



ARTICLE

Analysis and Experimental Demonstration of Amplitude Amplification for Combinatorial Optimization

Daniel Koch^{1,*}, Brian Pardo² and Kip Nieman²

¹Air Force Research Lab, Information Directorate, Rome, NY, USA

²National Academy of Sciences, Washington, DC, USA

*Corresponding Author: Daniel Koch. Email: daniel.koch.13@us.af.mil

Received: 20 January 2026; Accepted: 13 May 2026; Published: 26 June 2026

ABSTRACT: Quantum Amplitude Amplification (QAA), the generalization of Grover's algorithm, is capable of yielding optimal solutions to combinatorial optimization problems with high probabilities. In this work we extend the conventional 2-dimensional mathematical representation of Grover's (marked and non-marked orthogonal collective states) to oracle operators which encode cost functions, such as those shown in previous studies with QUBO (Quadratic Unconstrained Binary Optimization). We show that unconstrained linear cost functions (no quadratic or higher terms) are a special case whereby the symmetry of the system leads to an exact formula for determining optimal oracle parameter settings, the first known case for non-Grover QAA. Using simulations of problem sizes up to 40 qubits we demonstrate QAA's algorithmic performance using our derived equation for oracle parameter values across all possible solutions, with an emphasis on the closeness in Grover-like performance for solutions near the global optimum. We conclude with a first-of-its-kind experimental demonstration of generalized QAA on both IBMQ (superconducting) and IonQ (trapped ion) qubits, with and without error mitigation techniques from the respective hardware vendors. In addition to the first ever 5-qubit Grover's experimental demonstration on a trapped ion system, we show that the observed probabilities of each basis state agree with theoretical predictions for the full range of free parameter values in the oracle and diffusion operators.

KEYWORDS: Quantum algorithm; quantum computing; amplitude amplification; combinatorial optimization

1 Introduction

Quantum Amplitude Amplification (QAA) is the generalization of Grover's algorithm, first proposed nearly three decades ago [1] as a means of searching for a marked subset of an unstructured database. This problem is framed as having access to a black-box boolean oracle function $f : \{1, 2, 3, \dots, N\} \rightarrow \{0, 1\}$ mapping elements of the database to either 0 or 1, for which the solver must find the instance(s) $f(x) = 1$ in the fewest calls to f . An optimal classical algorithm requires $O(N)$ calls to the oracle function. Encoding f as a unitary operation, Grover's algorithm solves this problem in only $O(\sqrt{N})$ calls to f using a quantum computer. It was shown shortly thereafter that Grover's algorithm is optimal [2,3], and later that a deterministic (fixed-point) quantum search for marked states is possible with the same speedup [4-7].

Fast-forward nearly 30 years, and the mathematical framework of Amplitude Amplification has been analyzed and generalized in numerous ways [8-15], used to solve various problems beyond database searching [16-23], and incorporated into subroutines for other quantum algorithms [24-27]. Many of these

theoretical advancements were developed in the absence of physical hardware, but simultaneously there have been several successful implementations of QAA across multiple quantum technology platforms [28–35].

In this study we focus on a particular variant of QAA which solves combinatorial optimization problems [36–42]. This formulation of QAA uses an oracle operator similar to the phase-separator operator in QAOA [43–49], applying phases proportional to every solution of a discrete cost function (Hamiltonian), which we call cost oracles. Although we shall use the name oracle in this study to keep with normal convention, we stress that these operators contain no black-box element to them. The motivation for these oracle operations is their quantum circuit efficiency, addressing a common point of criticism of implementing Grover’s [50–52], namely the requirement of full N -Toffoli gate operators or quantum dictionaries [21,53] for the oracle.

QAA as defined in this study consists of applying two alternating operations: oracle and diffusion, each containing a free parameter which can in principle vary with each application. One way to determine these values is through measurement feedback and a classical optimizer, effectively QAOA using diffusion as the mixer [47–49]. Alternatively, it has been shown that cost oracle QAA can succeed in the same manner as Grover’s, using a single parameter value for each operator [37–42]. As in this study, using π as the diffusion free parameter setting has been shown to be capable of achieving high probabilities of measurement for optimal solutions [38,39,41,42] (90%+).

The challenge of QAA as described above is that every combinatorial optimization problem results in a unique oracle operation, which in turn requires determining the optimal free parameter setting(s) for the oracle, diffusion, and total iterations k . For both regular [1] and deterministic (fixed-point) Grover’s [5–7] optimal values for each are exactly computable, while here we show that unconstrained linear cost functions (no quadratic or higher terms) produce a symmetry of solutions which can be used to derive an exact formula for predicting oracle parameter settings. Using simulations of problem sizes up to 40 qubits we show peak probabilities and iterations k for finding all possible solutions, highlighting that algorithmic performance for globally optimal solution identification becomes increasingly Grover-like with problem size.

We conclude with experimental results showing a single iteration of generalized QAA up to 5 qubits, run on both IBMQ and IonQ, with and without noise mitigation techniques implemented by the respective vendors [54,55]. Each experiment is designed around varying the free parameter in either the oracle or diffusion operator, comparing the measured probabilities of each basis state with their predicted theoretical values. Our results show the first ever experimental demonstration of generalized QAA using cost oracles.

2 Amplitude Amplification

We begin by defining the QAA algorithm, given in Algorithm 1.

Algorithm 1: Amplitude amplification algorithm

```

Initialize Qubits:  $|\Psi\rangle = |0\rangle^{\otimes N}$ 
Prepare Equal Superposition:  $H^{\otimes N}|\Psi\rangle = |s\rangle$ 
for  $k$  iterations do
  Apply Oracle Operation:  $U_G(\phi)|\Psi\rangle$  or  $U_C(p_s)|\Psi\rangle$ 
  Apply Diffusion Operation:  $U_s(\theta)|\Psi\rangle$ 
end for
Measure  $|\Psi\rangle$ 

```

Algorithm 1 applies to both parametrized variations of QAA using Grover’s oracle $U_G(\phi)$ and cost oracles $U_C(p_s)$. Starting from the N -qubit equal superposition state $|s\rangle$ defined as

$$|s\rangle = \frac{1}{\sqrt{2^N}} \sum_i^{2^N} |Z_i\rangle, \tag{1}$$

where Z_i is a length- N binary string, we perform k iterations of alternating oracle (U_G or U_C) and diffusion (U_s) operations until concluding with a measurement on $|\Psi\rangle$ in the computational basis (the Bloch sphere z -axis of each qubit). Ideally, the result of this measurement yields the target state(s) $|Z_i\rangle$ solving the problem encoded by the oracle. The optimal runtime of Algorithm 1 using $U_G(\pi)$ together with $U_s(\pi)$ is $k \approx \frac{\pi}{4} \sqrt{2^N/N_m}$ for N_m solutions [1–3].

2.1 Oracle Operations

The Grover oracle $U_G(\phi)$ is a quantum operation defined by the unitary operator

$$U_G(\phi)|Z_i\rangle = \begin{cases} e^{i\phi}|Z_i\rangle & \text{if } |Z_i\rangle \in \text{marked,} \\ |Z_i\rangle & \text{otherwise} \end{cases}, \tag{2}$$

which applies a phase of $e^{i\phi}$ to marked target states. The number of marked states, which we denote N_m , can be as few as one and typically no more than $2^N/4$ [13]. Realizing this fully entangling operation on quantum hardware is a known challenge [50–52].

We shall define \mathbb{M} and \mathbb{N} as sets containing all marked and non-marked Z_i , respectively. The Grover oracle subdivides the state $|\Psi\rangle$ into a two-dimensional subspace spanned by the orthogonal basis states in \mathbb{M} and \mathbb{N} as defined by

$$|m\rangle \equiv \frac{1}{\sqrt{N_m}} \sum_{Z_i \in \mathbb{M}} |Z_i\rangle, \quad |n\rangle \equiv \frac{1}{\sqrt{N_n}} \sum_{Z_i \in \mathbb{N}} |Z_i\rangle. \tag{3}$$

For an N -qubit system one typically has $N_n + N_m = 2^N$, but more generally this summation can be any integer when using qudits [56] or qubits [57]. Writing $|\Psi\rangle$ in terms of the collective states $|n\rangle$ and $|m\rangle$ allows the oracle operation to be rewritten as

$$U_G(\phi) = |n\rangle\langle n| + e^{i\phi}|m\rangle\langle m|. \tag{4}$$

Now we shall extend the formalism of Eqs. (3) and (4) to oracles which apply more than two unique phases, which is how we define cost oracles in this study. Given a cost function $C(Z)$, we define C_i as the value obtained from evaluating the bit string Z_i ($C(Z_i) = C_i$). The result of applying a cost oracle to any basis state is given by

$$U_C(p_s)|Z_i\rangle = e^{iC(Z_i) \cdot p_s} |Z_i\rangle. \tag{5}$$

Next we can generalize the notion of the collective states $|n\rangle$ and $|m\rangle$ to $|C_i\rangle$, defined as

$$|C_i\rangle \equiv \frac{1}{\sqrt{N_i}} \sum_{Z_i \in \mathbb{C}_i} |Z_i\rangle. \tag{6}$$

Analogous to $|n\rangle$ and $|m\rangle$, which are defined by the sets of binary strings \mathbb{N} and \mathbb{M} in Eq. (3), here each collective state $|C_i\rangle$ is defined by the corresponding set \mathbb{C}_i . Specifically, each \mathbb{C}_i is the set of all Z_j which evaluate to the same cost function value $C(Z_j) = C_i$. We can then rewrite the cost oracle as

$$U_C(p_s) = \sum_i^D |C_i\rangle\langle C_i| e^{iC_i p_s}. \quad (7)$$

The unitary operator U_C has one free parameter p_s , which we use to stand for the *phase scale* [41,42] (equivalent to γ in the QAOA literature [47–49]). Analogous to U_G , applying U_C subdivides $|\Psi\rangle$ into a D -dimensional subspace spanned by the orthogonal collective states $|C_i\rangle$, where D is the number of unique evaluations obtainable from $C(Z)$. Importantly, the quantum circuit construction of U_C does not require any knowledge of these collective states or the $|Z_i\rangle$ contained within them (see [41,42] and Appendix E), which are assumed to be unknown to the solver.

To conclude, $D = 2$ is by definition a Grover oracle, while $D \in [3, 2^N]$ defines cost oracles, where D is problem dependent on $C(Z)$. Note that Eqs. (6) and (7) reduce to the form of (3) and (4) for the case $D = 2$, $p_s = \phi$, $\{|C_1\rangle, |C_2\rangle\} = \{|n\rangle, |m\rangle\}$, $\{C_1, C_2\} = \{0, 1\}$, $\{N_1, N_2\} = \{N_n, N_m\}$. Thus, our equations for cost oracles are a generalization of Grover's to higher dimensions $D > 2$.

2.2 Diffusion

We now turn to the diffusion operator $U_s(\theta)$, defined by

$$U_s(\theta) = \mathbb{I} - (1 - e^{i\theta})|s\rangle\langle s|, \quad (8)$$

corresponding to the quantum circuit construction shown in Fig. 1. This is the standard construction of $U_s(\theta)$ as originally proposed by Grover [1], also referred to as the Grover mixer in QAOA [46,47]. This multiqubit operation is the N -qubit equivalent to $P(\theta)$ shown in Eq. (9), applying a phase of $e^{i\theta}$ to the basis state $|1\rangle^{\otimes N}$, and no phase to all other $|Z_i\rangle$.

$$P(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \quad (9)$$

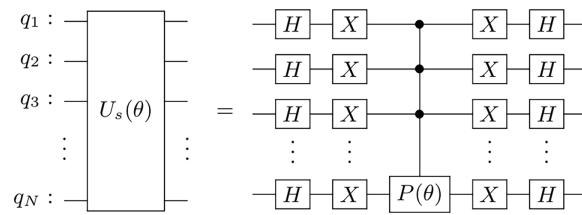


Figure 1: Quantum circuit for the diffusion operator $U_s(\theta)$ acting on N qubits.

Because oracle operations only apply phases, in QAA the diffusion operator is solely responsible for increasing and decreasing amplitude magnitudes (probabilities). The manner in which it does so is proportional to $\bar{\alpha}$, the arithmetic mean of all amplitudes α_i (amplitudes of each basis state: $\alpha_i = \langle Z_i | \Psi \rangle$), given by

$$\bar{\alpha} = \frac{1}{\sqrt{2^N}} \langle s | \Psi \rangle = \frac{1}{2^N} \sum_{i=1}^{2^N} \langle Z_i | \Psi \rangle \quad (10)$$

$$= \frac{1}{2^N} \sum_{i=1}^D N_i \cdot \alpha_i, \tag{11}$$

while

$$\langle Z_j | U_s(\theta) | \Psi \rangle = \alpha_j - (1 - e^{i\theta}) \bar{\alpha}. \tag{12}$$

Eq. (12) is the change in amplitude experienced by each basis state $|Z_i\rangle$ resulting from diffusion. The typical choice for diffusion is $\theta = \pi$, which from Eq. (12) results in $\Delta\alpha_i = \alpha_i - 2\bar{\alpha}$. This change in amplitude is then maximal when α_i and $\bar{\alpha}$ are π phase different, which is the situation created by $U_G(\pi)$ for the marked state(s) [1]. Next in Section 3, we show that this same π phase matching condition can be achieved for U_C encoding unconstrained linear cost functions.

3 Solving Linear Cost Functions

For U_G , the optimal ϕ , θ , and k values for Algorithm 1 are determined by the size of the unmarked and marked subspaces N_n and N_m [1,6,7]. For cost oracles U_C , these parameters are determined by C_i and N_i . In practice however, these values are unknown except through evaluations $C(Z)$, so a realistic implementation of QAA for combinatorial optimization needs to be able to determine p_s , θ , and k values without excessive classical precalculation. Here we demonstrate one such example for unconstrained linear cost functions and discuss the mathematical properties which make it possible.

3.1 Problem Symmetry and $\bar{\alpha}$

Consider the case of a set of N randomly selected, real-valued weights W_i with $\mathbb{W} = \{W_1, W_2, \dots, W_N\}$. We define linear cost functions in this study as

$$C(Z_i) = \sum_{i=1}^N W_i \cdot z_i, \tag{13}$$

where each binary variable z_i is assigned to one weight W_i . Implementing $C(Z)$ according to Eq. (13) as a cost oracle $U_C(p_s)$ is achieved through single qubit phase gates $P(W_i \cdot p_s)$ on each qubit [41,42] (Appendix E). The key feature of this $C(Z)$ that allows one to predict optimal p_s values is the symmetric distribution of the solution space about the arithmetic mean \bar{C} , given by

$$\bar{C} = \frac{1}{2^N} \sum_i C_i \tag{14}$$

$$= \frac{C(Z_j) + C(\neg Z_j)}{2}. \tag{15}$$

Eq. (14) is generic to all $C(Z)$, while (15) is a property of linear $C(Z)$ as defined by Eq. (13) (proof in Appendix C). For linear $C(Z)$ the value of \bar{C} can be obtained by evaluating any Z_j and its bitwise inverse $\neg Z_j$. The significance of this symmetry of C_i solutions about \bar{C} is that the resulting mean amplitude after the first oracle operation $U_C(p_s)|s\rangle$ is

$$\bar{\alpha} = |\bar{\alpha}| e^{i\bar{C} \cdot p_s}, \tag{16}$$

whereby the phase of $\bar{\alpha}$ is proportional to \bar{C} (Appendix C). Even though the value of $\bar{\alpha}$ itself requires complete information of all $|C_i\rangle$, for linear $C(Z)$ knowing the phase of $\bar{\alpha}$ does not. Thus, using Eq. (16), we can derive a

formula for p_s such that the phase of $\bar{\alpha}$ is made to be exactly π different from the phase applied to any target collective state $|C_i\rangle$ of one's choosing.

$$\bar{C} \cdot p_s - C_i \cdot p_s = \pm\pi, \quad (17)$$

$$p_s = \frac{\pm\pi}{\bar{C} - C_i} \text{ for } C_i \neq \bar{C} \quad (18)$$

Eq. (18) ensures a π phase difference between $\bar{\alpha}$ and the target state $|C_i\rangle$ after the first oracle operation, while in Appendix C we show that this phase difference holds for subsequent iterations when using $U_s(\pi)$. A visualization of this property is shown in Fig. 2, displaying a series of complex plane plots for $|\Psi\rangle$ after applying $U_C(p_s)$ according to Eq. (18) for the first five iterations, corresponding to the 10-qubit C(Z) given in Appendix A, Fig. A1 (\mathbb{W}_1) as well as Fig. A2. As indicated by the axes scale below each plot, the probability of measuring the target state is increasing with each subsequent iteration in a manner exactly analogous to Grover iterations $U_s(\pi)U_G(\pi)$ (π phase between $|m\rangle$ and $\bar{\alpha}$), but for an oracle encoding a cost function $U_s(\pi)U_C(p_s)$.

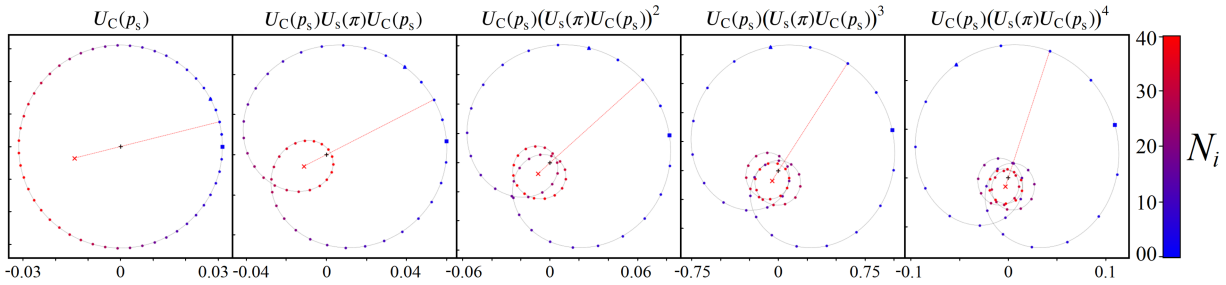


Figure 2: Five complex plane plots of $|\Psi\rangle$ after the application of the operators shown above each panel, corresponding to the 10-qubit C(Z) given in Appendix A. The value of p_s used in U_C is from Eq. (18), for $C_i = 2$. Each colored point in the plots represents the amplitudes contained within one collective state $|C_i\rangle$, with N_i indicated by the accompanying color scale. The extrema states $|C_{\min}\rangle$ and $|C_{\max}\rangle$ are plotted as square and triangle markers, respectively. In each panel the value of $\bar{\alpha}$ is depicted by the red \times , and the origin of the complex plane with a black $+$. A red-dotted line is drawn in each panel from $\bar{\alpha}$ to the collective state $|C_i = 2\rangle$, illustrating their π phase difference via the line's intersection with the origin.

3.2 Algorithmic Performance

The success of QAA for solving a combinatorial optimization problem encoded as U_C is dependent on correctly choosing the three parameters p_s , θ , and k . A reliable means of determining these parameters for problems such as QUBO or harder [37–42] remains an open research question, while here shall show that Eq. (18) for p_s together with $U_s(\pi)$ is sufficient for reliable algorithmic performance.

For the 20-qubit linear C(Z) composed of the integer weights \mathbb{W}_2 given in Appendix A, Fig. 3 shows the joint peak probabilities (solid-colored lines) obtainable for the ten most optimal $|C_i\rangle$ states as a function of p_s . Specifically, each colored curve represents the combined probability of measuring $|C_i = T\rangle$ or its inverse $|C_i = 2\bar{C} - T\rangle$ (for example, $|C_i = -223\rangle$ and $|C_i = 194\rangle$ shown in blue). Over the range of p_s values shown, each peak probability was obtained by simulating Algorithm 1 up to the iteration k where it peaks for the first time (see Fig. 4). The resonance-like shape of these curves is a well-understood feature of Grover's algorithm [8,58–60], also observed for harder combinatorial problems such as QUBO [41,42], and emerges from the condition for constructive interference for the target state (see Appendix D, Fig. A3 for more details).

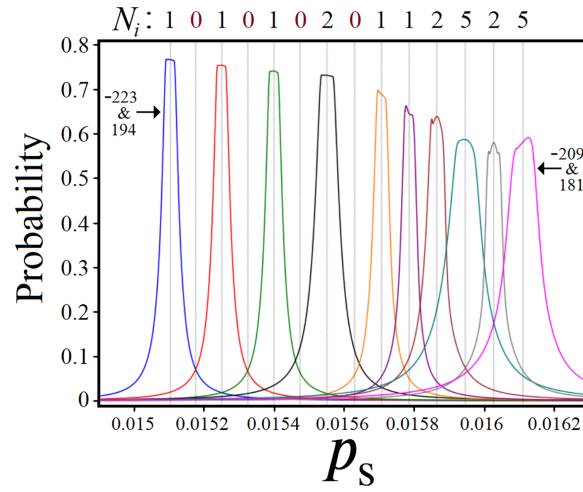


Figure 3: (solid-color lines) Combined peak probability as a function of p_s for the ten highest (lowest) $|C_i\rangle$ states, for the $N = 20$ linear $C(Z)$ composed of the weights \mathbb{W}_2 from Appendix A. (gray-dashed lines) The p_s values produced from Eq. (18) for the integers $T \in [-223, -209]$, with N_i values reported atop each line.

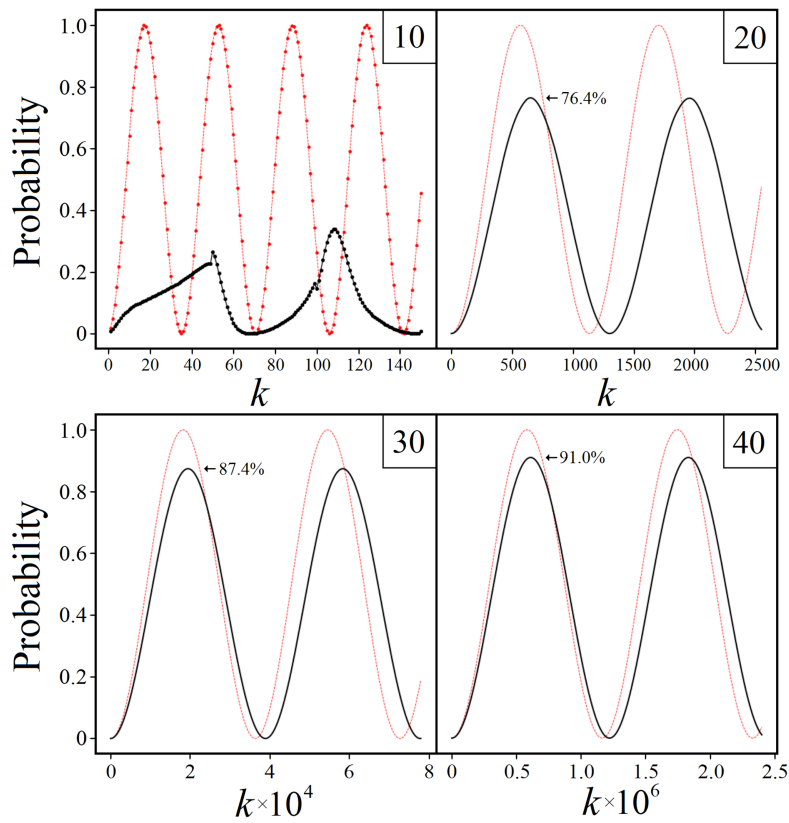


Figure 4: Simulations of QAA showing probability vs. iterations k for qubit sizes $N = 10, 20, 30, 40$ (top right corner), comparing the joint probability of $|C_i = 2\rangle$ and its inverse (solid-black lines) to $|m\rangle$ for standard Grover's (dashed-red lines) for $N_m = 2$. The p_s value used for each U_C is from Eq. (18), while the weights of each linear $C(Z)$ are \mathbb{W}_1 from Appendix A.

Accompanying the curves in the figure are vertical gray lines corresponding to p_s values obtained from Eq. (18) for the integers $T \in [-223, -209]$. Above each line is the corresponding N_i value for each $|C_i = T\rangle$ state (or zero if no such solution exists). As evidenced by the closeness of the lines to each of the ten probability curve peaks, linear $C(Z)$ with integer weights is the ideal scenario for QAA using Eq. (18). Specifically, a solver knows to only check integer values for Eq. (18). Although it is a simple problem instance, it demonstrates that at 20 qubits an exact equation for p_s is capable of producing measurement probabilities of 60%–75%+ for $|C_i\rangle$ near the extrema. However, these probabilities are low compared to larger qubit problem sizes, such as those shown in Fig. 4.

Fig. 4, together with Fig. 5 illustrate two key features of QAA algorithmic performance. Firstly, as qubit size increases the achievable probabilities of solutions near the extrema, and the iterations k to reach these probabilities both become increasingly Grover-like. Secondly, within a single problem instance the peak achievable probabilities of each $|C_i\rangle$ decrease for solutions further away from the extrema. This first feature is an important strength for cost oracle QAA, illustrating $U_C(p_s)$ is capable of producing comparable probabilities to $U_G(\phi)$ without the need for expensive oracle circuits [21,50–53], especially for extrema C_i which are typically the solutions of interest for combinatorial optimization problems.

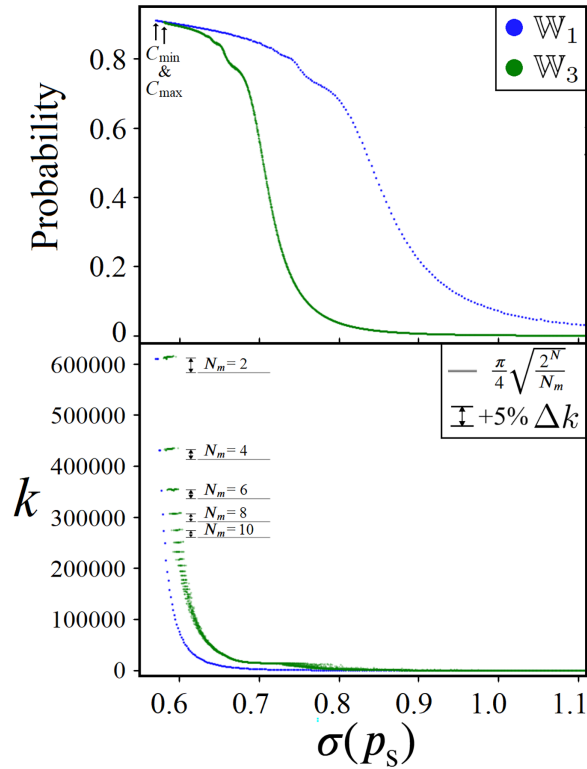


Figure 5: (top) Plot of peak achievable probabilities for each $|C_i\rangle$ state (colored circles) as a function of $\sigma(p_s)$ using p_s from Eq. (18), for the $N = 40$ cases \mathbb{W}_1 and \mathbb{W}_3 given in Appendix A. (bottom) Plot of iterations k corresponding to the probabilities shown in the top plot. For comparison, lines indicating the number of iterations for Grover's ($U_G(\pi)$) using various N_m are plotted. Beside each line is Δk , showing a 5% increase in k from Grover's.

Focusing now on Fig. 5, the top plot shows the complete spectrum of achievable probabilities for all $|C_i\rangle$ for two 40-qubit linear $C(Z)$ using p_s from Eq. (18), while the bottom plot shows the corresponding number of iterations k needed to achieve these probabilities. For completeness, given below in Eq. (19) is $\sigma(p_s)$, the standard deviation of all C_i solutions after scaling by p_s , which is a convenient way to plot all C_i solutions for both $C(Z)$ on the same axis.

$$\sigma(p_s) = \sqrt{\frac{\sum_i^{2^N} (C_i \cdot p_s - \bar{C} \cdot p_s)^2}{2^N}} \tag{19}$$

The significance of Fig. 5 is the bottom plot, particularly the black lines which show the value of k for standard Grover’s for various values of N_m . Beside each black line is a 5% increase in k , labeled Δk , illustrating that for these 40-qubit problem instances the optimal number of iterations for cost oracle QAA is consistently around 5% more than standard Grover’s. Thus, the results shown in Figs. 3–5 illustrate a special case of QAA, oracles encoding linear $C(Z)$ composed of integer weights, whereby a solver can reliably determine near optimal parameter settings for p_s , θ , and k . These results are particularly encouraging because the same underlying features have been observed for QAA using oracles encoding QUBO [37–42], suggesting that these harder problem instances may also have reliable strategies for approximating optimal parameter values as well. In particular, the important condition which led to Eq. (18) is solution space symmetry, which may be applicable to some NP-hard problem instances.

4 Experimental Demonstration

To conclude this study here we present experimental results demonstrating current hardware progress towards the realization of generalized QAA [12] on both IBMQ’s superconducting qubits as well as IonQ’s trapped ion qubits. Each experiment is a single iteration of $U_s(\theta)U_C(p_s)$ or $U_s(\theta)U_G(\phi)$, varying either θ or p_s across 100 different values and tracking the measured probabilities of each basis state $|Z_i\rangle$. At each parameter value we ran the quantum circuit 10,000 times, totaling one million circuit executions per experiment per qubit size. Our results confirm that these probabilities match the theoretically predicted values from our equations, to our knowledge the first ever such experimental demonstration of generalized QAA using cost oracles.

In total we conducted three unique experiments, which we refer to by number, with quantum circuits for each experiment given in Appendix E, Figs. A4 and A5 (for circuit depth and further details on circuit decomposition for U_G and U_s see Figs. A6–A8). Each experiment was implemented for qubit sizes $N = 2, 3, 4, 5$, run on five unique devices: IonQ’s arial ion trap, IBMQ’s fez and torino (heron), and IBMQ’s kyiv, and brisbane (eagle) superconducting processors, with and without error mitigation techniques provided by the respective hardware vendors [54,55]. The implementation of the multicontrolled phase gate $C^N-P(\theta)$ used for both $U_s(\theta)$ and $U_G(\pi)$ is adapted from [61,62], discussed in detail in Appendix E

4.1 The First Iteration

Here we present equations for the theoretical quantum states of experiments 1–3, corresponding to the first iteration of QAA using Grover and cost oracles, analogous to [7,12]. We begin with Eqs. (20) and (21) below.

$$\bar{\alpha}(\phi) = \left(\frac{1}{2^N}\right)^{\frac{3}{2}} \cdot (N_n + e^{i\phi}N_m) \tag{20}$$

$$\bar{\alpha}(p_s) = \left(\frac{1}{2^N}\right)^{\frac{3}{2}} \cdot \sum_j^D e^{iC_j p_s} N_j, \quad (21)$$

which are the average amplitudes $\bar{\alpha}$ (Eq. (11)) as a function of ϕ and p_s for the states $U_G(\phi)|s\rangle$ and $U_C(p_s)|s\rangle$, respectively. Using these equations in combination with (12), we obtain the quantum states after the first iteration for both oracle types given by

$$U_s(\theta)U_G(\phi)|s\rangle = \sqrt{N_n} \left(\frac{1}{\sqrt{2^N}} - \bar{\alpha}(\phi)(1 - e^{i\theta}) \right) |n\rangle + \sqrt{N_m} \left(\frac{e^{i\phi}}{\sqrt{2^N}} - \bar{\alpha}(\phi)(1 - e^{i\theta}) \right) |m\rangle \quad (22)$$

$$U_s(\theta)U_C(p_s)|s\rangle = \sum_j^D \sqrt{N_j} \left(\frac{e^{iC_j p_s}}{\sqrt{2^N}} - \bar{\alpha}(p_s)(1 - e^{i\theta}) \right) |C_j\rangle. \quad (23)$$

Experiments 1 and 2 correspond to Eq. (23), whereby in exp. 1 we prepared and measured 100 quantum circuits for the state $U_s(\pi)U_C(p_s)$ (varying the parameter p_s) and likewise $U_s(\theta)U_C(1)$ for exp. 2 (varying θ). Experiment 3 corresponds to states of Eq. (22) $U_s(\theta)U_G(\pi)$, $N_m = 1$, varying θ over the full range of 2π . Taking the squared magnitude of the amplitudes given in Eqs. (22) and (23) yields the expected theoretical measurement probabilities for each collective state, which we experimentally verify in the coming subsections.

4.2 Performance Metric

Because each experiment is a composition of 100 different quantum circuits, here we shall briefly describe the quantity f , which is a metric that pools the results from all 100 circuits into a single value comparing experimentally measured probabilities vs. theory, full mathematical description given in Appendix B. The quantity f is a ratio comparing experimentally observed measurement counts of each $|Z_i\rangle$ to probabilities predicted by Eqs. (22) and (23): $f = 1$ means that the basis state was observed with the same probability as theoretically expected, while $f = 0$ means an observed probability of $1/2^N$ (the equal superposition state $|s\rangle$), i.e., a completely decohered quantum state. Values of $f < 0$ occur when a basis state's measured probability was found to be on the opposite side of $1/2^N$ as predicted from theory, signaling that the qubits had reached an unintended final $|\Psi\rangle$ not attributable to decoherence. Shown in Fig. 6 is an example plot illustrating three theoretical values of f (no experimental data) for experiment 2, the quantum state $|\Psi\rangle = U_s(\theta)U_C(1)$ over the full 2π range of θ (see Fig. A1 for the $|C_i\rangle$ corresponding to each color).

4.3 Improvement from Error Mitigation

A recent advancement in both IBMQ and IonQ's commercially available quantum computing platforms is the addition of error mitigation as an option for users [54,55]. More specifically each service offers both Pauli gate twirling and dynamical decoupling. In this study we report on qubit performance for error mitigation 'on' and 'off', which refers to the use of both techniques together by the respective hardware vendors, or neither. An example of difference in performance for on vs. off is illustrated in Fig. 7. In each plot the solid-colored lines represent $f = 1$ (Eq. (23)), while the colored data points show the measured probability counts for each $|Z_i\rangle$ state (colored according to the $|C_i\rangle$ each basis state belongs to), with f_{exp} for the entire experiment reported in the top right (f_{exp} given in Appendix B). Examples of $f < 0$ for individual $|Z_i\rangle$ can be seen in the top plot (error mitigation off), illustrating noisy $|\Psi\rangle$ states with clear underlying structure (ripe for error mitigation/correction techniques) as opposed to complete decoherence.

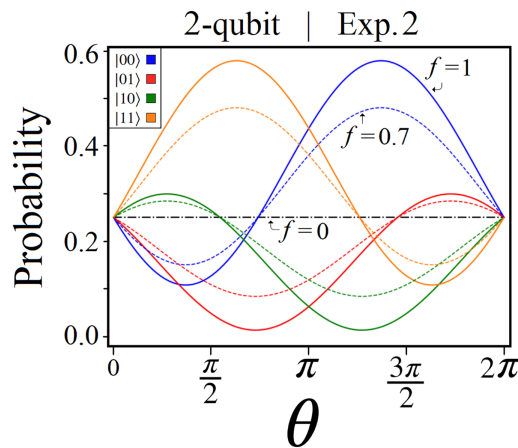


Figure 6: (solid-colored) Probabilities representing $f = 1$ for each basis state as a function of diffusion angle θ for experiment 2, $N = 2$, as predicted from Eq. (23) over the range $\theta \in [0, 2\pi]$. For comparison, probabilities corresponding to $f = 0.7$ (dashed-colored) and 0 (black-dash-dotted) are also plotted.

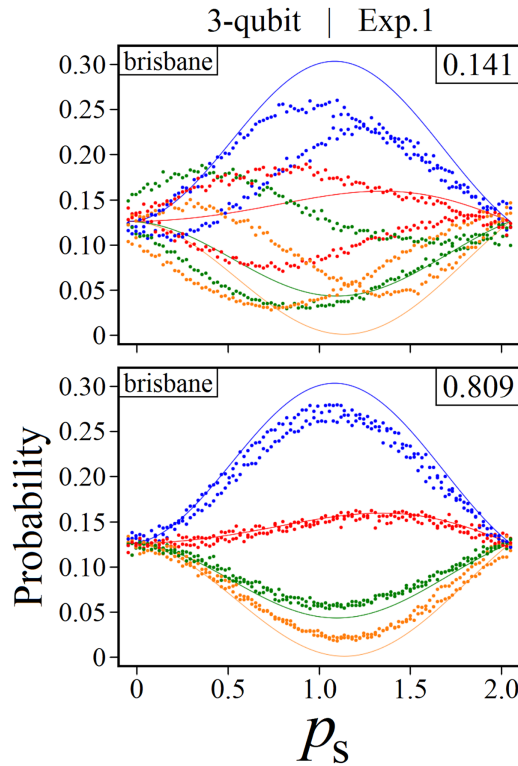


Figure 7: A comparison of measurement results for exp. 1, $N = 3$, on IBMQ’s brisbane with (bottom) and without (top) error mitigation. The solid-colored lines represent probabilities for $f = 1$ (Eq. (23)), while the colored data points are the experimental measurement probabilities obtained for each basis state. Reported in the top right corner is the cumulative f for the experiment, f_{exp} from Eq. (A9) in Appendix B.

4.4 Experimental Results

Given in Fig. 8 below are the full results for exps. 1–3, with error mitigation techniques turned on (top) and off (bottom). The numerical values shown in each table are f_{exp} from Eq. (A9) (Appendix B), the average

f value from all 2^N basis states. The color of each cell indicates the ranked performance of each device within a single experiment, from best to worst: [blue, green, yellow, orange, red]. Cells that are instead colored gray are for values $f_{\text{exp}} < 0.15$.

		Error Mitigation: On				
		ionq_aria1	ibm_fez	ibm_torino	ibm_kyiv	ibm_brisbane
2-qubit	Exp 1	0.949	0.935	0.915	0.930	0.936
	Exp 2	0.959	0.949	0.931	0.950	0.942
	Exp 3	0.956	0.930	0.901	0.879	0.936
3-qubit	Exp 1	0.700	0.871	0.790	0.800	0.809
	Exp 2	0.922	0.899	0.866	0.828	0.843
	Exp 3	0.616	0.850	0.783	0.742	0.706
4-qubit	Exp 1	0.792	0.804	0.739	0.677	0.656
	Exp 2	0.635	0.844	0.797	0.731	0.691
	Exp 3	0.394	0.680	0.629	0.455	0.419
5-qubit	Exp 1	0.281	0.561	0.436	0.248	-0.009
	Exp 2	0.528	0.675	0.612	0.447	0.153
	Exp 3	0.044	0.220	0.169	-0.122	-0.125

		Error Mitigation: Off				
		ionq_aria1	ibm_fez	ibm_torino	ibm_kyiv	ibm_brisbane
2-qubit	Exp 1	0.818	0.942	0.898	0.920	0.897
	Exp 2	0.889	0.927	0.929	0.939	0.917
	Exp 3	0.834	0.928	0.870	0.887	0.899
3-qubit	Exp 1	0.708	0.820	0.831	0.598	0.141
	Exp 2	0.822	0.860	0.870	0.484	0.346
	Exp 3	0.601	0.841	0.721	0.536	0.262
4-qubit	Exp 1	0.252	0.645	0.568	-0.206	-0.128
	Exp 2	0.590	0.779	0.719	0.099	-0.273
	Exp 3	-1.432	0.309	0.286	-1.002	-0.597
5-qubit	Exp 1	-0.379	-0.632	-0.136	-0.926	-1.013
	Exp 2	0.143	0.451	0.460	-0.109	-0.121
	Exp 3	-1.773	-1.005	-0.926	-0.543	-0.439

Figure 8: Two tables summarizing the experimental results of exps. 1–3 for error mitigation on (top) and off (bottom). The numerical value of each cell is f_{exp} from Eq. (A9), with colors indicating ranked performance. Gray cells indicate values less than 0.15.

A summary of the results shown in Fig. 8 is as follows. The effect of error mitigation provided by both vendors improved f_{exp} in 55 out of the 60 experiment runs, producing an overall average increase in f_{exp} by 0.39. Regarding the performance of the individual devices, IBMQ's heron processors fez and torino ranked first and second in 13 out of 18 experiments for qubit sizes $N \geq 3$. However, with error mitigation IonQ's arial trap was first in all three 2-qubit experiments, including the highest overall f_{exp} value of 0.959 as well as the highest 3-qubit value of 0.922. Shown in Fig. 9 are the highest f_{exp} achieved for all experiments and qubit sizes.

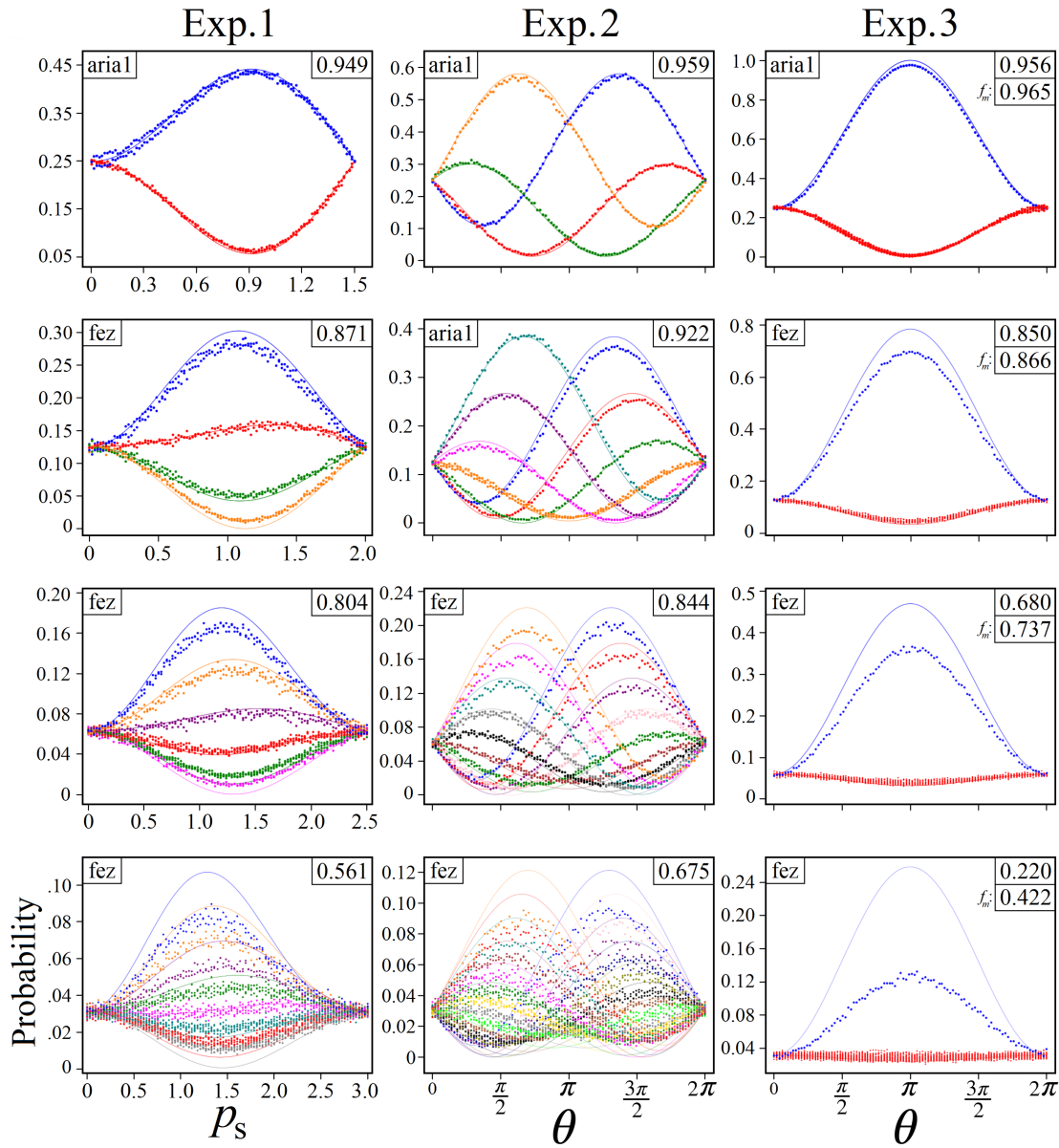


Figure 9: Experiment results showing the highest f_{exp} achieved for each combination of experiment and qubit size. The data points in each plot are $\text{Meas.}(|Z_i\rangle)$ from Eq. (A4), colored according to $|C_i\rangle$ states (see Appendix A). The solid-colored lines in each plot represent $f = 1$ for each collective state, obtained from Eqs. (22) and (23). Reported in the top right corner is the cumulative f for the experiment, f_{exp} from Eq. (A9) in Appendix B, as well as f_m for exp. 3 (f for only the blue data points corresponding to $|m\rangle$, the all $|1\rangle$ state).

5 Conclusion

In this study, we have generalized the common $|n\rangle$ and $|m\rangle$ collective state formalism of Grover's algorithm to cost oracles encoding combinatorial optimization problems [37–42], and shown that for the special case of unconstrained linear $C(Z)$ an exact equation exists for determining the free parameter value of the oracle operator. In Section 3, we used simulations of QAA up to 40 qubits to demonstrate the closeness in algorithmic performance of cost oracle QAA using our equation for p_s with standard Grover's. And finally, in Section 4, we verify our derived equations of generalized QAA [7,12] on two different qubit

technologies, showcasing progress in state-of-the-art commercial quantum devices and their respective error mitigation capabilities.

Outlook and Future Research

The motivation for studying cost oracle QAA [37–42] stems from the quantum circuit efficiency of U_C [63,64] as compared to the full N -Toffoli gates needed for U_G [21,23,31,32,50–53], especially when considering the additional overhead needed for error correction [65,66]. For both oracle cases however, implementing the diffusion operator still remains as a technological bottleneck. Progress in circuit depth efficiency [67–70] is one solution towards achieving large N -Toffoli gates, as well as going beyond 2-qubit gates [71–74]. Additionally there is also the potential for realizing QAA outside of the gate-based model, such as quantum programmable processors [75,76] for implementing fixed unitary operators such as diffusion.

In this work we focus on linear cost functions composed of integer weights to demonstrate ideal conditions for cost oracle QAA. Specifically, integer weights limit the possible values obtainable from the cost function, which in turn are used for selecting the free parameter setting of the oracle via our derived equation. The same algorithmic performance as demonstrated in this study has been shown for QUBO and more complex problems [38–42], but determining the free parameter setting of the oracle for these harder problem instances remains an open research question. Alternatively, one way to sidestep this issue at the cost of increased runtime is the Grover-Mixer QAOA [47–49] style of determining optimal parameter settings for each iteration via measurement results and a classical optimizer. The advantage of QAA as demonstrated in Section 3 is efficiency, requiring ideally only a few measurements to find the desired solution of a cost function.

Lastly, the experimental results of Section 4 show current commercial hardware's ability to achieve QAA up to $N = 5$ qubits, in essence demonstrating their ability to implement the N -Toffoli gate operation. Based on experimental trends of the last 5+ years [29–35] and progress in circuit efficiency [67–70], it does not appear likely that this number N will dramatically increase in the next several years without a significant technological breakthrough (extending N into the 10s or 100s of qubits). By contrast, current hardware (especially superconducting qubits which support parallel gates) is already powerful enough to implement cost oracle operations up to hundreds of qubits, making diffusion the technological bottleneck limiting QAA. Therefore, another avenue for future research is exploring QAA performance using less than full N -qubit diffusion, instead replaced by multiple low qubit-sized diffusions in parallel. This would significantly reduce circuit depth, but also introduces more θ degrees of freedom which complicate the algorithm. If small diffusions in parallel can produce probabilities comparable to full N -qubit diffusion, then QAA could potentially be a viable near term quantum algorithm alongside QAOA for solving combinatorial optimization problems.

Acknowledgement: Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL. This project was supported in part by an appointment to the NRC Research Associateship Program at AFRL, administered by the Fellowships Office of the National Academies of Sciences, Engineering, and Medicine.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Daniel Koch. Methodology, Daniel Koch, Brian Pardo. Formal Analysis, Daniel Koch, Kip Nieman. Experimental Data Generation, Daniel Koch, Brian Pardo. Writing, Reviewing and Editing, Daniel Koch, Brian Pardo, Kip Nieman. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data and code files that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Cost Functions & Distributions

Found in Sections 3 and 4 are references to a particular problem cases of linear cost functions which can be found here. Given in Eq. (A1) is \mathbb{W}_1 , the set of positive integers from 1 to N , which are the sets of weights for Figs. 2, 4, and 5, as well as experiments 1 and 2 in Section 4. Also shown in Fig. 5 is a second 40-qubit cost function corresponding to the weights \mathbb{W}_3 given in Eq. (A3).

$$\mathbb{W}_1 = \{1, 2, 3, \dots, N\} \tag{A1}$$

$$\mathbb{W}_2 = \{-44, -35, -33, -32, -23, -20, -11, -11, -10, -4, 2, 6, 9, 11, 11, 17, 21, 34, 40, 43\} \tag{A2}$$

$$\mathbb{W}_3 = \{-731, -722, -676, -668, -663, -662, -564, -563, -555, -409, -209, -189, -135, -43, 1, 3, 28, 48, 73, 127, 139, 156, 160, 286, 307, 308, 427, 461, 490, 512, 548, 551, 568, 583, 589, 642, 776, 917, 929, 948, 949\} \tag{A3}$$

For the results of experiments 1 and 2 found in Section 4, which use $U_C(p_s)$ encoding linear $C(Z)$ according to Eq. (A1), Fig. A1 below provides a table all of the collective states $|C_i\rangle$ along with the basis states $|Z_i\rangle$ contained within them.

N	$ C_i\rangle$			
2	$ 0\rangle = 00\rangle$	$ 1\rangle = 10\rangle$	$ 2\rangle = 01\rangle$	$ 3\rangle = 11\rangle$
3	$ 0\rangle = 000\rangle$ $ 4\rangle = 101\rangle$	$ 1\rangle = 100\rangle$ $ 5\rangle = 011\rangle$	$ 2\rangle = 010\rangle$ $ 6\rangle = 111\rangle$	$ 3\rangle = 110\rangle, 001\rangle$
4	$ 0\rangle = 0000\rangle$ $ 4\rangle = 1010\rangle, 0001\rangle$ $ 7\rangle = 1101\rangle, 0011\rangle$	$ 1\rangle = 1000\rangle$ $ 5\rangle = 0110\rangle, 1001\rangle$ $ 8\rangle = 1011\rangle$	$ 2\rangle = 0100\rangle$ $ 6\rangle = 1110\rangle, 0101\rangle$ $ 9\rangle = 0111\rangle$	$ 3\rangle = 1100\rangle, 0010\rangle$ $ 10\rangle = 1111\rangle$
5	$ 0\rangle = 00000\rangle$ $ 3\rangle = 11000\rangle, 00100\rangle$ $ 5\rangle = 01100\rangle, 10010\rangle, 00001\rangle$ $ 7\rangle = 11010\rangle, 00110\rangle, 01001\rangle$ $ 9\rangle = 01110\rangle, 10101\rangle, 00011\rangle$ $ 11\rangle = 01011\rangle, 11101\rangle$ $ 13\rangle = 10111\rangle$	$ 1\rangle = 10000\rangle$ $ 4\rangle = 10100\rangle, 00010\rangle$ $ 6\rangle = 11100\rangle, 01010\rangle, 10001\rangle$ $ 8\rangle = 10110\rangle, 11001\rangle, 00101\rangle$ $ 10\rangle = 11110\rangle, 10011\rangle, 01101\rangle$ $ 12\rangle = 00111\rangle, 11011\rangle$ $ 14\rangle = 01111\rangle$	$ 2\rangle = 01000\rangle$ $ 15\rangle = 11111\rangle$	

Figure A1: A table displaying all of the collective states $|C_i\rangle$ for experiments 1 and 2, corresponding to a linear $C(Z)$ composed of the weights \mathbb{W}_1 from Eq. (A1) up to $N = 5$, as well as the basis states $|Z_i\rangle$ contained within them.

Appendix B Fidelity Metric

Here we discuss in full mathematical detail the calculation of f , the metric of performance used in Section 4. We begin with Eq. (A4) for $\text{Meas.}(|Z_i\rangle)$, which is simply the experimentally observed probability of a basis state. ‘Shots’ refers to the number of times a quantum circuit was run, producing that number of measurement results, which in this study is 10,000. Given in Eq. (A5) is the the quantity ΔP_i , which is the difference in probability between measured counts of a $|Z_i\rangle$ basis state (Eq. (A4)) and its theoretical expected probability given by Eq. (23) or (22).

$$\text{Meas.}(|Z_i\rangle) = \frac{\text{Counts}(|Z_i\rangle)}{\text{shots}} \quad (\text{A4})$$

$$\Delta P_i = |\text{Meas.}(|Z_i\rangle) - |\langle Z_i|\Psi\rangle|^2| \quad (\text{A5})$$

The problem with using raw probability difference to evaluate performance is that good or bad is relative for each basis state. This was the motivation for introducing f , which compares each ΔP_i against probabilities corresponding to a completely decohered equal superposition state, given below in Eq. (A6) as $\Delta \tilde{P}_i$.

$$\Delta \tilde{P}_i = \left| \frac{1}{2^N} - |\langle Z_i|\Psi\rangle|^2 \right| \quad (\text{A6})$$

The computation of f is based on the ratio between root mean square calculations of ΔP_i and $\Delta \tilde{P}_i$ values from all 100 circuits, as shown in Eq. (A7) (note that i refers to the basis state while j refers to one of the 100 quantum circuits). For each basis state $|Z_i\rangle$ we compute f_i according to Eq. (A8) (where $\overline{\text{RMS}}$ is obtained from using $\Delta \tilde{P}_i$ values in Eq. (A7)). Averaging all $2^N f_i$ together yields f_{exp} , given in Eq. (A9).

$$\text{RMS}_i = \sqrt{\frac{1}{100} \sum_j \Delta P_{ij}^2} \quad (\text{A7})$$

$$f_i = 1 - \frac{\text{RMS}_i}{\overline{\text{RMS}}_i} \quad (\text{A8})$$

$$f_{\text{exp}} = \frac{1}{2^N} \sum_i f_i \quad (\text{A9})$$

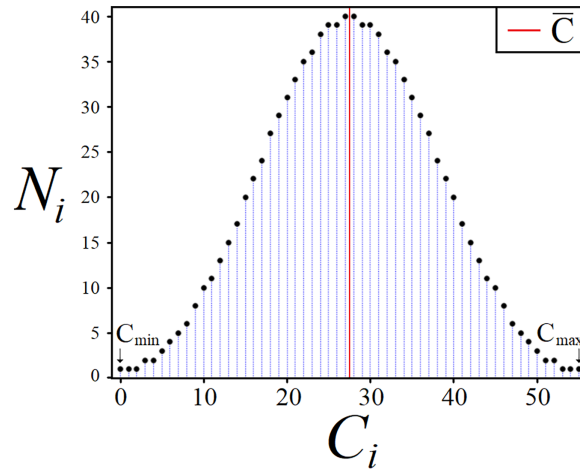


Figure A2: Histogram of all C_i solutions to a 10-qubit $C(Z)$ using the weights \mathbb{W} given in Eq. (A1). (red line) The value \bar{C} from Eq. (14).

Appendix C Symmetry Proofs for Linear $C(Z)$

To assist with the forthcoming derivations, given below in Fig. A2 is a histogram of all C_i solutions to the 10-qubit linear $C(Z)$ composed of the set of weights \mathbb{W}_1 from Eq. (A1). Each black circle shown in the figure corresponds to one collective state $|C_i\rangle$ containing N_i basis states.

Eq. (16) from Section 3 states that for any $U_C(p_s)$ encoding a linear $C(Z)$ the resulting phase of $\bar{\alpha}$ is equal to the $\bar{C} \cdot p_s$. Here we shall derive the equations leading to this relation, showing explicitly the computation of $\bar{\alpha}$ after the first two applications of $U_C(p_s)$. The results which follow are specifically for $C(Z)$ according to Eq. (13) and do not generalize to more complex problems such as QUBO. We begin with Eq. (A10) below which defines a pair of inverse solutions Z_i and $\neg Z_i$, two binary strings that have equal and opposite 0 and 1 values on every bit such that their binary sum equals $2^N - 1$ (base-10). For linear $C(Z)$ the evaluation of any pair of inverse Z_i is equal to the full sum of \mathbb{W} .

$$Z_i + \neg Z_i = 111\dots 1 = 2^N - 1 \tag{A10}$$

$$C(Z_i) + C(\neg Z_i) = \sum_i^N W_i = W_{\text{sum}} \tag{A11}$$

Using Eq. (A11) we can now prove Eq. (11), the mean cost function value \bar{C} according to Eq. (14) is always equal to half of the sum of any pair of inverse solutions, given by Eqs. (A12)–(A15) below.

$$\bar{C} = \frac{1}{2^N} \sum_i^{2^N} C(Z_i) \tag{A12}$$

$$= \frac{1}{2^N} \sum_i^{2^{N-1}} C(Z_i) + C(\neg Z_i) \tag{A13}$$

$$= \frac{2^{N-1} \cdot W_{\text{sum}}}{2} \tag{A14}$$

$$= \frac{C(Z_j) + C(\neg Z_j)}{2} \tag{A15}$$

For the next derivation let us define ΔC_i , given in Eq. (A16), which is simply the difference between the mean cost function value and any $C(Z_i)$.

$$\Delta C_i = \bar{C} - C(Z_i) \tag{A16}$$

Illustrated in Fig. A2 is the symmetry property of linear $C(Z)$ that all solutions come in equal and opposite pairs that are equidistant from \bar{C} . Given in Eqs. (A17)–(A19) below is the proof that these symmetric solutions are in fact pairs of inverse Z_i .

$$C(Z_i) + C(\neg Z_i) = 2\bar{C} \tag{A17}$$

$$\bar{C} - \Delta C_i + C(\neg Z_i) = 2\bar{C} \tag{A18}$$

$$C(\neg Z_i) = \bar{C} + \Delta C_i \tag{A19}$$

Using Eqs. (A16) and (A19), we can now prove the relation between \bar{C} and $\bar{\alpha}$ given in Eq. (16) of Section 3. The following equations use the amplitudes α_i corresponding to the state $|\Psi\rangle = U_C(p_s)|s\rangle$, the first oracle application acting on the equal superposition state. By rewriting the summation for $\bar{\alpha}$ first in terms of inverse Z_i pairs and then substituting in ΔC_i and \bar{C} equivalents, we arrive at Eq. (A24).

$$\bar{\alpha}_1 = \frac{1}{2^N} \sum_j^{2^N} \alpha_j \tag{A20}$$

$$= \frac{1}{2^N \sqrt{2^N}} \sum_j^{2^N} e^{iC(Z_j) \cdot p_s} \quad (\text{A21})$$

$$= \frac{1}{(2^N)^{3/2}} \sum_j^{2^{N-1}} (e^{iC(Z_j) \cdot p_s} + e^{iC(-Z_j) \cdot p_s}) \quad (\text{A22})$$

$$= \frac{1}{(2^N)^{3/2}} \sum_j^{2^{N-1}} e^{i\tilde{C} \cdot p_s} (e^{-i\Delta C_j \cdot p_s} + e^{i\Delta C_j \cdot p_s}) \quad (\text{A23})$$

$$= \frac{e^{i\tilde{C} \cdot p_s}}{(2^N)^{3/2}} \sum_j^{2^{N-1}} 2 \cos(\Delta C_j \cdot p_s) \quad (\text{A24})$$

To summarize, the equations above show that the value of $\bar{\alpha}_1$ after the first application of U_C is equal to $e^{i\tilde{C} \cdot p_s}$ times a constant, confirming the relation given by Eq. (16), which in turn is used to derive Eq. (17) for finding p_s values which create a π phase difference between $\bar{\alpha}$ and any $|C_i\rangle$. Here we have shown that the phase of $\bar{\alpha}$ is proportional to \tilde{C} , but the full computation of $\bar{\alpha}$ still requires complete knowledge of all C_i .

So far we have shown that Eq. (16) holds for the first oracle application, so next we shall prove that it holds for the second as well. Given in Eq. (A25) below are the amplitudes α'_j corresponding to the state $|\Psi\rangle = U_s(\pi)U_C(p_s)|s\rangle$.

$$\alpha'_j = \frac{e^{iC(Z_j) \cdot p_s}}{\sqrt{2^N}} - 2|\bar{\alpha}_1|e^{i\tilde{C} \cdot p_s} \quad (\text{A25})$$

Eqs. (A26)–(A29) below show the calculation of $\bar{\alpha}_2$, the mean amplitude following the second oracle application.

$$\bar{\alpha}_2 = \frac{1}{2^N} \sum_j^{2^N} \frac{e^{i2C(Z_j) \cdot p_s}}{\sqrt{2^N}} - 2|\bar{\alpha}_1|e^{i(\tilde{C}+C(Z_j)) \cdot p_s} \quad (\text{A26})$$

$$= \frac{1}{2^N} \sum_j^{2^{N-1}} \frac{(e^{i2C(Z_j) \cdot p_s} + e^{i2C(-Z_j) \cdot p_s})}{\sqrt{2^N}} - 2|\bar{\alpha}_1|e^{i\tilde{C} \cdot p_s} (e^{iC(Z_j) \cdot p_s} + e^{iC(-Z_j) \cdot p_s}) \quad (\text{A27})$$

$$= \frac{e^{i2\tilde{C} \cdot p_s}}{2^N} \sum_j^{2^{N-1}} \frac{(e^{-i2\Delta C_j \cdot p_s} + e^{i2\Delta C_j \cdot p_s})}{\sqrt{2^N}} - 2|\bar{\alpha}_1|(e^{-i\Delta C_j \cdot p_s} + e^{i\Delta C_j \cdot p_s}) \quad (\text{A28})$$

$$= \frac{e^{i2\tilde{C} \cdot p_s}}{2^{N-1}} \sum_j^{2^{N-1}} \frac{\cos(2\Delta C_j \cdot p_s)}{\sqrt{2^N}} - 2|\bar{\alpha}_1|\cos(\Delta C_j \cdot p_s) \quad (\text{A29})$$

In Eq. (A29) above we see that the phase of $\bar{\alpha}_2$ is equal to $2\tilde{C} \cdot p_s$, but once again the full value of $\bar{\alpha}_2$ requires complete knowledge of every amplitude value. To conclude, we note that the derivation of two iterations given above is general, showing that phase relation given in Eq. (16) holds, also supported by the simulation results of Section 3, but does not constitute a complete proof by induction.

Appendix D Resonance Width in Grover's

The QAA algorithm using iterations of $U_s(\theta)U_G(\phi)$ in Algorithm 1 is optimal when both free parameters are $[\theta, \phi] = [\pi, \pi]$ [1–3]. However, it is important to understand the behavior of the algorithm when these phases differ, particularly when considering noisy gates which lead to imperfect implementations

of the oracle [8,58]. In this appendix we show the relation between peak achievable probabilities as a function of ϕ around π following closely the approach of [59]. The resonance phenomena demonstrated here for Grover's at $\phi = \pi$ is the same as those shown in Fig. 3 for each of the individual $|C_i\rangle$ states [38,41,42]. Eq. (18) from Section 3 produces near optimal p_s values for linear $C(Z)$, but in general determining p_s values which align with $|C_i\rangle$ resonance peaks for QUBO and harder problem instances is an open research question [40]. We begin with Eq. (A30) below, the equal superposition state $|s\rangle$ expressed as a column vector. For simplicity the initial amplitudes of $|m\rangle$ and $|n\rangle$ are $\sin \beta = \sqrt{N_m/2^N}$ and $\cos \beta = \sqrt{N_n/2^N}$, respectively.

$$|s\rangle = \sin \beta |m\rangle + \cos \beta |n\rangle = \begin{bmatrix} \sin \beta \\ \cos \beta \end{bmatrix}. \tag{A30}$$

Next we need the 2×2 matrix form of $U_s(\theta)U_G(\phi)$, which we call G in Eq. (A31) below.

$$G \equiv U_s(\theta)U_G(\phi) \tag{A31}$$

$$= \begin{bmatrix} e^{i\phi}(1+(e^{i\theta}-1)\sin^2(\beta)) & (e^{i\theta}-1)\sin(\beta)\cos(\beta) \\ e^{i\phi}(e^{i\theta}-1)\sin(\beta)\cos(\beta) & 1+(e^{i\theta}-1)\cos^2(\beta) \end{bmatrix} \tag{A32}$$

Since G given in Eq. (A32) is a unitary matrix, it can be represented generically by the matrix decomposition given in Eq. (A33), where U is some unitary 2×2 matrix and D is a diagonal matrix with values λ_+ and λ_- . The matrix U has columns given by the normalized eigenvectors of Eq. (A34), with eigenvalues given in Eqs. (A35) and (A36).

$$G = U^*DU, \tag{A33}$$

$$u_+ = \begin{bmatrix} e^{-i\frac{\phi}{2}} \cos x \\ \sin x \end{bmatrix} \quad u_- = \begin{bmatrix} -\sin x \\ e^{i\frac{\phi}{2}} \cos x \end{bmatrix} \tag{A34}$$

$$\lambda_{\pm} = -e^{i\frac{(\theta+\phi)}{2} \pm w} \tag{A35}$$

$$\cos w = \cos\left(\frac{\phi-\theta}{2}\right) - 2 \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin^2 \beta. \tag{A36}$$

Using Eqs. (A33)–(A36), we can write the general form of the Grover operator G after t iterations, given in Eq. (A37).

$$G^t = \begin{bmatrix} e^{iwt} \cos^2 x + e^{-iwt} \sin^2 x & i e^{-i\frac{\phi}{2}} \sin(wt) \sin(2x) \\ i e^{i\frac{\phi}{2}} \sin(wt) \sin(2x) & e^{iwt} \sin^2 x + e^{-iwt} \cos^2 x \end{bmatrix} \tag{A37}$$

Next we need expressions for the angle x , which can be found by using the equation $Gu_+ = \lambda_+u_+$, or equivalently by setting the off-diagonal elements of U^*GU to 0. Solving this first expression yields the equations for $\sin x$ and $\cos x$ given in Eqs. (A38)–(A40) below.

$$\sin x = l_m^{-\frac{1}{2}} \sin \frac{\theta}{2} \sin(2\beta) \tag{A38}$$

$$\cos x = l_m^{-\frac{1}{2}} \left[\sin w + \sin\left(\frac{\phi-\theta}{2}\right) + 2 \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin^2 \beta \right] \tag{A39}$$

$$l_m = \left(\sin \frac{\theta}{2} \sin(2\beta) \right)^2 + \left[\sin w + \sin\left(\frac{\phi-\theta}{2}\right) + 2 \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin^2 \beta \right]^2 \tag{A40}$$

Using the equations above for G^t , we can compute the amplitude of $|m\rangle$ at any iteration t using Eq. (A41) below.

$$\langle m|G^t|s\rangle = \sin\beta \cos(wt) + i\left(e^{i\phi/2} \cos\beta \sin(2x) + \sin\beta \cos(2x)\right) \sin(wt) \quad (\text{A41})$$

Taking the absolute value squared of Eq. (A41) yields $P_m(t) = |\langle m|G^t|s\rangle|^2$, the probability of measuring $|m\rangle$ after t iterations. In the large 2^N limit, this probability simplifies significantly to Eq. (A42) below.

$$P_m(t) \approx \sin^2(2x) \sin^2(wt) \quad (\text{A42})$$

The probability $P_m(t)$ is maximal at $t = \frac{\pi}{2w}$, which after substituting into the equation above yields Eq. (A43) for P_{\max} , the maximum achievable probability for $|m\rangle$ as a function of ϕ and θ .

$$P_{\max} \approx \frac{4 \sin^2 \frac{\theta}{2}}{2^N \sin^2\left(\frac{\theta - \phi}{2}\right) + 4 \sin \frac{\theta}{2} \sin \frac{\phi}{2} \cos\left(\frac{\theta - \phi}{2}\right)} \quad (\text{A43})$$

And finally, we are interested in the case of $\theta = \pi$, yielding Eq. (A44) for P_{\max} as a function of U_G oracle angle ϕ . Shown in Fig. A3 are four plots for various problem sizes N , illustrating the resonance behavior sharply peaked around $\phi = \pi$, becoming narrower with increasing problem size N .

$$P_{\max} = \frac{1}{\sin^2 \frac{\phi}{2} + \frac{2^N}{4} \cos^2 \frac{\phi}{2}}. \quad (\text{A44})$$

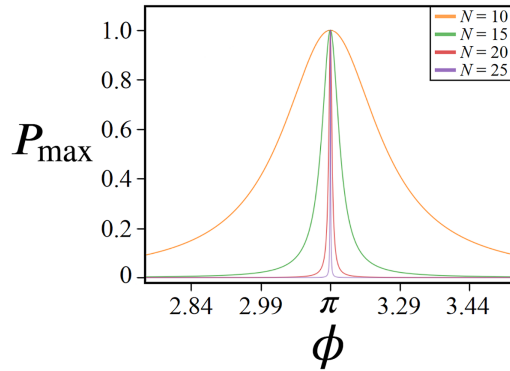


Figure A3: Peak achievable probability P_{\max} from Eq. (A44) as a function of oracle phase ϕ for various N -qubit problem sizes. Each plot shows the resonance behavior around $\phi = \pi$ for Grover's algorithm using $U_s(\pi)$ for diffusion searching for a single marked state $N_m = 1$.

Fig. A3 illustrates the degree of precision in ϕ necessary for Grover's algorithm as problem size N increases, also representing the required accuracy in p_s for cost oracles [38,42]. We can quantify this required precision with Eq. (A45) below, the full width half maximum (FWHM) for problem sizes $N > 2$.

$$\delta\phi = 2 \cos^{-1} \left(\frac{2}{\sqrt{2^N - 4}} \right) \quad (\text{A45})$$

Appendix E Quantum Circuits

Here we present the quantum circuits for experiments 1–3 from Section 4, with further details on the decomposition of the C^N - $P(\theta)$ gate given in the next subsection. Shown in Fig. A4 below is the quantum circuit for experiments 1 and 2, corresponding to the state $U_s(\theta)U_C(p_s)|s\rangle$ given in Eq. (23). The oracle U_C here encodes \mathbb{W}_1 from Eq. (A1). Experiment 1 uses $\theta = \pi$ while varying p_s , while conversely experiment 2 uses $p_s = 1$ while varying θ . The value of N' is given by Eq. (A46) below.

$$N' = \sum_{i=1}^N i = \frac{1}{2}(N^2 + N) \tag{A46}$$

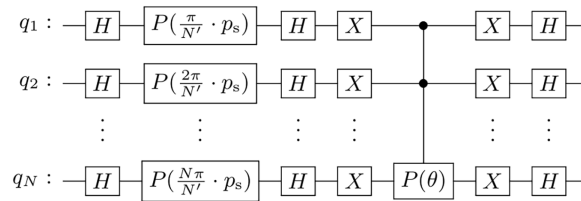


Figure A4: Quantum circuit for experiments 1 and 2.

The quantum circuit for experiment 3 is given in Fig. A5, using $U_G(\pi)$ as the oracle while varying θ in the diffusion operator. The C^N - Z operator shown in the figure was implemented using C^N - $P(\pi)$ as discussed in the next subsection. The quantum state produced from this circuit is Eq. (22). Given in Fig. A6 is a table detailing the circuit depth and 2-qubit gate count for all three experiments on IBMQ’s four processors, where circuit depth is defined as parallel gate layers [63].

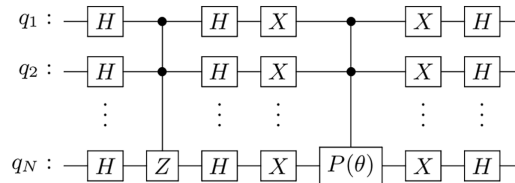


Figure A5: Quantum circuit for experiment 3.

Experiment	Qubits	Fez		Torino		Kyiv		Brisbane	
		Depth	2q Gates	Depth	2q Gates	Depth	2q Gates	Depth	2q Gates
1	2	23	2	23	2	28	2	28	2
	3	51	9	57	9	71	9	56	9
	4	113	20	113	20	111	20	124	20
	5	239	48	239	48	235	48	259	48
2	2	23	2	23	2	28	2	28	2
	3	51	9	51	9	60	9	61	9
	4	113	20	119	20	111	20	130	20
	5	239	48	239	48	228	48	296	63
3	2	32	4	32	4	42	4	27	4
	3	95	21	95	21	124	21	127	21
	4	219	43	219	43	232	43	246	43
	5	493	126	534	126	626	126	615	126

Figure A6: Circuit depth and number of 2 qubit gates across all experiments and devices.

The most resource-intensive component of both the Grover oracle and the diffusion operator is the fully-entangling C^N - $P(\theta)$ gate. This multiqubit gate must be transpiled into 1 & 2-qubit instructions, for which there are several approaches. In this work, we opt for a simple decomposition that is not optimized for any particular architecture. We send the same quantum circuit to both IBMQ and IonQ, decomposing $C^N P(\theta)$ into 2-qubit CX and single qubit $P(\theta)$ operations. For IBMQ we transpiled directly to the backend's qubits, while for IonQ qubit selection and gate transpilation were left to the hardware provider.

The decomposition procedure used in this study is adapted from [61,62]. Consider a sequence of binary variables z_i for $i \in [1, N]$. We want to apply a 2×2 unitary gate U , in our case the phase gate $P(\theta)$ from Eq. (9), to the N^{th} qubit if and only if $z_i = 1$ for all $i \in [1, N - 1]$. This operation can be expressed by the identity given in Eq. (A47).

$$2^{m-1}(z_1 \wedge z_2 \wedge \dots \wedge z_m) = \sum_{k_1} z_{k_1} - \sum_{k_1 < k_2} (z_{k_1} \oplus z_{k_2}) + \dots + (-1)^{m-1}(z_1 \oplus \dots \oplus z_m) \quad (\text{A47})$$

The operation above can be implemented using a Gray sequence to determine the control and target qubits. An N -bit Gray code is the sequence of 2^N binary numbers with adjacent elements in the sequence differing by a single bit flip. The particular Gray sequence that we use can be determined recursively as follows. We denote a Gray sequence by $G_N = (g_1, g_2, \dots, g_{2^N})$, with the base case Gray sequence given by $G_1 = (0, 1)$. We can recursively define a Gray sequence given by $G_{N+1} = (0G_N, 1\tilde{G}_N)$, where \tilde{G} is the reverse ordering of G . For example, this produces $G_2 = (00, 01, 11, 10)$ followed by $G_3 = (000, 001, 011, 010, 110, 111, 101, 100)$. To implement this code as gates, we note that the leading 1 corresponds to the target qubit and the remaining 1's correspond to the XOR combination for which we apply the gate V or V^* . The parity of the bits determine which is applied: V if even or V^* if odd. Given in Fig. A7 below is the quantum circuit used in this study for implementing a C^N - $P(\theta)$ gate, with further details given in Fig. A8.

We note that the approach used in this study requires $O(2^N)$ CNOT gates, and thus is only practical for small N instances. However, our approach did give shallower circuit depth and fewer gate counts than the native C^N - $P(\theta)$ implementation by Qiskit at the time of running (pre-Qiskit 2.0). For example, at $N = 5$ our transpilation for diffusion was 48 CNOTs as compared to Qiskit's `MCPHASE` gate which transpiled to 70.

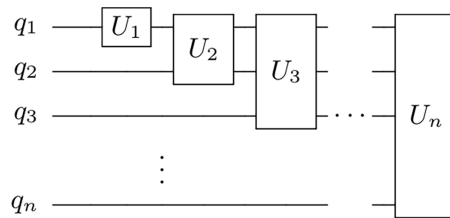


Figure A7: Quantum circuit for implementing the gray code.

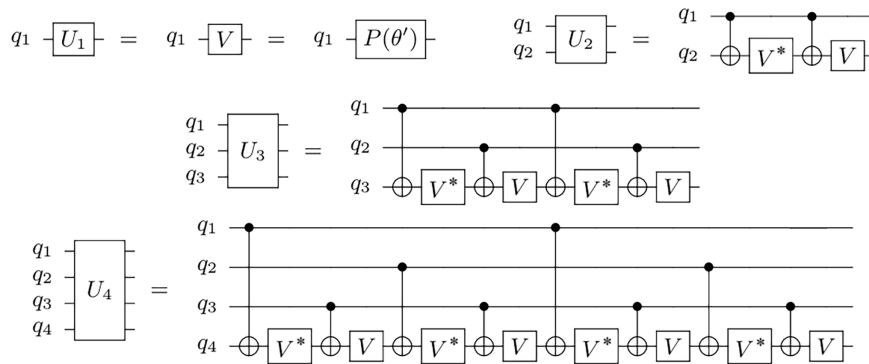


Figure A8: Quantum circuits for each of the unitaries U_n as shown in Fig. A7. For achieving an N -qubit C^N - $P(\theta)$ phase gate, the operators V and V^* are the single qubit phases gates $P(\theta')$ and $P(-\theta')$, respectively, where $\theta' = \theta/2^{N-1}$.

References

1. Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing; 1996 May 22–24; Philadelphia, PA, USA. p. 212–29.
2. Boyer M, Brassard G, Høyer P, Tapp A. Tight bounds on quantum searching. Fortschritte Phys. 1998;46:493–506. doi:10.1002/(sici)1521-3978(199806)46:4/5<493::aid-prop493>3.0.co;2-p.
3. Zalka C. Grover’s quantum searching algorithm is optimal. Phys Rev A. 1999;60:2746–51.
4. Long GL. Grover algorithm with zero theoretical failure rate. Phys Rev A. 2001;64:022307. doi:10.1103/physreva.64.022307.
5. Grover LK. Fixed-point quantum search. Phys Rev Lett. 2005;95:150501. doi:10.1103/physrevlett.95.150501.
6. Yoder TJ, Low GH, Chuang IL. Fixed-point quantum search with an optimal number of queries. Phys Rev Lett. 2014;113(21):210501. doi:10.1103/physrevlett.113.210501.
7. Roy T, Jiang L, Schuster DI. Deterministic Grover with a restricted oracle. Phys Rev Res. 2022;4:L022013.
8. Long GL, Li YS, Zhang WL, Niu L. Phase matching in quantum searching. Phys Lett A. 1999;262(1):27–34. doi:10.1016/s0375-9601(99)00631-3.
9. Høyer P. Arbitrary phases in quantum amplitude amplification. Phys Rev A. 2000;62(5):052304. doi:10.1103/physreva.62.052304.
10. Biham E, Biham O, Biron D, Grassl M, Lidar DA, Shapira D. Generalized Grover’s quantum search algorithms using recursion equations. Phys Rev A. 2001;63(1):012310. doi:10.1103/physreva.63.012310.
11. Byrnes T, Forster G, Tessler L. Generalized Grover’s algorithm for multiple phase inversion states. Phys Rev Lett. 2018;120(6):060501. doi:10.1103/physrevlett.120.060501.
12. Kwon H, Bae J. Quantum amplitude amplification operators. Phys Rev A. 2021;104(6):062438. doi:10.1103/physreva.104.062438.
13. Szablowski PJ. Understanding the mathematics of Grover’s algorithm. Quantum Inf Process. 2021;20(5):191. doi:10.1007/s11128-021-03125-w.
14. Sun Y, Wu LA. Quantum search algorithm on weighted databases. Sci Rep. 2024;14(1):30169. doi:10.1038/s41598-024-81701-7.
15. Suzuki Y, Gluza M, Son J, Tiang BH, Ng NHY, Holmes Z. Grover’s algorithm is an approximation of imaginary-time evolution. arXiv:2507.15065. 2025.
16. Brassard G, Høyer P, Mosca M, Tapp A. Quantum amplitude amplification and estimation. In: Contemporary mathematics. Vol. 305. London, UK: AMS; 2002. p. 53–74.
17. Childs AM, Goldstone J. Spatial search by quantum walk. Phys Rev A. 2004;70:022314.
18. Aaronson S, Ambainis A. Quantum search of spatial regions. Theory Comput. 2005;1:47–79. doi:10.1109/sfcs.2003.1238194.

19. Janmark J, Meyer DA, Wong TG. Global symmetry is unnecessary for fast quantum search. *Phys Rev Lett.* 2014;112(21):210502. doi:10.1103/physrevlett.112.210502.
20. Anikeeva G, Markovic O, Borish V, Hines JA, Rajagopal SV, Cooper ES, et al. Number partitioning with Grover's algorithm in central spin systems. *PRX Quantum.* 2021;2(2):020319. doi:10.1103/prxquantum.2.020319.
21. Gilliam A, Woerner S, Gonciulea C. Grover adaptive search for constrained polynomial binary optimization. *Quantum.* 2021;5:428.
22. Nieman K, Durand H, Patel S, Koch D, Alsing PM. Investigating an amplitude amplification-based optimization algorithm for model predictive control. *Digit Chem Eng.* 2024;10:100134.
23. Zhang Z, Paredes R, Sundar B, Quiroga D, Kyrillidis A, Duenas-Osorio L, et al. Grover-QAOA for 3-SAT: quadratic speedup, fair-sampling, and parameter clustering. *Quantum Sci Technol.* 2025;10:015022.
24. Harrow AW, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. *Phys Rev Lett.* 2009;103(15):150502. doi:10.1103/physrevlett.103.150502.
25. Brassard G, Høyer P, Tapp A. Quantum counting. Vol. 1443. Berlin/Heidelberg, Germany: Springer; 1998. p. 820–31.
26. Grinko D, Gacon J, Zoufal C, Woerner S. Iterative quantum amplitude estimation. *npj Quant Inf.* 2021;7:52.
27. Muser T, Zapusek E, Bellis V, Reiter F. Provable advantages of kernel-based quantum learners and quantum preprocessing based on Grover's algorithm. *Phys Rev A.* 2024;110(3):032434. doi:10.1103/physreva.110.032434.
28. Chen K, Li CM, Zhang Q, Chen YA, Goebel A, Chen S, et al. Experimental realization of one-way quantum computing with two-photon four-qubit cluster states. *Phys Rev Lett.* 2007;99(12):120503. doi:10.1103/physrevlett.99.120503.
29. Figgatt C, Maslov D, Landsman KA, Linke NM, Debnath S, Monroe C. Complete 3-qubit Grover search on a programmable quantum computer. *Nat Commun.* 2017;8:1918. doi:10.1038/s41467-017-01904-7.
30. Mandviwalla A, Ohshiro K, Ji B. Implementing Grover's algorithm on the IBM quantum computers. In: *Proceedings of the 2018 IEEE International Conference on Big Data; 2018 Dec 10–13; Seattle, WA, USA.* p. 2531–7.
31. Zhang K, Rao P, Yu K, Lim H, Korepin V. Implementation of efficient quantum search algorithms on NISQ computers. *Quantum Inf Process.* 2021;20(7):233. doi:10.1007/s11128-021-03165-2.
32. Zhang K, Yu K, Korepin V. Quantum search on noisy intermediate-scale quantum devices. *Europhys Lett.* 2022;140(1):18002. doi:10.1209/0295-5075/ac90e6.
33. Pokharel B, Lidar DA. Better-than-classical Grover search via quantum error detection and suppression. *npj Quantum Inf.* 2024;10(1):23. doi:10.1038/s41534-023-00794-6.
34. Thorvaldson I, Poulos D, Moehle CM, Misha SH, Edlbauer H, Reiner J, et al. Grover's algorithm in a four-qubit silicon processor above the fault-tolerant threshold. *Nat Nanotechnol.* 2025;20(4):472–7. doi:10.1038/s41565-024-01853-5.
35. AbuGhanem M. Characterizing Grover search algorithm on large-scale superconducting quantum computers. *Sci Rep.* 2025;15(1):1281. doi:10.1038/s41598-024-80188-6.
36. Shyamsundar P. Non-Boolean quantum amplitude amplification and quantum mean estimation. *Quantum Inf Process.* 2023;22(12):423. doi:10.1007/s11128-023-04146-3.
37. Satoh T, Ohkura Y, Meter RV. Subdivided phase oracle for NISQ search algorithms. *IEEE Trans Quantum Eng.* 2020;1:3100815. doi:10.1109/tqe.2020.3012068.
38. Benchasattabuse N, Satoh T, Hajdušek M, Van Meter R. Amplitude amplification for optimization via subdivided phase oracle. *arXiv:2205.00602.* 2022.
39. Tani K, Tsuchiya S, Tani S, Takeuchi Y. Quantum algorithm for unstructured search of ranked targets. *Phys Scr.* 2025;100(7):075114. doi:10.1088/1402-4896/ade377.
40. Zhukov A, Lebedev A, Pogosov W. Grover's search meets Ising models: a quantum algorithm for finding low-energy states. *Comput Phys Commun.* 2025;313:109627.
41. Koch D, Cutugno M, Karlson S, Patel S, Wessing L, Alsing PM. Gaussian amplitude amplification for quantum pathfinding. *Entropy.* 2022;24(7):963. doi:10.3390/e24070963.
42. Koch D, Cutugno M, Patel S, Wessing L, Alsing PM. Variational amplification for solving QUBO problems. *Quantum Rep.* 2023;5(4):625–58. doi:10.3390/quantum5040041.
43. Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm. *arXiv:1411.4028.* 2014.

44. Hadfield S, Wang Z, O’Gorman B, Rieffel EG, Venturelli D, Biswas R. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*. 2019;12(2):34. doi:10.3390/a12020034.
45. Maciejewski FB, Bach BG, Dupont M, Lott PA, Sundar B, Bernal Neira DE, et al. A multilevel approach for solving large-scale QUBO problems with noisy hybrid quantum approximate optimization. In: *Proceedings of the 2024 IEEE High Performance Extreme Computing Conference (HPEC)*; 2024 Sep 23–27; Wakefield, MA, USA. p. 1–10.
46. Blekos K, Brand D, Ceschini A, Chou C-H, Li R-H, Pandya K, et al. A review on quantum approximate optimization algorithm and its variants. *Phys Rep*. 2024;1068(2):1–66. doi:10.1016/j.physrep.2024.03.002.
47. Bärtschi A, Eidenbenz S. Grover mixers for QAOA: shifting complexity from mixer design to state preparation. arXiv:2006.00354. 2020.
48. Bridi GA, Marquezino FdL. Analytical results for the quantum alternating operator ansatz with Grover mixer. *Phys Rev A*. 2024;110(5):052409. doi:10.1103/physreva.110.052409.
49. Xie N, Xu J, Chen T, Lee X, Saito Y, Asai N, et al. Performance upper bound of a Grover-mixer quantum alternating operator ansatz. *Phys Rev A*. 2025;111(1):012401. doi:10.1103/physreva.111.012401.
50. Stoudenmire EM, Waintal X. Opening the black box inside Grover’s algorithm. *Phys Rev X*. 2024;14(4):041029. doi:10.1103/physrevx.14.041029.
51. Babbush R, McClean JR, Newman M, Gidney C, Boixo S, Neven H. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum*. 2021;2(1):010103. doi:10.1103/prxquantum.2.010103.
52. Hoefler T, Haener T, Troyer M. Disentangling hype from practicality: on realistically achieving quantum advantage. arXiv:2307.00523. 2023.
53. Gilliam A, Venci C, Muralidharan S, Dorum V, May E, Narasimhan R, et al. Foundational patterns for efficient quantum computing. arXiv:1907.11513v3. 2021.
54. IBMQ. Error suppression and mitigation techniques. [cited 2024 Nov 5]. Available from: <https://docs.quantum.ibm.com/guides/error-mitigation-and-suppression-techniques>.
55. IonQ. Debiasing and sharpening. [cited 2024 Nov 11]. Available from: <https://ionq.com/resources/debiasing-and-sharpening>.
56. Wang Y, Hu Z, Sanders BC, Kais S. Qudits and high-dimensional quantum computing. *Front Phys*. 2020;8:589504. doi:10.3389/fphy.2020.589504.
57. Shukla A, Vedula P. An efficient implementation of a quantum search algorithm for arbitrary N. *Eur Phys J Plus*. 2025;140(6):575. doi:10.1140/epjp/s13360-025-06518-3.
58. Long GL, Li YS, Zhang WL, Tu CC. Dominant gate imperfection in Grover’s quantum search algorithm. *Phys Rev A*. 2000;61(4):042305. doi:10.1103/physreva.61.042305.
59. Hsieh JY, Li CM, Chuu DS. An improved phase error tolerance in a quantum search algorithm. *Chin J Phys*. 2004;42(5):585. doi:10.5772/14391.
60. Romanelli A, Auyuanet A, Donangelo R. Quantum search with resonances. *Phys A Stat Mech Its Appl*. 2006;360(2):274–84. doi:10.1016/j.physa.2005.05.101.
61. Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, Shor P, et al. Elementary gates for quantum computation. *Phys Rev A*. 1995;52(5):3457–67. doi:10.1103/physreva.52.3457.
62. Schuch N. *Implementation of quantum algorithms with Josephson charge qubits*. Regensburg, Germany: Universität Regensburg; 2002.
63. McKay DC, Hincks I, Pritchett EJ, Carroll M, Govia LCG, Merkel ST. Benchmarking quantum processor performance at scale. arXiv:2311.05933. 2023.
64. Pelofske E, Bärtschi A, Eidenbenz S. Short-depth QAOA circuits and quantum annealing on higher-order ising models. *npj Quantum Inf*. 2024;10:30. doi:10.21203/rs.3.rs-3238348/v2.
65. Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*. 2023;614(7949):676–81.
66. Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature*. 2025;638(8052):920–6.
67. He Y, Luo M, Zhang E, Wang HK, Wang XF. Decompositions of n-qubit Toffoli gates with linear circuit complexity. *Int J Theor Phys*. 2017;56(7):2350–61. doi:10.1007/s10773-017-3389-4.

68. Dreier F, Fleckenstein C, Aigner G, Fellner M, Stahn R, Lanthaler M, et al. Connectivity-aware synthesis of quantum algorithms. In: Proceedings of the 2025 IEEE International Conference on Quantum Computing and Engineering (QCE); 2025 Aug 30–Sep 5; Albuquerque, NM, USA. p. 394–95.
69. Nie J, Zi W, Sun X. Quantum circuit for multi-qubit Toffoli gate with optimal resource. arXiv:2402.05053. 2024.
70. Silva JDS, da Silva AJ. Logarithmic depth decomposition of approximate multi-controlled single-qubit gates without ancilla qubits. arXiv:2507.00400. 2025.
71. Maslov D. Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. *Phys Rev A*. 2016;93(2):022311. doi:10.1103/physreva.93.022311.
72. Kim Y, Morvan A, Nguyen LB, Naik RK, Jünger C, Chen L, et al. High-fidelity three-qubit Toffoli gate for fixed-frequency superconducting qubits. *Nat Phys*. 2022;18(7):783–8. doi:10.1038/s41567-022-01590-3.
73. Warren CW, Fernández-Pendás J, Ahmed S, Abad T, Bengtsson A, Biznárová J, et al. Extensive characterization and implementation of a family of three-qubit gates at the coherence limit. *npj Quantum Inf*. 2023;9(1):44. doi:10.1038/s41534-023-00711-x.
74. Katz O, Cetina M, Monroe C. Programmable N-body interactions with trapped ions. *PRX Quantum*. 2023;4:030311.
75. Harris NC, Carolan J, Bunandar D, Prabhu M, Hochberg M, Baehr-Jones T, et al. Linear programmable nanophotonic processors. *Optica*. 2018;5(12):1623–31. doi:10.1364/optica.5.001623.
76. Mohit F, Guanzon J, McKinlay J, Weinhold TJ, Myers CR, Almeida MP, et al. Quantum mechanics can find a needle in a haystack every time. arXiv:2506.06435. 2025.