a time to solution (TTS) given an initial approximation $D_H = (1-r)N$ for various fractions of r below.

Table C.2 Fractional Hamming distance approximation $P(D_H \leq rN)$.

Guessing error r	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
1/3	3.4e-2	2.2e-2	**-8.3e-5**
3/10	3.0e-2	9.5e-4	-1.6e-2
1/4	1.4e-2	**-8.9e-3**	-3.9e-2
1/8	**-1.0e-2**	-3.6e-2	–

Table C.3 Fractional Hamming distance approximation $P(D_H \leq rN/2)$.

Guessing error r	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
1/3	7.8e-2	4.4e-2	**-3.0e-2**
3/10	6.3e-2	2.4e-2	-2.2e-2
1/4	4.0e-2	**1.3e-3**	-4.7e-2
1/8	**5.1e-3**	-3.0e-2	–

C.2 Performance of IST-SAT to any ground state

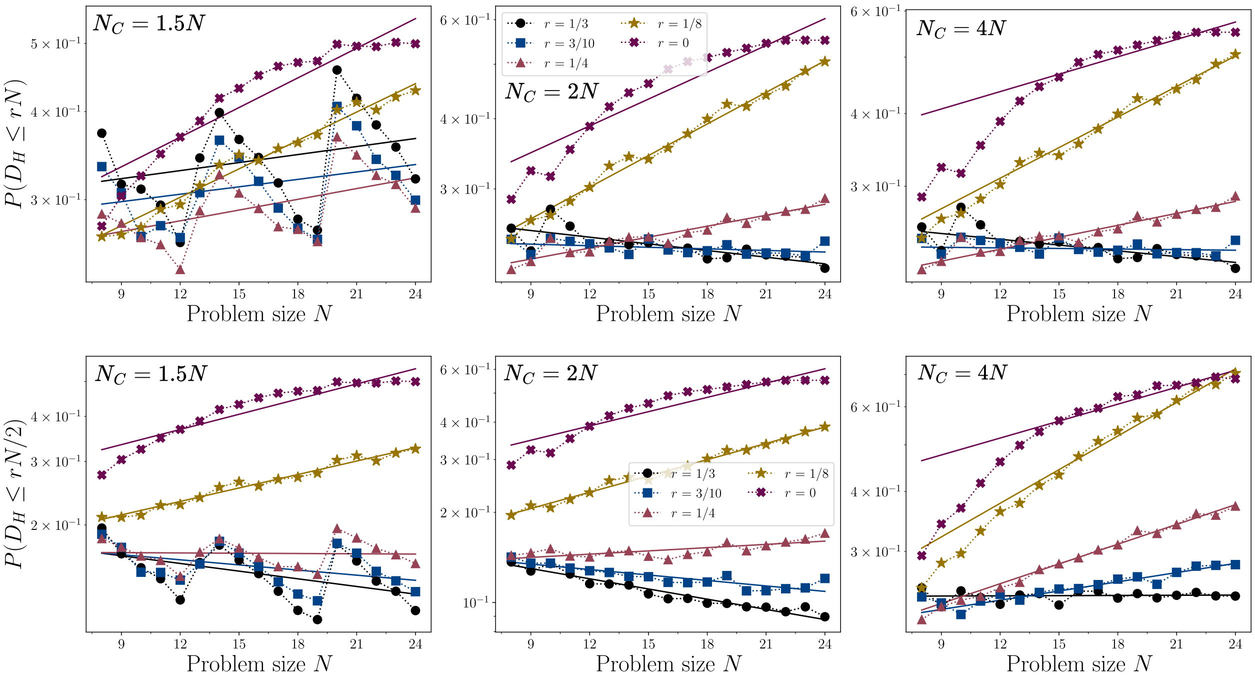


Figure C.2 Convergence of IST-SAT to the any ground state in the degenerate energy manifold. Using seed states which approximate the planted solution by some fraction $r \in \{1/8, 1/4, 3/10, 1/3\}N$ in Hamming distance (or the exact planted solution in the $r = 0$ case), we report the probabilities of approximating any ground state (top row) $P(D_H \leq rN)$, and (bottom row) $P(D_H \leq rN/2)$.

As mentioned in the main text, we include the performance of IST-SAT to any ground state per problem instance (averaged over 1000 random problem instances) for each constraint density. As shown in Figure C.2, the results for $N_C/N = 1.5$ shows that the empirical radius of converge we observe for reaching the planted ground state, both exactly and approximately in Hamming distance, is distinct from reaching *any* ground state. We report the exponents from lines of best fit below, which demonstrate that finding any ground state is often easier than finding the planted solution, in particular. Hence, even when IST-SAT is seeded with a bitstring that approximates the planted solution, the probability of reaching any ground state remains robust. This result is supported by the statistics provided in Figure C.3, which report the average the number of ground states per problem instance, and relative Hamming distance between ground states. For lower constraint densities, the number of ground states is significant, while the planted solution tends to be the unique ground state for $N_C/N = 4$. We identify the constraint density of $N_C/N = 2$ to be an intermediate regime, where size of the degenerate manifold is non-zero, but not so large that the performance of IST-SAT degrades significantly. Thus, these results further characterize the energy landscape of MAX-3-XORSAT, which may be useful in the future development of quantum (approximation) algorithms.

Table C.4 Numerical fit exponents b using the same set-up as the tables above. Here we report the fits from the probabilities reported to any ground state in Figure C.2.

Table C.5 Fractional Hamming distance $P(D_H \leq rN)$ to any ground state.

Guessing error r	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
1/3	1.3e-2	-1.3e-2	-1.1e-2
3/10	1.2e-2	-3.1e-3	-1.1e-3
1/4	1.7e-2	2.2e-2	2.3e-2
1/8	4.5e-2	6.2e-2	5.8e-2
P_{GS}	4.5e-2	5.3e-2	3.3e-2

Table C.6 Fractional Hamming distance $P(D_H \leq rN/2)$ to any ground state.

Guessing error r	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
1/3	-2.4e-2	-3.8e-2	4.5e-4
3/10	-1.6e-2	-2.0e-2	2.1e-2
1/4	-8.7e-4	1.2e-2	4.6e-2
1/8	4.1e-2	6.0e-2	7.7e-2
P_{GS}	4.7e-2	5.3e-2	3.9e-2

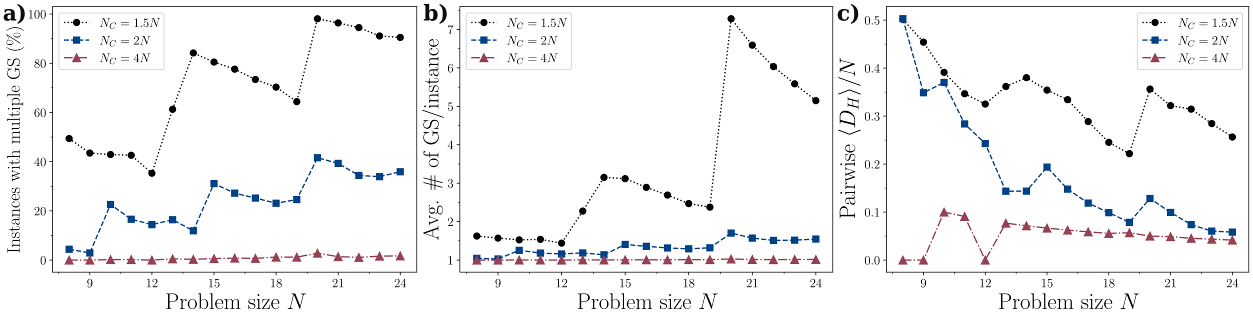


Figure C.3 (a) Percentage of problem instances (from the total 1000 instances) with more than one ground state, including the PPS. For each system size $N = 8 - 24$, we computed exact ground states using brute-force methods. (b) Average number of ground states per problem instance. (c) Pairwise Hamming distance D_H within the set of ground states. The $N = 8, 9, 12$ cases have only one ground state, which is the PPS.

C.3 Approximation performance for seed algorithms on MAX-3-XORSAT

Table C.7 In this table, we report the parameters b obtained from fitting the function $a2^{bN}$ to the probabilities $P(D_H \leq dN)$ of finding states close in Hamming distance D_H for different distances dN or fewer flips away from the ground state. We include the fitting results obtained from both (a) spectral folding and (b) TAQC for the set of constraint densities $N_C = \{1.5, 2, 4\}N$. For the particular case of no distance from the ground state ($d = 0$), we report the parameters for P_{GS} fit to the function $aN2^{bN}$. We denote the fraction d for each constraint density associated to r_c in bold face text. In (a), we note the probability to any degenerate ground state with an asterisk (*), for the particular case of $N_C = 1.5N$. The readers may refer to [120] for more details regarding quadratic SFQO.

Table C.8 Quadratic spectrally folded quantum optimization (SFQO) approximation performance.

Fractional Hamming distance d	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
2/5	*2.8e-3	5.5e-3	1.4e-4
1/3	*-2.5e-3	5.0e-3	**3.4e-3**
1/4	*-6.3e-3	**-5.6e-3**	6.1e-3
1/5	*-7.1e-3	-1.0e-2	9.1-3
1/8	***-1.5e-2**	3.2e-3	1.9e-2
$0(P_{GS})$	–	-4.2e-1	-4.6e-1

Table C.9 TAQC approximation performance.

Fractional Hamming distance d	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
2/5	-3.0e-2	-2.6e-2	-2.0e-2
1/3	-5.0e-2	-4.1e-2	**-2.6e-2**
1/4	-7.0e-2	**-5.5e-2**	-2.9e-2
1/5	-8.4e-2	-6.4e-2	-3.2e-2
1/8	**-1.1e-1**	-8.2e-2	-4.0e-2
$0(P_{GS})$	-3.0e-1	-2.6e-1	-2.8e-1

Table C.10 Greedy classical algorithm approximation performance.

Fractional Hamming distance d	$N_C = 1.5N$	$N_C = 2N$	$N_C = 4N$
2/5	-3.2e-2	-3.1e-2	-2.9e-2
1/3	-5.7e-2	-5.4e-2	**-4.5e-2**
1/4	-9.0e-2	**-8.3e-2**	-5.2e-2
1/5	-1.1e-1	-10.0e-2	-5.3e-2
1/8	**-1.5e-1**	-1.2e-1	-5.5e-2
$0(P_{GS})$	-2.3e-1	-1.9e-1	-7.5e-2

C.4 Performance of TAQC and it's use as a seed algorithm for IST-SAT

In this appendix, we examine the performance of TAQC as a seed algorithm for IST-SAT. In particular we include the results for approximating the ground state in terms of Hamming distance in Figure C.4 and energy in Figure C.5 obtained from quantum simulations in formulation of TAQC.

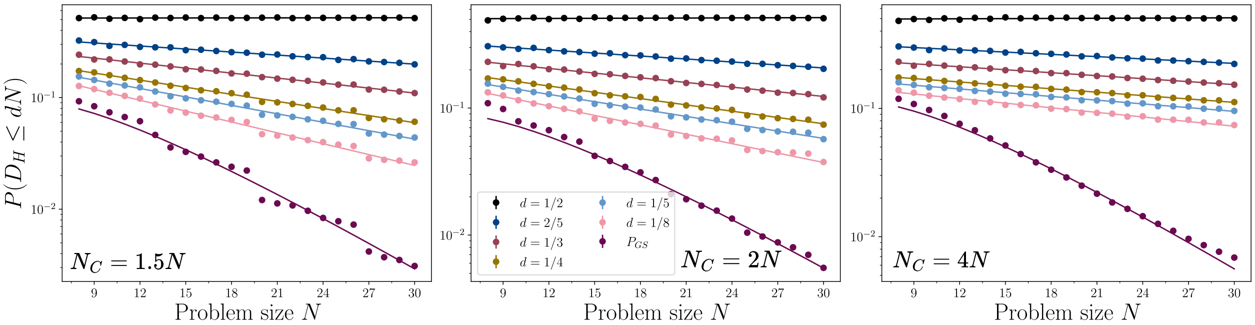


Figure C.4 Probabilities of returning strings within a Hamming distance $D_H \leq dN$ from the ground state G with TAQC, for (left to right) $N_C = \{1.5, 2, 4\}N$ and other parameters listed in the text. Lines are fits to $a2^{bN}$ ($aN2^{bN}$ for the probability of finding G). As discussed in the text, the probability of finding G is roughly independent of N_C/N but the probabilities of finding states close to it are much more sensitive it.

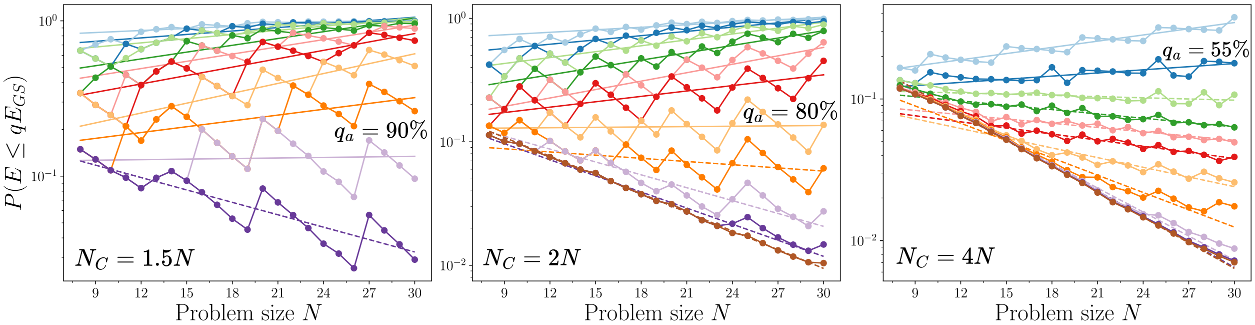


Figure C.5 Probability of TAQC returning states that approximate the ground state energy E_{GS} by a percentage $d = \{50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100(P_{GS})\}\%$. We include constraint densities $N_C = \{1.5N, 2N, 4N\}$ (left to right) denoted in the respective figure panels. Lines of best fit were computed according to a simple exponential function $f(N) = a2^{bN}$, where a is a constant. Exponentially decaying fits are denoted by dashed lines, while fits with a positive exponent are denoted by solid lines. We identify the highest percentages that are not exponentially decaying by $q_a = \{90, 80, 55\}\%$ associated to constraint densities $N_C = 1.5N, 2N, 4N$ respectively.

C.5 Quasi-greedy classical algorithm with warm starts

To further approximate the radius of convergence for our PPSPs we applied a classical greedy descent to random problem instances of constraint densities $1.5N$, $2N$, and $4N$. The algorithm is a modified version of the simple greedy algorithm introduced in [45] and later applied to hypergraphs in [120]. To approximate the radius of convergence, we ran the greedy algorithm on random hypergraphs for the stated constraint densities. The initial states started from different values of $r * N$ flips away from the planted solution to show how well the algorithm performs when beginning at different distances away from the ground state.

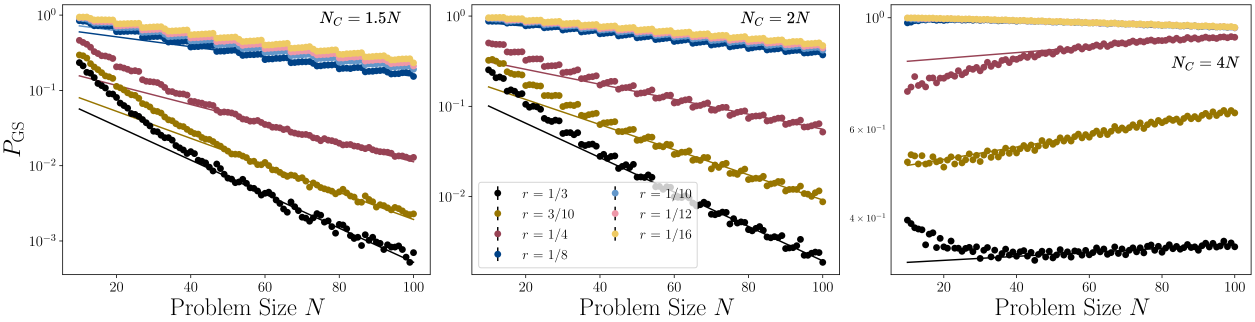


Figure C.6 Probabilities of finding the ground state of random hypergraphs for $N_C = \{1.5N, 2N, 4N\}$ using 100,000 trials on each constraint density and each value of rN. Also plotted are line fits to $a2^{bN}$. The plot demonstrates the increase in improvement as the classical algorithm is seeded with bitstrings closer in hamming distance to the ground states.