Quantum-classical tradeoffs and multi-controlled quantum gate decompositions in variational algorithms

Teague Tomesh[1,2], Nicholas Allen[1,3], Daniel Dilley[4,5], and Zain H. Saleem[5]

[1]Princeton University, Princeton, NJ, 08540, USA
[2]Infleqtion, Chicago, IL, 60604, USA
[3]University of Waterloo, Institute for Quantum Computing, Waterloo, ON N2L 3G1, Canada
[4]Quantum Quadrivium Technologies LLP, Evergreen Park , IL, 60805, USA
[5]Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, IL, 60439, USA

arXiv:2210.04378v4 [quant-ph] 1 Oct 2024

The computational capabilities of near-term quantum computers are limited by the noisy execution of gate operations and a limited number of physical qubits. Hybrid variational algorithms are well-suited to near-term quantum devices because they allow for a wide range of tradeoffs between the amount of quantum and classical resources used to solve a problem. This paper investigates tradeoffs available at both the algorithmic and hardware levels by studying a specific case — applying the Quantum Approximate Optimization Algorithm (QAOA) to instances of the Maximum Independent Set (MIS) problem. We consider three variants of the QAOA which offer different tradeoffs at the algorithmic level in terms of their required number of classical parameters, quantum gates, and iterations of classical optimization needed. Since MIS is a constrained combinatorial optimization problem, the QAOA must respect the problem constraints. This can be accomplished by using many multi-controlled gate operations which must be decomposed into gates executable by the target hardware. We study the tradeoffs available at this hardware level, combining the gate fidelities and decomposition efficiencies of different native gate sets into a single metric called the *gate decomposition cost*.

Teague Tomesh: teague.tomesh@infleqtion.com
Zain H. Saleem: zsaleem@anl.gov

1 Introduction

Today's quantum computing platforms are capable of executing single- and two-qubit gates with fidelities that have steadily improved year over year Gambetta et al. [2017], Wright et al. [2019], Arute et al. [2019], Arrazola et al. [2021]. However, implementing multi-qubit operations between three or more qubits presents significant challenges, although some progress has been made in this direction Monz et al. [2009], Isenhower et al. [2011], and it remains unclear what level of fidelity must be achieved before multi-qubit operations may outperform one- and two-qubit operations when executing a quantum algorithm.

The limitations of today's Noisy Intermediate-Scale Quantum (NISQ) computers severely restrict the set of executable quantum algorithms Preskill [2018]. Hybrid variational algorithms are a class of applications which are well-suited to NISQ systems because they make use of both quantum and classical resources when solving a problem, and this allows the quantum computer to focus on a single, well-defined task Cerezo et al. [2021]. For a given computational workload, a variational algorithm may make algorithmic level tradeoffs (e.g., varying the number of classical parameters, the size of the variational ansatz, or iterations of classical optimization) that exchange quantum and classical resources. Furthermore, current NISQ computers support a wide variety of native gate sets which enables different choices for decomposing the algorithmic level circuits into hardware executables Murali et al. [2019].

The Quantum Approximate Optimization Algorithm (QAOA) Farhi et al. [2014] is a hy-

brid variational algorithm that solves both constrained and unconstrained combinatorial optimization problems. For constrained combinatorial optimization problems there are two methods for dealing with the added problem constraints. One choice is to convert the constrained problem into an unconstrained one by introducing a Lagrange multiplier into the objective function. This can then be executed on quantum hardware using only single- and two-qubit gates, however, this method often requires an additional hyper-parameter optimization step in order to correctly set the value of the Lagrange multiplier Saleem et al. [2023]. Alternatively, the constraint may be introduced within the ansatz itself which then requires the use of multi-controlled quantum gates Hadfield [2018], Hadfield et al. [2019], Saleem [2020]. The constraint-aware ansatz ensures that the constraints are satisfied at all times during the optimization and the extremization is only performed over the set of feasible states. In this paper we focus solely on the latter case which uses the constraint-aware ansatz and therefore requires the use of multi-controlled gates.

Before the QAOA may be executed, the algorithmically defined multi-qubit gates must be decomposed into equivalent sets of gate operations which are natively supported by the quantum hardware. Depending on the specific native gate set, the final overhead gate cost incurred by the decomposition may vary greatly. Quantum gate decompositions are a broad and extensive area of study Barenco et al. [1995], Vartiainen et al. [2004], Martinez et al. [2016], Gokhale et al. [2019], Rakyta and Zimborás [2022]. Strategies for general unitary decompositions include algorithmic rule-based constructions Barenco et al. [1995], Vartiainen et al. [2004], Li et al. [2013], direct synthesis via numerical optimization Younis et al. [2021], and optimal control Shi et al. [2019]. Oftentimes the multi-qubit gates appearing in the constrained QAOA contain a large amount of structure (e.g., they may be expressed simply as sets of single-qubit rotations and multi-controlled NOT's) that allows for more efficient decompositions Barenco et al. [1995], Gokhale et al. [2019].

In this paper, we study the quantum and classical tradeoffs available to variational quantum algorithms, specifically focusing on the case of constrained QAOA applied to Maximum Independent Set (MIS) problems. We consider both algorithmic and hardware level design choices that influence the final amount of classical and quantum resources that are required.

At the algorithmic level, we consider three variants of the QAOA which differ in the number of classical parameters, rounds of optimization, and gate operations they require. In the original formulation of the QAOA, the ansatz is composed of alternating layers of parameterized gates and all of the gates within a single layer are parameterized by the same variable. We refer to this version as single-angle QAOA (SA-QAOA). Recent work has shown that, for both unconstrained optimization problems, such as MaxCut, and constrained problems, such as MIS, parameterizing the gates within each layer by a vector of variables can result in higher approximation ratios and shorter circuits at the expense of additional classical parameters Herrman et al. [2022], Saleem et al. [2023]. We refer to this implementation of the QAOA as multi-angle QAOA (MA-QAOA). Finally, the number of multi-controlled quantum gates in both the SA-QAOA and MA-QAOA scales linearly with the problem size. To mitigate this large quantum resource requirement, a new approach called the Dynamic Quantum Variational Ansatz (DQVA) was introduced in Saleem et al. [2023]. The DQVA removes this scaling dependency on the problem size by making the number of parameters and multi-controlled gates a tunable hyper-parameter in exchange for additional iterations of classical optimization.

At the hardware level, we study the cost of decomposing the large multi-qubit gates into different native gate sets containing different levels of controlled gates and higher dimensional qudits. Our consideration of native gate sets with higher degree multi-qubit gates is motivated by the recent emergence of quantum hardware capable of supporting such operations. These include neutral atoms Isenhower et al. [2011], Saffman [2016], trapped ions Espinoza et al. [2021], Katz et al. [2022], Rasmussen et al. [2020], and recent demonstrations of Toffoli gates in superconducting quantum hardware Kim et al. [2022]. Simultaneously, multiple experiments have demonstrated the viability of using *qutrits*, containing three logical states $|0\rangle$, $|1\rangle$, and $|2\rangle$, within quantum computations Morvan et al. [2021], Gokhale et al. [2020]. Prior work has already shown that more expressive gate sets, containing mul-

tiple two-qubit entangling operations from different gate families, are beneficial for the compilation of quantum applications Lao et al. [2021]. We extend this intuition here by studying the impact that natively supported multi-controlled gates and three-level qutrits can have on the decomposition overheads for the QAOA.

In summary, our contributions include:

- Numerical simulations of the QAOA on 20-node graphs, highlighting the tradeoffs that can be made between classical and quantum resources while achieving similar performance amongst the three QAOA variants.
- Asymptotic gate counts with exact scaling coefficients for the decomposition of the multi-controlled single-qubit rotations appearing in the constrained QAOA circuits. We consider scenarios where varying numbers of ancillae and different native gate sets are made available for the decompositions.
- We introduce a new metric, the gate decomposition cost, which shows that we can use lower fidelity native gates in one QAOA to outperform the higher fidelity gates in another.

The goal of this paper is to aid future system designs by calculating the fidelity of native multi-qubit (or qutrit) gates that is necessary to achieve performance improvements over the typical two-qubit gates for realistic workloads.

The rest of the paper is organized as follows, we begin in Section 2 by introducing the different flavors of the constrained QAOA applied to the MIS problem. Section 3 describes the different decompositions of multi-controlled single-qubit rotations. In Section 4 we compare the variants of the constrained QAOA and perform resource analysis. Finally, in Section 5 we summarize our conclusions and suggest future directions.

2 Quantum Approximate Optimization for Maximum Independent Set

The QAOA is a hybrid quantum-classical algorithm developed for the purpose of finding approximate solutions to combinatorial optimization problems. The objective of the QAOA is to prepare the ground state of a problem-dependent quantum operator using a variational ansatz, i.e., a parameterized quantum circuit. This quantum operator is defined on n-bit strings and is diagonal in the computational basis

Algorithm	**Number of parameters, $\lvert\boldsymbol{\theta}\rvert$**	**Rounds of variational optimization**
SA-QAOA	Constant, $\lvert\boldsymbol{\theta}\rvert = 2p$	Single
MA-QAOA	Scales with problem size, $\lvert\boldsymbol{\theta}\rvert =$ up to $2pn$	Single
DQVA	Tunable with problem size, $\lvert\boldsymbol{\theta}\rvert = \nu$	Multiple

Table 1: The three flavors of the QAOA considered in this work. Each makes different tradeoffs with respect to the classical (e.g., number of parameters, rounds of optimization) and quantum (e.g., circuit depth) resources they require. We consider the Maximum Independent Set (MIS) problem over graphs with n nodes.

$$C_{obj}|b\rangle = C(\mathbf{b})|b\rangle \tag{1}$$

where $\mathbf{b} = \{b_1, b_2, b_3 \dots b_n\} \in \{0,1\}^n$. The expectation value of this operator is measured on a quantum computer after preparing the state

$$|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle = e^{-i\beta_p M} e^{-i\gamma_p C} \cdots e^{-i\beta_1 M} e^{-i\gamma_1 C} |s\rangle \tag{2}$$

where $|s\rangle$ is the initial state, $e^{i\gamma C}$ is the phase separator unitary, diagonal in computational basis, $e^{i\beta M}$ is the mixing unitary that mixes the state among one another during optimization and p controls the number of times the unitary operators are applied. The expectation value of C_{obj} with respect to this variational state,

$$E_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})|C_{obj}|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle, \tag{3}$$

is evaluated on a quantum computer before being passed to a classical optimizer which attempts to find the optimal parameters which maximize $E_p(\boldsymbol{\gamma},\boldsymbol{\beta})$. This maximization is achieved for the states corresponding to the optimal solutions of the original optimization problem.

2.1 Unconstrained QAOA

The Maximum Independent Set (MIS) problem involves finding the largest set of nodes in a graph such that no two nodes are connected to each other. Only a subset of all possible solutions (i.e., all n-bit strings) are feasible solutions which satisfy the problem constraints. However, we may still apply the QAOA to this problem by adding a penalty term to the objective function

$$C_{obj} = H - \lambda\, C_{pen} = \sum_{i\in V} b_i - \lambda \sum_{i,j\in E} b_i b_j, \quad (4)$$

and

$$b_i = \frac{1}{2}\left(1 - Z_i\right) \quad (5)$$

where V and E are the set of all the nodes and edges in the graph, Z_i is the Pauli-Z operator acting on the ith qubit and λ is the Lagrange multiplier. The first term H is the Hamming weight of the bit string and the second term C_{pen} is the penalty term which penalizes every time two nodes are connected by an edge. This objective function defines an unconstrained optimization problem which can be solved by the QAOA and can be implemented using only single- and two-qubit gates, but it has the disadvantage that the optimization is performed over both feasible and infeasible states. Finding the optimal value for λ, which effectively balances the trade-off between optimality and feasibility in the objective function, can pose a considerable challenge.

2.2 Constrained QAOA

In constrained versions of the QAOA the structure of the ansatz is altered to ensure that the variational optimization takes place only over the set of feasible solutions Hadfield [2018], Hadfield et al. [2019], whereas the Lagrange multiplier method includes optimization over a larger set of states that may not correspond to an independent set. Since the problem constraints are handled at the circuit level, the objective function is simply the Hamming weight operator,

$$C_{obj} = H = \sum_{i\in V} b_i. \quad (6)$$

The initial state may be any feasible state or a superposition of feasible states. The phase separator unitary $U_C(\gamma) := e^{i\gamma H}$ is determined by the objective function. The mixing unitary is specially designed to ensure that the "independent set" constraint is maintained at all times during optimization. Let $U_M(\beta) := \prod_j e^{i\beta M_j}$, where $M_j = X_j \bar{B}$ and we have defined

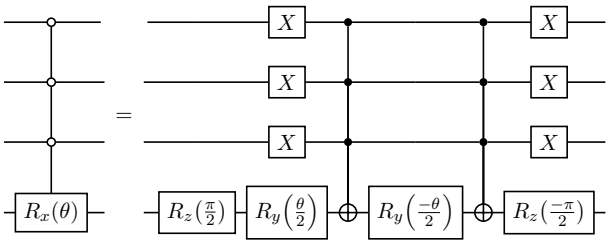

Figure 1: Decomposition of the partial mixers used in the constrained QAOA for MIS into two multi-controlled Toffolis and single-qubit rotations Barenco et al. [1995]. The partial mixer enforces the MIS constraint by only performing the single-qubit rotation if all neighbors are in the $|0\rangle$ state.

$$\bar{B} := \prod_{k=1}^{\ell} \bar{b}_{v_k}, \quad \bar{b}_{v_k} = \frac{1 + Z_{v_k}}{2}, \quad (7)$$

where v_k are the neighbors and ℓ is the number of neighbors for the kth node. We can also write the mixer as

$$\begin{aligned} U_M(\beta) &= \prod_{j=1}^{N} V_j(\beta) \\ &= \prod_{j=1}^{N} \left(I + \left(e^{-i\beta X_j} - I\right) \bar{B}\right), \end{aligned} \quad (8)$$

where we have used $\bar{b}_{v_j}^2 = \bar{b}_{v_j}$. The unitary mixer above is a product of N partial mixers V_i, and in general they may not all commute with each other $[V_i, V_j] \neq 0$. The partial mixers are executed on a digital quantum computer using the circuit shown in Figure 1.

Additionally, the mixing unitary may be parameterized by a list of angles $\boldsymbol{\beta}$, such that each partial mixer can have a different classical variable as a parameter, and is defined up to a permutation of the partial mixers:

$$U_M(\boldsymbol{\beta}) \simeq \mathcal{P}\left(V_1(\beta_1) V_2(\beta_2) \cdots V_N(\beta_N)\right). \quad (9)$$

To distinguish between the QAOA whose mixing unitary has a single angle, Eq. (8), or multiple angles, Eq. (9), we denote these cases as SA-QAOA and MA-QAOA, respectively.

2.3 Dynamic Quantum Variational Ansatz

An extension of the MA-QAOA, called the Dynamic Quantum Variational Ansatz (DQVA),

was introduced in Saleem et al. [2023] which enables a tradeoff between the number of partial mixers and rounds of classical optimization. The DQVA may be initialized in any superposition of feasible states, potentially even using the output of a classical optimization algorithm $|\mathbf{c}\rangle = |c_1 c_2 \cdots c_n\rangle$. This warm-start approach was recently proposed for Max-Cut and QUBOs in Egger et al. [2021]. Then, the phase separator and mixing unitaries are applied to construct the state,

$$\psi(\boldsymbol{\gamma}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p) = U_C(\gamma_p) U_M^{\sigma_p}(\boldsymbol{\alpha}_p) \cdots U_C(\gamma_1) U_M^{\sigma_1}(\boldsymbol{\alpha}_1) |\mathbf{c}\rangle \tag{10}$$

where, similarly to Eq. (9), each of the individual mixers is given by

$$U_M^{\sigma_k}(\boldsymbol{\alpha}_k) = V_{\sigma_k(n)}\left(\alpha_k^{\sigma_k(n)}\right) \cdots V_{\sigma_k(1)}\left(\alpha_k^{\sigma_k(1)}\right) \tag{11}$$

for $k = 1, 2, \cdots, p$. Here for simplicity we have fixed all permutations $\sigma_1 = \sigma_2 = \cdots = \sigma_p$, and henceforth drop the permutation index on mixers.

In Saleem et al. [2023] it was found that the increased expressibility of Eq. (11) resulted in improved performance with smaller circuit depth on constrained optimization problems, and in Herman et al. [2022] similar results were found for unconstrained problems. The increased expressibility of the unitary mixer has further advantages. We can set some of the partial mixers "off" by setting $\alpha_i = 0$, effectively reducing the quantum resources required by the ansatz. In this paper, we control the number of nonzero parameters in the variational ansatz, including both the phase separator γ and mixer $\boldsymbol{\beta}$ parameters, using a hyperparameter ν.

The final component of the DQVA algorithm is the dynamic ansatz update. This is a classical outer loop where in every iteration, called a *mixer round*, the order of the partial mixers appearing in Eq. (11) is randomly permuted. Each mixer round is composed of multiple *inner rounds* where the ansatz is initialized with the best independent set found thus far, variationally optimized, and then uses the output of that optimization as the input for the next inner round. As more and more nodes become part of the independent set, the ansatz is continually updated as detailed in Saleem et al. [2023] until the entire graph has been traversed, increasing the size of the independent set at each iteration.

The DQVA algorithm has previously been used in the quantum local search (QLS) approach in Tomesh et al. [2022a]. In QLS the DQVA is applied to local neighbourhoods and the whole graph is traversed neighbourhood by neighbourhood, increasing the size of the independent set at each step. The DQVA was also applied in the quantum divide and conquer approach in Tomesh et al. [2023] in which the graph is first partitioned and then the DQVA is implemented on each of the partitions separately. This allows the potential for distributed computing as well since each of the partitions can be run in parallel. The analysis of multi-controlled gate decompositions presented in this work is complimentary to these higher-level algorithmic approaches — enabling more accurate resource estimates for these algorithms under different assumptions of the underlying native gate set.

3 Multi-controlled Gate Decompositions

Many quantum algorithms are expressed using multi-controlled quantum gates Draper et al. [2004], Chakrabarti and Sur-Kolay [2008], Arrazola et al. [2022], Crow et al. [2016], Perlin et al. [2023]. However, most currently available hardware platforms, like IBM or Rigetti's superconducting systems Murali et al. [2019] or IonQ's trapped ion systems IonQ [2022], only expose single- and two-qubit primitive operations to the user. Thus, to execute large quantum gates on physical hardware, one needs to use a sequence of single- and two-qubit gates whose composite action is equivalent to the larger gate.

Methods for deriving these gate sequences are known as gate decomposition schemes. By imposing various restrictions on the physical system available to us (e.g. native gate set, number of available ancilla qubits), we can analyze the resources required to implement various operations by analyzing the gate complexity of these decompositions. We can then compare the resource efficiency between different algorithms or different physical systems, building upon the decomposition schemes given by Barenco *et al.* in Barenco et al. [1995].

3.1 Definitions and notation

In what follows, we denote a k-controlled U gate by $C^k(U)$. We can then define the gate set $\mathcal{S}_n^d$ to be the collection of quantum gates on d-level qudits given by

$$\mathcal{S}_n^d = \left\{U(d), C(X), C^2(X), \ldots, C^{n-1}(X)\right\}, \tag{12}$$

consisting of single qudit gates and every multi-controlled NOT from one control to $(n-1)$ controls. The gates contained in any gate set are known as basis gates. We make the assumption that the available native gates on the hardware include only these generalized Toffoli gates, whereas, in reality, actual hardware may employ a variety of gates such as CPHASE gates, $\sqrt{\text{SWAP}}$ gates, cross-resonance gates, etc. Fortunately, this assumption does not usually pose a problem since any two-qubit entangling gate can be simulated by another two-qubit entangling gate with a constant overhead. The primary gate sets we consider are:

1. $\mathcal{S}_2^2 = \{U(2), C(X)\}$, the simplest universal gate set, and representative of what most architectures are capable of.
2. $\mathcal{S}_3^2 = \{U(2), C(X), C^2(X)\}$, which includes native Toffoli gates.
3. $\mathcal{S}_2^3 = \{U(3), C(X_{+1}), C(X_{-1})\}$, which consists of single- and two-*qutrit* gates. See Gokhale et al. [2019] for further background information.

When we consider the following gate decomposition schemes, we will generally assume that the ancilla qubits available for every computation are zeroed – meaning that they start initialized to the $|0\rangle$ state, and they are required to be returned to the $|0\rangle$ state after each decomposition. This is to avoid the experimental overhead of needing to re-initialize the ancilla qubits back to $|0\rangle$ in the middle of a computation. Here, we focus on the decomposition schemes for a $C^n(U)$ gate with access to either one or n ancilla qubits.

3.2 Gate counts for decompositions

In all three of the variational algorithms we will focus on – SA-QAOA, MA-QAOA, and DQVA – the only multi-qubit entangling gate needed is the multi-controlled rotation gate, i.e. the $C^n(R_x(\theta))$ gate, which we use to encode the constraints of the problem. For MIS, we use controlled rotations to encode information about the edges within the input graph.

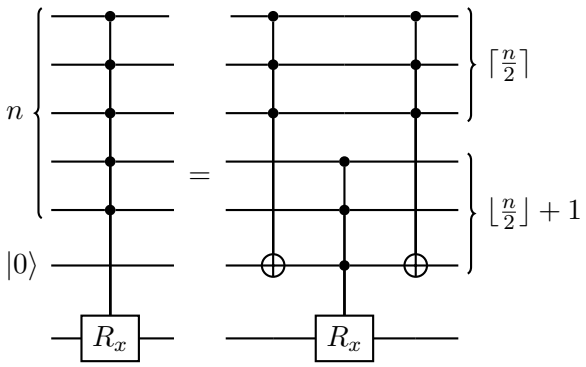

Figure 2: Splitting a multi-controlled gate with one zeroed ancilla into three multi-controlled gates with half the number of controls Gidney [2015a].

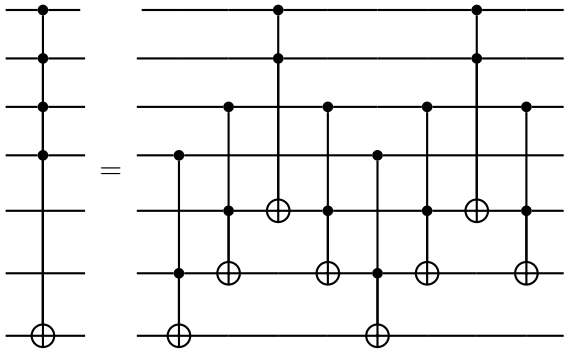

Figure 3: Decomposition scheme for a multi-controlled X gate with access to two borrowed ancilla qubits that get returned to the state $|0\rangle$ Barenco et al. [1995].

In what follows, we use decomposition schemes from Barenco et al. [1995] and Iten et al. [2016] to give the total number of entangling gates required to decompose a $C^n(R_x(\theta))$ gate. Since the rotation gate is a special unitary operator $R_x(\theta) \in SU(2)$, we can make use of simpler decomposition schemes than would be possible for a general single-qubit gate U. We will now give a general overview of the decomposition schemes used and show how the entangling gate counts were calculated.

3.2.1 Decompositions with one ancilla

Our first step in the decomposition process will be to split the single multi-control gate with n controls into three multi-control gates with $n/2$ controls, following the zeroed bit construction from

	One ancilla		n ancillae		No ancilla
	$\mathcal{S}_2^2$	$\mathcal{S}_3^2$	$\mathcal{S}_2^2$	$\mathcal{S}_3^2$	$\mathcal{S}_2^3$
$n=1$	2	2	2	2	2
$n=2$	6	2	6	2	4
$n=3$	18	4	18	4	14
$n=4$	42	10	24	6	16
$n=5$	72	16	30	8	26
$\vdots$	$16n-8$	$8n-24$	$6n$	$2n-2$	$6n-4$ (n odd)
	($n \geq 5$)		($n \geq 3$)		$6n-8$ (n even)

Table 2: Exact entangling gate counts for a $C^n(R_x)$ gate given access to zeroed ancilla using the decomposition schemes described in Section 3.

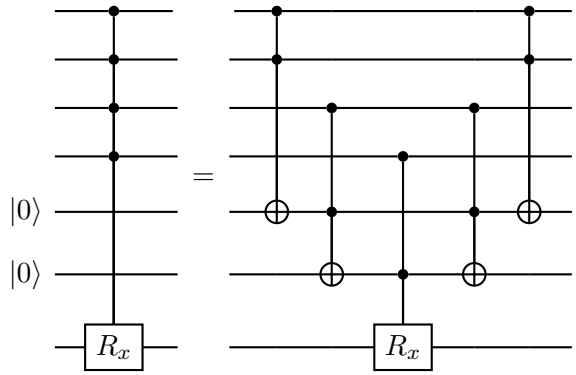

Figure 4: Decomposing a multi-controlled $R_x \in SU(2)$ gate with access to two zeroed ancilla qubits.

Gidney [2015a]. This step is illustrated for $n = 5$ in Figure 2.

Next, as was shown in Figure 1, since $R_x \in SU(2)$ we can split up the resulting $C^{\lfloor n/2 \rfloor+1}(R_x)$ gate into two multi-controlled Toffolis and four single qubit gates following Lemma 5.1 of Barenco et al. [1995].

Then, we can decompose the resulting $C^{\lceil n/2 \rceil}(X)$ gates into a series of Toffoli gates by borrowing ancillae from the set of qubits not participating in the current gate, as described in Lemma 7.2 of Barenco et al. [1995] and shown in Figure 3.

From this decomposition we can derive the asymptotic entangling gate counts. First, we consider the case where we have access to native Toffoli gates (i.e. $\mathcal{S}_3^2$). Working backwards, the decomposition scheme shown in Figure 3 with k controls requires $4k - 8$ Toffolis to implement, since there are 4 Toffoli gates for each of the $k-2$ controls. Then the decomposition in Figure 2 followed by the decomposition in Figure 1 gives two Toffolis with $\lceil n/2 \rceil$ controls and two Toffolis with $\lfloor n/2 \rfloor + 1$ controls, so in total the number of entangling gates used is

$$\begin{aligned} 2\left(4\left\lceil \frac{n}{2} \right\rceil - 8\right) + 2\left(4\left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right) - 8\right) \\ = 8(n+1) - 32 \\ = 8n - 24, \end{aligned} \tag{13}$$

for $n \geq 5$. For $n < 5$, there are too few controls for the decomposition scheme to work, but these cases can be calculated by hand.

Next, we consider the case where we do not have access to native Toffoli gates (i.e. $\mathcal{S}_2^2$). The decomposition scheme shown in Figure 3 with k controls instead requires $8k - 6$ CNOT's to implement (see Lemma 8 of Iten et al. [2016]) and then the same algebra as above gives an entangling gate count of $16n - 8$, again for $n \geq 5$. These results, along with the base cases of $n < 5$ are shown in Table 2.

3.2.2 Decompositions with n ancillae

If we have access to many ancilla qubits – i.e. roughly the same number as there are controls – then we can use an even simpler decomposition scheme, where we decompose directly into 2-controlled Toffoli gates. The scheme is similar to that shown in Figure 3, and is shown in Figure 4.

We can then decompose the $C^2(R_x)$ gate in Figure 4 into four single qubit gates and two Toffoli gates. Then, directly counting the number of

Gate set	# of ancillae	$C^n(R_x)$	$C^n(X)$
$\mathcal{S}_2^2$	No ancillae	$(28n-24, 28n-60)$	—
	One ancilla	$(16n+20, 16n-6)$	$(8n+8, 8n-4)$
	n ancillae	$(8n-8, 6n-6)$	—
$\mathcal{S}_3^2$	No ancillae	$(12, 6, 14n-38)$	—
	One ancilla	$(4, 2, 8n-24)$	$(0, 0, 4n-12)$
	n ancillae	$(6, 2, n-2)$	$(0, 0, n-1)$
$\mathcal{S}_m^2$	No ancillae	$\sim\left(\frac{16}{m-2}\right)n$ C^{m-1}NOT's	—
	One ancilla	$\sim\left(\frac{8}{m-2}\right)n$ C^{m-1}NOT's	$\sim\left(\frac{4}{m-2}\right)n$ C^{m-1}NOT's
	n ancillae	$\sim\left(\frac{1}{m-2}\right)n$ C^{m-1}NOT's	$\sim\left(\frac{1}{m-2}\right)n$ C^{m-1}NOT's
$\mathcal{S}_2^3$	No ancillae	$(10n-10, 6n-4)$	—

Table 3: Asymptotic gate counts for decompositions into different gate sets with differing amounts of available ancillae. The i-th entry in the ordered tuple shows how many i-qudit basis gates are required in $\mathcal{S}_m^d$ to decompose that type of gate (e.g. (a, b) would require a single qubit gates and b CNOT's). Cells with no entry have no better decomposition than the equivalent $C^n(R_x)$ decomposition in the same row. For schemes with gate counts that depend on the parity of n, we simply report the larger gate count. Note that the asymptotic counts reported here consider *burnable* ancillae and may differ from the counts given in Table 2 which considered *zeroed* ancillae. The derivation of the gate counts is contained in Appendix B.

Toffoli gates used for the $\mathcal{S}_3^2$ count, there are two Toffoli gates for each of the $n-1$ controls, giving $2n-2$ Toffoli gates required for $n \geq 3$. To get the $\mathcal{S}_2^2$ count, we use the tricks shown in the proof of Lemma 8 of Iten et al. [2016] to lower the number of CNOT's required to only $6n$ for $n \geq 3$. These results, along with the base cases, are also shown in Table 2.

3.2.3 Additional counts

Using similar methodology to that shown above, one can derive exact gate counts for decomposing such gates with access to burnable ancilla – meaning that the ancilla qubits are initialized to $|0\rangle$ and do not need to be returned to this state – we have carried this analysis out, following the decomposition schemes given in Barenco et al. [1995], Gidney [2015a,b,c], with the results summarized in Table 3. The derivations of the gate counts can be found in Appendix B as they may prove useful for future reference.

We also include the asymptotic gate counts for the $\mathcal{S}_2^3$ gate set, which includes single- and two-qutrit gates, based on the decomposition schemes given in Gokhale et al. [2019]. These decompositions require no ancillae but we include them in our comparisons in Section 4 as an example of an interesting tradeoff between resource efficiency and the complexity of physical implementation. Intuitively, the qutrits (with basis states $|0\rangle, |1\rangle,$ and $|2\rangle$) can use the extra $|2\rangle$ state as an efficient scratch space to store parity information. This enables the $\mathcal{S}_2^3$ gate set to achieve the same asymptotic cost as the $\mathcal{S}_2^2$ gate set *with n ancillae* when decomposing multi-controlled R_x and X gates.

3.3 Fidelity thresholds

We can also use the results from Table 3 to derive gate set thresholds – i.e. under what conditions it is asymptotically beneficial to decompose into one gate set over another. For example, if the i-qubit basis gate of $\mathcal{S}_m$ has gate fidelity $0 \leq \mathcal{F}_i \leq 1$ (e.g. $\mathcal{F}_2$ is the CNOT fidelity, etc.), then by comparing the one ancilla decompositions for $\mathcal{S}_2^2$ and $\mathcal{S}_3^2$ in Table 3, at large n the process fidelity for a $C^n(R_x)$ gate in $\mathcal{S}_2^2$ is $\mathcal{F}_1^{16n}\mathcal{F}_2^{16n}$ while the process fidelity in $\mathcal{S}_3^2$ is $\mathcal{F}_3^{8n}$. In order for $\mathcal{S}_3^2$ to

Threshold	Requirement	$\mathcal{F}_1 = 95\%$ $\mathcal{F}_2 = 90\%$	$\mathcal{F}_1 = 99\%$ $\mathcal{F}_2 = 95\%$	$\mathcal{F}_1 = 99.9\%$ $\mathcal{F}_2 = 99\%$	$\mathcal{F}_1 = 99.99\%$ $\mathcal{F}_2 = 99.9\%$	$\mathcal{F}_1 = 99.999\%$ $\mathcal{F}_2 = 99.99\%$
$\mathcal{S}_3^2 \succ \mathcal{S}_2^2$	$\mathcal{F}_3 > \mathcal{F}_1^2\mathcal{F}_2^2$	73%	88%	97.8%	99.78%	99.978%
$\mathcal{S}_4^2 \succ \mathcal{S}_3^2$	$\mathcal{F}_4 > \mathcal{F}_3^2$	53%	78%	95.7%	99.56%	99.956%
$\mathcal{S}_5^2 \succ \mathcal{S}_4^2$	$\mathcal{F}_5^2 > \mathcal{F}_4^3$	39%	69%	93.6%	99.34%	99.934%
$\mathcal{S}_6^2 \succ \mathcal{S}_5^2$	$\mathcal{F}_6^3 > \mathcal{F}_5^4$	29%	61%	91.2%	99.12%	99.912%
$\mathcal{S}_7^2 \succ \mathcal{S}_6^2$	$\mathcal{F}_7^4 > \mathcal{F}_6^5$	21%	54%	89.6%	98.91%	99.890%
$\mathcal{S}_8^2 \succ \mathcal{S}_7^2$	$\mathcal{F}_8^5 > \mathcal{F}_7^6$	15%	48%	87.6%	98.69%	99.868%

Table 4: Summary of gate set thresholds, their gate fidelity requirements, and the required largest gate fidelity under various assumptions for $\mathcal{F}_1$ and $\mathcal{F}_2$, the one- and two-qubit gate fidelities, respectively. For larger gate sets, we assume that each of the smaller gates just meet the fidelity threshold (for example, to calculate the threshold for $\mathcal{F}_5$, we assume recursively that $\mathcal{F}_3$ and $\mathcal{F}_4$ equal their respective threshold values). The center column with values of $\mathcal{F}_1 \approx 99.9\%$ and $\mathcal{F}_2 \approx 99\%$ is about where current NISQ technologies stand Tomesh et al. [2022b], AI [2023].

be better than $\mathcal{S}_2^2$, we must have

$$\mathcal{F}_3^{8n} > \mathcal{F}_1^{16n}\mathcal{F}_2^{16n} \implies \mathcal{F}_3 > \mathcal{F}_1^2\mathcal{F}_2^2.$$

In other words, in order for it to be asymptotically beneficial to stop decompositions early using native Toffolis, the Toffoli fidelity must be greater than the product of the squares of the single qubit and CNOT fidelities.

A summary of these results, as well as some experimental targets for various gate fidelity estimates, is given in Table 4. The central column, where $\mathcal{F}_1 \approx 99.9\%$ and $\mathcal{F}_2 \approx 99\%$, most closely represents the capabilities of many current NISQ computers Tomesh et al. [2022b], AI [2023]. Using these systems as an example, if they natively supported a three-qubit gate with fidelity $\mathcal{F}_3 > 97.8\%$, it would be asymptotically advantageous to decompose large multi-controlled gates into $\mathcal{S}_3^2$ instead of $\mathcal{S}_2^2$. This agrees with other theoretical results and simulations that suggest that such gate set optimizations would be possible, for example in Gulliksen et al. [2015].

4 Analyzing Classical and Quantum Resource Tradeoffs

Hybrid quantum-classical algorithms offer architectural tradeoffs which allow both the CPU and QPU to share the burden of a computation. For a given instance of a target problem (e.g., solving MIS on a specific graph) the exact value of these tradeoffs that will maximize performance depends on the characteristics of the specific algorithm and hardware that are used. In this section we consider the impact that the number of variational parameters, number of rounds of optimization, and native gate set supported by the hardware backend have on the final performance, runtime, and circuit depth for three flavors of the QAOA. We begin by examining the tradeoff between the size of the independent sets produced by the algorithms on a suite of benchmark graphs and the classical computational resources they require which includes the number of variational parameters and the number of rounds of classical optimization. We then compare the number of quantum operations required by the different algorithms as a function of the native gate set supported by hardware.

4.1 Performance vs. Classical Resources

We consider three variations of the QAOA: SA-QAOA, MA-QAOA, and DQVA as defined in Section 2. Figure 5 shows the maximum, minimum, and average approximation ratios achieved by the three algorithms as a function of the number of classical parameters used within each variational ansatz. We simulated the execution of each algorithm on both 3-regular and Erdös-Rényi graphs containing 20 nodes. For the Erdös-Rényi graphs the probability of any edge existing between two nodes was set to 50%. An open-source implementation of the code used to construct the quantum circuits and simulate the execution of the algorithms is available on Github Tomesh [2022].

The simulation results were collected over ensembles of 50 benchmark graphs and for each graph the best performing execution over 30 repetitions was saved. Prior work has shown that the number of trials needed to converge to good

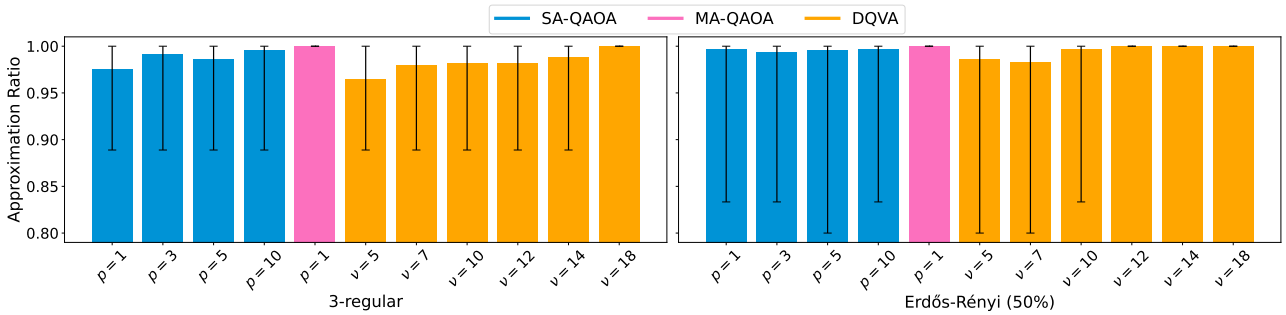

Figure 5: The average approximation ratio attained by each QAOA variant with respect to the optimal MIS over 50 20-node benchmark graphs. For each benchmark graph, every algorithm is executed 30 times and the best performing run is saved. The error bars show the maximum and minimum approximation ratio over the 50 benchmark graphs. Each algorithm is able to find the optimal solution at least once. This is even true for some algorithms when the number of parameters is much less than the size of the graph.

solutions can scale strongly with the number of paramters. For example, while studying parameter transfer for solving MaxCut problems with QAOA, Shaydulin et al. [2023] allocated 50, 200, and 1,500 trials for $p = 1, 2, 3$ SA-QAOA, respectively. In this work our aim is to analyze the tradeoff between performance – as measured by the final approximation ratio – and the classical resources required to execute the QAOA. An equal-repetition comparison helps to capture this tradeoff and serves to represent the difficulty of optimizing ansatzes with an increasing number of classical parameters. Indeed, Figure 5 shows that each algorithm is able attain comparably high approximation ratios so long as the number of classical parameters is not too small.

Even though each algorithm is evaluated on a target graph for an equal number of trials, the classical resources required within a single trial is also variable. This is qualitatively shown in Table 1 where both the SA-QAOA and MA-QAOA require a single round of classical optimization but the DQVA can require multiple. Figure 6 quantitatively illustrates this point by showing the average number of rounds of classical optimization required until the DQVA reaches its final solution. Instances of the DQVA which use few parameters, $\nu < \frac{n}{2}$, require many additional rounds of optimization (between 4 to 9) and on average they find independent sets smaller than what the SA-QAOA and MA-QAOA are able to find in a single round of optimization as seen in Figure 5. However, for larger values of $\nu \gtrsim \frac{n}{2}$ comparable performance can be achieved with approximately 1 to 3 additional rounds of optimization. The DQVA balances this additional cost in classical resources by requiring quantum circuits which contain many fewer gates than the SA-QAOA and MA-QAOA as discussed below.

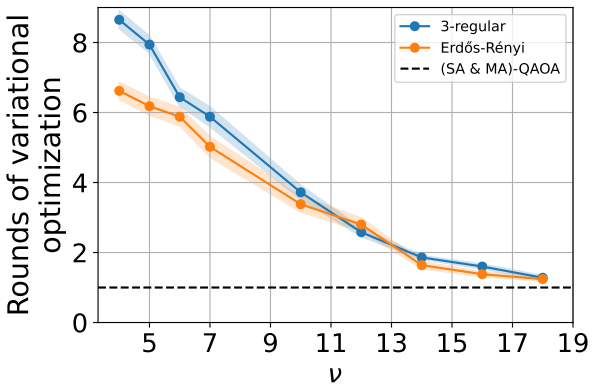

Figure 6: Average number of rounds of variational optimization incurred by the DQVA as a function of the number of parameters ν across both the 3-regular and Erdös-Rényi benchmark graphs. The shaded areas denote one standard error from the mean. DQVA makes a tradeoff between the classical and quantum resources needed to solve the MIS problem. Fewer parameters and partial mixers can be used at the cost of additional rounds of variational optimization. At $\nu \gtrsim 10$ the DQVA incurs $2-4$ rounds of optimization and finds approximation ratios comparable to the SA-QAOA and MA-QAOA.

4.2 Native Gate Set vs. Quantum Resources

The exact gate counts for the benchmark graphs used in Figure 5 are contained in Appendix A. To better understand the circuit sizes that would be encountered in more realistic workloads targeting larger graphs, we used the exact gate counts given in Table 2 to compute the number of entangling gates required across algorithms and na-

tive gate sets. Since variational circuit simulations quickly become intractable beyond 20 to 30 qubits, we were unable to verify the values of p and ν needed by SA-QAOA, MA-QAOA, and the DQVA, respectively, to ensure adequate performance as the graph size increased. Instead, we selected the $p = 10$ SA-QAOA, $p = 1$ MA-QAOA, and $\nu = \frac{n}{2}$ DQVA which showed comparable performance on the 20-node Erdös-Rényi graphs in Figure 5 and calculated the gate counts for these ansatzes when targeting average degree $d = 6$ Erdös-Rényi graphs with up to 640 nodes. Figure 7 shows the final number of entangling gates of each ansatz for the S_2^2, S_3^2, and S_2^3 gate sets. Additionally we report the cost for separate scenarios where only a single ancilla or n ancilla are available. Note that this only impacts the qubit (S_2^2 and S_3^2) decompositions because the qutrit S_2^3 decomposition requires no extra ancilla. In the future, we expect the n ancilla case to be more common as quantum processors scale to larger sizes since it is unlikely that every program will require 100% of the qubits available on a device Gokhale et al. [2019].

The impact of the constant factors in front of the asymptotic scaling given in Tables 2 and 3 appear clearly in Figure 7. The S_3^2 gate set consistently produces the shortest circuits across graph sizes and amount of available ancillae. Interestingly, the gate counts for the S_2^2, with n ancillae, and S_2^3, with no ancilla, decompositions are nearly equivalent.

4.3 Decomposition Cost

In order to compare the usefulness of the different decomposition methods, we define a metric in Equation (14) which we call the Gate Decomposition Cost (GDC).

$$\text{GDC} \coloneqq -\log\left(\prod_{i}^{\text{Types}} \mathcal{F}_i^{n_i}\right) \tag{14}$$

The GDC takes into account the number, n_i, and type (e.g., C^1X, C^2X, etc.) of basis gates that we decompose the large multi-controlled gates into and also takes into account their respective fidelities, $\mathcal{F}_i$.

This is a good metric that is generalizable beyond this paper since gate fidelity products are an effective approach in comparing the output of the quantum circuit to the ideal output when fidelities are all 1. Even though gate fidelities are not invariant for different physical setups, the GDC will always decrease for smaller fidelities and increase for larger ones. The costs and benefits of each decomposition will always be captured by the product in the logarithmic function. We can similarly define a Gate Decomposition Score (GDS) as 1-GDC to compare how well one circuit helps solve the MIS problem to another one with a different decomposition.

Figure 8 shows the average GDC of the different QAOA variants when targeting 100-node Erdös-Rényi graphs. Note that the gate fidelity given along the X-axis is assumed for every entangling operation within the gate set, however, in practice, we expect two-qubit gates to have higher fidelities than three-qubit gates for most hardware platforms. This assumption mainly impacts the GDC for the $\mathcal{S}_3^2$ gate set which produces both two- and three-qubit gates in the final circuit decompositions, but we expect the impact to be minimal as we observed that the number of three-qubit gates always dominated the two-qubit gate count by a factor of approximately 1000.

The GDC provides a useful means of combining the gate decomposition results with the gate fidelities we expect to see in future hardware. Comparing the different gate set decompositions of a particular QAOA variant along equal GDC lines indicates the easing fidelity requirements afforded by lower basis gate overheads. These numerical results confirm the asymptotic values provided in Table 4 and reflect the diminishing gains won by the $\mathcal{S}_3^2$ gate set as the quality of the two-qubit gates approaches unity.

5 Conclusions

Hybrid variational algorithms are a promising means for extending the capabilities of classical computation with noisy, intermediate-scale, quantum resources. While the practical utility of heuristic approaches such as the QAOA remains an open question, recent work, both theoretical Basso et al. [2021] and empirical Shaydulin et al. [2024], has provided interesting avenues for future investigation. For these applications, we consider tradeoffs at an algorithmic level, which exchange fewer quantum resources for additional classical computation. Concretely, we consider variants of

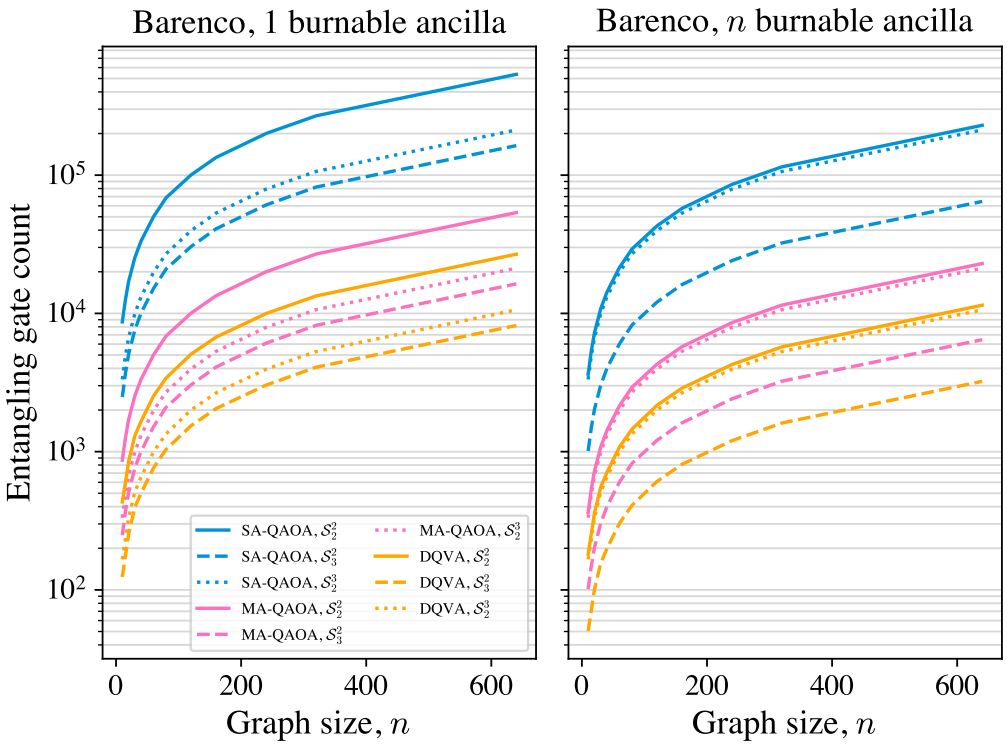

Figure 7: Total entangling gate count on Erdös-Rényi graphs with constant average density $d = 6$ (i.e., the edge probability is always set to $\frac{d}{m-1}$) for the $p = 10$ SA-QAOA (blue) with 20 variational parameters, $p = 1$ MA-QAOA (pink) with m variational parameters, and $\nu = \frac{m}{2}$ DQVA (orange). We compare the gate costs incurred by decomposing the ansatz into the $\mathcal{S}_2^2$ (solid) and $\mathcal{S}_3^2$ (dashed) gate sets with access to 1 and n burnable ancillae. We also include the gate overhead associated with decompositions targeting the S_2^3 (dotted) qutrit gate set. Note that the decompositions in this gate set require no ancilla and the same counts are shown in both panels. The S_3^2 gate set always produces the most efficient decompositions across all graph sizes. The plot on the right highlights the resource efficiency of qutrits — the S_2^3 gate set incurs a slightly smaller gate cost than the S_2^2 qubit gate set while requiring only half as many qutrits since its decomposition requires no ancilla.

the QAOA applied to the MIS problem and study how increasing or decreasing the number of parameters — which is proportional to the depth of the quantum circuits — affects performance. We find that the MA-QAOA, which parameterizes each individual partial mixer, outperforms the SA-QAOA while requiring much shorter quantum circuits. Furthermore, the DQVA extends this idea by allowing some of those individual parameters to be turned off. This enables the DQVA to further reduce the quantum cost compared to the MA-QAOA at the expense of additional iterations of classical optimization.

In addition to these algorithmic tradeoffs, we also consider the effect that the hardware's native gate set has on the overhead incurred during gate decomposition. We analyze the gate decomposition schemes introduced in prior work Barenco et al. [1995], Gokhale et al. [2019], and provide asymptotic as well as numerical gate counts specific to the multi-controlled R_x rotations which are ubiquitous in constrained QAOA for MIS. While our asymptotic gate counts show that all of the decompositions have the same $\mathcal{O}(n)$ scaling, our numerical calculations reveal the practical impact of the constant scaling factors as the size of the problem graph grows. For example, for the 600-node graphs, the gate counts for the same QAOA circuit under different decompositions may differ by as much as 1.6×10^5 entangling gates.

The asymptotic gate counts reported in Section 3 show the advantages, in terms of the gate decomposition overhead, of augmenting a quantum computer's typical single- and two-qubit gate set with higher-degree native gates as well as additional qudit states. This is reinforced by the numerical simulations in Section 4 which suggest that the gains in decomposition efficiency can outweigh the effects of lower-fidelity three-qubit

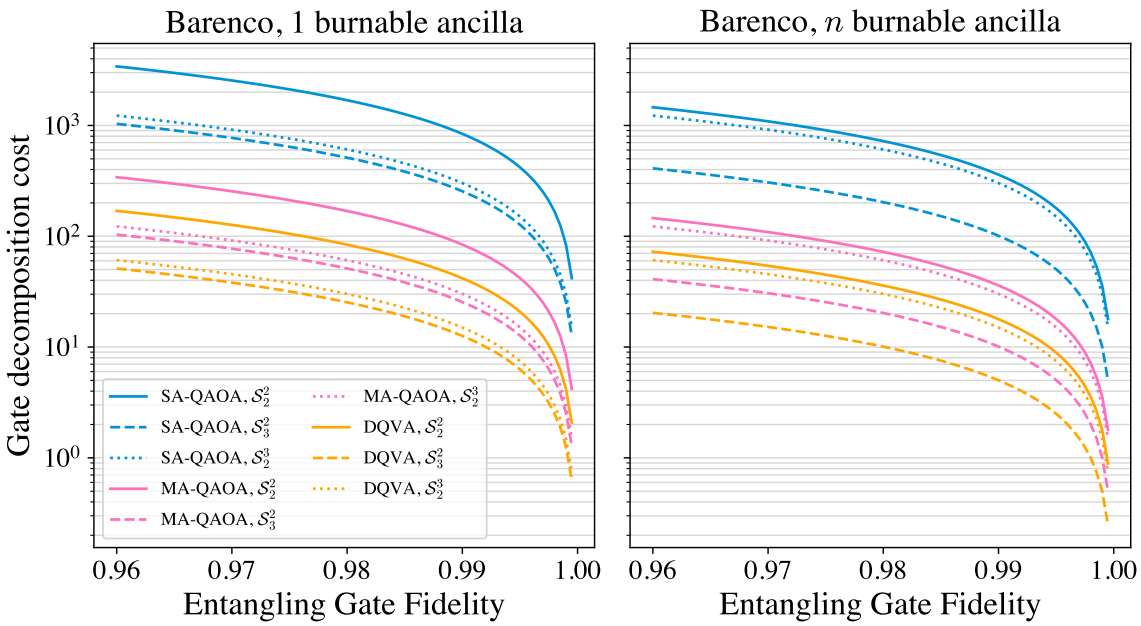

Figure 8: Gate decomposition cost (GDC) for the QAOA variants as a function of entangling gate fidelity. These values were computed using 100-node Erdös-Rényi graphs with an average density $d = 6$. The more efficient decompositions produced by the $\mathcal{S}_3^2$ and $\mathcal{S}_2^3$ gate sets allow for lower fidelity entangling gates while maintaining an equivalent GDC as compared to the $\mathcal{S}_2^2$ decompositions.

and two-qutrit gates.

However, our analysis of the tradeoffs in the decomposition overheads is not specific to this task and is applicable to many other quantum algorithms which rely on these gate operations including adders Draper et al. [2004], Chakrabarti and Sur-Kolay [2008], variational quantum eigensolvers with particle-conserving unitaries Arrazola et al. [2022], and measurement-free error correction Crow et al. [2016], Perlin et al. [2023].

The trainability across the three QAOA variants exhibits relatively minor variations. The primary distinction, and a focal point of our research, lies in the total count of quantum gates and optimization rounds required. Our work primarily concentrates on enhancing the implementation of these algorithms by furnishing system designers with clear fidelity targets. This aids in discerning the advantages of supporting more intricate gate operations, particularly those involving many-body or higher-dimensional scenarios. These contributions are seen as complementary to ongoing initiatives addressing trainability issues in Variational Quantum Algorithms (VQAs). By elevating hardware fidelity and lowering the GDCs, we anticipate a reduction in the overall number of shots required to measure the relevant operator for other optimization schemes, be it the gradient or any other operator.

Constrained combinatorial optimization problems, and particularly MIS, have widespread applications including portfolio optimization in finance Kalra et al. [2018], the logistics of job scheduling Lawler [1973], and the modeling of genomics data Auyeung [2005]. Obtaining accurate solutions to these problems in a time and cost efficient manner is a significant challenge. Many quantum-classical algorithms have been developed to target these problems, and they are often inspired by classical approaches such as divide and conquer and local search Tomesh et al. [2023, 2022a]. The analysis presented here is complementary with these algorithmic approaches as they use the same basic quantum gates.

6 Future Directions

Hardware experiments demonstrating the performance of constrained-QAOA are especially exciting. The crux of constrained-QAOA is the subspace-restricted evolution within the set of feasible solutions. Therefore, it is critically important to study the noise effects when running on real quantum hardware and developing mitigation methods. New hardware is scheduled to be made available which natively supports multi-

qubit gates and we plan to experimentally test these algorithms.

More recently, some architectures have proposed methods to natively support highly-controlled gates in the quantum instruction set architecture. However, as the number of controls increases, these native gates will experience lower fidelity due to the increased experimental complexity. In neutral atom quantum computers, these gates can be implemented using Rydberg interactions Isenhower et al. [2011], Saffman [2016]. In fact, using these native Rydberg gates allows for more efficient implementations of Grover's search algorithm Molmer et al. [2011], Petrosyan et al. [2016], and others have shown via quantum Monte Carlo simulations that native Toffoli gates implemented via Rydberg blockades may have higher fidelity than the decomposition into one- and two-qubit Rydberg gates Gulliksen et al. [2015], in spite of the Toffoli's naturally lower fidelity. Using numerical simulations and error analysis, error rates of $1 - \mathcal{F} < 10\%$ for $k = 35$ Isenhower et al. [2011] and $1 - \mathcal{F} < 1\%$ for $k = 20$ Khazali and Molmer [2020] are possible using these Rydberg gates.

Highly-controlled gates can also be implemented using Ising interactions similar to the two-qubit Molmer-Sorensen gates typically used in trapped ion systems Espinoza et al. [2021], Katz et al. [2022]. These gates are based on a single-step implementation protocol that scales with the number of qubits Rasmussen et al. [2020]. They demonstrate high fidelity ($\mathcal{F} > 99\%$) gates on 3 to 7 qubits with gate times < 1 ms Espinoza et al. [2021].

Various approaches to natively implement highly-controlled gates also exist in superconducting systems. One approach involves using adiabatic evolution of excitation-exchange eigenstates, similar to those used in Rydberg atoms Khazali and Molmer [2020]. Other demonstrations have relied on a driven Ising-type model similar to those used in trapped ions Rasmussen et al. [2020], and the native cross-resonance effect on fixed-frequency transmons Kim et al. [2022]. The new developments and experimental demonstrations of natively supported multi-controlled gates presents an exciting opportunity for implementing and testing algorithms such as the constrained QAOA on near-term quantum computers.

7 Acknowledgments

T.T. and N.A. is supported in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730082. Z.S. is supported by the National Science Foundation under Award No. 2037984, and QNEXT.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

8 References

Jay M Gambetta, Jerry M Chow, and Matthias Steffen. Building logical qubits in a superconducting quantum computing system. npj Quantum Information, 3(1):1–7, 2017. https://doi.org/10.1038/s41534-016-0004-0.

K Wright, KM Beck, Sea Debnath, JM Amini, Y Nam, N Grzesiak, J-S Chen, NC Pisenti, M Chmielewski, C Collins, et al. Benchmarking an 11-qubit quantum computer. Nature communications, 10(1):1–6, 2019. https://doi.org/10.1038/s41467-019-13534-2.

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. Nature, 574(7779):505–510, 2019. https://doi.org/10.1038/s41586-019-1666-5.

JM Arrazola, V Bergholm, K Brádler, TR Bromley, MJ Collins, I Dhand, A Fumagalli, T Gerrits, A Goussev, LG Helt, et al. Quantum circuits with many photons on a programmable nanophotonic chip. Nature, 591 (7848):54–60, 2021. https://doi.org/10.1038/s41586-021-03202-1.

Thomas Monz, Kihwan Kim, Wolfgang Hänsel, M Riebe, AS Villar, Philipp Schindler, Michael Chwalla, Markus Hennrich, and Rainer Blatt. Realization of the quantum toffoli gate with trapped ions. Physical review letters, 102 (4):040501, 2009. https://doi.org/10.1103/PhysRevLett.102.040501.

Larry Isenhower, Mark Saffman, and Klaus Molmer. Multibit C_k NOT quantum gates via Rydberg blockade. Quantum Information Processing, 10(6):755–770, 2011. https://doi.org/10.1007/s11128-011-0292-4.

John Preskill. Quantum computing in the NISQ era and beyond. Quantum, 2:79, 2018. https://doi.org/10.22331/q-2018-08-06-79.

Marco Cerezo, 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, 3(9):625–644, 2021. https://doi.org/10.1038/s42254-021-00348-9.

Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. Full-stack, real-system quantum computer studies: Architectural comparisons and design insights. In 2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA), pages 527–540. IEEE, 2019. https://doi.org/10.1145/3307650.3322273.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. arXiv preprint arXiv:1411.4028, 2014. https://doi.org/10.48550/arXiv.1411.4028.

Zain H Saleem, Teague Tomesh, Bilal Tariq, and Martin Suchara. Approaches to Constrained Quantum Approximate Optimization. SN Computer Science, 4(2):183, 2023. https://doi.org/10.1007/s42979-022-01638-4.

Stuart Andrew Hadfield. Quantum Algorithms for Scientific Computing and Approximate Optimization. PhD thesis, Columbia University, 2018. https://doi.org/10.7916/D8X650C9.

Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. Algorithms, 12(2):34, 2019. https://doi.org/10.3390/a12020034.

Zain Hamid Saleem. Max-independent set and the quantum alternating operator ansatz. International Journal of Quantum Information, 18(04):2050011, 2020. https://doi.org/10.1142/S0219749920500112.

Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. Physical review A, 52 (5):3457, 1995. https://doi.org/10.1103/PhysRevA.52.3457.

Juha J Vartiainen, Mikko Möttönen, and Martti M Salomaa. Efficient decomposition of quantum gates. Physical review letters, 92(17):177902, 2004. https://doi.org/10.1103/PhysRevLett.92.177902.

Esteban A Martinez, Thomas Monz, Daniel Nigg, Philipp Schindler, and Rainer Blatt. Compiling quantum algorithms for architectures with multi-qubit gates. New Journal of Physics, 18 (6):063029, 2016. https://doi.org/10.1088/1367-2630/18/6/063029.

Pranav Gokhale, Jonathan M Baker, Casey Duckering, Natalie C Brown, Kenneth R Brown, and Frederic T Chong. Asymptotic improvements to quantum circuits via qutrits. In Proceedings of the 46th International Symposium on Computer Architecture, pages 554–566, 2019. https://doi.org/10.1145/3307650.3322253.

Péter Rakyta and Zoltán Zimborás. Approaching the theoretical limit in quantum gate decomposition. Quantum, 6:710, 2022. https://doi.org/10.22331/q-2022-05-11-710.

Chi-Kwong Li, Rebecca Roberts, and Xiaoyan Yin. Decomposition of unitary matrices and quantum gates. International Journal of Quantum Information, 11(01): 1350015, 2013. https://doi.org/10.1142/S0219749913500159.

Ed Younis, Koushik Sen, Katherine Yelick, and Costin Iancu. QFAST: Conflating Search and Numerical Optimization for Scalable Quantum Circuit Synthesis. In 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pages 232–243. IEEE, 2021. https://doi.org/10.1109/QCE52317.2021.00041.

Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I Schuster, Henry Hoffmann, and Frederic T Chong. Optimized Compilation of Aggregated Instructions for Realistic Quantum Computers. In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, pages 1031–1044, 2019. https://doi.org/10.1145/3297858.3304018.

Rebekah Herrman, Phillip C Lotshaw, James Ostrowski, Travis S Humble, and George Siopsis. Multi-angle quantum approximate optimization algorithm. Scientific Reports, 12 (1):1–10, 2022. https://doi.org/10.1038/s41598-022-10555-8.

Mark Saffman. Quantum computing with atomic qubits and Rydberg interactions: progress and challenges. Journal of Physics B: Atomic, Molecular and Optical Physics, 49(20):202001, 2016. https:doi.org/10.1088/0953-4075/49/20/202001.

Juan Diego Arias Espinoza, Koen Groenland, Matteo Mazzanti, Kareljan Schoutens, and Rene Gerritsma. High-fidelity method for a single-step N-bit Toffoli gate in trapped ions. Physical Review A, 103(5):052437, 2021. https://doi.org/10.1103/PhysRevA.103.052437.

Or Katz, Marko Cetina, and Christopher Monroe. N-Body Interactions between Trapped Ion Qubits via Spin-Dependent Squeezing. Phys. Rev. Lett., 129:063603, Aug 2022. https://doi.org/10.1103/PhysRevLett.129.063603.

SE Rasmussen, K Groenland, R Gerritsma, K Schoutens, and NT Zinner. Single-step implementation of high-fidelity n-bit Toffoli gates. Physical Review A, 101(2):022308, 2020. https://doi.org/10.1103/PhysRevA.101.022308.

Yosep Kim, Alexis Morvan, Long B Nguyen, Ravi K Naik, Christian Jünger, Larry Chen, John Mark Kreikebaum, David I Santiago, and Irfan Siddiqi. High-fidelity three-qubit iToffoli gate for fixed-frequency superconducting qubits. Nature Physics, pages 1–6, 2022. https://doi.org/10.1038/s41567-022-01590-3.

Alexis Morvan, VV Ramasesh, MS Blok, JM Kreikebaum, K O'Brien, L Chen, BK Mitchell, RK Naik, DI Santiago, and I Siddiqi. Qutrit Randomized Benchmarking. Physical review letters, 126(21): 210504, 2021. https://doi.org/10.1103/PhysRevLett.126.210504.

Pranav Gokhale, Ali Javadi-Abhari, Nathan Earnest, Yunong Shi, and Frederic T Chong. Optimized Quantum Compilation for Near-Term Algorithms with OpenPulse. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pages 186–200. IEEE, 2020. https://doi.org/10.1109/MICRO50266.2020.00027.

Lingling Lao, Prakash Murali, Margaret Martonosi, and Dan Browne. Designing Calibration and Expressivity-Efficient Instruction Sets for Quantum Computing. In 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), pages 846–859. IEEE, 2021. https://doi.org/10.1109/ISCA52012.2021.00071.

Daniel J Egger, Jakub Mareček, and Stefan Woerner. Warm-starting quantum optimization. Quantum, 5:479, 2021. https://doi.org/10.22331/q-2021-06-17-479.

Teague Tomesh, Zain H Saleem, and Martin Suchara. Quantum Local Search with the Quantum Alternating Operator Ansatz. Quantum, 6:781, 2022a. https://doi.org/10.22331/q-2022-08-22-781.

Teague Tomesh, Zain H. Saleem, Michael A. Perlin, Pranav Gokhale, Martin Suchara, and Margaret Martonosi. Divide and Conquer for Combinatorial Optimization and Distributed Quantum Computation. In 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), volume 01, pages 1–12, 2023. https://doi.org/10.1109/QCE57702.2023.00009.

Thomas G Draper, Samuel A Kutin, Eric M Rains, and Krysta M Svore. A logarithmic-depth quantum carry-lookahead adder. arXiv preprint quant-ph/0406142, 2004. https://doi.org/10.48550/arXiv.quant-ph/0406142.

Amlan Chakrabarti and Susmita Sur-Kolay. Designing quantum adder circuits and evaluating their error performance. In 2008 International Conference on Electronic Design, pages 1–6. IEEE, 2008. https://doi.org/10.1109/ICED.2008.4786689.

Juan Miguel Arrazola, Olivia Di Matteo, Nicolás Quesada, Soran Jahangiri, Alain Delgado, and Nathan Killoran. Universal quantum circuits for quantum chemistry. Quantum, 6:742, 2022. https://doi.org/10.22331/q-2022-06-20-742.

Daniel Crow, Robert Joynt, and Mark Saffman. Improved Error Thresholds for Measurement-Free Error Correction. Physical review letters, 117(13):130503, 2016. https://doi.org/10.1103/PhysRevLett.117.130503.

Michael A. Perlin, Vickram N. Premakumar, Jiakai Wang, Mark Saffman, and Robert Joynt. Fault-tolerant measurement-free quantum error correction with multiqubit gates. Phys. Rev. A, 108:062426, Dec 2023. https://doi.org/10.1103/PhysRevA.108.062426.

IonQ. API Documentation: Supported Gates. https://docs.ionq.com/#section/Supported-Gates, 2022.

Craig Gidney. Constructing large controlled nots. Available from: https://algassert.com/circuits/2015/06/05/Constructing-Large-Controlled-Nots.html, 2015a.

Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum circuits for isometries. Physical Review A, 93(3):032318, 2016. https://doi.org/10.1103/PhysRevA.93.032318.

Teague Tomesh, Pranav Gokhale, Victory Omole, Gokul Subramanian Ravi, Kaitlin N Smith, Joshua Viszlai, Xin-Chuan Wu, Nikos Hardavellas, Margaret R Martonosi, and Frederic T Chong. SupermarQ: A Scalable Quantum Benchmark Suite. In 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pages 587–603. IEEE, 2022b. https://doi.org/10.1109/HPCA53966.2022.00050.

Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. Nature, 614(7949):676–681, 2023. See the supplementary information for gate fidelities. https://doi.org/10.1038/s41586-022-05434-1.

Craig Gidney. Constructing large increment gates. Available: https://algassert.com/circuits/2015/06/12/Constructing-Large-Increment-Gates.html, 2015b.

Craig Gidney. Using quantum gates instead of ancilla bits. Available: https://algassert.com/circuits/2015/06/22/Using-Quantum-Gates-instead-of-Ancilla-Bits.html, 2015c.

Jake Gulliksen, Durga Bhaktavatsala Rao Dasari, and Klaus Molmer. Characterization of how dissipation and dephasing errors accumulate in quantum computers. EPJ Quantum Technology, 2(1):1–10, 2015. https://doi.org/10.1140/epjqt17.

Teague Tomesh. quantum-constrained-optimization. https://github.com/teaguetomesh/quantum-constrained-optimization, 2022.

Ruslan Shaydulin, Phillip C. Lotshaw, Jeffrey Larson, James Ostrowski, and Travis S. Humble. Parameter Transfer for Quantum Approximate Optimization of Weighted MaxCut. ACM Transactions on Quantum Computing, 4 (3), April 2023. https://doi.org/10.1145/3584706.

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 preprint arXiv:2110.14206, 2021. https://doi.org/10.48550/arXiv.2110.14206.

Ruslan Shaydulin, Changhao Li, Shouvanik Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, et al. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. Science Advances, 10 (22):eadm6761, 2024. https://doi.org/10.1126/sciadv.adm6761.

Angad Kalra, Faisal Qureshi, and Michael Tisi. Portfolio Asset Identification Using Graph Algorithms on a Quantum Annealer. Available at SSRN 3333537, 2018. https://dx.doi.org/10.2139/ssrn.3333537.

Eugene L Lawler. Optimal Sequencing of a Single Machine Subject to Precedence Constraints. Management science, 19(5):544–546, 1973. https://doi.org/10.1287/mnsc.19.5.544.

Andy Auyeung. A new phylogenetic tree model for fuzzy characters. In International

Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II, volume 1, pages 2–7. IEEE, 2005. https://doi.org/10.1109/ITCC.2005.22.

Klaus Molmer, Larry Isenhower, and Mark Saffman. Efficient Grover search with Rydberg blockade. Journal of Physics B: Atomic, Molecular and Optical Physics, 44 (18):184016, 2011. https://doi.org/10.1088/0953-4075/44/18/184016.

David Petrosyan, Mark Saffman, and Klaus Molmer. Grover search algorithm with Rydberg-blockaded atoms: quantum Monte Carlo simulations. Journal of Physics B: Atomic, Molecular and Optical Physics, 49(9): 094004, 2016. http://dx.doi.org/10.1088/0953-4075/49/9/094004.

Mohammadsadegh Khazali and Klaus Molmer. Fast multiqubit gates by adiabatic evolution in interacting excited-state manifolds of rydberg atoms and superconducting circuits. Physical Review X, 10(2):021054, 2020. https://doi.org/10.1103/PhysRevX.10.021054.

Yao-Min Di and Hai-Rui Wei. Elementary gates for ternary quantum logic circuit. arXiv preprint arXiv:1105.5485, 2011. https://doi.org/10.48550/arXiv.1105.5485.

A Gate Counts for Benchmark Graphs

	Average $C^k(R_x(\theta))$ gate counts for 20 node graphs													
Circuit	$C^3(R_x)$	$C^4(R_x)$	$C^5(R_x)$	$C^6(R_x)$	$C^7(R_x)$	$C^8(R_x)$	$C^9(R_x)$	$C^{10}(R_x)$	$C^{11}(R_x)$	$C^{12}(R_x)$	$C^{13}(R_x)$	$C^{14}(R_x)$	$C^{15}(R_x)$	$C^{16}(R_x)$
SA-QAOA $p=1$	(20, 0.08)	0.16	0.38	0.94	2.14	2.9	3.74	3.7	2.84	1.62	0.96	0.42	0.06	0.06
SA-QAOA $p=5$	(100, 0.4)	0.8	1.9	4.7	10.7	14.5	18.7	18.5	14.2	8.1	4.8	2.1	0.3	0.3
SA-QAOA $p=10$	(200, 0.8)	1.6	3.8	9.4	21.4	29.0	37.4	37.0	28.4	16.2	9.6	4.2	0.6	0.6
MA-QAOA $p=1$	(20, 0.08)	0.16	0.38	0.94	2.14	2.9	3.74	3.7	2.84	1.62	0.96	0.42	0.06	0.06
DQVA $\nu=5$	(4, 0.01)	0.01	0.06	0.14	0.33	0.54	0.71	0.77	0.67	0.41	0.25	0.09	0.01	0.01
DQVA $\nu=7$	(6, 0.01)	0.03	0.07	0.17	0.45	0.87	1.15	1.2	0.92	0.56	0.36	0.17	0.02	0.03
DQVA $\nu=10$	(9, 0.01)	0.01	0.12	0.24	0.62	1.12	1.73	1.8	1.59	0.94	0.52	0.24	0.04	0.02
DQVA $\nu=12$	(11, 0.01)	0.04	0.13	0.33	0.81	1.41	1.97	2.22	1.93	1.08	0.7	0.32	0.04	0.04
DQVA $\nu=14$	(13, 0.01)	0.03	0.08	0.38	1.01	1.66	2.44	2.63	2.22	1.31	0.76	0.37	0.05	0.05
DQVA $\nu=18$	(17, 0.01)	0.06	0.08	0.47	1.07	1.87	2.72	3.12	2.53	1.51	0.91	0.41	0.06	0.06

Table 5: Average number of $C^k(R_x(\theta))$ gates that appeared in the variational circuits for SA-QAOA, MA-QAOA, and DQVA across all 20-node benchmark 3-regular and Erdös-Rényi graphs used to generate Figure 5. The quantum circuits targeting the 3-regular graphs are composed entirely of $C^3(R_x(\theta))$ gates so two values are listed in that column (3-regular counts on the left, Erdös-Rényi on the right). All other gate counts correspond to quantum circuits targeting the benchmark Erdös-Rényi graphs. Across both graph types and number of controls, the DQVA produces variational circuits requiring fewer quantum resources.

Table 5 reports the number and kind of partial mixers that appear in the variational circuits over all of the 20-node benchmark graphs. The structure of the 3-regular graphs is consistent across each random instance which leads to stable gate counts in the corresponding quantum circuits. The Erdös-Rényi graphs show much more variation and so the average number of each partial mixer within the circuits varies within and between each algorithm. Across all of the 20-node graphs, the DQVA circuits consistently contain the fewest number of partial mixers.

B Gate Count Derivations for Decomposition Schemes

In this section, we will provide references and derivations for each of the asymptotic gate count values provided in Table 3. We will break into cases depending on the number of ancillae being considered.

B.1 No ancillae

In the case where we have no ancillae, the gate counts for $\mathcal{S}_2^2$ and $\mathcal{S}_3^2$ come from Theorem 5 of Iten et al. [2016]. The asymptotic gate count for $\mathcal{S}_m^2$ comes from first decomposing the $C^n(R_x)$ gate into two $C^{n-1}(X)$ gates via Lemma 7.9 of Barenco et al. [1995]. Then, we can split the $C^{n-1}(X)$ gates in half using a similar circuit to that of Figure 2. The only difference is that the ancilla is now borrowed, so we need to split into four half-sized multi-controlled NOT gates, as shown in Figure 9.

Then, we can use a generalization of the decomposition scheme shown in Figure 3 where we decompose into $\mathrm{C}^{m-1}\mathrm{NOT}$ gates instead of Toffolis. We show this scheme where $m-1=3$ in Figure 10. It is not too difficult to see that each $\mathrm{C}^k\mathrm{NOT}$ gate requires $\sim 4k/(m-2)$ $\mathrm{C}^{m-1}\mathrm{NOT}$ gates to implement, and so putting this altogether the gate set $\mathcal{S}_m^2$ uses

$$4\left(\frac{4}{m-2}\right)\left(\left\lfloor\frac{n}{2}\right\rfloor+\left\lceil\frac{n}{2}\right\rceil\right)=\sim\left(\frac{16}{m-2}\right)n \quad (15)$$

$\mathrm{C}^{m-1}\mathrm{NOT}$ gates as desired.

B.2 One ancilla

In the case where we have one burnable ancilla and we want to decompose a $C^n(R_x)$ gate, we can break up the gate into a $C^{n-1}(X)$ gate and a $C(R_x)$ gate to then decompose separately, as in Lemma 7.11 of Barenco et al. [1995] (here, since our ancilla is burnable, we don't need the extra $C^{n-1}(X)$ gate). We can use the same decompo-

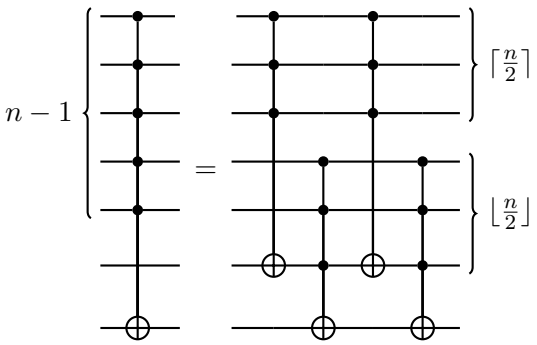

Figure 9: Splitting a multi-controlled not "in half," into four multi-controlled not gates with half the number of controls Gidney [2015a]. Generalization of the decomposition scheme shown in Figure 2.

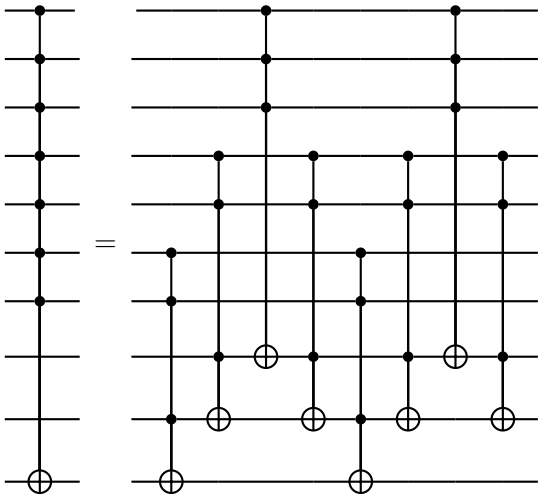

Figure 10: Decomposition scheme for a multi-controlled X gate with access to two borrowed ancilla qubits that uses C^{m-1}NOT gates (here $m-1=3$) Barenco et al. [1995]. Generalization of the decomposition scheme shown in Figure 3 that uses gates larger than Toffoli's.

sition scheme as shown in Figure 9 to split the $C^{n-1}(X)$ in half, and then use the decomposition scheme in Figure 3 to split it into Toffolis. The counts for $\mathcal{S}_2^2$ and $\mathcal{S}_3^2$ come from Lemma 8 of Iten et al. [2016] and Corollary 7.4 of Barenco et al. [1995], respectively. The counts for $\mathcal{S}_m^2$ can be computed in much the same way as we did for the no ancillae case – since there is only one $C^{n-1}(X)$ gate instead of two, we obtain the count of $\sim (8/(m-2))n$ C^{m-1}NOT gates as desired.

In the case where we want to decompose a $C^n(X)$ gate, we can just directly split the gate in half using the scheme in Figure 9, except we only need one of each "half-sized" multi-controlled not gate, since the ancilla is burnable. Then, the same procedures give the counts in the table

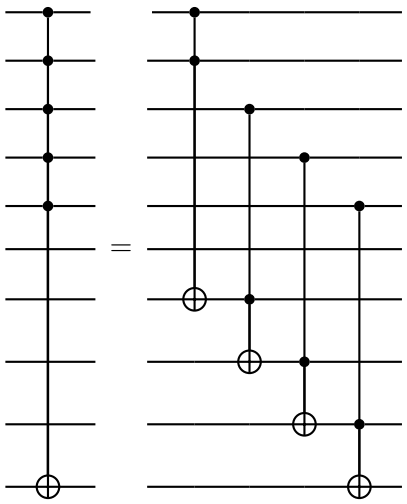

Figure 11: Modified decomposition scheme from Figures 3 and 10 that now uses burnable ancillae. As a result, we don't have to return the ancillae to their original states, allowing us to quarter the number of gates required.

for $\mathcal{S}_2^2$ and $\mathcal{S}_3^2$ as desired (notice also that now we do not have the extra $C(R_x)$ gate to decompose, which is what gave the 4 extra single qubit gates and 2 extra two qubit gates in the $\mathcal{S}_3^2$ case, by Corollary 5.3 of Barenco et al. [1995]). The counts for $\mathcal{S}_m^2$ follow similarly, since we have again halved the number of $C^{O(n)}(X)$ gates again.

B.3 n ancillae

In the case where we have n burnable ancillae, we can actually modify the decomposition scheme in Figures 3 and 10 to use about one fourth as many Toffolis. We show this modified scheme in Figure 11. This gives us the $n-1$ count for the gate set $\mathcal{S}_3^2$, but for the other counts, we need to treat the last Toffoli as a $C^2(R_x)$ gate. The same techniques as those above give us the remaining counts. The gate counts for $\mathcal{S}_m^2$ come from generalizing the decomposition scheme from Figure 11 in the same way as we did to get Figure 10.

B.4 Gate counts for $\mathcal{S}_2^3$

The general tree-like decomposition for a multi-controlled unitary using qutrits is given in Gokhale et al. [2019] (see Figure 5 of Gokhale et al. [2019]). The three-qutrit gates that appear in the decomposition — consisting of two controls on either the $|1\rangle$ or $|2\rangle$ qutrit states and an increment or decrement operation applied to the final qutrit — are then further decomposed into single-

and two-qutrit operations using the decomposition found in Figure 10 of Di and Wei [2011]. For each three-qutrit operation, the decomposition of Di and Wei [2011] yields 6 two-qutrit and 7 single-qutrit operations. Therefore, for a multi-controlled gate with n controls, $n-1$ three-qutrit operations will appear due to the decomposition of Gokhale et al. [2019], and each of these yields 6 two-qutrit operations due to Di and Wei [2011], finally two extra two-qutrit operations are required to implement the single-controlled unitary. This yields results in an asymptotic scaling of $6n - 4$ two-qutrit operations required to implement n-controlled R_x gates. The single-qutrit gate costs similarly follow using the same decomposition given in Di and Wei [2011].